



DOCTORAL THESIS

Design of a hyper-heuristics based
control framework for
modular production systems

Submitted by

Marcel Panzer

in fulfillment of the requirements
for the *doctor rerum politicarum*

at the

CHAIR OF BUSINESS INFORMATICS, ESPC. PROCESSES AND SYSTEMS,
FACULTY OF ECONOMICS AND SOCIAL SCIENCES,
UNIVERSITY OF POTSDAM

Thesis defense conducted on April 12, 2024.

Unless otherwise indicated, this work is licensed under a Creative Commons License Attribution – NonCommercial – ShareAlike 4.0 International.

This does not apply to quoted content and works based on other permissions.

To view a copy of this licence visit:

<https://creativecommons.org/licenses/by-nc-sa/4.0>

I would like to thank my thesis supervisors:

1. Univ.-Prof. Dr.-Ing. habil. Norbert Gronau, as my primary supervisor,
2. Prof. Dr. Hanna Krasnova, as my secondary supervisor.

I am very grateful to the esteemed reviewers of my thesis:

1. Prof. Dr.-Ing. habil. Peter Nyhuis,
2. Prof. Dr.-Ing. Marcus Grum.

Published online on the

Publication Server of the University of Potsdam:

<https://doi.org/10.25932/publishup-63300>

<https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-633006>

Acknowledgement

My sincere appreciation goes to Univ.-Prof. Dr.-Ing. habil. Norbert Gronau, my supervisor, for his invaluable guidance and expertise. His mentorship has been instrumental in lending clarity and confidence to my research journey, significantly shaping the direction and quality of this thesis. The countless constructive discussions we've had have been immensely enriching and pivotal to the development of my ideas and understanding.

I extend my gratitude to Prof. Dr.-Ing. habil. Peter Nyhuis and Jun. Prof. Dr.-Ing. Marcus Grum for the interest in my work and for their invaluable contributions as reviewers and supervisors of this thesis.

I would like to thank my colleagues, particularly M.A. Malte Rolf Teichmann, Dr. Benedict Bender, and M.Sc. Clementine Bertheau, for their valuable insights, collaboration, and friendship. Your contributions have been instrumental in bringing this thesis to completion.

I would like to extend my thanks to Dr.-Ing. Sander Lass and B.Sc. Stephan Sailer, the masterminds and driving forces behind the *Center for Industry 4.0*. As my primary contacts for technical matters, their expertise and guidance have been invaluable to my work. Additionally, I extend many thanks to M.Sc. David Kotarski for consistently and reliably resolving IT-issues with remarkable speed.

I extend my gratitude to my family and friends for their unwavering support during this journey. Special thanks goes to my parents, Verena and Karl-Heinz, and my sister, Melissa, for their encouragement and strength, which have been crucial to the completion of this thesis. Additionally, I extend my gratitude to my partner and my better half, Mercedes, for her constant support, motivation, and understanding.

Finally, I express my deep gratitude to all others who contributed to this thesis and journey, regardless of their level of involvement. Your support and motivation have significantly impacted both my research and personal growth.

Potsdam, 08.01.2024

Marcel Panzer

Abstract

Volatile supply and sales markets, coupled with increasing product individualization and complex production processes, present significant challenges for manufacturing companies. These must navigate and adapt to ever-shifting external and internal factors while ensuring robustness against process variabilities and unforeseen events. This has a pronounced impact on production control, which serves as the operational intersection between production planning and the shop-floor resources, and necessitates the capability to manage intricate process interdependencies effectively. Considering the increasing dynamics and product diversification, alongside the need to maintain constant production performances, the implementation of innovative control strategies becomes crucial.

In recent years, the integration of Industry 4.0 technologies and machine learning methods has gained prominence in addressing emerging challenges in production applications. Within this context, this cumulative thesis analyzes deep learning based production systems based on five publications. Particular attention is paid to the applications of deep reinforcement learning, aiming to explore its potential in dynamic control contexts. Analysis reveal that deep reinforcement learning excels in various applications, especially in dynamic production control tasks. Its efficacy can be attributed to its interactive learning and real-time operational model. However, despite its evident utility, there are notable structural, organizational, and algorithmic gaps in the prevailing research. A predominant portion of deep reinforcement learning based approaches is limited to specific job shop scenarios and often overlooks the potential synergies in combined resources. Furthermore, it highlights the rare implementation of multi-agent systems and semi-heterarchical systems in practical settings. A notable gap remains in the integration of deep reinforcement learning into a hyper-heuristic.

To bridge these research gaps, this thesis introduces a deep reinforcement learning based hyper-heuristic for the control of modular production systems, developed in accordance with the design science research methodology. Implemented within a semi-heterarchical multi-agent framework, this approach achieves a threefold reduction in control and optimisation complexity while ensuring high scalability, adaptability, and robustness of the system. In comparative benchmarks, this control methodology outperforms rule-based heuristics, reducing throughput times and tardiness, and effectively incorporates customer and order-centric metrics. The control artifact facilitates a rapid scenario generation, motivating for further research efforts and bridging the gap to real-world applications. The overarching goal is to foster a synergy between theoretical insights and practical solutions, thereby enriching scientific discourse and addressing current industrial challenges.

Zusammenfassung

Volatile Beschaffungs- und Absatzmärkte sowie eine zunehmende Produktindividualisierung konfrontieren Fertigungsunternehmen mit beträchtlichen Herausforderungen. Diese erfordern eine Anpassung der Produktion an sich ständig wechselnde externe Einflüsse und eine hohe Prozessrobustheit gegenüber unvorhersehbaren Schwankungen. Ein Schlüsselement in diesem Kontext ist die Produktionssteuerung, die als operative Schnittstelle zwischen der Produktionsplanung und den Fertigungsressourcen fungiert und eine effiziente Handhabung zahlreicher Prozessinterdependenzen sicherstellen muss. Angesichts dieser gesteigerten Produktionsdynamik und Produktvielfalt rücken innovative Steuerungsansätze in den Vordergrund.

In jüngerer Zeit wurden daher verstärkt Industrie-4.0-Ansätze und Methoden des maschinellen Lernens betrachtet. Im Kontext der aktuellen Forschung analysiert die vorliegende kumulative Arbeit Deep-Learning basierte Produktionssysteme anhand von fünf Publikationen. Hierbei wird ein besonderes Augenmerk auf die Anwendungen des Deep Reinforcement Learning gelegt, um dessen Potenzial zu ergründen. Die Untersuchungen zeigen, dass das Deep Reinforcement Learning in vielen Produktionsanwendungen sowohl herkömmlichen Ansätzen als auch anderen Deep-Learning Werkzeugen überlegen ist. Diese Überlegenheit ergibt sich vor allem aus dem interaktiven Lernprinzip und der direkten Interaktion mit der Umwelt, was es für die dynamische Produktionssteuerung besonders geeignet macht. Dennoch werden strukturelle, organisatorische und algorithmische Forschungslücken identifiziert. Die überwiegende Mehrheit der untersuchten Ansätze fokussiert sich auf Werkstattfertigungen und vernachlässigt dabei potenzielle Prozesssynergien modularer Produktionssysteme. Ferner zeigt sich, dass Multi-Agenten- und Mehr-Ebenen-Systeme sowie die Kombination verschiedener algorithmischer Ansätze nur selten zur Anwendung kommen.

Um diese Forschungslücken zu adressieren, wird eine auf Deep Reinforcement Learning basierende Hyper-Heuristik für die Steuerung modularer Produktionssysteme vorgestellt, die nach der Design Science Research Methodology entwickelt wird. Ein semi-heterarchisches Multi-Agenten-System ermöglicht eine dreifache Reduktion der Steuerungs- und Optimierungskomplexität und gewährleistet gleichzeitig eine hohe Systemadaptabilität und -robustheit. In Benchmarks übertrifft das Steuerungskonzept regelbasierte Ansätze, minimiert Durchlaufzeiten und Verspätungen und berücksichtigt kunden- sowie auftragsorientierte Kennzahlen. Die entwickelte Steuerungsmethodik ermöglicht einen schnellen Szenari엔entwurf, um dadurch weitere Forschungsbemühungen zu stimulieren und die bestehende Transferlücke zur Realität weiter zu überbrücken. Das Ziel dieser Forschungsarbeit ist es, eine Synergie zwischen theoretischen Erkenntnissen und Praxis-relevanten Lösungen zu schaffen, um sowohl den wissenschaftlichen Diskurs zu bereichern als auch Antworten auf aktuelle industrielle Herausforderungen zu bieten.

Contents

1	Introduction	1
1.1	Motivation	4
1.2	Research questions and objective	6
1.3	Research methodology	11
1.4	Thesis structure	14
2	Fundamentals	17
2.1	Production process design	17
2.1.1	Job-shop production processes	18
2.1.2	Matrix production systems	19
2.1.3	Agent-based systems	21
2.1.3.1	Production agents and environment	21
2.1.3.2	Multi-agent based production - organization	22
2.1.4	Section conclusion	24
2.2	Production planning and control	25
2.2.1	Basic concepts of production control	26
2.2.2	Heuristics control and optimization strategies	28
2.2.2.1	Priority-rule based heuristics	28
2.2.2.2	Meta-heuristics	31
2.2.2.3	Hyper-heuristics	31
2.2.3	Section conclusion	31
2.3	Basic concepts of machine learning	33
2.3.1	Basic of reinforcement learning	33
2.3.1.1	Temporal-difference learning and value-based algorithms	35
2.3.1.2	Optimization model	36
2.3.2	Deep reinforcement learning	38
2.3.2.1	Neural networks	39
2.3.2.2	Algorithmic DQN peculiarities	40
2.4	Chapter conclusion and initial research gap	41

3	Publications and research paradigm	43
3.1	Research paradigm	43
3.2	Bundle of publications 1 - identification and structuring of the research gap . .	46
3.2.1	Review methodology	47
3.2.2	Publication 1 - deep reinforcement learning based production systems .	48
3.2.3	Publication 2 - organizational deep learning production perspectives . .	49
3.3	Bundle of publications 2 - addressing the research gap	51
3.3.1	Artifact construction methodology	54
3.3.2	Publication 3 - a deep learning based simulation framework	55
3.3.3	Publication 4 - benchmarking and real-world transfer	57
3.3.4	Publication 5 - economic performance evaluation	58
4	Publication 1 - Deep reinforcement learning based production	61
4.1	Introduction	62
4.2	Introduction to reinforcement learning	63
4.3	Research methodology	65
4.3.1	Review focus	66
4.3.2	Literature search	67
4.3.2.1	Phase 1 - database and iterative keyword selection	67
4.3.2.2	Phase 2 - defining inclusion and exclusion criteria	67
4.3.2.3	Phase 3 - conducting the literature search	68
4.3.2.4	Phase 4 - data gathering	69
4.3.3	Analysis of yearly and outlet related contributions	69
4.4	Literature analysis	70
4.4.1	Process control	70
4.4.2	Production scheduling and dispatching	72
4.4.2.1	Production scheduling	72
4.4.2.2	Production dispatching	74
4.4.3	(Intra-) Logistics	75
4.4.4	Assembly	76
4.4.5	Robotics	78
4.4.6	Maintenance	78
4.4.7	Energy management	80
4.4.8	(Process) Design	81
4.4.9	Quality control	81
4.4.10	Further applications	82
4.5	Implementation challenges and research agenda	83
4.5.1	Implementation challenges and research gaps	83

4.5.2	Future research agenda	85
4.6	Discussion	88
4.6.1	Managerial implications	88
4.6.2	Limitations	89
4.7	Conclusion	89
4.8	References - publication 1	90
5	Publication 2 - An architectural deep learning production perspective	107
5.1	Introduction	108
5.2	Neural network-based production planning and control	110
5.2.1	Neural networks	110
5.2.2	ML-based PPC	110
5.2.3	MA system organization	111
5.3	Methodology	112
5.3.1	Review focus	112
5.3.2	Literature search	113
5.3.2.1	Phase 1 - database and iterative keyword selection	113
5.3.2.2	Phase 2 - defining inclusion and exclusion criteria	113
5.3.2.3	Phase 3 - conducting the literature search	114
5.3.2.4	Phase 4 - data gathering	114
5.3.3	Analysis of yearly and outlet-related contributions	115
5.4	Analysis	115
5.4.1	Production planning	116
5.4.1.1	Plain NN planning approaches	117
5.4.1.2	Embedded NN planning approaches	117
5.4.1.3	Multi-agent planning approaches	118
5.4.2	Forecasting	119
5.4.2.1	Plain NN forecasting approaches	119
5.4.2.2	Embedded NN forecasting approaches	120
5.4.2.3	Multi-agent forecasting approaches	121
5.4.3	Production control	121
5.4.3.1	Plain NN control approaches	121
5.4.3.2	Embedded NN control approaches	122
5.4.3.3	Multi-agent control approaches	122
5.4.4	General analysis	123
5.5	Taxonomy	125
5.6	Implementation challenges and research agenda	129
5.6.1	Implementation challenges	129

5.6.2	Future research agenda	131
5.7	Discussion	133
5.7.1	Managerial implications	134
5.7.2	Limitations	134
5.8	Conclusion	134
5.9	References - publication 2	137
5.10	Supplements and detailed review tables	155
6	Transition - from research gap definition to prototype design	163
6.1	Identification of the research gap and structuring research requirements	163
6.1.1	Structural perspective	164
6.1.2	Organizational perspective	164
6.1.3	Algorithmic perspective	166
6.2	Definition of design specifications and artifact construction	168
6.2.1	Structural perspective	169
6.2.2	Organizational perspective	171
6.2.3	Algorithmic perspective	173
6.3	Artifact design summary	177
6.4	Technical implementation - low-barrier artifact design	179
7	Publication 3 - A deep learning based production control framework	183
7.1	Introduction	184
7.2	Related work	186
7.2.1	Discrete-event based production simulation	187
7.2.2	Basics of (deep) reinforcement learning and hyper-heuristics	189
7.2.3	Deep RL based production dispatching	190
7.2.4	Research highlights and key contributions	192
7.3	Simulation design	193
7.3.1	Simulation framework design	193
7.3.2	Simulation components	195
7.4	Hyper-heuristics based control framework	196
7.4.1	Hyper-heuristics control mechanism	198
7.4.2	Reward function design	199
7.4.3	State and action space design	200
7.5	Demonstration and transfer of results	201
7.5.1	Simulated case-study	201
7.5.2	Exemplary simulation results	203
7.5.3	Training process analysis	203

7.5.4	Analysis of customer related indicator benchmarks	206
7.5.5	Evaluation of adaptability and resilience	206
7.6	Framework discussion	208
7.7	Conclusion	209
7.8	References - publication 3	211
8	Publication 4 - A hyper-heuristics based modular production control	221
8.1	Introduction	222
8.2	Problem statement	223
8.2.1	Modular and semi-heterarchical production systems	223
8.2.2	Deep reinforcement learning based hyper-heuristic	224
8.2.3	Deep RL and multi-agent based production control	226
8.2.4	Problem formulation and contribution	228
8.3	Conceptual design	230
8.3.1	Simulation approach	231
8.3.2	Variable hyper-heuristic design	232
8.3.2.1	State space design	233
8.3.2.2	Action space design	234
8.3.2.3	Reward function design	235
8.4	Demonstration	236
8.4.1	Experimental settings	236
8.4.2	Experimental results	239
8.4.2.1	Training process	239
8.4.2.2	Benchmark results	240
8.4.2.3	Analysis of optimization robustness and scalability	244
8.5	Simulation to reality transfer	245
8.6	Discussion	248
8.7	Conclusion	249
8.8	References - publication 4	251
9	Publication 5 - Economic efficiency in modular production	259
9.1	Introduction	260
9.2	Related work	261
9.2.1	Modular production systems	261
9.2.2	Deep reinforcement learning	262
9.2.3	Deep reinforcement learning-based production control	262
9.3	Proposed algorithm - a deep RL control framework	263
9.3.1	Agent state space	264

Contents

9.3.2	Agent action space	265
9.3.3	Designing the reward function	266
9.4	Simulation results and analysis	266
9.4.1	Experimental settings	266
9.4.2	Experimental results	267
9.5	Conclusion	269
9.6	References - publication 5	271
10	Discussion	275
10.1	Integration of the results	275
10.1.1	Requirements for deep learning based control optimization	275
10.1.2	Managing decision-making and optimization complexity	277
10.1.3	Generalizability of the developed control framework	279
10.1.4	Answering the central research question	280
10.2	Transferability of the results	281
10.2.1	Research contribution	281
10.2.2	Research transfer to operational practice	283
10.2.3	Managerial insights	288
11	Summary	291
11.1	Critical appraisal of the thesis	292
11.2	Fields for future research	293
12	Conclusion	299
	References	301
	List of Figures	327
	List of Tables	331
	Declaration on Honour	334
	Statements by the co-authors	336

1 Introduction

In times of increasing market uncertainties, sharp fluctuations in demand, and dynamic production processes, production planning and control systems must guarantee reliable and robust production processes (ElMaraghy et al., 2012a; Durão et al., 2019). Particularly in the current economy, companies are under increasing pressure to reduce costs, leverage process potentials, and at the same time maintain a consistently high level of product quality and production adaptability to changing market conditions (Omar et al., 2019; ElMaraghy et al., 2021). In addition, increased social responsibility and stringent sustainability regulations are becoming increasingly important, not only in terms of reducing emissions but also in the overall utilization and reuse of existing resources for redesigned processes and products (Jamwal et al., 2021). Therefore, to enhance competitiveness at both local and global levels, optimal scheduling and balanced control operations of shop-floor resources are essential to remain competitive. These must ensure adherence to customer demands and facilitate effective resource management (Koç et al., 2022; Geurtsen et al., 2023).

Central to this is the production planning and control domain, as illustrated in Figure 1.1. It integrates production tasks, ranging from the long-term production program to short-term dispatching activities (Chapman, 2006; Mönch et al., 2013; Jacobs et al., 2018). It addresses the challenge of reconciling conflicting objectives like maximizing mean resource utilization while maintaining low inventory levels (Lödding, 2016). In operations, orchestrating and overseeing production processes are key, which require continuous synchronization of interdependent processes to create optimal schedules. The process demands cohesive interaction across the planning and control levels and timescales, to ensure a harmonized and resilient operations design (Jacobs et al., 2018; Oluyisola et al., 2020). Based on long-term sales forecasts, production planning progressively develops a detailed plan, setting targets from a broad strategic perspective to a specific material requirements plan. This plan considers not only raw materials but also the necessary machinery and workforce for scheduled product production. This aims to optimize machine and labor utilization, ensuring a balanced production output that accounts for available resources, capacities, and other relevant factors (Chapman, 2006; Lödding, 2016).

Production planning generally adopts a strategic or mid-term approach, while short-term production control is more tactical in nature. This tactical aspect typically begins with scheduling and dispatching after the order release. Both these stages are directly linked to actual production operations, having a direct impact on the efficiency and effectiveness of the production system. This involves making numerous decisions, but with limited decision-making values on the production process. The execution of the production plan aims to fulfill the objectives set by the planning phase. Given the dynamic nature of prevailing production systems, it is essential for this execution to possess sufficient robustness and adaptability to accommodate both planned

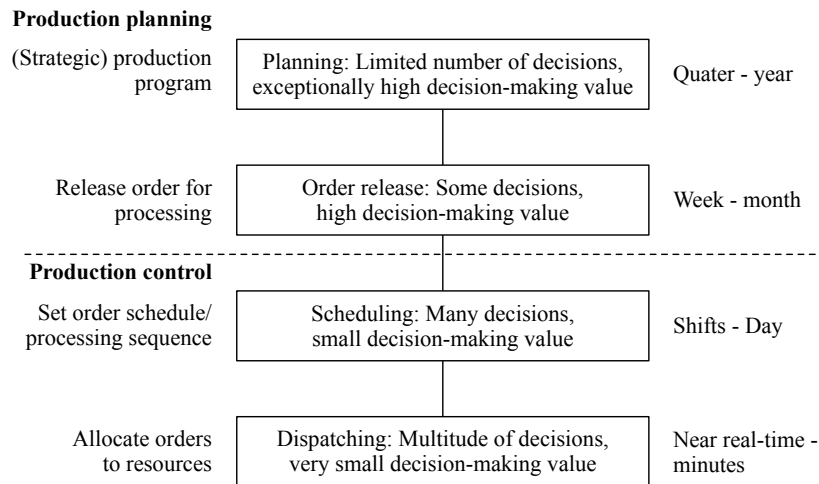


Figure 1.1 Production planning and control hierarchy, adapted from Mönch et al. (2013); Kuhnle (2020)

and unplanned events. Such events include planned machine maintenance but also exceptions like sudden machine breakdowns or the processing of unexpected rush orders (Chapman, 2006; Mönch et al., 2013; Schmidt and Nyhuis, 2021).

The early recognition and adaptation to such unexpected events and failures are crucial in minimizing stoppages and enhancing operational excellence. This need becomes more pressing with additional system loads resulting from continually shortening development cycles and escalating product complexities, including the shift towards fully customized products (Monostori et al., 2004; Sabadka et al., 2019). Moreover, the trend of shortening product life cycles not only encompasses new product introductions but also more customer-specific configurations, impacting production processes significantly. This trend towards a greater variety of product specifications is illustrated in Figure 1.2 (Koren, 2010). Processes altered by the observed increase in the amount of personalized products require quick and resilient integration. In response, robust strategies are essential for optimizing key performance indicators. These indicators span beyond quantitative metrics like throughput times or tardiness. They increasingly encompass quality standards, customer satisfaction, and sustainability factors (Ghobakhloo, 2020). To satisfy these arising demands and meet emerging challenges, advanced production technologies within the *Industry 4.0* framework have been increasingly deployed in recent years to realize flexible and adaptive production systems (Kagermann et al., 2013; Zheng et al., 2020; Marcucci et al., 2022).

Industry 4.0, also known as the fourth industrial revolution, refers to the integration of advanced technologies such as artificial intelligence, *Internet of Things*, and advanced automation mechanisms into manufacturing and production processes (Kagermann et al., 2013). Although *Industry 4.0* has the potential to significantly increase efficiency and productivity (Waschneck et al., 2017; Peres et al., 2020), it also poses challenges in terms of data-efficient processing and the

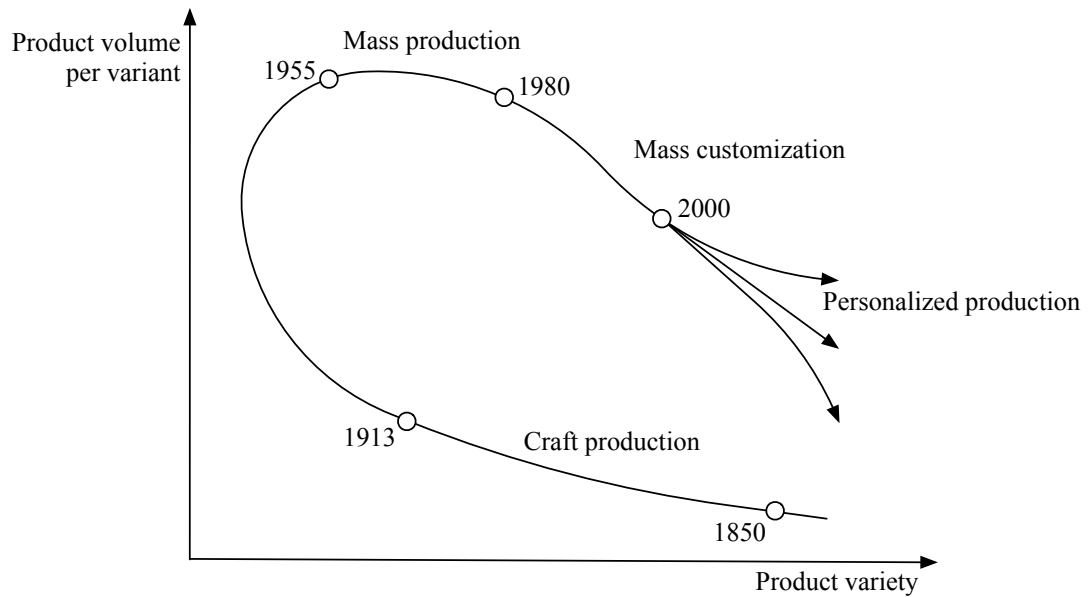


Figure 1.2 Trend towards higher product variety, adapted from Koren (2010)

complexity and transferability of the approaches into practical concepts (Schuh et al., 2017; Adadi, 2021; Alzubaidi et al., 2021). The progressive collection and accumulation of raw data and the limitations of conventional algorithms in dealing with large data sets have led to an intense discussion about the importance of data and the development of more data-efficient algorithms (Fan et al., 2014; Adadi, 2021). This is particularly relevant given the ever-increasing amounts of generated data. Consequently, production planning and control approaches face the challenge of processing this data efficiently to use it for informed decision-making processes and data-driven manufacturing analytics solutions (Schwab, 2016; Zhong et al., 2017; Tao et al., 2018).

A promising response to these data-centric challenges in production optimization involves employing advanced artificial intelligence technologies that support continuous learning and adaptation in planning and control systems. As early as 1996, Liu and Dong recognized the potential of using machine learning and artificial neural networks in production for its significant impact. Subsequent research by scholars such as Kádár et al. (2003); Mahadevan and Theocharous (1998); Kang et al. (2020) explored various approaches, resulting in improvements in performance indicators like inventory management and throughput time reduction. By analyzing the collected data from sensors and actors, an intelligent planning and control system can discover patterns and trends that may not be immediately apparent, leveraging this insight to make well-informed decisions that enhance overall system performance (Zhang et al., 2021).

1.1 Motivation

Modern production environments are characterized by increasingly dynamic and cross-system interrelationships. As ElMaraghy et al. (2009) point out, these dynamics with inherent process uncertainties lead to increased complexities in production. In this context, it is also important to consider the correlation between product and production complexity as described by Wiendahl and Scholtissek (1994). The increasing complexity of individualized products and inter-connected processes poses new challenges, especially with regard to decision-making. In order to maintain stable production systems and to avoid order backlogs or queues, production systems require efficient decision-making that must extend to real time.

The production control itself is directly connected to the resources on the shop-floor and manages the system participants through direct interaction and feedback. In order to handle the system dynamics and to guarantee learning capabilities, the control system must be able to adapt continuously to internal and external circumstances to maximize the production efficiency, without large ongoing efforts. In addition to the system complexity, the control design must be able to balance a large number of input and output parameters, even in high-dimensional solution spaces, to stabilize and optimize predefined key performance parameters. These parameters must not be static but, like the system itself, can be dynamic and must be defined and evaluated context-specifically (Nyhuis, 2008; Schwartz, 2014; Zhong et al., 2017).

The use of conventional control approaches, such as dispatching rules, encounters difficulties when dealing with the aforementioned dynamics or handling large amounts of data (Csáji et al., 2006). As such, they may not be able to cope with future flexible manufacturing processes or require high initial and ongoing implementation and maintenance efforts (as described by Zhou et al., 2020). They tend to be hard coded and require humans as supervisors to cope with system dynamics and to conduct online parameter optimization. They also often assume static environments in which not only ongoing processes are defined in advance, but also the information about the production environment is fully known (Luo, 2020). The reduced ability to incorporate randomly occurring and unplanned events, such as machine failures, often results in a discrepancy between the actual and the intended planned production state (Schneeweiss, 2003), which must then be either reacted to or predicted beforehand (Ouelhadj and Petrovic, 2009). To address these challenges, online reaction and optimization techniques, benefiting significantly from advancements in machine learning, are increasingly being applied to cope with the inherent dynamics of manufacturing systems (Arinez et al., 2020; Bertolini et al., 2021). Machine learning, encompassing a diverse array of algorithms, includes reinforcement learning, which stands out due to its interactive and real-time operation capabilities, offering extensive prospects for online optimization (Dey, 2016; Sutton and Barto, 2017). As early as 1998, Mahadevan and Theocharous demonstrated the efficacy of reinforcement learning in inventory

minimization compared to traditional Kanban systems.

Recently, the focus shifted towards deep learning, a specialized subset of machine learning characterized by the use of multi-layer neural networks (Alzubaidi et al., 2021). This shift is partly motivated by its successes in the Atari environments by van Hasselt et al. (2016) and Google DeepMind's AlphaGo (Silver et al., 2017). Deep learning enables the efficient processing of large amounts of data and the implementation of situational actions. Despite the potential advantages of deep learning over conventional heuristic approaches, its use has so far been limited to certain fields of application. Especially in production control, there is a need for deep learning based approaches to leverage adaptability and robustness (Arinez et al., 2020; Peres et al., 2020). However, the lack of interpretability and high computational complexities of deep learning in production continue to pose challenges that need to be overcome (Malhan and Gupta, 2023). Against this background, the question arises as to how the increasing complexity of optimization and the increased requirements in production can be effectively managed in practice using data-centric and machine learning based approaches. In production control, which is closely linked to production resources and demands quick decision-making, the inability to undertake exhaustive re-planning after every system change thereby presents a remaining challenge (Bueno et al., 2020; Ghaleb et al., 2020).

Given this challenge, it becomes important to explore how the escalating optimization complexity, often too demanding for non-linear optimization methods, can be effectively addressed using data-centric and machine learning based approaches (Bottou et al., 2018). In particular, the division of complexity into restricted and problem-oriented segments is becoming increasingly important for neural networks in order to effectively manage their scope and structure (Amer and Maul, 2019). Even with neural networks, the computational effort increases exponentially with the amount of input data, accompanied by a growing state space that represents the production at a certain point in time and grows with each new input value. According to Bellman (1957), this is called the *Curse of Dimensionality* in the field of multi-dimensional optimization problems and raises the question of a meaningful use and processing of data. It not only requires a reduction of the data points to be processed but also a structural analysis of the problem. By splitting the overall problem into comprehensible (neural) sub-fragments or modules, that are limited in scale, the overall complexity might be reduced and made more manageable (Maulana et al., 2015; Amer and Maul, 2019).

Such decomposition of problems or optimization schemes was already analyzed in a variety of applications, including energy systems (Kotzur et al., 2021), product development for part design optimization in additive manufacturing (Oh et al., 2018), as well as in the scheduling domain (Hu et al., 2021; Yonaga et al., 2022). Specifically in scheduling, the multi-objective evolutionary algorithm is widely deployed because of its high computational efficiency (Huang et al., 2021).

It divides an optimization problem into several discrete optimization sub-problems (Zheng et al., 2018). This approach was applied in flow-shop scheduling (Wang et al., 2021a) and milk-run scheduling (Zhou and Zhao, 2022), and was able to outperform existing scheduling benchmarks. Nevertheless, an optimal decomposition, even if ideally performed in equal-sized work packages, cannot be achieved in polynomial time. This constrains its application in real-time environments, especially for large-scale optimization problems (Kotzur et al., 2021). However, an approach that breaks down complexity and offers a sufficiently high degree of autonomy promises to increase system performances (Philipp et al., 2007; Bendul and Blunck, 2019).

The motivation of this thesis is therefore grounded on the following three main focus areas.

- First, this thesis examines the potential of deep learning in production, aiming to structure the research field and identify existing limitations. The emphasis is on deep reinforcement learning, which combines the capabilities of neural networks with the interactive and responsive reinforcement learning technique. While (deep) reinforcement learning is a significant focus field, the analysis also encompasses further deep learning methodologies. The findings from this analysis will drive the development of the problem-oriented artifact.
- Second, the thesis addresses the need for adaptive production control mechanisms. This also aims to achieve a high degree of generalizability. The motivation is to tackle large-scale problems through improved scalability and efficient use of a foundational knowledge base.
- Third, the thesis aims on reducing the optimization complexity of current deep learning based control approaches. It conceptualizes a framework that shall be based on multiple pillars to deconstruct the overall problem complexity into smaller, more manageable segments. This encompasses both the theoretical exploration and the practical application of deep learning in production control, a smart factory domain that has not yet been extensively explored or reviewed.

1.2 Research questions and objective

The development of the desired artifact with its comprehensively integrated knowledge base should not only enable the step towards a data-driven smart factory as described by Zhong et al. (2017) or Zhou et al. (2018), but also serve as a basis for further research. In this regard, the planning of a research project is the first step to enable the creation of structural knowledge and is initially reflected in the formulation of research questions (Armstrong et al., 2011; Hunt et al., 2018). The process of formulating adequate research questions not only facilitates the identification of attractive research topics and the guaranteed yield of scientific insights but also serves to define the scope of research under consideration (Armstrong et al., 2011). This also

includes the omission of possible sub-areas to focus on the essential core of the research work. Building on the initial motivation and introductory section, this thesis primarily focuses on the continuous optimization of predefined key performance indicators within complex production systems through novel data-driven and autonomous control approaches. Such approaches might leverage machine learning techniques, including deep learning, to effectively optimize control- and data-intensive processes. The aim is to overcome the static limitations that are inherent in conventional approaches while ensuring adequate learning ability and adaptability for various applications. This might also involve integrating principles of autonomous agents, decentralized decision-making, and the implementation of an adaptive organization. Another important aspect to emphasize is that the artifact to be developed shall not resemble a common black box model. Rather, it should enable comprehensible decision-making, which can increase user acceptance. This leads to the central research question, which will be detailed further on.

How can a data-driven and autonomous control optimization be designed for adaptive production systems?

From the primary research question, sub-questions can be derived for better fragmentation of the research field and structuring of the thesis. A thorough exposition of these in-depth research questions and an elaboration of the resulting artifact requirements is provided in Chapter 6. Based on an extensive literature review, not only the exact research gap must be defined, but an identification and classification of existing approaches and research streams should be made. This facilitates the deduction of direct requirements and design indications for the research intention, whether they are algorithm-, optimization-, or application-centered. In general terms, the approach is not only intended to enable robust production but also to be adaptive in each differing dimension.

The aspects of adaptability that must be addressed in the further course of this thesis can be interpreted in a context-specific way, but they can be attributed to the basic concept of a production environment that has to respond to internal and external requirements and disruptions. These can be classified into separate sub-dimensions, which encompass a range of decisive forces that influence production processes. Such forces include fluctuating procurement and sales markets, a changing volume and type of products to be produced, modifications of particular product specifications due to changing customer requirements, or the replacement of a product in the portfolio, among many others (Simpson et al., 2006; ElMaraghy et al., 2021). It is therefore necessary to consider criteria for process-, product-, technology-, and market-specific adaptability. Its utmost expression manifests itself in extreme adaptability, which, without preventive measures to increase resilience and robustness, would cause significant impairments in the processes in case of major deviations in the aforementioned criteria. Resilience in this context refers to the ability to prevent and handle disturbances, while robustness describes the ability to cope with internal

disturbances without system adaptations (ElMaraghy et al., 2021). Adaptability, in aggregate terms, can be defined through attributes such as system modularity, scalability, compatibility, and universality, as suggested by Heger (2007), Wiendahl et al. (2015), and ElMaraghy and Wiendahl (2019). Wiendahl et al. (2015) and ElMaraghy and Wiendahl (2019) additionally emphasize system mobility, while Heger (2007) includes system collaboration in the definition. In particular, modularity proves to be a pivotal adaptability factor, as it imposes the requirement of independent modules that can be configured and inter-connected as needed. This modularity affects the other characteristics mentioned and promotes adaptability overall, as discussed by Simpson et al. (2006) or Caesar et al. (2019). Thereby, it is important to make a clear distinction between production flexibility and adaptability, which is illustrated in Figure 1.3. Flexibility refers to the ability to change capacitive indicators without system modifications, such as through changed routing, scheduling, or augmentation, as discussed in VDI (2017) and Bitsch and Senjic (2022).

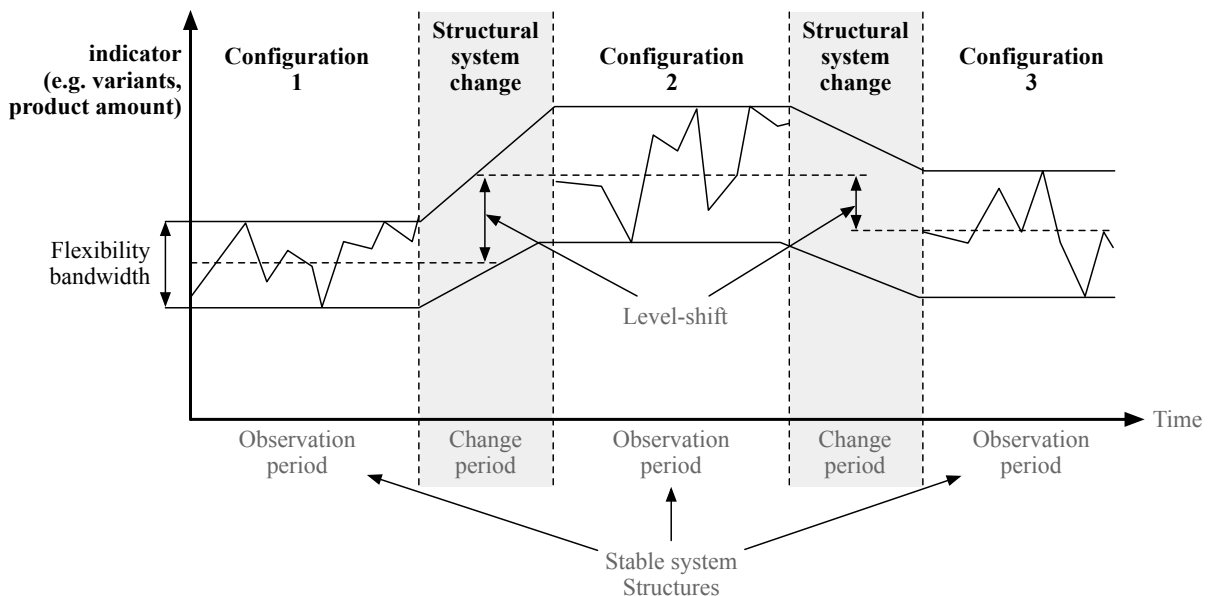


Figure 1.3 System adaptability and flexibility measures, adapted from VDI (2017); Bitsch and Senjic (2022)

In the remainder of this thesis, one focus will be on adaptability, emphasizing that the artifact to be developed should not only be designed for a specific environment. An additional focus will be on generalizability, particularly concerning the initialization, training, and re-use of potential autonomous elements. These elements should not only operate in defined environments but also enable knowledge transfer and long-term learning. A feasible deep learning approach, which has often been characterized by its ability to map complex control principles in network structures, might support the control of flexible production processes in various scenarios. The inherent adaptability of deep learning might also be decisive to autonomously react to changing system parameters and thus strengthen system robustness. In addition, the scalability and aforementioned

attributes offer extended possibilities for operation and evaluation on a larger scale and also in other problem domains. The research objective thus extends beyond the scenarios and production environments addressed in this thesis and encompasses the ability to adapt sensitively to changing production conditions. This supports the development of a comprehensively adaptable artifact. However, prior to designing such an artifact, it's necessary to define the design requirements, building upon existing research approaches. This step aims to identify strategies to mitigate overall problem complexity, enabling the realization of the adaptable artifact. Collectively, the outlined research problem leads to the formulation of several sub-research questions (S-RQ), guiding the subsequent investigations and explorations.

- **S-RQ1:** What requirements do production systems impose on deep learning based control optimization methodologies?
- **S-RQ2:** How can the decision-making and optimization complexity of large systems be distributed among autonomous system components?
- **S-RQ3:** How can a high level of control generalizability be ensured across varying production scenarios?

By answering the formulated sub-research questions above, the cornerstones for the design and implementation of the artifact are defined. Based on this, the artifact is constructed in an iterative development process. This not only addresses the artifact creation but also its continuous optimization. Subsequently, in order to evaluate the success of the research process, not only quantitative metrics must be analyzed, but the artifact should satisfy additional customer- and process-related target criteria. The production planning and control design framework of Bendul and Blunck (2019) summarizes essential design criteria in Figure 1.4, which can likewise be conceived as objectives and have to be met by the artifact accordingly. In general, Bendul and Blunck distinguishes between design-, scheduling-, and control-related parameters. From this, direct indicators for the formulation of the research objectives can be derived within the production planning and control domain.

As discussed above, deep learning methodologies provide viable tools to achieve a high degree of automation and to facilitate data management in processes with high order loads. Based on the defined research questions and motivation, the following general research objectives according to Bendul and Blunck (2019) shall be addressed. Beginning with the design criteria at the top of Figure 1.4, the operations baseline is established, and basic organizational decisions are set. The green marked lines indicate the design objectives defined for this thesis. First, within the realm of operational flexibility, the suitability of a deep learning based control framework should be evaluated in terms of handling multiple path alternatives and its proficiency in managing a medium level of structural complexity. Second, in light of intricate production requisites, the prevailing complexity should be reduced through appropriate decomposition strategies. To

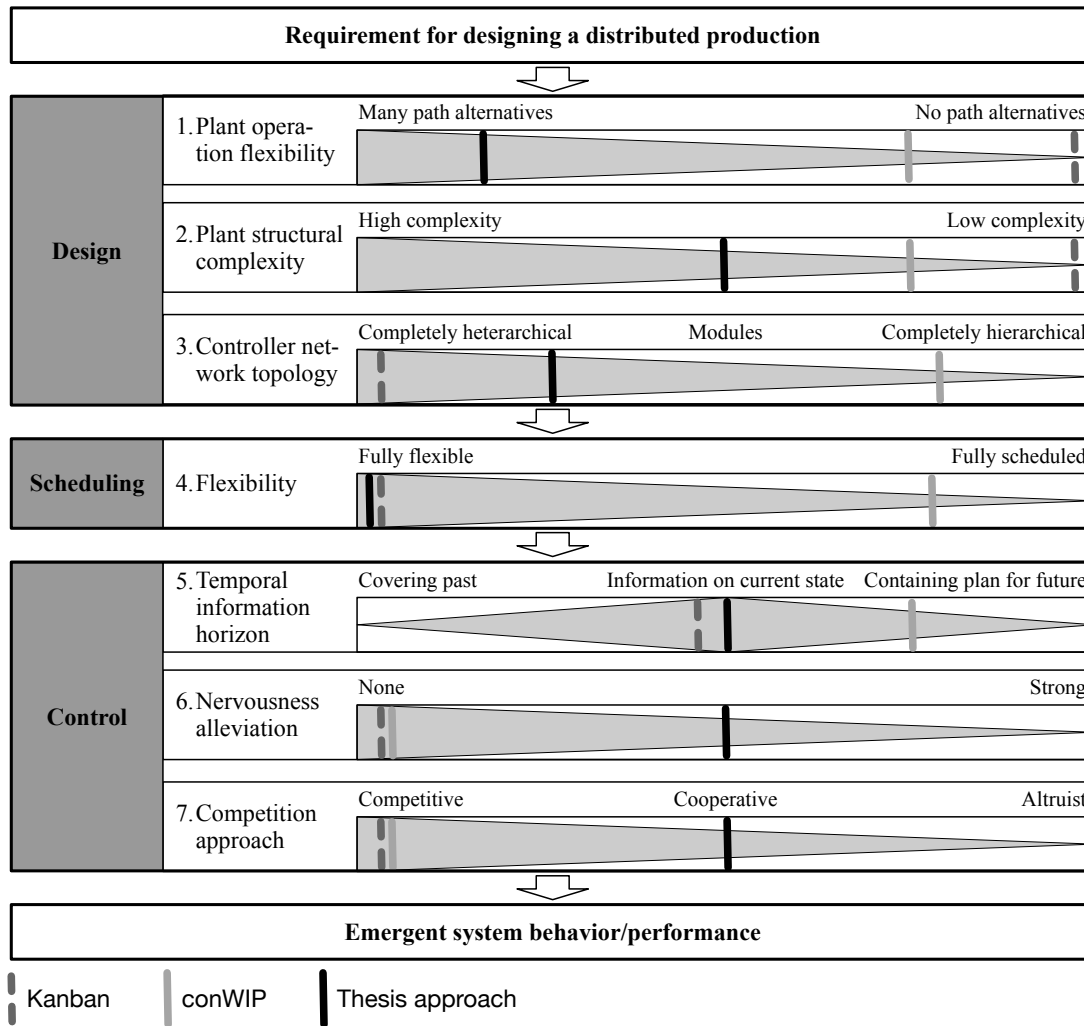


Figure 1.4 Production planning and control design framework, adapted from Bendul and Blunck (2019)

facilitate this reduction in structural complexity, the adoption of sophisticated matrix, modular, or more advanced production models is targeted as proposed by Scholz-Reiter et al. (2011). These models, thirdly, pave the way for inter-agent communication both within and across all system layers. Lastly, as the design aims to refine production metrics via a deep learning-centered approach, it's imperative that the agents exhibit an adequate level of operational flexibility.

The control portion of the framework in Figure 1.4 determines the behavior and decision logic of the agents and the artifact as a whole. The pursued control approach should make decisions based on current information as a representation of the current production state and should be able to react flexibly to unplanned incidents (5.). Past information should be integrated into the decision logic as experience data to facilitate a sufficiently high degree of system learning behavior. The artifact should have an average nervousness and not slip into chaotic states. Nevertheless, decisions should be revisable on the basis of new information (6.). Finally, the agents should reach global objectives and avoid local optima (7.). A detailed framework assessment, along

with the derivation of feasible design requirements, is available in the transition (see Chapter 6).

In summary, the three main objectives of this thesis are listed below.

- Objective 1: Design of a performant and adaptive deep learning based production control framework
- Objective 2: Enable a flexible key performance parameter optimization within a broad range of production scenarios
- Objective 3: Facilitate a sufficiently high degree of scalability and generalizability to allow coping with multi-level production systems

1.3 Research methodology

The following section provides an introduction to the *Design Science Research Methodology (DSRM)* by Peffers et al. (2007), which is adopted in the further course of this thesis. The *DSRM* methodology seeks to structure and optimize the development of scientific-technological artifacts for their ability to solve practical problems. In this context, an artifact describes a technically developed solution for a constrained problem. The approach is application-oriented and describes, on the one hand, how the solution for this problem is developed and, on the other hand, how it is tested in order to validate and iteratively optimize its impact on the given environment and problem. The methodology in Figure 1.5 clarifies precisely these steps as well as the associated iterative loops and refers to the respective chapters in this thesis that focus on a corresponding step.

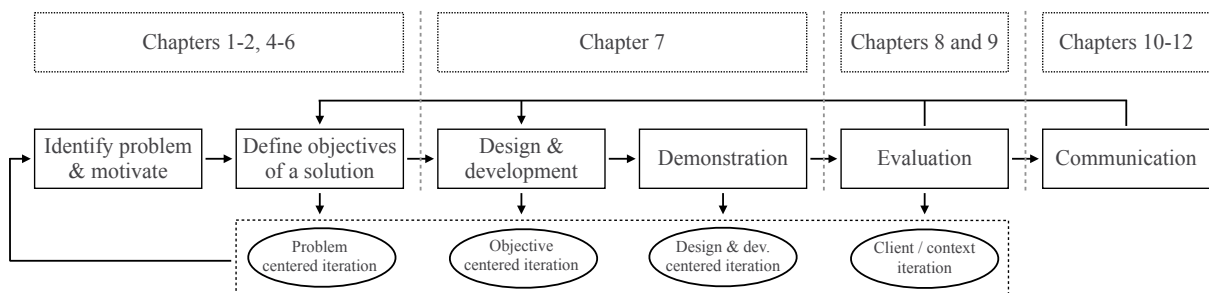


Figure 1.5 Pursued design science research methodology (Peffers et al., 2007)

Concerning this thesis and the approach outlined in Figure 1.5, the process is initiated by the identification and definition of the research problem and the accompanying motivation to highlight the relevance of finding a solution for this specific problem. This step provides a detailed definition of the research gap to derive corresponding solution design criteria. Based on the composed design criteria and requirements, the artifact can be designed and developed to be tested in the defined scope for its ability to solve the problem and to evaluate the degree of

objective fulfillment. During the evaluation phase, a corresponding iteration back to the design and development phase can be carried out if objectives were not met or further optimization potentials were recognized. Finally, the results are published in subject-related journals or similar media to disseminate the findings.

In Hevner et al. (2004), seven further guidelines summarize the objectives of a design science research procedure, which will be referred to in the further course of this thesis and are listed in Table 1.1. In the context of a problem-solving process, these serve as overarching objectives to ensure coherent pursuit of the conducted research. The right column of the table lists the corresponding equivalents from this thesis, outlining how the guidelines and associated objectives will be accomplished and materialized. Thereby, Hevner et al. (2004) addresses central elements of the design science research process, from the actual development (1), through the determination of the problem relevance (2) and its evaluation (3), to the assessment of the scientific contribution (4) and its methodological underpinning (5). Just as in the *DSRM* model, this process is iteratively performed (6) and is subsequently communicated in a broad and discipline-non-specific manner (7).

Guideline	Objective	Consideration in this thesis
(1) Design as an artifact	A functional artifact (e.g., model or construct) should be the outcome.	In this thesis, a functional and transferable control model will be developed.
(2) Problem relevance	The artifact should address a relevant problem with a tech-based solution.	The relevant problem will be identified based on a comprehensive theoretical analysis and addressed using an innovative control approach.
(3) Design evaluation	An evaluation must be conducted to demonstrate its quality, efficacy and utility within the considered scope.	The performance metrics will be assessed in simulated scenarios, and the control will be transitioned to a real-world testing.
(4) Research contributions	The developed artifact must offer a significant contribution in the respective field of research.	The artifact will address various production challenges, demonstrating requisite adaptability, thereby reducing optimization complexity.
(5) Research rigor	The design and validation must be conducted through the use of profound research methods.	Overarching methodologies will be used, with sub-methodologies employed for reviewing, taxonomy formulation, and evaluation.
(6) Design as a search process	The search for an appropriate solution is iterative and requires well-founded search strategies.	Comprehensive reviews and simulations enable iterative evaluations and refinement of research strategies and outcomes.
(7) Communication of research	Not only specialized scholars should be addressed through out the research process, but also business-oriented ones.	In addition to technical parameters, financial metrics will be considered and managerial insights are given.

Table 1.1 Research guidelines for design science by Hevner et al. (2004) and coverage in this thesis

The described guidelines are conceived by Hevner (2007), which iteratively links the interrelationships and emerging requirements between practice, research, and the considered problem and its solution finding through three cycles. The design and development iteration, which was described by Peffers et al. (2007) is iteratively addressed by Hevner (2007), who integrates it as a *design cycle* in his information systems research framework. It serves as the inter-connective

part of consistent evaluation and corresponding follow-up development, and vice versa. The other successive iterations (see Figure 1.6), represent the *relevance cycle* and the *rigor cycle*, which connect the design science research process with the practical environment and existing knowledge base, respectively.

The central element of the design science research box in Figure 1.6 includes the *design cycle*, which evaluates the artifact in terms of its objective fulfillment while constantly deriving research and design needs. Based on the defined needs as well as the findings of empirical testing, the artifact is continuously adapted. The *relevance cycle* takes into account the practical and business-related scope of the research problem and considers the practical and business-related subject matter of the research problem. By defining the external technical requirements, the involved actors as well as the affected organizational structures, the application context is specified. Through the *relevance cycle*, which can be supported by simulations or real-world transfers, fundamental improvements in production performance shall be obtained in a pre-defined environment to provide a significant added value for practical applications. The *rigor cycle*, on the other hand, ensures the scientific integration of the artifact and knowledge into the targeted research field, where the acquired knowledge can be abstracted and disseminated, parallel to the final methodological step in Peffers et al. (2007). This can be reflected in a comprehensive approach that combines algorithmic and organizational methods and provides innovative perspectives on how to deal with optimization complexity. This can be embodied in a tool, taxonomy, or equivalent framework that will support and guide future research fields.

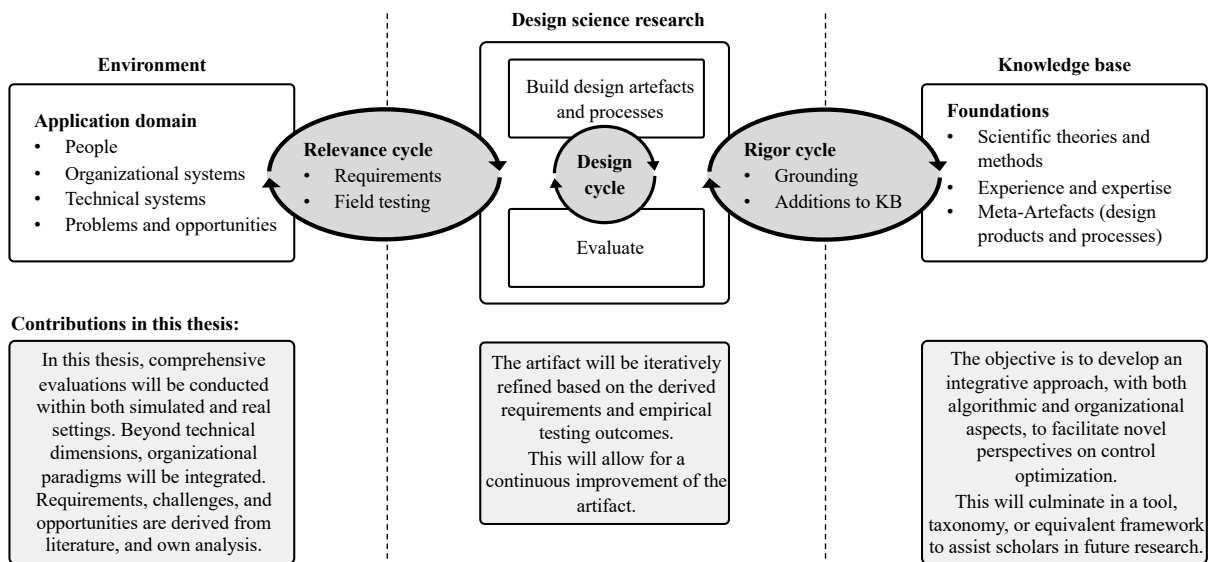


Figure 1.6 Pursued methodology for the scientific approach (Hevner, 2007)

1.4 Thesis structure

In this chapter, the motivation and the fundamental methodology are outlined and the intended research objectives were briefly presented. The subsequent chapters follow the structure outlined in Figure 1.7. Chapter 2 outlines the essential principles of prevailing production systems and research, introducing a unified approach to deep learning based optimization strategies. It also describes two initial research gaps, which are expanded upon in subsequent chapters. Chapter 3 provides a cumulative exposition of conducted research and discusses the contents of the underlying publications of this thesis. This comprises the more specific methodological foundations that constitute to an efficient and streamlined research process. The first two publications in 4 and 5 examine the current state of research on deep learning based production systems and seek to identify in-depth algorithmic and organizational requirements for the artifact construction, adhering to the DSRM approach. In doing so, application domains, algorithms, and a taxonomy for classifying deep learning approaches are presented.

In Chapter 6, based on the reviews and developed taxonomy, the specific research gap is highlighted and design requirements for artifact construction are derived. Chapter 7 contains the third publication with the design and development of the deep learning based control framework. This includes the implemented control framework, the multi-agent interaction design, and the associated training framework. Chapter 8 discusses the fourth publication with its in-depth benchmarking results and a optimization robustness analysis. It further integrates the control framework into a hybrid test-bed. The last publication in Chapter 9 discusses techno-financial aspects, in particular the impact on revenues and profits, to evaluate its attractiveness from both a technical and economic perspective.

An integrated perspective, encompassing and extending beyond the scope of this thesis, is elaborated in the discussion in Chapter 10. In Chapter 11, the remaining research gaps are presented and a critical evaluation of the results is given. The thesis concludes in Chapter 12.

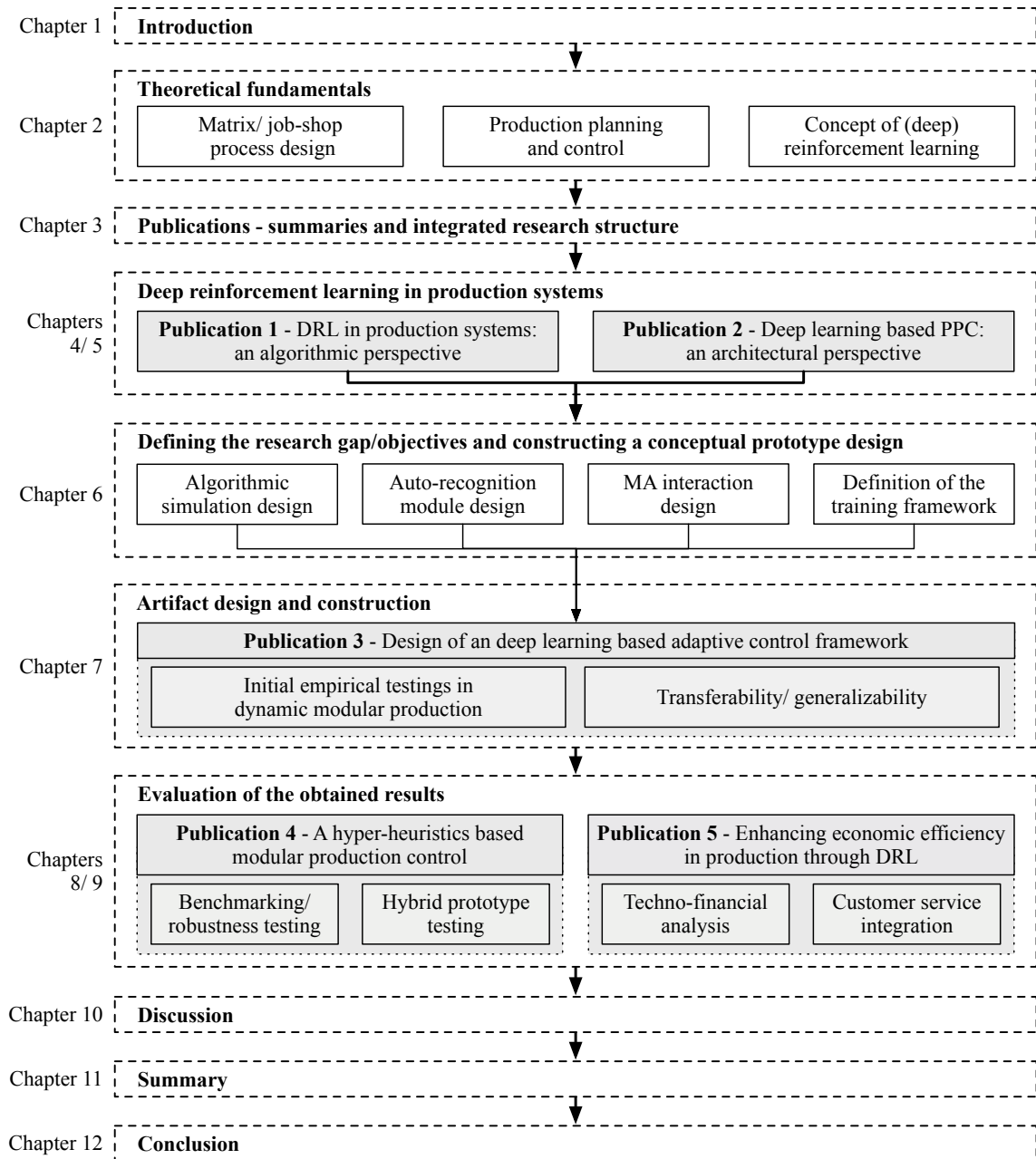


Figure 1.7 Thesis structure

2 Fundamentals

This chapter provides an introduction to the fundamental principles of production systems and their planning and control, presented in detail in Sections 2.1 and 2.2. Given the increasing complexity and dynamics in production systems, as discussed in the introduction, Section 2.3 focuses on the basic principles of machine learning. The use of these techniques seeks to facilitate robust and data-driven decision-making processes, which contributes significantly to the understanding of the artifact developed in later chapters. Finally, Section 2.4 identifies initial research gaps. This establishes the groundwork for formulating the central research problem of this thesis and sets the scope for the comprehensive reviews in Chapters 4 and 5. The structure of the chapter is illustrated in Figure 2.1.

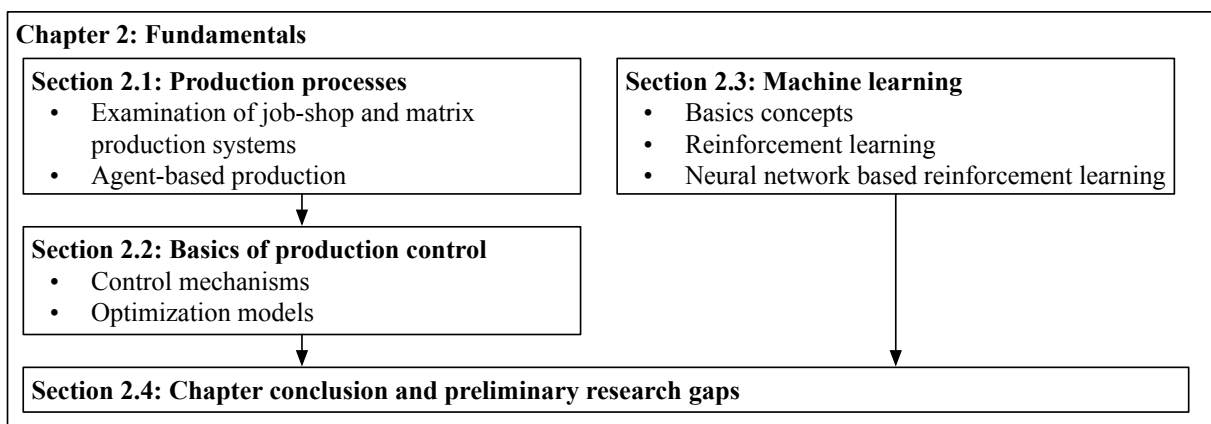


Figure 2.1 Structure of the fundamentals chapter

2.1 Production process design

Production processes can be differentiated with regard to various criteria and, depending on the exact specifications, not only have separate requirements but also necessitate their consideration in the design of production planning and control algorithms. Table 2.1 lists some of these process categories, which differ mainly in the volume of production and the variety of goods produced. According to Chapman (2006), this also results in a clear differentiation of the respective production workflow and the emerging skills of the workforce and management planning. In repetitive processes, equipment is adapted to highly specialized operations and manpower to standardized tasks of low complexity to achieve maximum efficiency at high volumes (Spencer and Cox, 1995). In contrast, job-shop processes require more general-purpose equipment and highly skilled staff to enable a wide range of processing types. A further differentiation of the manufacturing processes can be derived from the type of customer order processing. A job-shop is also suitable for engineer-to-order (ETO) or make-to-order (MTO) processes due to its generalized mode of operation, whereas repetitive processes often adhere to assembly-to-order or

make-to-stock procedures (Hayes and Wheelwright, 1984; Chapman, 2006; Helkiö and Tenhiälä, 2013).

	Job-shop processes	Batch processing	Repetitive processes
Equipment	General purpose	Semi-specialized	Highly-specialized
Labor skills	Highly skilled	Semi-skilled	Low skills
Managerial approach	Technical solver	Team leadership	Efficiency
Volume output	Low	Medium	High
Design variety	High	Medium	Low
Design environment	ETO, MTO	MTO, ATO, MTS	ATO, MTS
Flow of work	Variable, jumbled	More defined	Highly defined and fixed

Table 2.1 Production process categories, according to Chapman (2006)

Nowadays, decisions regarding the selection of the presented production forms become outdated in numerous industries. Despite significant throughputs in modern repetitive and serial production systems, there is an growing need to meet individual customer requirements and incorporate future product generations. Yet, this poses challenges for recurrent environments like repetitive and batch manufacturing systems (Helkiö and Tenhiälä, 2013; Hofmann and Knébel, 2013). It is this product individuality, where job shops can demonstrate their distinct advantages, offering a more flexible approach to meet these evolving demands.

2.1.1 Job-shop production processes

Figure 2.2 illustrates an example job-shop, where three distinct orders undergo processing across multiple shared machines, resulting in the production of three respective products. Another order, which would only require the already installed machines for basic processing, could simply be added to this job-shop. In serial production, this would not be possible without further efforts, since the machines are installed in a predefined sequence and thus structurally constrain the process. The idea of leveraging flexible job-shop properties with batch production designed for efficiency was already proposed by Browne et al. (1982).

The above-mentioned material flow design offers great flexibility in terms of the range of products that can be produced, as well as the possibility of adding or removing machines, or changing their arrangement to circumvent bottlenecks (Mason et al., 2002; Schmidtke et al., 2021). There is also the possibility of providing a machine redundancy to increase throughputs, but also to minimize the impact of machine breakdowns. On the other hand, the machines can be used for a wide range of operations due to their very broad functional base and the selection of the functionally optimal machine for a particular process step can be facilitated (Chapman, 2006).

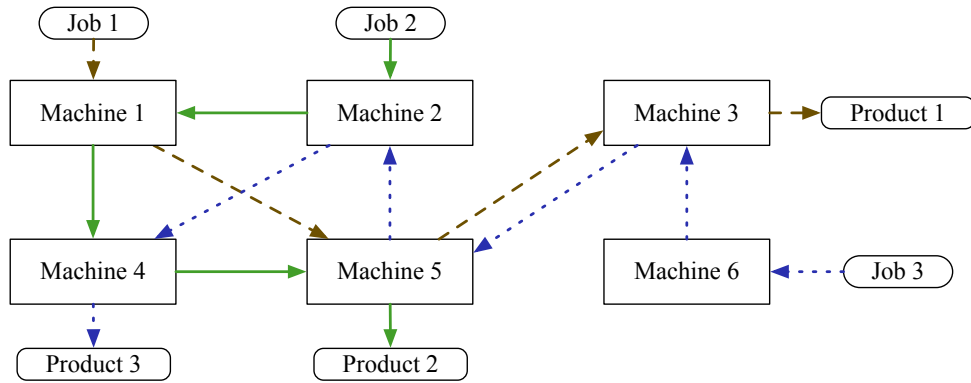


Figure 2.2 Exemplary job-shop process, adapted from Wang et al. (2021b)

On the other hand, the process-related flexibility results in a large number of possible routing and control options, which makes finding the optimum significantly more difficult with large layouts and often requires heuristic optimization algorithms. This also includes the consideration of tool changes, which can be very high in a job-shop with many machining options, as well as the consideration of transportation routes between processing steps. The complexity and arising system dynamics could thereby cause inefficient workflows and low utilization rates if production planning and control are not optimized and scaled to the actual system (Schenk et al., 2010; Liaqait et al., 2021; Schmidtke et al., 2021).

2.1.2 Matrix production systems

In recent years, particularly within deep learning research, emphasis has not only been placed on job-shop production but also to a limited extent on matrix-based production, due to its particularly flexible processing capabilities (Hofmann et al., 2020; Gankin et al., 2021). A major difference compared to the job-shop is the line independent flow control of the processes within the system, which results in arbitrary material flows with varying cycle times. This also circumvents a major disadvantage of sequential flow-shops, the unbalanced cycle-time due to blocking and downtime in multi-variant production lines (Schönemann et al., 2015). Instead of a fixed order sequence, jobs are released on a short-term basis in matrix production. With the concept of a cycle-independent flow, matrix production tries to combine the economic advantages of a classic flow production with the flexibility advantage of a job-shop production (Schenk et al., 2010; Schönemann et al., 2015; Greschke, 2016).

The matrix production concept is primarily characterized by its close-meshed logistics network, through which the workstations are interconnected (see Figure 2.3). The interconnected workstations provide the basis for the deployment of autonomous logistics devices that enable each order to follow its own individual processing path (Perwitz et al., 2022). However, the closed-meshed structure and cycle-independent concept would not prevent machines from being

2 Fundamentals

blocked, necessitating redundant resources. It might be reasonable to equip workstations with the same operations and capabilities to increase the availability of a workstation for processing an order. Simultaneously, the configuration of different machines within a workstation can significantly increase the range of functions and enable further processing without increased transportation times (Schönemann et al., 2015; Greschke, 2016).

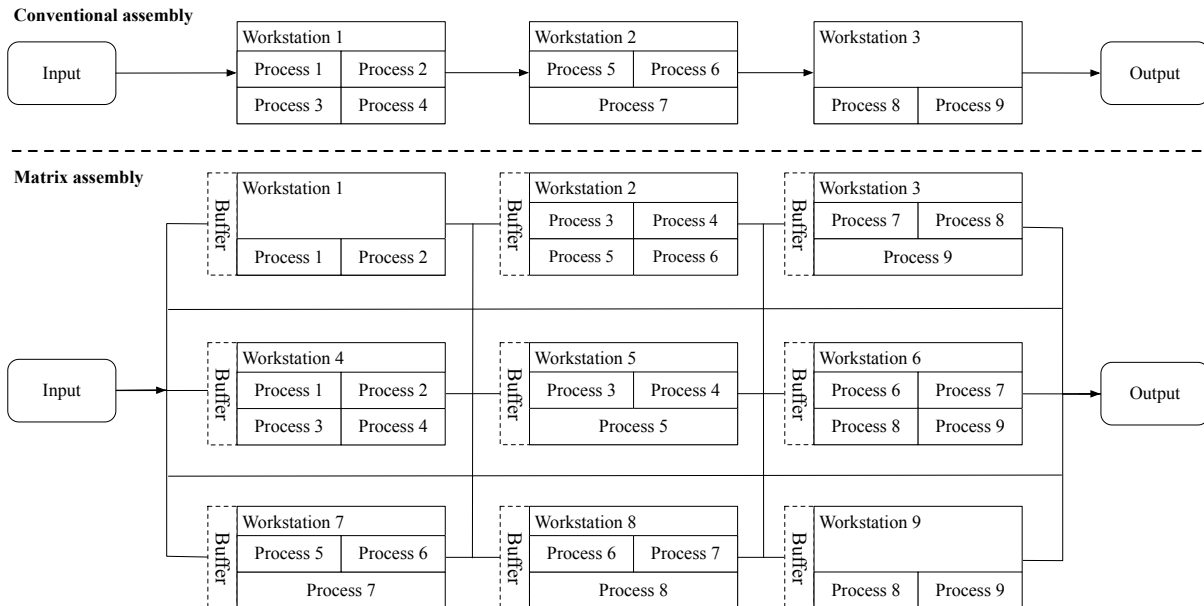


Figure 2.3 Comparison of a conventional and matrix assembly, adapted from Schönemann et al. (2015)

Figure 2.3 highlights the machine or process redundancy caused by duplicate process steps. The obvious redundancy in this case would have to be checked in real use-cases with regard to machine utilization. If appropriate, workstations and their allocation would need to be modified or re-organized. Due to the loose design of the overall system, this could be done by adding and removing workstations, thus guaranteeing adequate system scalability and resilience to react to system changes (Hofmann et al., 2019; Trierweiler et al., 2020). This makes it possible, for instance, to respond to fluctuating customer demands or to quickly incorporate individual process modifications (Greschke et al., 2014). While there are advantages in term of flexibility and adaptability, a matrix production necessitates high capital expenditures and can result in increased space consumption due to high routing requirements. In comparison to a job-shop, there's reduced order flexibility, even though it handles moderate order volumes. Standardization and streamlined processes provide a level of customization, but this comes at the cost of process synergies.

The presented job-shop and matrix concepts primarily highlight the structural and procedural aspects of production systems. A further dimension of flexibility and efficiency in modern production systems is introduced through the use of agent-based concepts. These represent another approach to leverage production logistics and offer a foundation to tackle modern

challenges of process automation (Karageorgos et al., 2003; Barbati et al., 2012).

2.1.3 Agent-based systems

Logistics resources, such as autonomous mobile robots, can move between workstations along predefined paths or even autonomously. Here, a production system is theoretically unlimited in the definition of the number of process participants. This raises the questions of how the individual participant of a system is, first, interacting with his environment and, second, which position he has in the overall production organization.

2.1.3.1 Production agents and environment

With an increasing availability of distributed computation resources, already Parunak et al. (1986) discussed the necessity of not transferring the decision-making process to a central authority, but to allocate the intelligence to the various participating production agents. A single agent can thereby be characterized by several properties. These are indicated in Figure 2.4 on the right and essentially comprise, related to the individual agent, its reasoning, perception of the environment, and type of actions (Balaji and Srinivasan, 2010). This thesis introduces an advanced control approach focused on data-driven, real-time decision-making. Thereby it's crucial for the agents to efficiently gather and process information from their environment, especially in the dynamic context of production control and shop-floor operations. This need is particularly evident for deep reinforcement learning controlled agents, where learning through direct interaction with the environment is fundamental (Zhang et al., 2022a). As these agents process information through neural networks, their data processing capabilities become increasingly complex (Sarang and Poullis, 2023). Consequently, the effective extraction and communication of relevant information is crucial to their performance.

Another relevant part is the type of action and interaction with its environment. Balaji and Srinivasan (2010) divide the interaction into the categories of communication and negotiation. Márkus et al. (1996), in contrast, consider the agent interaction in a production context and aims to differentiate between order and resource agents. Within the assembly domain, Seliger and Kruetzfeldt (1999) went further in an early approach and subdivided the overall structure or society of agents into manufacturing, transportation, and assembly agents. These were further divided into stock, control, and negotiation agents, which negotiate vertically to the next layer in each case and horizontally across negotiation agents. During execution, agents can thereby understand their environment and behavior, and adapt their behavior to pursue a single or multiple goals.

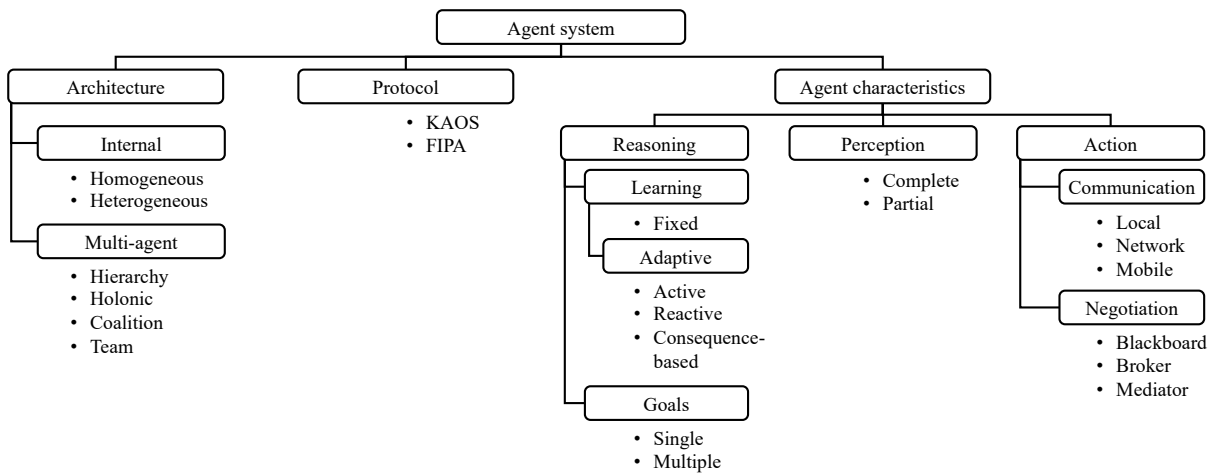


Figure 2.4 Classification of agent based systems, adapted from Balaji and Srinivasan (2010)

2.1.3.2 Multi-agent based production - organization

Whereas the communication of the individual agents was considered above, they can be organized in different structures within a multi-agent system. This is not just about the division of a task per se, such as the three-part split in Seliger and Kruezfeldt (1999), but about how an agent is positioned and operates within the overall system. In this context, Weiss (2001) particularly emphasizes the flexible and reconfigurable properties of such multi-agent based decentralized control structures. In a more recent review, Herrera et al. (2020) further outline the relevance of multi-agent systems for existing and planned real-world applications.

A specific differentiation of multi-agent systems is made by classifying them into hierarchical or heterarchical structures, depending on the allocation, grouping, and interaction of agents. While a hierarchy is characterized by a multitude of master-slave relationships, a heterarchy consists primarily of peer-level relationships (Baker, 1998; Bongaerts et al., 2000). As listed in Table 2.2, an intermediate option between the hierarchical and heterarchical systems can be reached through semi-heterarchical and/ or holonic systems (Sallez et al., 2010; Borangiu et al., 2015). For these intermediate control organizations, some approaches to multi-agent manufacturing systems have been proposed, that attempt to exploit the distributed capabilities locally and globally to leverage system performances. Examples are the holonic control approaches *PROSA* and *ADACOR* for distributed manufacturing systems (Van Brussel et al., 1998; Leitão and Restivo, 2006), *ADACOR-2*, a further development to enable dynamic configuration in online operation (Barbosa et al., 2015), or *Pollux*, whose structure is composed of a control and a reconfiguration mechanism to reach higher degrees of adaptation (Jimenez et al., 2017). The holonic concept, introduced by Koestler (1970), defines a holon as an entity that is both an independent whole and part of a larger system. These holons are characterized by a high degree of autonomy and can act both independently and in cooperation with other entities. In contrast, the semi-heterarchical approach

describes a control concept which combines hierarchical control elements with heterarchical flexibility, enabling both top-down management and peer-level collaboration (Sallez et al., 2010; Grassi et al., 2020). Later on in this thesis, a semi-heterarchical organisation is included in the control framework, which draws on the idea of autonomous agents inside of decentralized holons according to Fischer et al. (2003).

A detailed discussion and illustration of the concepts and their optimization potentials are presented in Table 2.2. It clarifies that hierarchical systems are particularly suitable for the optimization of long-term objectives due to their top-down planning and control capabilities. By the communication of high-level objectives, these can be fragmented, monitored and coordinated by the agents on the respective levels. In a fully heterarchical system, on the other hand, fragmentation of goals is only possible to a limited extent due to an high decision autonomy, up to the complete independence of the agents. This control design is more suitable for short-term optimization. Such a framework can be described as very reactive due to the autonomy of the operating agents but suffers from local optimization tendencies and myopic behavior due to the lack of master-slave relationships. A semi-heterarchical system seeks to combine the advantages of the two former organizations (Trentesaux, 2009; Sallez et al., 2010; Borangiu et al., 2015).

Aspect	Hierarchical system	Heterarchical system	Semi-heterarchical system
Optimization focus	Long-term objectives	Short-term objectives	Combines long- and short-term objectives
Planning and control	Top-down approach	Decentralized with high autonomy	Mixture of centralized and decentralized control
Goal fragments	High-level objectives are fragmented, monitored, and coordinated	Limited fragmentation due to high decision autonomy	Combines goal fragmentation and autonomous decision-making
System reactivity	Less reactive due to top-down approach	Highly reactive and resilient	Balances reactivity with structure
Optimization tendencies	More suited for stable, predictable systems	Autonomous responses in stochastic systems	Seeks to adapt to dynamic environments
Behavioral characteristics	Structured approach with coordinated efforts	Myopic behavior, local optimization tendencies	Attempts to mitigate local optimization

Table 2.2 Organization dependent optimization, according to Trentesaux (2009); Sallez et al. (2010); Borangiu et al. (2015)

In an early explanatory approach, the combination of hierarchical and heterarchical production control was considered by Bongaerts et al. (2000), with cooperating agents on different layers to ensure sufficient system flexibility. An approach that used semi-heterarchical features within a single control structure and to establish domain-wise clustering of them has been implemented by Borangiu et al. (2009, 2010) in the field of product-driven scheduling or by Rey et al. (2013) in the field of flexible manufacturing control. By using a 2-layered semi-heterarchical approach,

Borangiu et al. fulfilled different objective horizons and obtained a comparably higher resilience and agility of the system. Rey et al. (2013) was further able to facilitate a increased control level over the otherwise myopic behavior of the agents.

However, a reduction of system complexity is not only achieved by distributing it among several agents but also by a breakdown into task-related and structural units. In the fundamental study by Bertrand et al. (1990), it is described that complex manufacturing systems are typically hierarchically divided into the levels of flow control at the top level and detailed scheduling at the level of the production units. By the subdivision into structured units, the disadvantage is prevented that the individual agents resemble black-boxes and only optimize their associated subsystem (Monostori et al., 2006). The agents should rather be able to organize themselves in such a way that they reach goals together and optimize the targeted parameters iteratively by adaptation. This makes it necessary, however, that each agent, also on the lower levels, receives global state information to avoid local optima (Philipp et al., 2007).

2.1.4 Section conclusion

In this section, fundamental production structures of job-shop and matrix productions, along with the multi-agent systems and system organizations, are outlined. The presented approaches are associated with respective advantages and disadvantages regarding production volume and variety, which require a detailed evaluation to chose the right organizational model for a specific use-case. In selecting the later agent-based control approach for this thesis, it is necessary to address the requirements and objectives outlined in the introduction. These include exhibiting high adaptability and accommodating increased order volumes and product varieties.

The trend towards a wide variety of products, extending to mass individualization, presents a challenge due to decreasing batch sizes and volumes. A production system must be able to manufacture a broader range of similar products despite possible lower demands for these sub-products. This does not necessarily require new or additional machines, but flexible and adaptive process flows. Thereby, Hayes and Wheelwright (1984) point out the relationship between order variety and volume in Figure 2.5, while also considering the costs that would arise from deviations from standardized procedures. It should be possible to enable certain volumes in throughput without generating large additional efforts and costs. However, the matrix's top right and bottom left segments highlight the struggles, underscoring these as contemporary challenges.

With an increasing degree of freedom within the process flows, the planning and continuous control in job-shop and matrix production systems become significantly more complex due to the prevailing non-standardized processes and high modification rates. In multi-agent production systems, with multiple autonomous mobile robots, a further planning and control complexity increase can occur. In this context, efficient production planning and control concepts, including

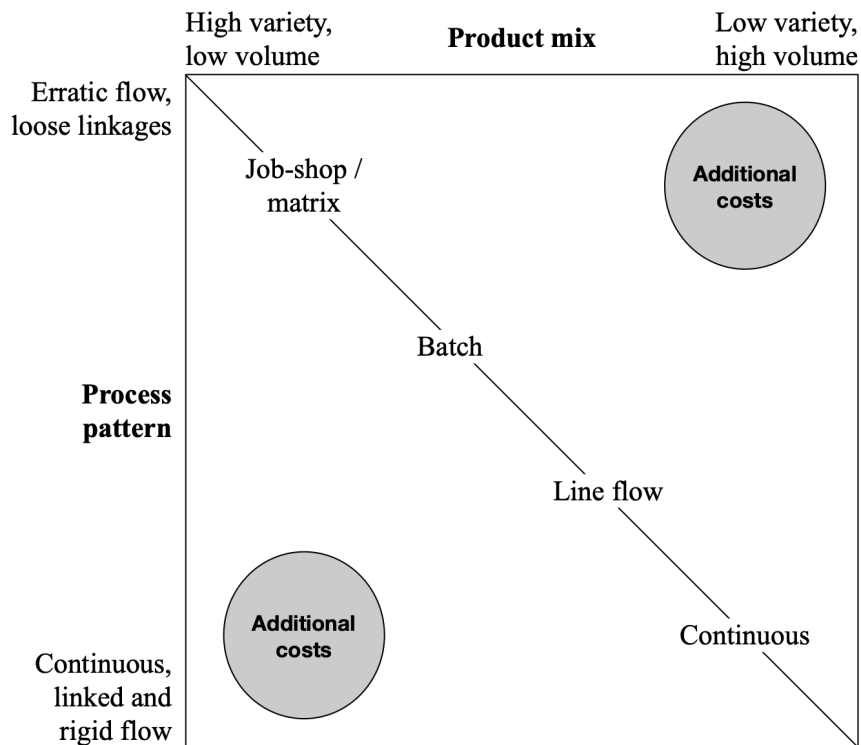


Figure 2.5 Hayes-Wheelwright Matrix, according to Hayes and Wheelwright (1984)

the use of multi-agent systems and advanced production organizations, can help in coping with the rising system complexity and the quickly shifting market conditions and customer demands.

2.2 Production planning and control

The last section was about production processes and forms of organization therein. Conversely, within such organizations, a large number of participants must be managed, which can make centralized or decentralized decisions, but are still expected to optimize the system as a whole. For the holistic integration of the subject matter of this thesis, the aspired production control approach is contextualized within the general production planning and control framework. This is where the widely used Aachener production planning and control model classifies the so-called in-plant planning and control within the core tasks of a factory or production, as indicated in Figure 2.6 (Luczak et al., 1998; Schuh and Kampker, 2012). In addition to the network tasks for overarching activities shown on the left in this figure and the cross-sectional tasks such as storage tasks shown on the right, core tasks also include strategic program planning, requirements, and procurement planning and control. The focus of this thesis is highlighted in grey, comprising the in-plant production planning and control domain.

Besides the Aachen model, the Y-model of Scheer (1997), and other conceptual frameworks, Chapman (2006) condensed in-depth production planning and control activities based on the

2 Fundamentals

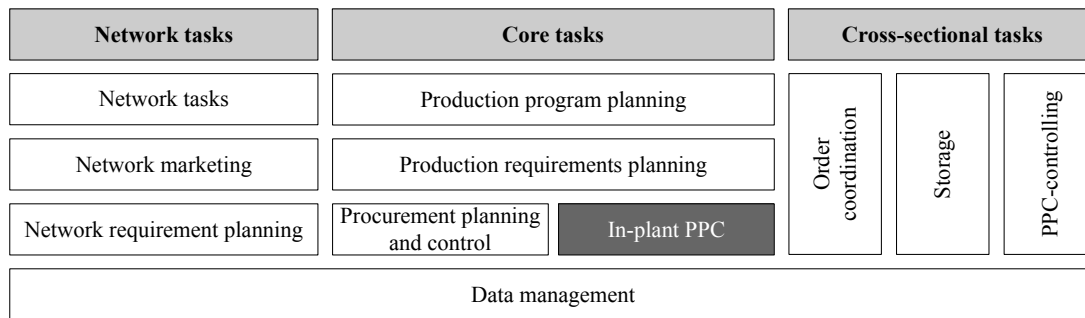


Figure 2.6 Aachen production planning and control model, adapted from Luczak et al. (1998); Schuh and Kampker (2012)

information flows. In this model, the determinants of demand and given resources are embedded as central influence parameters for production planning. Similar to the Aachen model, the flow of information begins with strategic production planning. Based on this, the plan is detailed to the mid-term generation of a master plan and material requirements planning (Chapman, 2006). The production control then proceeds with the execution of the established plans, typically after order release (Scherer, 1998; Lödding, 2016). This order release point also represents the transition from planning to executive tasks (Zäpfel, 2001). The production control comprises operational tasks involved on the shop floor regarding the processing of an order and includes scheduling, dispatching, and the ongoing monitoring and tracking of orders and inventory. It is intended to fulfill given plans and increase the actual throughput, but also to reduce costs and tardiness and maximize production efficiency (Bertrand et al., 1990; Chapman, 2006; Schuh et al., 2012).

In job-shop, matrix or other dynamic production systems, the given flexibility and loose process restrictions result in significantly more complex planning and shopfloor control optimization, if compared to pre-defined process flows. Thus, it is necessary to further classify such tasks within the control context and other preceding and subsequent tasks need to be considered in their mutual interaction with regard to their specific responsibility and functional scope.

2.2.1 Basic concepts of production control

For the purpose of reducing the overall system complexity and dividing it into manageable fragments, control problems are often structured into two levels. The top-level primarily comprises the overall flow of goods, whereas the lower level comprises detailed adjustments within the individual production units (Bertrand et al., 1990). The individual production control and the dynamic production planning positioned thereon can be described as reactive closed control loops as depicted in Figure 2.7. In these control loops the reference values derived from the respective objectives of the planning and the production control loop are compared with the actually reached values. If necessary, operational adjustments are made in case of deviations from the reference values (Pritschow and Wiendahl, 1995).

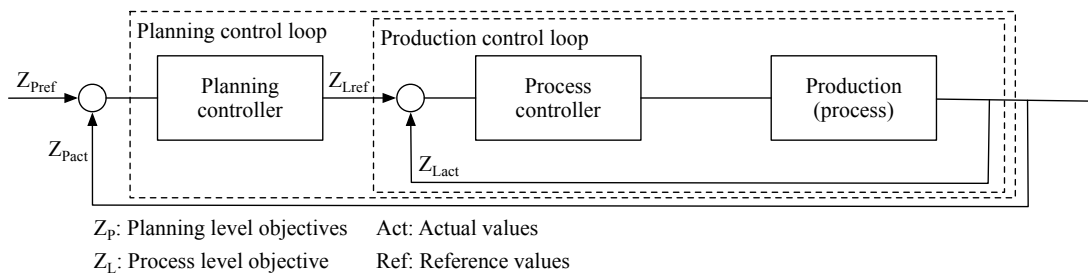


Figure 2.7 Closed loop production control model, adapted from Pritschow and Wiendahl (1995)

Given the conditions on the shop floor, the production control loop needs to be considerably faster than the planning loop. As already indicated in the introductory section and especially in Figure 1.1, this could mean a time-frame ranging from minutes to seconds, including real-time capabilities (Mönch et al., 2013). In the event of unforeseen circumstances, like a machine failure, the production control must have the capability to reallocate semi-finished orders to alternative options promptly, ensuring robust processes and order completion without delays (Pritschow and Wiendahl, 1995). The so-called dispatcher, which carries out production control tasks, has the closest range of impact to production and must react as first instance to process deviations. Thus negative effects for the total production should be contained by the dispatcher and options for the optimization of the current production should be sought. Likewise, the dispatcher receives immediate feedback about production, but not complete information about the total production state as it might be available to the scheduler (McKay and Wiers, 2003).

Considering these issues, it is crucial to define dispatching strategies that improve the decision making accuracy in production control. As such, both combinatorial and continuous approaches, including linear and non-linear methods, play a central role. Also they can be further distinguished into linear and non-linear approaches, which also including methods such as gradient-based search or simple methods. With the combinatorial methods, a distinction is made between exact (such as branch-and-bound, or dynamic programming) and approximate methods (Baker, 1998; Reddy and Nagesh Kumar, 2020). An example of an exact method is the application of mathematical optimization to identify the optimal solution for a problem. However, these problems are often NP-hard, leading to an exponential increase in computational efforts as the problem size grows (Klemmt et al., 2009). In contrast to heuristics, this can result in significantly longer computation times to generate an executable action, hindering the ability to respond in real-time. This has already been countered by decomposition methods, but these do not necessarily promise a fast finding of an optimal solution due to the prior splitting. In addition, such methods have difficulty dealing with discrete event parameters and with multi-variable environments (Coello et al., 2007). For these reasons, in-depth consideration will not be given to mathematical optimization. Instead, the further focus will be on heuristics, which may facilitate near-optimal decision-making and higher generalizability in operations (Fuchigami et al., 2018;

Kallestad et al., 2023).

2.2.2 Heuristics control and optimization strategies

A heuristic defines a methodology to find an analytical and acceptable solution to a problem despite an incomplete information set and limited system knowledge as well as a short time horizon. In this context, the priority-rule heuristic serves as a baseline for problem-solving, which, however, does not guarantee an optimal solution to be found. Rather, the advantage resides in finding fast and feasible solutions even for large-scale problems, for which mathematical optimization and other modelling methods would require significantly increased computation times (Kolisch and Hartmann, 1999; Klemmt et al., 2009). The following sections begin with an explanation of the basic heuristic approaches based on priority rules in Section 2.2.2.1. Subsequently, meta-heuristic methods, which attempt to find optimal solutions within a problem solution search space, and hyper-heuristic methods, which attempt to find a solution within a set of heuristics, are explained in Sections 2.2.2.2 and 2.2.2.3.

2.2.2.1 Priority-rule based heuristics

Priority-rule based heuristics can be differentiated into construction and improvement heuristics. The former starts without an initial elaborated order sequence in the case of scheduling and adds one order per iteration to the processing list. The so-called dispatching rules such as first in first out (FIFO), last in first out (LIFO), and other rules can be assigned to this category and are widely applied in practice (Schneider and Kirkpatrick, 2006; Schuh and Schmidt, 2014). A heuristic serves as an approximation for finding a solution, which can satisfy given constraints and is significantly more efficient and faster in its computation than finding an optimal solution (Zimmermann, 2008; Epitropakis and Burke, 2018). Thus, as an example, in the field of semiconductor job-shop processes, conventional dispatching rules are commonly used construction heuristics, which can cope with the fast-paced and flexible manufacturing processes (Pinedo, 2012; Waschneck et al., 2017; Nasiri et al., 2017).

The right dispatching rule significantly affects a production system's performance. For example, underutilizing a bottleneck resource can hinder the entire system's efficiency, a responsibility falling on the dispatcher who typically only plans for the immediate and next job of a machine. This necessitates a balance between reactive dispatching and predictive scheduling approaches in dynamic processes (McKay and Wiers, 2003). While predictive scheduling can reduce tardiness in systems with machine failures (Mehta, 1999), finding the right parameters is challenging and must be regularly updated for unforeseen events to maintain efficiency. Conversely, reactive scheduling can entirely rely on dispatch rules, recalculating priorities for each order post-event based on current production status and order details (McKay and Wiers, 2003; Ouelhadj and

Petrovic, 2009). This necessitates the quick derivation of clear sequences for material and product flow, taking into account resource- and order-related events as categorized by Vieira et al. (2003) and Ouelhadj and Petrovic (2009).

- **Order related:** rush orders, order cancellations, due date changes, orders arriving early or late, order priority changes, order cycle time changes, etc.
- **Resource related:** machine failures, operator illness, unavailability or failure of tools, load limits, etc.

To calculate the priorities of individual orders, it is possible to use all available information about the order and its production parameters. The exact calculation can be carried out on the basis of pre-existing rules or on the basis of custom specifications and specific process conditions. A non-exhaustive but thoroughly representative list of potential parameters to calculate and derive the specific order priority is provided in the following breakdown, which was adapted from Wiendahl (1997), Bergmann et al. (2014), and Lödging (2016).

- **Order related:** local/ global arrival time, local/ global processing time, due date, order tardiness (or remaining time), externally defined order priority (rush order)
- **Resource/process related:** setup cost, setup time, buffer length, processing cost, storage cost

Based on these criteria, a large number of widely used dispatching rules was embedded into production processes, which evaluate essential operational parameters and allow quick conclusions to be drawn about the priority of the orders and the corresponding order sequence. In this context, the use of the aforementioned priorities is applied in several production tasks that require fast responsiveness. Such tasks can be lot sizing, resource allocation, sequence design, and order release (Selke, 2005; Herrmann et al., 2021). In these tasks, appropriate processing and dispatch sequences are generated after the occurrence of an event described above and the parameters mentioned. Potential target parameters can be machine-centered utilization, but also process-centered such as the reduction of the maximum or average makespan. Further, the evaluation can also be order-centric such as reducing the maximum or mean tardiness (Sculli and Tsang, 1990; Pinedo, 2012; Xie et al., 2019).

Depending on the variable and objective measure under consideration, other process indicators are prioritized. Time-based dispatching rules such as the shortest-processing-time rule, i.e., optimize flow times or work-in-progress levels, but neglect other important parameters such as minimizing order tardiness (Blackstone et al., 1982; Gonzalez et al., 2010). Table 2.3 presents a non-exhaustive overview of essential and basic dispatching rules and associated parameters. Throughout the evaluation phase in Chapter 8, a selection of the listed dispatching rules will be implemented for the benchmarking and validation of the developed artifact.

2 Fundamentals

Category	Name	Description	Parameter	Type
Entry-time oriented	FIFO	First in first out	Global system entry/ local buffer entry	Static
	LIFO	Last in last out		
Process-time oriented	SPT/ LPT	Shortest / longest processing time	Local/ global order processing time	
	TSPT	Truncated SPT	Orders above a pre-determined time are given priority	
Date oriented	EDD	Earliest due date	Time until the order must be completed	
	ERD	Earliest release date	Time, when the order was released into the system	
Priority oriented	CP	Customer priority	Orders with the highest priority are processed first	
Setup time oriented	MST	Minimum setup time	Machine setup time for the respective order	
Waiting time	SWT/ LWT	Shortest/ longest waiting time	Total order waiting time in the system	

Table 2.3 Popular dispatching rules, adapted from Blackstone et al. (1982); Kaban et al. (2012); Bergmann et al. (2014)

A further distinction is drawn by the subdivision into static and dynamic dispatching rules. Accordingly, priority values such as the arrival time of an order into the system or the local buffer are static and do not change accordingly (i.e. FIFO). The waiting time, on the other hand, increases with time and must be considered again locally with each entry of a new order and may require dynamic rescheduling (Kaban et al., 2012). In addition to basic single-parameter and single-objective rules, other multi-parameter dispatching rules have emerged that aim to optimize multiple parameters concurrently. As demonstrated by Le-Anh and De Koster (2005) or Paul et al. (2018), among others, multi-parameter approaches require higher efforts for initial adjustments, but outperform basic single-parameter rules regarding the simultaneous optimization of order waiting times and machine utilization. In addition to multi-parameter concatenation, the coupling of dispatching rules by mathematical operations is feasible and might lead to improvements in performance (Durasević and Jakobović, 2019). However, the disadvantages of the static and individualized rule design prompt the question of how to optimize the solution-finding process, thereby justifying the transition to the more complex meta-heuristics.

2.2.2.2 Meta-heuristics

Meta-heuristics serve as higher-level approximation strategies that control the solution process, aiming to find near-optimal solutions by guiding lower-level heuristics. In this case, the optimal solution is obtained by tailoring the subordinate heuristics and includes techniques such as tabu search or simulated annealing (Jones et al., 2002). Such meta-heuristic approaches have in common that they can be implemented as abstract problem solvers, which can be adapted to various problems (Glover, 1977; Voß, 2008). The procedure generally initiates with the determination of a first solution, from which the best one is selected, and then a search is made for other possibly better ones in the vicinity of the first solution. Thereby, in simulated annealing, the deterioration rate is continuously reduced in the course of the iterations to avoid local optima. Nevertheless, there is no meta-heuristic that, in average, performs better than all others in every problem. Thus, the rule selection at the beginning of the problem solution always remains, which is also referred to as the no free lunch theorem (Wolpert and Macready, 1997). However, one issue in multi-agent systems is a potential local optimization of the system, which may be insufficient on a global scale. Here, random actions can contribute to breaking out of this pattern, to discover potentially better solution spaces (Voß, 2008).

2.2.2.3 Hyper-heuristics

Unlike meta-heuristics, hyper-heuristics do not explore the problem's solution space. Rather, they employ a set of predefined low-level heuristics, as shown in Figure 2.8, or create new heuristic rules from basic elements. As a top-level algorithm, the hyper-heuristic solves optimization problems by combining underlying low-level heuristics into an effective operational sequence (Burke et al., 2010, 2013). The term hyper-heuristic was originally defined by Cowling et al. (2001) and it was initially implemented using a machine learning algorithm to determine an optimal action order for the sales summit problem. In recent years, machine learning and deep algorithms are often applied as top-level algorithms, which can flexibly adapt to the optimization task and thus exploit their domain knowledge and the capabilities of the underlying heuristics in a case-specific manner. This allows for a design process automation and to leverage the knowledge of an on- or offline machine learning algorithm as an optimizer to derive near-optimal scheduling and dispatching policies based on established and comprehensible heuristics (Cowling et al., 2001; Burke et al., 2010, 2019; Drake et al., 2020).

2.2.3 Section conclusion

The previous section summarizes fundamental principles of planning and control optimization, covering different kinds of heuristics. While priority rule-based heuristics typically focus on optimizing a single parameter in a greedy manner, meta-heuristics offer near-optimal solutions

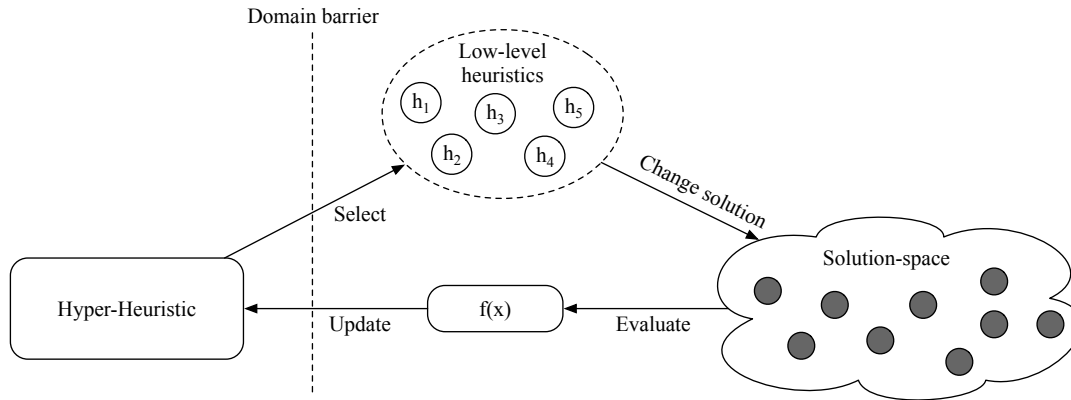


Figure 2.8 Hyper-heuristics principle, adopted from Cowling et al. (2001); Swiercz (2017)

and surpass conventional heuristics in solution quality. Especially for mid- and long-term strategic optimization tasks, meta-heuristics are well suited to generate appropriate results. In an operating production environment, in which fast decisions up to real-time capabilities are required, meta-heuristics and sophisticated mathematical optimization methods, would require too much time and computation resources. Besides, they require in-depth knowledge and are complex in initialization (Zimmermann, 2008; Rauf et al., 2020; Zhou et al., 2020).

Priority-rule based heuristics, in contrast, can be implemented quickly and are easy to understand, but they do not provide globally optimal solutions for optimization problems and have difficulties in adapting to other problems or problem instances (Burke et al., 2013). Nevertheless, due to their low capacity and technical know-how requirements as well as fast response times to find an acceptable solution, they are broadly used, also in large-scale systems (Chen and Matis, 2013). However, myopic patterns occur in such systems, which can be prevented by predictive and global planning (Ouelhadj and Petrovic, 2009). Fine-tuned heuristics are also limited in their ability to optimize various performance measures simultaneously (Grabot and Geneste, 1994) and imply a low degree of coordination at the global level if the information is processed at a local scale (Uzsoy et al., 1993; Holthaus and Rajendran, 1997). In practical settings, it has been noted that the development of advanced scheduling heuristics typically spans several years. Conversely, evolutionary hyper-heuristics demonstrate a markedly faster development pace (Geiger et al., 2006). For these reasons, too, hyper-heuristics were introduced that combine low-level heuristics with top-level strategies to make selections on a problem- and scenario-specific basis. Nevertheless, with the increasing scope of the problem, these also reach their limits under decreasing performances (Nasiri et al., 2017).

Building on the limitations and potentials of these control approaches, it becomes clear that exploring alternative methods, particularly those leveraging the capabilities of machine learning, could provide a solution for autonomous production processes (Weichert et al., 2019; Kang et al., 2020). However, machine learning methods should not be viewed exclusively as a separate

optimization approach, it can also function as a complementary combination with meta- and hyper-heuristics.

2.3 Basic concepts of machine learning

Machine learning approaches can be classified into the three fields of supervised, unsupervised, and reinforcement learning as listed in Table 2.4. Whereas supervised learning algorithms require a labeled set of data and perform task-specific classification, unsupervised learning performs data-driven clustering based on an unlabeled set of data. In contrast, a reinforcement learning based agent reacts to its environment on a continuous basis, issuing corresponding instructions for action, making it highly adaptable to dynamic conditions in real-time production environments. An RL agent learns and operates through direct interaction with its environment, thereby effectively managing uncertainty and establishing itself as an attractive active online optimization method for complex and unpredictable production scenarios (Dey, 2016; Sutton and Barto, 2017; Waschneck et al., 2018). Therefore, the subsequent sections will explore the fundamental concepts of reinforcement learning, encompassing both the basic optimization principles and its extensions into deep reinforcement learning.

	Supervised learning	Unsupervised learning	Reinforcement learning
Principle	Task-driven	Data driven	Reaction driven
Input	Labeled data	Unlabeled data	Current state-action pair
Output	Classification	Clustering	Next action

Table 2.4 Overview of machine learning methods (Dey, 2016; Sutton and Barto, 2017)

2.3.1 Basic of reinforcement learning

Reinforcement learning is characterized by its particularly dynamic learning in interaction with its environment. Based on recently collected and analyzed sensor data, reinforcement learning leverages data-driven decisions, that can be made in real-time, and promotes a responsive and adaptive system design (Han and Yang, 2020). It learns on a trial-and-error basis without requiring a previously collected database or human guidance, and is able to flexibly adapt to uncertain conditions (Sutton and Barto, 2017). This enables volatile and complex production processes to be successfully managed and, in particular, enables resilient production operations even in the case of unexpected events such as machine failures (as in Huang et al., 2020). Based on a trial-and-error concept, the agent explores the problem space without the intervention of a supervisor. Unaware of the consequences of its actions, the agent must learn which action to perform in a particular state to maximize the expected rewards. This action-observation loop of performing an action and then obtaining a new state and reward is illustrated in Figure 2.9.

2 Fundamentals

The reward itself does not only reflect the immediate action impact but also long-term related elements in the form of future incoming rewards for potential further actions. Since every action at time t also affects future states, the agent must be able to estimate transition effects. The exploration-exploitation dilemma is often mentioned in this context, where a choice must be made between exploiting the current knowledge or discovering a potentially more beneficial (or worse) policy. It is precisely this principle of future rewards as well as the trial-and-error principle that distinguishes reinforcement learning from other machine learning methods (Sutton and Barto, 2017).

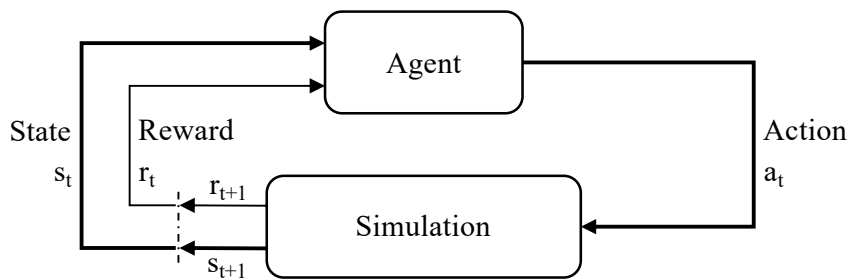


Figure 2.9 Agent - environment interaction loop, adapted from Sutton and Barto (2017)

Reinforcement learning can be further distinguished into dynamic programming, Monte-Carlo methods, and temporal difference learning. These approaches, as well as associated properties, are listed in Table 2.5. Dynamic programming decomposes a task into small sub-tasks and requires a perfect model of the environment. Monte-Carlo methods do not require such a model, but they update the policy only after an entire learning episode. In contrast, temporal difference learning combines the advantages of the previous methods and, as a model-free method, does not require a model of the environment and is updated after each step. This flexibility makes it possible to learn strategies through interaction with the environment without the need for detailed prior problem or process knowledge (Sutton and Barto, 2017). For this reason, temporal difference learning will be focused in the remainder of this thesis. In the later implementations and evaluation, the agent will not receive a production model, as this would be inflexible and scenario-specific. Instead, it is model-free and can deal with stochastic rewards and transitions, as described in Mnih et al. (2015) and Sarker (2021).

	Dynamic programming	Monte-Carlo methods	Temporal difference learning
Needed model	Perfect one	None	None
Policy update	After every step	After every episode	After every step

Table 2.5 Overview of reinforcement learning methods (Sutton and Barto, 2017)

2.3.1.1 Temporal-difference learning and value-based algorithms

An agent that operates based on temporal difference learning updates its policy π after each action. It is assumed that the underlying problem in this thesis can be described as a Markov decision process (Malus et al., 2020). This implies that the Markov assumption is met and that future states only depend on the present state. Being unaware of the environment, dynamic processes, and consequences of its actions, the agent must figure out which decision leads to the highest possible reward in its current state. To facilitate this, the agent's behavior is driven by its policy, which is adjusted based on the reward signal with each action and constitutes the core of the reinforcement learning algorithm. Depending on the state s , it determines the behavior of the agent and outputs corresponding action instructions. The policy is thereby updated based on the acquired experience and incrementally completes the task or adapts to a new problem (Sutton and Barto, 2017).

At this point, a further segmentation into value-based and policy-based algorithms can be established. As illustrated in Figure 2.10 on the right, policy-based algorithms map an explicit representation of the actual policy and output a single action value. Thus, the policy would be parameterized directly, which is particularly suitable for continuous action spaces (Doya, 2000). Value-based algorithms, on the other hand, exploit a discrete action space but do not operate directly on a policy but process the output of a value function. The policy can be derived directly from this value function, for example by selecting the action that yields the highest value. In the further course, the focus will be on value-based algorithms due to their relevance for pre-defined discrete dispatching action spaces and their proven superiority in planning and control related production research, as detailed in the first publication (see Section 4). For an exhaustive introduction into these basics, one can refer to the fundamental reinforcement learning literature by Sutton and Barto (2017).

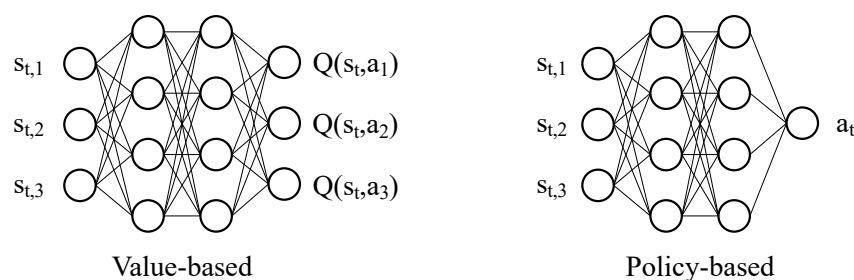


Figure 2.10 Value- and policy-based neural networks

For value-based reinforcement learning, the reward function, as an immediate gratification, and the value function determine the learning process and significantly affect system dynamics. The reward function quantifies the scenario-specific reward and can be designed on the basis of various production objectives. It could, for instance, be designed to reward fast throughput times, but also to penalize incorrect actions. The value function does not only consider the current

reward but also future states that are associated with a long-term evaluation horizon of the agent. Starting from its initial state s_t , the agent can perform an action a_t and subsequently observes a new state s_{t+1} . In that state s_{t+1} , the agent again has possible options for action that have implications for the rewards in the states s_{t+1+n} . The value function considers those very future rewards with their likelihood of occurrence and calculates the resulting cumulative reward. For this reason, states with low rewards may have a high value because they promise high future rewards, or vice versa. This is why decisions about actions should be made based on the value function rather than on direct rewards. However, values are much more difficult to calculate and the value function must be determined iteratively to make it more accurate with each action performed (Sutton and Barto, 2017).

2.3.1.2 Optimization model

To clarify the implementation in Chapter 7, based on Sutton and Barto (2017), the following subsection briefly outlines central learning mechanisms in temporal difference reinforcement learning. One of these is bootstrapping, which allows estimating the values for future states of a system after each step based on the prior gained experience and thereby extracted system knowledge. Beginning with Eq. 2.1, it is indicated how the total expected and cumulative reward G can first be calculated based on the expected rewards at a given step t . With the Monte Carlo method, this would mean waiting until the end of the episode to calculate and update the applied policy. In temporal difference learning, bootstrapping (as in dynamic programming) is applied at this point, and the actual total expected reward G_{t+1} is replaced by the value function as an estimate for the next state S_{t+1} . Here γ denotes the discount factor and determines the relevance of the future estimates. $\gamma = 0$ would mean that only instantaneous rewards would count, whereas $\gamma = 1$ would assign the same relevance and influence to all future rewards.

$$\begin{aligned}
 G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\
 &= R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \dots) \\
 &= R_{t+1} + \gamma G_{t+1} \\
 &= R_{t+1} + \gamma V(S_{t+1}) \leftarrow \textit{Bootstrapping}
 \end{aligned} \tag{2.1}$$

Next, the value function estimates how beneficial it is for an agent to be in a particular state. According to the received rewards and the estimate of the next state, the value function (see Eq. 2.2) is adjusted by means of the temporal difference error (first line). This can be seen as an estimate of how far the agent was mistaken with his original estimate. On the basis of the temporal difference error, the original estimation of the value function (2.1/2.2 in Eq. 2.2) can be adjusted accordingly. The impact of the alteration by the temporal difference error δ_t in this context is also driven by the learning rate factor α . The future reward R_{t+1} in this case

corresponds to the reward received by the agent and will be denoted by r in the further course.

$$\begin{aligned}
 [1.] \delta_t &= r + \gamma V(S_{t+1}) - V(S_t) \\
 [2.1] V(S_t) &\leftarrow V(S_t) + \alpha \delta_t \\
 [2.2] V(S_t) &\leftarrow V(S_t) + \alpha [r + \gamma V(S_{t+1}) - V(S_t)]
 \end{aligned} \tag{2.2}$$

In either case, the goal of an agent should not only be to be able to estimate the value function correctly or very accurately but also to maximize the received rewards over time by selecting optimal actions. At this point, therefore, a differentiation needs to be drawn between the state-value (see 1. in Eq. 2.3) and the action-value function. The action-value function is also called Q-function and estimates the quality of executing an action a in a state s . Thereby, π describes the used policy of an agent.

$$\begin{aligned}
 [1] V_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\
 [2] Q_\pi(s, a) &= \mathbb{E}_\pi[G_t | S_t = s, A_t = a]
 \end{aligned} \tag{2.3}$$

The state value function $V_\pi(s)$ can be described as the cumulative return of rewards for a state s considering all possible actions a . Mathematically, this is the sum of all possible action values $q(s, a)$ times their probability $\pi(a|s)$ for choosing the respective action in state s , as given in Eq. 2.4 below (Sutton and Barto, 2017).

$$V_\pi(s) = \sum_a \pi(a|s) Q_\pi(s, a) \tag{2.4}$$

To guarantee an optimal decision-making process, the agent should follow the maximum state- and action-value functions, which promise maximum rewards and respective state and action values (see Eq. 2.5).

$$\begin{aligned}
 [1] V_*(s) &= \max V_\pi(s) \\
 [2] Q_*(s) &= \max Q_\pi(s, a)
 \end{aligned} \tag{2.5}$$

In reinforcement learning, this is where the greedy policy is frequently mentioned and introduced as a potential policy. The greedy approach explicitly reinforces the relationship mentioned in Eq. 2.5 to always execute the optimal next step that yields the highest reward. Thus, Eq. 2.4 can be adapted accordingly, since the probabilities are no longer demanded, and the maximum action value has the now decisive influence on the state value. This modified relationship is outlined in the following equation.

$$V_*(s) = \max Q_\pi(s, a) \tag{2.6}$$

Substituting Eq. 2.6 into 2.1, G and V can be replaced accordingly as outlined in Eq. 2.7.

$$Q(s_t, a_t) = r + \gamma \max_{\pi} Q_{\pi}(s_{t+1}, a_{t+1}) \quad (2.7)$$

Eq. 2.7 represents a modified form of the Bellman equation. This results in the iterative relationship of formula 2.2 and constitutes the following principle of Q-learning.

$$\begin{aligned} [1] \quad & Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \delta \\ [2] \quad & Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r + \gamma \max_{\pi} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \end{aligned} \quad (2.8)$$

Q-learning is a model-free, reward-based reinforcement learning approach, used to learn the best action in a given state within a Markov decision process. Based on this Q-learning update principle, a potential state-action Q-value table can already be iteratively updated for a specific problem. In such conventional Q-learning, a Q-table stores all quality Q-values for executing an action in a certain state. However, in a large or multidimensional problems, a q-table grows quickly and becomes difficult to process (Arulkumaran et al., 2017). Therefore, deep learning based reinforcement learning approaches have emerged in recent years in various applications Alzubaidi et al. (2021); Gronauer and Diepold (2021). The integration of a neural network eliminates the dependency on a Q-table and tries to combine the interactive adaptive learning of reinforcement learning with the processing properties of neural networks (Sutton and Barto, 2017). The following section outlines how neural networks can be integrated into reinforcement learning to represent the policy in a more compact and robust form.

2.3.2 Deep reinforcement learning

When comparing the general framework of the deep reinforcement learning algorithm in Figure 2.11 with the conventional one in Figure 2.9, there are no significant differences. The agent continues to receive the state S_t and the reward R_t to determine an appropriate action. After executing the action, the next state and reward are observed, and the agent-environment interaction loop continues (Sutton and Barto, 2017). The main difference lies in the processing of the incoming state through the neural network (van Hasselt et al., 2016).

With its first implementation in 2013, Mnih et al. demonstrated how deep neural networks can be deployed to leverage reinforcement learning to solve high-dimensional problems with superior performances. The neural network functions to approximate the action-value function, where the input layer receives either a raw or pre-processed state vector. This vector is then processed through hidden layers, culminating in the output layer, which suggests an action. In Q-learning contexts, employing a deep neural network is known as deep Q-learning or the deep Q-network (DQN).

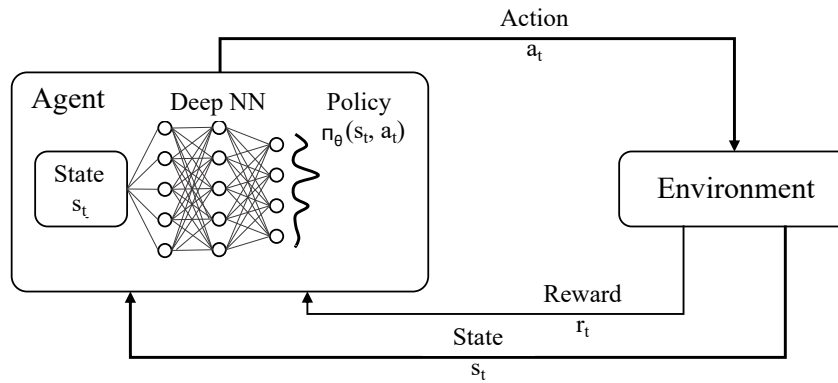


Figure 2.11 Neural network-based policy approximation in deep RL

2.3.2.1 Neural networks

Regarding the composition of such a network, there are several possible network architectures available with highly different characteristics and possible use cases (Alzubaidi et al., 2021). Three of the most common architectures are the (deep) feed-forward network, the recurrent network, and the long-short-term-memory network as illustrated in Figure 2.12. In a feed-forward network, all neighboring layers are connected and the (processed) information is only passed forward. In this case, the neurons of the recurrent neural network also take into account past experience in particular but suffer from a vanishing gradient that decreases exponentially with time. Long-short-term-memory networks provide a solution to this problem by introducing gates that help preserve information (van Veen, 2017; Sherstinsky, 2020). The neural network is important for the performance of a reinforcement learning agent, as it enables it to recognize complex patterns and create operating policies. Its ability to generalize and adapt to different environments contributes significantly to the efficiency and accuracy of operational decision making (Mnih et al., 2015; Alzubaidi et al., 2021). Thus, selecting the network type is a crucial aspect of the reviews in Chapters 4 and 5, corresponding to Publications 1 and 2.

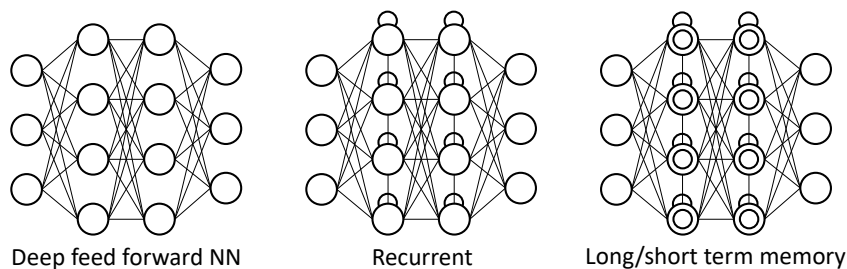


Figure 2.12 Types of neural network

2.3.2.2 Algorithmic DQN peculiarities

Eq. 2.8, introduced in Section 2.3.1.2, for the update of Q-values can also be implemented in the field of deep reinforcement learning. Referring to the DQN, the action a that promises the maximum Q-value within the output layer is adopted. The maximum action value Q thus directly implies the optimal strategy for the agent in a particular state s . In the output layer, each index is assigned a fixed action, which the agent executes if selected. In contrast to Q-learning, in which a single table entry is updated, a large number of affecting parameters are updated during the DQN update process. This can lead to a process of the so called *catastrophic forgetting*, which, although a good policy has already been learned, again ends in a poor performance of the agent. Due to the resulting risk of instability during operation, if the online network is used for the computation of the target and the Q-values at the same time, an additional target network is deployed in parallel during the computation process. The target network shares the same network structure as the online network but contains different network weights θ^- , as outlined in Eq. 2.9. L_i denotes the loss to update the neural network (Mnih et al., 2015). This equation utilizes the target network to compute the temporal difference target based on the new state s_{t+1} .

$$L(\theta) = \mathbb{E}_{s_t, a_t, s_{t+1}, r} \left(\underbrace{r_{s_t, a_t} + \gamma \max_{a' \in A} Q(s_{t+1}, a', \theta^-)}_{TD\text{-}target} - Q(s_t, a_t, \theta) \right)^2 \quad (2.9)$$

$\underbrace{\hspace{10em}}_{TD\text{-}error}$

Where:

- θ : Online network parameters
- θ^- : Target network parameters

The online network is in turn used to determine the action to be executed by the agent and thus to observe the occurring new state s_{t+1} after execution. In addition, the online network is used to calculate the action value of the state s and the action a (see Figure 2.13). If the targets were updated continuously, the operational stability would not be positively affected. For this reason, it is only updated every C steps to prevent unstable behavior (Mnih et al., 2015). During the update, the weights of the online network are transferred to those of the target network, which is also called a hard update.

For the training, an experience replay proved to be efficient, which will be implemented in the further course of the thesis. During training not only the current/next state, action, and reward pair is included, but with each update step, a mini-batch is retrieved from the memory. The advantage of this batch replay is that not only current experiences are internalized on the basis of the received state data, but also past state changes or experiences are always reconsidered. Although the DQN in its initial form demonstrated superior performances in numerous papers,

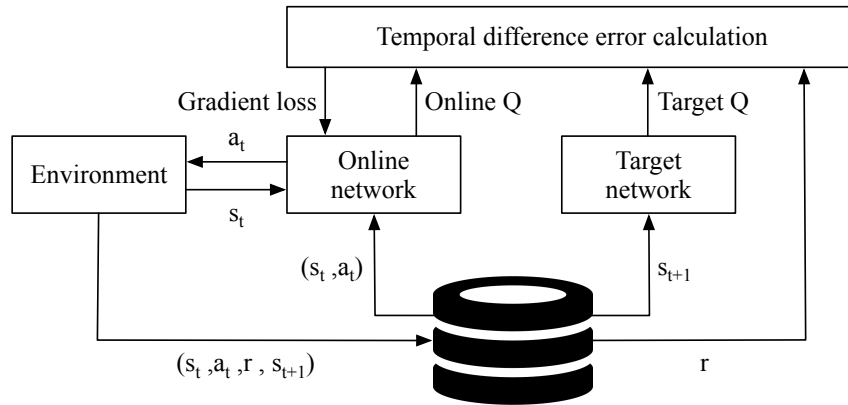


Figure 2.13 DQN operating with target network (Arwa and Folly, 2020)

there are different possibilities for further controlling and leveraging the learning process. On the one hand, events considered to be particularly relevant can be prioritized within the experience replay (Schaul et al., 2016). On the other hand, the performance of the algorithms can be significantly improved by tuning the parameters. This training and optimization process is explained in more detail in the third publication in Chapter 7.

2.4 Chapter conclusion and initial research gap

This chapter delved into the central elements of production processes, also focusing on their planning and control principles. It also presents an in-depth explanation of reinforcement learning, emphasizing its combination with deep neural networks. This introduction is essential for laying the scientific groundwork of the thesis. The chapter concludes with a brief summary of two identified research gaps. These gaps are discussed more thoroughly in Chapter 3 and are addressed in the first two publications, detailed in Chapters 4 and 5, following the *DSRM* approach.

Deep reinforcement learning based production systems. Deep reinforcement learning has proven effective in a variety of research fields, including the communications sector (Luong et al., 2019), economics (Mosavi et al., 2020), and electrical utilities (Mishra et al., 2020). Despite these successes in related research fields, the specific application of deep reinforcement learning in production systems remains to be reviewed and structured. This aspect is also reflected in the reviews by Cadavid et al. (2019), Kang et al. (2020), and Arinez et al. (2020), which explore general machine learning in production but do not concentrate on deep reinforcement learning. In current job-shop and matrix production systems, primarily conventional rule-based heuristics are used, which enable a fast and acceptable decision-making process, but are inferior to advanced heuristics for global operations and multi-factorial requirements (Greschke, 2016).

To identify appropriate dispatching mechanisms, Gonzalez et al. (2010) suggests that approaches like dynamic rule identification could be promising.

Deep reinforcement learning, particularly, could facilitate dynamic control of problems with large state spaces and achieve high adaptability. For practical applications, it's a significant advantage that neither extensive process or production knowledge nor a process model are required (Sutton and Barto, 2017). However, a thorough exploitation of deep reinforcement learning's potential in production systems remains to be done and initially necessitates a detailed review. This review should not only focus on the specific fields of application and the algorithms and parameters used, but also highlight the resulting benefits and risks. Consequently, the first review aims to provide a comprehensive overview of the opportunities and challenges associated with deep reinforcement learning based production systems.

Deep learning based production optimization organisation. In addition to the application of advanced deep reinforcement learning, the question arises as to how decision-making production agents are structured and how they interact with each other. In Gronauer and Diepold (2021) a comprehensive analysis of multi-agent systems employing deep reinforcement learning (DRL) is presented, highlighting an increasing emphasis on multi-agent systems. However, various inter- and intra-organizational as well as algorithmic aspects of production agents may significantly influence the flexibility, adaptability, and resilience of a production system. Emerging methodologies, such as the semi-heterarchical integration of hierarchical and heterarchical systems, may improve production flexibility and scalability, as emphasized by Balaji and Srinivasan (2010).

Concludingly, there's a gap in focused analysis on the organization, collaboration, and training of deep learning based production agents. With the increased use of modern concepts like swarm intelligence and the expansion of machine interfaces and data sources, it's crucial to understand the deployment and collaboration of these agents. Thus, the scope of the second review extends beyond deep reinforcement learning to include other deep learning based approaches like genetic algorithms and simulated annealing, also focusing on their organizational integration.

3 Publications and research paradigm

Reflecting the motivation and basics chapter, advanced production models and control mechanisms have recently emerged to cope with growing complexity and an increasing number of process interdependencies. This has led to an increased use of intelligent, data-driven, and autonomous approaches in production, which contribute significantly to dynamic decision-making (Weichert et al., 2019; Kang et al., 2020; Peres et al., 2020). After outlining fundamental concepts in the previous chapter, the following sections thoroughly describe the integrated publications in this thesis and contextualize them in an interrelated manner in Section 3.1.

This chapter and the thesis are structured into two primary sections, each having a distinct focus. The first section, which includes the first bundle of publications detailed in Section 3.2, concentrates on analyzing the research field and developing a taxonomy. The aim is to foster a comprehensive understanding and in-depth exploration of the subject area. Subsequently, the second section, along with the second bundle of publications presented in Section 3.3, focuses on constructing and empirically evaluating the developed artifact. This section emphasizes the practical application and validation of the theoretical concepts established earlier

Each bundle of publications addresses specific sub-research questions within the broader context. The first bundle of publications, focusing on S-RQ1, examines the general requirements for complex production environments, particularly their planning and control aspects. It prioritizes two aspects, firstly, the production-oriented structuring of the deep reinforcement learning research field, and secondly, analyzing the organizations and agent configurations in general deep learning based production systems. The objective is to identify specific research needs and to derive corresponding design requirements. Building on this, the second bundle of publications concentrates on implementing the identified requirements to address the two remaining sub-research questions, S-RQ2 and S-RQ3. S-RQ2 targets reducing decision and optimization complexity, incorporating insights from reviews, identifying trends, and advancing proven concepts. This includes considering potential algorithms, flexible layouts, multi-agent concepts, and other design elements. The objective is a non-specific scenario implementation to enhance the generalizability of the developed control strategy, aligning with S-RQ3.

3.1 Research paradigm

In the recent past, a variety of methodologies have emerged to optimize production, many falling under the *Industry 4.0* paradigm. This shift extends beyond mere process automation and notably incorporates autonomous operations and streamlined data management. In this context, the enhanced computing capabilities of individual resources have become increasingly crucial. Rather than relying solely on central or cloud computing resources, this approach involves

3 Publications and research paradigm

distributing decision-making to enable local autonomy of single or multiple agents. As a result, tasks across different domains can be allocated to specific agents (Bongaerts et al., 2000; Lee and Kim, 2008; Buckhorst et al., 2022). This strategy aligns with the concept of intelligent agents, mediators, or facilitators that autonomously make decisions based on their intrinsic (intelligent) logic or strategy (Christensen, 1994; Cowling et al., 2001). In this field, machine learning and deep learning oriented decision frameworks have gained traction, as outlined in reviews by Kang et al. (2020) and Peres et al. (2020).

Building upon deep learning driven and autonomous decision-making, this thesis aims to address the specified research questions and achieve the outlined objectives through a combined bundle of publications. These publications are organized in a multi-layered structure, guided by the *DSRM*, as depicted in Figure 3.1. The initial step of the *DSRM*, encompassing the general problem identification and motivation, is covered in the introduction and therefore not included separately. In the subsequent step, the first two publications identify a specific research gap and collect essential information to establish the foundational requirements for the artifact design. The third publication addresses these foundational design aspects, focusing on the design and development of the deep learning based control framework. The fourth and fifth publications are dedicated to the empirical evaluation of this control framework, including its application in real-world scenarios and a comprehensive financial analysis.

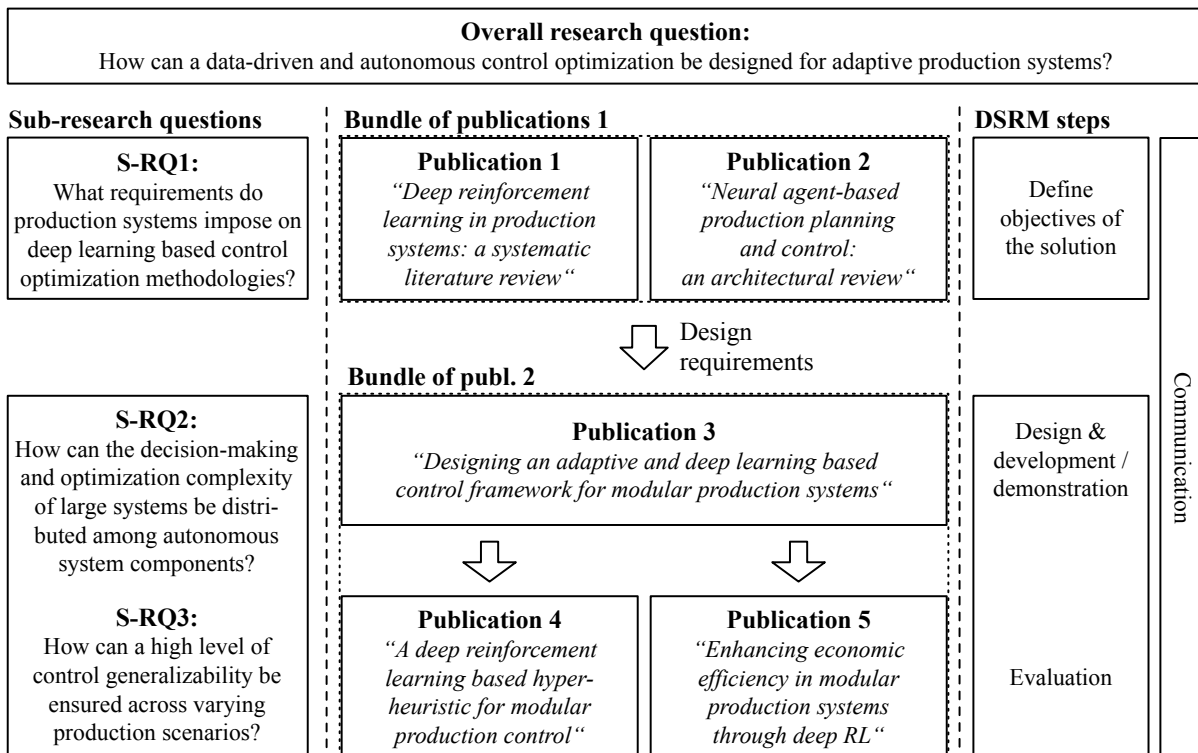


Figure 3.1 Elaboration of the fundamental research base

The initial objective is to understand, frame, and structure the application of deep learning within the production domain, with a sharper focus on deep reinforcement learning. This involves synthesizing earlier methodologies and comparing them in depth based on their application categories, associated algorithms, and optimization objectives. Furthermore, the integration and orchestration of participants in a multi-agent system emerge as a central research field, requiring thorough investigation. As such, the initial two publications of this thesis focus on two core aspects, first, an algorithmic analysis of deep reinforcement learning, and, second, an organizational study of deep learning based production.

The first publication highlights applications of deep reinforcement learning solutions in the production domain, emphasizing the superiority and scope of value-based reinforcement learning algorithms in production control and other real-time applications, such as robotics assembly. In the second publication, a comprehensive architectural study systematically examines prevalent deep learning based approaches, encompassing but not limited to deep reinforcement learning, in production planning, control, and forecasting. These approaches are then categorized based on their agent-specific and organizational structures. Using the devised taxonomy, a notable deficit of integrated multi-agent systems is identified. Within these multi-agent systems, no embedded approach is found that intrinsically combined multiple optimization methods within an agent. The objective of the embedded approach is to tackle the problem at the level of an individual agent by decomposing the whole task, thereby reducing its complexity, and subsequently identifying an optimal solution out of the smaller solution spaces. Therefore, based on the algorithmic and organizational review and the agent-based system taxonomy, the first two publications facilitate the clarification of the first sub-research question S-RQ1 based on the dominant algorithms and main agent organizations and orchestrations. These findings substantially narrow the specified research domain and drive the development of the artifact in subsequent publications.

In the second bundle of publications, during the design, development, and evaluation phase, sub-research questions S-RQ2 and S-RQ3 are addressed. Beginning with the third publication, a multi-embedded agent control optimization is implemented. The modularity allows a product- and process-bundling of resources which can be further customized. Regarding the chosen algorithm, a sequential embedded hyper-heuristics structure is applied. This combines deep reinforcement learning with conventional heuristics, which, among other advantages, increases the learning speed and proactively prevents the execution of faulty actions. Furthermore, a conceptual learning framework is developed that can reuse trained neural networks to reduce training times. Initial testings confirm the superiority of the approach over conventional dispatching rules.

In the fourth publication, the hyper-heuristics based control approach is further optimized to address both technical and customer-specific process parameters. Several testings demonstrated the superiority over commonly used dispatching rules and the approach was successfully inte-

grated into a hybrid test-bed. Both in the simulation and in the real environment, order tardiness and throughput times were significantly reduced, with increased control stability. Training and control robustness, in the face of volatile work-in-progress levels and machine failures, is demonstrated, ensuring consistent and seamless management across a multi-layer system with multiple manufacturing and distribution cells.

In the fifth publication, a techno-economical evaluation of the presented control framework is undertaken. Thereby, financial indicators such as revenue and costs are interconnected to order parameters to evaluate the proposed production control. Testings reveal that financial advantages can be realized with reduced delay penalties and an improved processing of rush orders. Additionally, the capability of the system to incorporate tailored services was underscored, facilitating the emergence and successful integration of novel business prospects. It is demonstrated that the representative reward function proficiently addressed both technical and economic objectives.

3.2 Bundle of publications 1 - identification and structuring of the research gap

The first bundle of publications emphasizes the preliminary organization of the research domain, the extraction of distinct design guidelines, and the extraction of pivotal production challenges associated with the application of deep learning, particularly deep reinforcement learning. In this context, a hybrid strategy is adopted. First, a theoretical construct is formulated, contributing to the resolution of the research objectives underpinning this thesis. Second, exemplary methods are consolidated, facilitating the refinement of the selected strategy and the conceived artifact. Correspondingly, the initial two stages of the *DSRM*, the problem identification and the deduction of artifact-oriented objectives, are initiated and addressed. The insights gained not only offer specific, tangible benefits for the construction of the artifact but are also shared with the scientific community. This dissemination takes the form of combined reviews of algorithms and their applications, as well as a taxonomy for agent-based systems. The practice-oriented evaluations also invoke the application domain from Hevner et al. (2004). Thereby, the development of the taxonomy expands the current knowledge base, which, when combined with both the *relevance* and *rigor cycle*, constitutes a crucial informational foundation for artifact construction.

While Xu et al. (2018), Arinez et al. (2020), and other scholars provide a comprehensive perspective on *Industry 4.0* and artificial intelligence in the production domain, there exists a notable gap. A detailed and systematic examination of deep learning and deep reinforcement learning within production was not thoroughly conducted. To bridge this gap and provide a clear understanding, a tailored procedure is indispensable. This is crucial for the methodological exploration of the subject and is detailed in Section 3.2.1. The central outcomes of this exploration are detailed in publications 1 and 2, as summarized in Sections 3.2.2 and 3.2.3, respectively.

3.2.1 Review methodology

A systematic review of the related literature serves to establish a scientifically sound research basis. The objective is to clearly identify the current state of research and to allow the derivation of potential research objectives to be addressed later in this thesis. Based on this, the review serves to address the first *DSRM* step of Peffers et al. (2007) from the previous subsection. For a detailed content analysis of the publications within the review scope, the guidelines provided by Tranfield et al. (2003) and Thomé et al. (2016) are adopted. This not only facilitates an optimized consolidation of conducted research to refine the latest state-of-the-art contents but also enhances the general evaluation and classification of the publications under consideration. To conduct both, a systematic and representative review, the eight-step approach, proposed by Thomé et al. (2016) was deployed (see Figure 3.2).

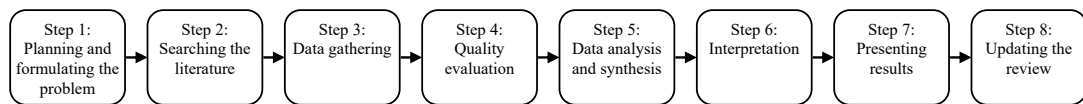


Figure 3.2 SLR review steps, Thomé et al. (2016)

The motivation and planning of the research problem are mentioned in the introduction and will be continued in Chapter 6. Further steps are the development of the literature database, the collection of relevant data a the quality assessment, which is followed by the analysis, interpretation nand presentation. For the respective SLR throughout this thesis, the same taxonomy framework, adapted from Cooper (1988), is always applied and is illustrated in Figure 3.3.

Characteristic	Categories			
(1) Focus	Research outcomes	Research methods	Theories	Applications
(2) Goal	Integration		Criticism	Central issues
(3) Perspective	Neutral representation		Espousal of position	
(4) Coverage	Exhaustive	Exhaustive and selective	Representative	Central / pivotal
(5) Organisation	Historical		Conceptual	Methodological
(6) Audience	Specialized scholars	General scholars	Practitioners / politicians	General public

Figure 3.3 Pursued taxonomy framework

In accordance with the introduced taxonomy, the survey primarily (1) focused on the gathering of relevant research results and applications, with a particular focus on the current state of research in the field of deep learning based production as well as deep reinforcement learning based production planning and control. The overall goal (2) was to provide a coherent representation of the results achieved, the algorithms used, and generate prospects for future research efforts, thereby highlighting the key research questions. Each analysis is based on a neutral (3) and exhaustive (4) examination, which takes into account all publications in the respective research

area. Furthermore, it is intended to analyze current concepts that allow the derivation of design goals and requirements not only for the readership but also directly for this thesis (5). The primary target audience is specialized scholars (6), although a broader scientific and practical community is not excluded. This is particularly reflected in the inclusion of analyses that extend beyond technical aspects, encompassing financial ratios, resilience, and other relevant metrics.

The following subsections provide summarized reviews on the first two publications. The former primarily focuses on exploring the modalities of deep reinforcement learning implementation, including state and action design, as well as possibilities for reward function configurations (Section 3.2.2). The latter aims to analyze the interactions among multiple agents, drawing conclusions about the various implementations of such systems and their optimal interaction and organization (Section 3.2.3).

3.2.2 Publication 1 - deep reinforcement learning based production systems

In this first publication, the applicability and performance of deep reinforcement learning in production systems was investigated. Out of a total of 1,255 papers, 120 were identified, primarily in the fields of production process planning, scheduling, and assembly. It is noteworthy that the relevance of the publications, measured by their number, increased from three in 2017 to 69 in 2020.

Based on the analysis, it appears that deep reinforcement learning is already being applied in diverse production settings, and often outperformed conventional methods, fostering a data-driven, flexible process while minimizing implementation efforts and dependency on expert knowledge. Deep reinforcement learning, through its inherent adaptability, learning behavior, and real-time response capabilities, reveals a high potential to address these challenges in assembly planning, robotics and other domains. It is emphasized that deep reinforcement learning continuously interacts with its environment and quickly responds to received sensor data which promotes a prompt and unbiased adaptation to system changes.

Nevertheless, the existing literature poses significant research gaps. A comprehensive scenario coverage or detachment from specific problem scopes was often neglected which leads to a lack of actionable guidelines. It is evident that many approaches were implemented in small-scale and simulated contexts, and their generalizability remains unexplored. Illustratively, as a concise example - while real-world tests were conducted in fields such as assembly, similar endeavors are missing in production planning and control. Therefore, the prevailing emphasis on restricted methodologies potentially decreases system efficacy, which potentially hinders the overarching progression toward smart production systems. Also, like in Rummukainen and Nurminen (2019), the lack of concrete implementation guidance must be criticized, which could provoke a high dependence on standardized algorithms. Given this perspectives, four major research imperatives

emerge, that can guide future research.

1. Reduce production optimization complexity through scale reduction
2. Prioritize the translation of simulated results to actual production settings
3. Master the creation and use of standardized, yet adaptable frameworks
4. Emphasize the advancement of methodological techniques and the incorporation of novel training strategies

In summary, this publication exposes the significant impact and added value of deep reinforcement learning for its application in production environments. It is revealed that various conventional methods, that are currently widely used in real systems, were outperformed. Nevertheless, it becomes evident that additional efforts are necessary to implement this specific form of deep learning on a broader scale, beyond the widely considered job-shop scenarios, particularly in production planning and control applications. This and the previous insights obtained in this first publication are synthesized in the following working hypothesis.

Deep reinforcement learning is well-suited for handling complex and dynamic optimization problems due to its high adaptability. However, current research lacks a control framework that consolidates research findings and facilitates flexible performance optimization in various production scenarios.

Beyond the solely algorithmic perspective, this publication initially highlights aspects that shed a first light on the importance of organizational design, especially in the context of collaborative multi-agent architectures. It is this organizational consideration that foregrounds the interaction and collaboration of autonomous agents which will be addressed in more detail in the next publication.

3.2.3 Publication 2 - organizational deep learning production perspectives

The previous publication highlighted the dominance, flexibility, and broad applicability of deep reinforcement learning in the production systems. Having outlined its benefits, an extended algorithmic focus is now given to the production planning and control domain in order to better understand integrative approaches in this specific domain. In doing so, this publication focuses on the classification of deep learning based approaches, their organizational composition, and deployment of agents, together with potential benefits and challenges. In doing so, the focus shifts from an algorithmic-based exploration to a more integrated and comprehensive review.

The obtained findings suggest that generic deep learning based agents are increasingly deployed in versatile setups and are also increasingly adopted in embedded or multi-agent scenarios. This not only serves to increase performance compared to conventional benchmarks but also

to minimize dependency on human expertise through combined algorithmic and collaborative problem-solving. To classify the different approaches, a taxonomy was devised that systematizes the underlying concepts and compares their advantages and disadvantages. The framework primarily differentiates between two dimensions, the number of agents and the number of utilized algorithmic methods, which are summarized in subsequent sections.

First, the monolithic or plain approach (1) primarily focuses on implementation speed and high optimization specification. In this case, an intelligent agent or other decision-making instance is implemented to solve the entire problem. A fast deployment and minimal intrinsic dependencies allow initial efforts to be significantly reduced. This efficiency enables the rapid evaluation of prototypical use cases, from which valuable empirical values can be derived. Despite the efficiency of this method, scaling problems can occur in large-scale implementations. In contrast, the embedded approach (2) aims to divide optimization problems into separate segments to either streamline their processing or increase performance. In this context, complex problems can be subdivided into manageable subsets. The decomposition of demand forecasts can be exemplified by segregating them into long-term and short-term components, which are subsequently aggregated. Beyond this parallel computation, algorithms can operate in a sequential manner, where the output of an initial algorithm serves as input for the succeeding one. For instance, an analytical model's prediction of a base rate can be employed as an input to an artificial intelligence model, which is then augmented by a dynamic component. The multi-agent approach (3), on the other hand, emphasizes the combination and interaction of multiple agents. These agents, often having a plain intrinsic design, act as independent, autonomous entities. Every agent receives environment data, that is consistently acquired from the associated system without an overwhelming influx, thus facilitating streamlined processing and enhancing adaptability and reactivity in dynamic production environments. A hybrid exploitation (4) of the aforementioned approaches (2-3), integrating diverse organizational and interaction modalities, could potentially compensate for the weaknesses inherent to the individual strategies.

However, no approach currently combines multi-agent systems with embedded control optimization. The analysis further reveals that in multi-agent systems, intensified monitoring of the learning behavior of the agents involved is essential to prevent simultaneous learning instabilities and mitigate the potential negative effects resulting from the dynamic and non-stationary behavior of the agents. Within more complex multi-agent systems, the prospect of sequential agent training was analyzed. Although this sequential approach offers methodological advantages, it probably increases the overall training duration. It proves to be a strategic need to further investigate the synergistic effects of multi-agent systems to reduce optimization complexities but also to strengthen the robustness of learning paradigms. Additionally, the challenges delineated in the preliminary study were reconfirmed in this extended analysis within the broader deep learning domain. Only one embedded and one multi-agent scheduling approach were implemented in

3.3 Bundle of publications 2 - addressing the research gap

a real or hybrid environment (Kumar and Dimitrakopoulos, 2021; Zhou et al., 2021). Hence, future research endeavors should focus on the following pivotal aspects.

1. Systematic categorization and implementation of multi-agent based methods and selective exploitation of their potential for dynamic and distributed control systems
2. Integration of embedded approaches to synergistically exploit their advantages, aiming to reduce decision-making complexity.
3. Encapsulate the production system's organization within the multi-agent system design to further reduce decision-making complexity

It is imperative to recognize that sophisticated methodologies, including multi-agent systems, and the organizational taxonomy should not be examined in isolation. Instead, they necessitate an integrated execution to facilitate precision-oriented and seamless deployment. While deep learning methodologies have manifested in impressive outcomes, most of them were realized within constrained single-agent frameworks. Notwithstanding the trend towards systems with larger scales, like matrix systems, the intricacies of advanced deep learning based production systems remain largely unexplored, leading to the following working hypothesis of the second publication.

Although there is a noticeable trend towards embedded and multi-agent systems, it manifests an imperative need for more integrative efforts. These should combine advanced approaches in a synergetic manner to leverage potentials and adequately reflect the production structure, striving to minimize control and optimization complexity.

3.3 Bundle of publications 2 - addressing the research gap

The second bundle of publications builds on the findings from the first two and aims to empirically answer the sub-research questions S-RQ2 and S-RQ3, seeking to address the hypotheses postulated in the previous publications. In particular, it addresses the *DSRM* phases 3 to 5, design and development, demonstration, and evaluation. In this regard, an iterative development approach is proposed that, according to Hevner (2007), emphasizes the *design cycle* and seeks a continuous improvement process. Drawing from literature reviews, and informed by the formulated taxonomy, along with discerned research gaps and challenges, specific research imperatives can be delineated. These directly affect the development of the artifact and, according to the *relevance cycle* of Hevner (2007), have multiple implications for subsequent implementation and application of the artifact in the simulation and real-world system.

It was found that the use of deep learning in production environments, in particular deep reinforcement learning, can lead to significant performance improvements and high robustness

and adaptability in a variety of applications. However, results in the field of production control, a particularly dynamic and operation-critical discipline, were mainly achieved in simulations with specific and restricted problem and optimization scopes. Moreover, only a few approaches were realized in multi-agent systems, and no approach has yet been implemented in a modular system. From an algorithmic perspective, the standard DQN is commonly employed. While it performs well in straightforward applications, it struggles with increased state dimensions and optimization complexities in larger systems.

The analysis identifies a notable research gap, underscoring the need to create a central deep learning artifact that facilitates an adaptive control operation and optimization. Such an artifact must be flexible enough to be adaptable to various performance metrics and encompass a wide application scope. Thereby, the objective is to address the second and third sub-research questions, the reduction of decision and optimization complexity, and the improvement of the production systems' generalizability. For this purpose, a control approach was designed that takes into account the identified research gaps and prevailing system design requirements from the first bundle of publications. This further ensures a holistic inclusion of the specifications derived from the first sub-research question. The control and optimization complexity in this thesis, as derived from the prior analysis, can be structured through three perspectives, a structural (1), an organizational (2), and an algorithmic perspective (3). Subsequently, a delineation of these perspectives is presented. This aims to secure a coherent integration and comprehensive understanding of the artifact's thematic classification and significance within the context of the thesis.

Structural perspective (1): In recent years, as analyzed in the first two publications, most deep learning approaches focused on job-shops, while matrix systems were only occasionally considered. Although both exhibit a high degree of adaptability, i.e. by adding or re-ordering machines, job-shops offer limited structural synergies between production resources and product groups. In these, machines are generally arranged in a function-oriented manner, rather than in a product-specific manner (Groover and Jayaprakash, 2016). While this leads to a high degree of process and product individuality, it also results in low throughputs, extended transport distances, and increased control complexities (Zhang et al., 2019). This gives rise to the potential for modularization in order to optimize performance indicators such as throughout times or different (Scholz-Reiter et al., 2011). These modules can be designed in a variety of ways, with some emulating specific functionalities of a job-shop, while others integrate a variety of functions for manufacturing large parts of product groups. Implementing these modules also limits the scope of optimization, allowing significant scalability of the control approach through constrained decision space within every module. This advantage is particularly evident from the organizational perspective.

Organizational perspective (2): Although studies such as Mayer et al. (2021) have conducted an initial investigation of deep learning based control systems for a variety of machines, a deficit can be identified in current research regarding advanced organizational structures. Previous approaches have focused primarily on one level of organization involving machine processes, such as sequentially arranged machines in a flow shop (Heger and Voss, 2021). A more sophisticated approach that considers both production and distribution processes at different levels can contribute significantly to the decomposition of process complexity and performance optimization (ElMaraghy et al., 2012b). In this context, specific and relevant systems and target parameters at the respective sub-levels could be used for decision-making, leading to modules in which manufacturing-specific capacities and control policies can be efficiently applied and bundled. By using standardized modules, collective optimal policies can be trained which, in contrast to centralized systems, can continue to be used after system changes and do not require re-training.

It also became apparent that only a few multi-agent systems have been implemented due to the high control complexity. The number of agents used was also often limited, such as 5 autonomous mobile robots in Malus et al. (2020). This emphasizes the need for an approach that distributes complexity across both the overall organization and individual entities, thus reducing the training, control, and optimization complexity. This could not only realize organizational benefits but also increase scalability through the use of flexible agents. Logistical performance indicators could be directly leveraged by adding or removing the agents. Such distributed intelligence enables optimization at a high level of modular granularity, while still taking global objectives into account.

Algorithmic perspective (3): The application of genetic algorithms or simulated annealing techniques, which find near-optimal solutions, has already gained acceptance in deep learning based production as indicated in the second publication. However, these approaches prove to be sub-optimal for real-time environments (Rauf et al., 2020; Zhou et al., 2020). Therefore, a novel approach is presented as the third perspective of this thesis. It addresses the combination of deep learning methodologies and conventional heuristics, an aspect that has not yet been addressed in deep learning based production control research. The central focus is on differentiating potential action policies from optimal action policies to address the optimization problem independently of the underlying process logic. While deep reinforcement learning techniques operate adaptively in dynamic contexts, conventional algorithms address specific problems in a precise and efficient manner. Integrating both methodologies into a hyper-heuristic offers the potential for fundamental adaptability combined with high optimization efficiency in complex production contexts.

A more comprehensive outline of these perspectives is given in Chapter 6, which seamlessly

transitions the results from the first to the second bundle of publications. The previously mentioned perspectives are combined in this thesis in the form of an artifact that enables a simulative evaluation of the deep learning based control approach. The control framework is structured in multiple layers and integrates several agents, each controlled by a hyper-heuristic. Considering the artifact construction, the methodological challenge arises of how to systematically design the control framework. This question is discussed in Section 3.3.1, followed by the scientific publications of the artifact design in Section 3.3.2 as well as the two validation and add-on publications in Section 3.3.3 and 3.3.4.

3.3.1 Artifact construction methodology

As an essential part of the demonstration and evaluation phase of the *DSRM*, the simulated control framework is of particular importance for the research objective and the later communication. A simulation aims to replicate a system and contains dynamic processes in a model to derive insights that can be transferred to reality. It also permits the replication of systemic interrelationships and allows the evaluation of new production strategies in various disciplines. Due to the defined scope and the regulated system boundaries, process risks can be reduced and monitored, which makes it possible to make conclusions about the efficiency of the tested strategies as early as in an initiatory phase. This not only reduces the technical but also the financial risk by preventing undesired developments and the inefficient use of cost-intensive real systems (Choi and Kang, 2013; White and Ingalls, 2018).

The simulation model reflects the underlying rationale and purpose of the system under consideration. For instance, it allows for the analysis of material flows, the testing of consequences to machine failures, or the evaluation of production ramp-ups through modified parameters. In a discrete event simulation, the production system is described by a static state, which is modified through dynamic changes, such as resource shifts, completion of a production process, or random dynamic events such as new customer orders. The simulation iterates until a target value, such as the predefined simulation time, is reached, after which it can provide the data basis for conducting performance analysis. This analysis can identify bottlenecks, inefficiently used resources, or excess inventory, and contributes to an iteratively increasing system efficiency while adapting real-world parameters (Fowler et al., 2015; Qiao and Wang, 2021). Furthermore, simulation allows the evaluation of system robustness through targeted disturbances and, in the context of a cost- and expenditure-specific consideration, the estimation of system adaptability for changed layouts (Kurinov et al., 2020; Pinho et al., 2021). This not only increases system performance but also promotes a deeper understanding of the system, especially in systems with numerous interdependencies in which system participants act in an increasingly synergetic manner (Uhlemann et al., 2017; Mourtzis, 2020).

To create such a simulation, a number of software solutions are available, including *Arena*, *Siemens Plant Simulation* or *AnyLogic*. In this thesis, an exiting simulation based on the open-source and Python-based simulation environment *SimPy* is used. In particular, this approach supports the dissemination of the artifact according to the *DSRM*, as many researchers and practitioners already rely on Python. The forthcoming deep learning based control artifact and the underlying *SimPy* simulation are freely available, inviting for further research efforts. *SimPy*, compared to alternative software, is notably lightweight yet compatible with Python libraries, notably TensorFlow. Furthermore, with its inherent discrete event traits, *SimPy* permits state extraction for deep reinforcement learning algorithms and executes discrete actions upon process alterations.

The *SimPy* framework has already fostered applications in numerous other fields, including telecommunications (Tinini et al., 2020), supply chain management (Pinho et al., 2021), and many others, in which the computational efficiency of *SimPy* was demonstrated. Additionally, in the field of deep reinforcement learning based production control, some approaches were already successfully implemented using *SimPy*, as in Kuhnle et al. (2020), Samsonov et al. (2022), and Schuh et al. (2023). This highlights the flexibility and suitability of *SimPy* as a training environment for deep learning models. As deep reinforcement learning does not require an existing data-set due to its learning-by-doing training, it does require the availability of a dynamic and interactive training environment.

Rather than an exact replication of a specific system, the simulation is intended to serve as a tool for addressing the research objectives, especially S-RQ2 and S-RQ3. This should ensure the integration and training of the deep reinforcement learning algorithm and enable an iterative optimization cycle. Ensuring sufficient generalizability, scalability, and transferability within the simulation is crucial and must be considered during its design.

3.3.2 Publication 3 - a deep learning based simulation framework

The third publication represents the central control artifact design and development step and demonstrates its control capabilities. In this phase, the developed dynamic production control framework is integrated into a *SimPy* simulation. In contrast to other approaches, no specific scenario was focused on, but the framework was designed to enable a wide range of modular production scenarios.

The first two publications and the associated working hypotheses highlighted the need for an integrative solution that takes into account the identified structural (1), organizational (2), and algorithmic (3) perspectives in complexity reduction. In line with the first working hypothesis from the first publication, a deep reinforcement learning algorithm was chosen to provide both a real-time optimization design as well as an adaptive learning mechanism. To address the

limitations of centralized organizations and single-stage approaches, which require a complete re-training of the neural networks after each layout adjustment, a double-stage exploration of the control paradigm was initiated. The focus was not primarily on optimization, but secondarily on finding an efficient and reliable solution for the integration of the process logic. By means of a hyper-heuristic and the definition of a set of low-level heuristics as the underlying decision instance, it became possible to select appropriate actions already in an untrained state, without intermediate action-masking steps. This contributes to avoiding strong fluctuations in learning stability, especially in multi-agent systems. Although the rules may not initially be optimal, but they are fine-tuned in their sequence during the course of training, always depending on the current production and order states. The basic algorithmic model also offers the flexibility to adjust a wide range of target parameters through the integration of diverse low-level heuristics rule sets. As such, the hyper-heuristic is characterized by a high degree of adaptability to structural changes.

According to the second publication, the hyper-heuristic is applied to multiple agents within the modules. Each agent has its own neural network for decision-making and can retrain, evolve, or simply apply its policy. The neural network only receives those state inputs that are relevant to the module in which it operates. In addition, the reward function is specifically aligned to optimize pre-defined parameters. Agents are seamlessly integrated into the framework, and mechanisms for automatic agent generation that configure the required parameters such as state input sizes and hidden layers are incorporated. This approach not only promotes dissemination by minimizing coding barriers but is also the first to develop a multi-agent based and semi-heterarchical framework that automatically differentiates between manufacturing and distribution agents. The framework supports the integration of agents at different levels, while always keeping the optimization scales within a manageable scope by structurally and organizationally dividing them into modules and assigning them to different agents. As a result, during the course of all simulations, each agent exhibited clear optimization tendencies.

In the further course of the analysis, the focus was particularly on the training process, whereby a positive development of selected parameters, such as the reduction of lead time, was observed. Another central topic was the explainability of the chosen actions, mainly in the context of the considered order backlogs. Since deep learning models are often perceived as black boxes, the structuring in discrete optimization and decision spaces allowed for increased comprehensibility. This allowed, for instance, a detailed analysis of the actions taken when dealing with priority orders. The robustness of the framework was demonstrated while adhering to the given optimization parameters, especially with regard to its ability to handle both machine failures and volatile and significantly increased order volumes.

Also for the first time in deep learning based production control, there was a focus on additional

customer-centric optimization parameters, including the processing of priority and rush orders, which are gaining importance in times of prime services. Such customer-centric parameters were integrated into the reward function and merged with technical parameters to yield a single local and global objective indicator. In an initial and truncated benchmarking, the order-specific multi-criteria optimization was eventually confirmed. From this publication, the following summary statement is derived, wherein *CoBra* signifies *ControlBrain* and alludes to the control framework features rooted in deep learning.

The CoBra framework integrates structural, organizational, and algorithmic perspectives, enabling a threefold control complexity reduction. For the first time, multi-stage processes are integrated into a self-configurable and deep learning based control framework. The underlying hyper-heuristic is characterized not only by its robustness and explainability but also by its high adaptability and the ability to optimize freely definable parameters.

3.3.3 Publication 4 - benchmarking and real-world transfer

The fourth publication builds upon the developed control framework with a primary objective of extensively demonstrating and critically evaluating the artifact. Within this scope, a production system composed of three layers, comprising the manufacturing layer and two distribution layers, was implemented. Established dispatching rules, like the FIFO rules, presently prevalent in the semiconductor industry, served as benchmarks. Throughout the training phase, notable robustness was exhibited, particularly against pronounced fluctuations in WIP figures. In addition, the achievement of improved performance was facilitated by a comparatively simple and modular designed reward function. This function begins by normalizing all the considered influencing variables using a *min-max* normalization. Following this, each variable is individually weighted within its respective value range using powers. Consequently, outliers, like a notably delayed product, are given more emphasis compared to the majority of the key figures that have lesser decision criticality.

In the benchmarks, the hyper-heuristic surpassed an expanded set of benchmark rules (including FiFo local, EDD, etc.) in nearly all categories, thereby improving average performance metrics. Notably, tardiness was reduced by almost 40%. Beyond the performance metrics, the stability of operational optimization was evidenced by the consistently stable rewards received. It was evident that the rules currently in use exhibit notable variations in achieving individual and multiple objectives. Yet, the hyper-heuristic consistently delivers high optimization results irrespective of the order parameters. This consistency is reflected again in the substantially higher rewards. In addition, its scalability was emphasized, adjusting a distribution agents demands merely a fraction of the training effort in contrast to a full retraining, as observed in

centralized control strategies.

Finally, the absence of real application scenarios, noted in the initial two publications, was addressed. The methodology, previously executed in the simulation, was transferred to the hybrid testing environment within the *Center for Industry 4.0*. This center emulated a multi-stage production, wherein the pre-existing simulated machines were transitioned into modules. System outputs, inputs, and storage were instituted with two operational distribution levels at the top level. Even with the subordinate processing of less crucial orders, the high and medium-priority orders were managed more effectively in the real setting.

The proposed hyper-heuristic control framework exhibits robustness and optimization superiority over common dispatching rules, optimizing real-world performance indicators through its decentralized decision-making.

3.3.4 Publication 5 - economic performance evaluation

Previous publications considered numerous technical process parameters but lacked an financial evaluation. There is an interest, especially in corporate practice, in analyzing innovative solutions from an economic perspective. This allows a more comprehensive assessment of the business challenges of a potential integration and consequently a more accurate assessment of the overall risk. Thereby, an advantage of deep reinforcement learning, as described in the previous publication, is the possible integration of a wide range of metrics that are or can be directly or indirectly linked to financial measures.

In this publication, therefore, a novel techno-financial analysis for deep learning based production control was carried out. Customer priority and delivery urgency information were augmented to the order entries, and revenues were dynamically calculated via base and additional fees or penalties. Subsequent examination of the lead time and delay of these orders resulted not only in increased revenue, but more importantly in a 6% increase in profits due to a significantly higher rate of orders processed on time. Despite significantly higher rates of penalties on priority and rush orders, profits from this segment increased significantly, in particular, due to a targeted process focus on these particularly profitable orders. Particularly in times of a growing offering of individual customer services, opportunities are opening up to identify additional sources of revenue. However, this leads to increased process and decision complexity. Thereby, the hyper-heuristic control framework offers an effective approach to integrate customer and finance-oriented metrics into production with high efficiency and optimization performance.

Through the integration of techno-financial parameters into the control framework, taking into account customer-centric services, additional revenue streams and profits can be generated in an easy manner.

In the following, the algorithmic analysis is presented in Chapter 4, succeeded by the organizational examination in Chapter 5. Following the transition detailed in Chapter 6, the artifact is delineated in Chapter 7. It is then subjected to a technical evaluation in Chapter 8, and a financial evaluation in Chapter 9.

4 Publication 1

Deep reinforcement learning in production systems: a systematic literature review

Marcel Panzer^{1a} and Benedict Bender^a

^a *Chair of Business Informatics, Processes and Systems, University of Potsdam,
Karl-Marx-Street 67, 14482 Potsdam, Germany*

ABSTRACT

Shortening product development cycles and fully customizable products pose major challenges for production systems. These not only have to cope with an increased product diversity but also enable high throughputs and provide a high adaptability and robustness to process variations and unforeseen incidents. To overcome these challenges, deep Reinforcement Learning (RL) has been increasingly applied for the optimization of production systems. Unlike other machine learning methods, deep RL operates on recently collected sensor-data in direct interaction with its environment and enables real-time responses to system changes. Although deep RL is already being deployed in production systems, a systematic review of the results has not yet been established. The main contribution of this paper is to provide researchers and practitioners an overview of applications and to motivate further implementations and research of deep RL supported production systems. Findings reveal that deep RL is applied in a variety of production domains, contributing to data-driven and flexible processes. In most applications, conventional methods were outperformed and implementation efforts or dependence on human experience were reduced. Nevertheless, future research must focus more on transferring the findings to real-world systems to analyze safety aspects and demonstrate reliability under prevailing conditions.

Keywords

Machine learning, reinforcement learning, production control, production planning, manufacturing processes, systematic literature review

¹Corresponding author

Submitted to the International Journal of Production Research on 6 April 2021, accepted on 18 August 2021.

4.1 Introduction

Nowadays, companies must cope with mass customization and shortening development cycles that pose major challenges for smart production facilities. They must be capable to operate in highly uncertain market conditions and satisfy the increasingly challenging standards of product quality and sustainability in the shortest possible time. To meet these challenges, Germany launched the Industry 4.0 initiative in 2013 to support the development of flexible and adaptive production systems (Kagermann et al., 2013). Although the initiative's potential and possible impact is huge, Xu et al. (2018) indicate that many of today's Industry 4.0 implementations are not yet applying corresponding advanced techniques such as machine learning. This also becomes apparent in Liao et al. (2017), who states that while modeling, virtualization, or big data techniques are increasingly in the focus of production research, machine learning is not. This impression has already been countered by Kang et al. (2020), who highlighted the broad application landscape of machine learning in modern production and their ability to reach state-of-the-art performance. Going further into detail, our review specifically considers deep Reinforcement Learning (RL) as an online data-driven optimization approach and highlights its beneficial properties for production systems.

The field of machine learning consists of (semi-) supervised, unsupervised, and reinforcement learning. Whereas supervised and unsupervised learning require a (pre-labeled) set of data, RL differs in particular by the learning in direct interaction with its environment. It learns by a trial-and-error principle without requiring any pre-collected data or prior (human) knowledge and has the ability to adapt flexibly to uncertain conditions (Sutton and Barto, 2017). Considering these flexible and desired features in modern production, our paper aims to capture the current state-of-the-art of real or simulated deep RL applications in production systems. Besides, we seek to identify existing challenges and help to define future fields of research.

Already in 1998, Mahadevan and Theodorou (1998) demonstrated the potential of RL in production manufacturing and its superiority in inventory minimization compared to a Kanban system. In recent years, since neural networks are emerging, neural network based RL reached impressive success with Google DeepMind's AlphaGo (Silver et al., 2017), and is now increasingly being transferred to production systems. Based on recently collected sensor data, deep RL enables online data-driven decisions in real-time and supports a responsive reaction-driven and adaptive system design (Han and Yang, 2020). It can increase production stability and robustness and reaches superior performances compared to state-of-the-art heuristics (as in Li et al., 2020).

However, in production related reviews, deep RL has often been considered only in the context of other machine learning techniques as in Kang et al. (2020) or Arinez et al. (2020) and is not mentioned in an industrial intelligence context in Peres et al. (2020), lacking in consolidation of the already obtained results. This is also apparent in other technology fields such as energy

(Mishra et al., 2020), process industry (Lee et al., 2018), or tool condition monitoring (Serin et al., 2020).

In contrast, other disciplines have consolidated the obtained research findings of deep RL and highlighted its adaptive behaviour and the ability to generalize past experiences. This includes communications and networking (Luong et al., 2019), cyber-physical-systems (Liu et al., 2019), economic applications (Mosavi et al., 2020), internet of things (Lei et al., 2020), object grasping (Mohammed et al., 2020), power and energy systems (Cao et al., 2020), robotics (Khan et al., 2020), robotic manipulations tasks (Nguyen and La, 2019), and dynamic task scheduling (Shyalika et al., 2020), which reflects the broad range of research and underlines the ongoing focus on implementing deep RL applications to significantly increase the adaptability and robustness of the respecting processes.

To the best of our knowledge, this is the first attempt to capture general applications of deep RL in production systems. We intend to provide a systematic overview of ongoing research to assist scholars in identifying deep RL research directions and potential future applications. The review also serves practitioners in considering possible deployment scenarios and motivate them to transfer research findings to real-world systems. For this purpose, we attempt to answer the following research questions.

- RQ1: What are deep RL applications in specific production system domains?
- RQ2: What are current implementation challenges of deep RL in production systems?
- RQ3: What future research needs to be conducted to address existing challenges of deep RL in production systems?

The paper is structured as follows. Section 2 describes the basics of deep RL and gives an overview of essential algorithms. Section 3 defines the methodology and the conceptual framework that guides the literature review. Section 4 answers RQ1 based on the conducted review and provides the basis for Section 5, which analyzes specific barriers and challenges (RQ2) and outlines fields for future research to address these (RQ3). Section 6 discusses the results and provides managerial insights and given limitations. Finally, a conclusion is given in Section 7.

4.2 Introduction to reinforcement learning

Reinforcement learning (RL) is a subcategory of machine learning and distinguishes itself from supervised and unsupervised learning in particular by the trial and error learning approach in direct interaction with its environment (Sutton and Barto, 2017). It does not need supervision or a pre-defined labeled or unlabeled set of data and comes into consideration whenever challenges have to be met in dynamic environments that require a real-time and reaction-driven decision-

making process. It is able to generalize its previously learned knowledge (Wang et al., 2020) and enables an online adaptation to changing environmental conditions by sequential decision-making (as in Palombarine et al., 2019).

In RL, the agent learns a policy that outputs an action according to the received state as illustrated in Figure 4.1. To achieve this, conventional RL often employs a Q-table to map the policy, which requires discretization of state and action spaces. The Q-table lists Q-values that quantify the action quality of performing an action in a given state, which are updated through ongoing training of the agent. In many cases, Q-learning outperformed several conventional approaches such as FIFO in flow production scheduling, which reduced makespan, whereby states were described by machine runtimes and buffer occupancy, and executable actions by unit movements (Lee and Kim, 2021). Other successful examples are the superior performance compared to multiple scheduling approaches in an adaptive assembly process (Wang et al., 2020) by choosing scheduling rules based on waiting queues, interval times, remaining processing time, and processing status, or the condition-based maintenance control in Xanthopoulos et al. (2018) that reduced costs compared to a Kanban method by authorizing maintenance actions based on finished goods, backorders, and facility deterioration states.

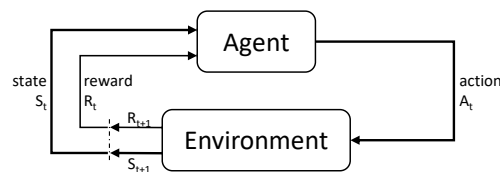


Figure 4.1 Agent-Environment Interaction; Sutton and Barto (2017)

However, the required action and state discretization impose the curse of dimensionality in high-dimensional problem spaces, which causes an exponentially increasing table size and leads to high iterative computational costs, low learning efficiencies, and degraded performances (Bellman, 1957). To address this, and as proposed by Lee and Kim (2021), among others, deep RL attempts to solve this problem by combining the advantages of RL with those of deep learning. In deep RL, the policy is mapped by a neural network as a function approximator, which is capable of processing large amounts of unsorted and raw input data (Lange et al., 2012). (Deep) RL can be further classified into model-free and model-based algorithms. Model-based algorithms such as the AlphaZero get or learn a model of the environment to predict next values or states (Silver et al., 2017). In contrast, model-free algorithms neither learn the dynamics of the environment nor a state-transition function (Sutton and Barto, 2017). Model-free algorithms, as the major group in this review, can be further classified into policy-based, value-based, and hybrid algorithms. Policy-based algorithms such as a PPO provide a continuous action space and try to directly map a state to an action by building a representation of the actual behavior policy (Sewak, 2019b). In contrast, value-based algorithms such as a DQN learn a

value function for discrete action spaces to evaluate each of the potential actions (Watkins and Dayan, 1992). Algorithms like the DDPG utilize a hybrid actor-critic structure which combines previous methods advantages (Lillicrap et al., 2016). Other possible modifications such as a prioritized experience replay, which takes particular account of important experiences during updates, can be integrated into the deep RL framework (Schaul et al., 2016).

Besides basic algorithmic settings, particular consideration is required for the choice of hyperparameters. The discount factor, which determines the relevance of short-term or distant future rewards, the learning rate, which determines the balance between learning speed and stability, and other algorithmic as well as neural network parameters in deep RL strongly affect the final performance. Specific considerations should also reflect the optimal design of the state/action space and reward design. Appropriate interference between these can lead to optimal system behavior and help in the search for optimal control strategies (Sewak, 2019a). In particular, the reward function must be designed concerning the agent's objective and system dynamics and must be able to account for short- as well as long-term outcomes. For further algorithmic insights we would like to refer to Wang et al. (2020) or Naeem et al. (2020) for an extended introduction and in-depth analysis of (deep) RL algorithms.

Initially limited to the Atari platform in Mnih et al. (2013), deep RL is being deployed in an increasing number of applications which benefit from its flexibility and online adaption capabilities. Potential applications such as smart scheduling benefit from the distributed multi-agent capabilities and collaborative properties, which could significantly increase robustness as proposed in Rossit et al. (2019). It makes deep RL being a promising technique to improve the performance of modern production systems and enable the transition towards industry 4.0. However, unlike other algorithmic overviews or the general descriptions of machine intelligence applications in production, the intersection of deep RL in different production system domains was not specifically covered. To address this gap and highlight the benefits, an representative review of the intersection might assist to identify individual applications, challenges, and future fields of research.

4.3 Research methodology

This section outlines the basic literature review process of deep RL applications in production systems. To ensure a systematic and representative review, we follow Tranfield et al. (2003) and Thomé et al. (2016) who provide guidelines for the content analysis. This enables a consolidation and evaluation of existing literature and provides the state-of-the-art in the focused domain at a given time. The consolidation shall assist researchers and others to identify research gaps and provides research incentives and managerial insights (Petticrew and Roberts, 2006).

According to the guideline proposed by Thomé et al. (2016), the systematic literature review

(SLR) can be organized into 8 (iterative) steps. These main steps are outlined sequentially in Figure 4.2 and will be considered in the subsequent review process.

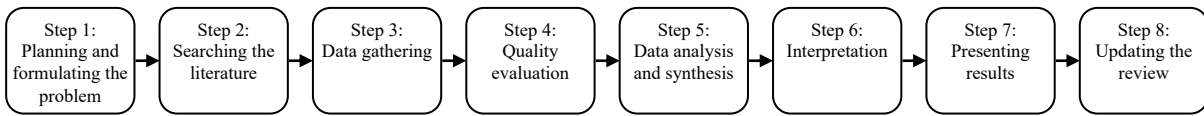


Figure 4.2 Eight step approach to conduct a SLR

4.3.1 Review focus

The formulation of the research questions and clarification of the problem, is outlined in Section 4.1. The composition of the review team consisted of the two authors who worked through each step separately and finally combined their work. To define the scope of the problem and simplify the review process, the more in-depth planning relies on Brocke et al. (2009) and follows the associated taxonomy framework by Cooper (1988), Table 4.1. The gray highlighted cells represent the selection of underlying characteristics of this SLR and the associated goals and foci. Following the taxonomy, this SLR focuses on presenting existing applications and

Characteristic	Categories			
(1) Focus	Research outcomes	Research methods	Theories	Applications
(2) Goal	Integration		Criticism	Central issues
(3) Perspective	Neutral representation		Espousal of position	
(4) Coverage	Exhaustive	Exhaustive and selective	Representative	Central / pivotal
(5) Organisation	Historical		Conceptual	Methodological
(6) Audience	Specialized scholars	General scholars	Practitioners / politicians	General public

Table 4.1 Taxonomy framework of the SLR

achieved research results of deep RL in production systems (1). Its goal (2) is to present existing research in an integrative and synthesizing manner while highlighting central future application and maturity issues. We try to maintain a neutral perspective (3) and provide a representative coverage of our focused content (4). The organization of the review is conceptually designed (5). In particular, the application concept in the respective discipline shall be reflected rather than the historical or methodological organization. Finally, we try to address a broad audience (6). We do not explain technical details in-depth, which benefits general scholars and practitioners, and at the same time, we try to give specialized scholars an overview of their quickly expanding research field. Altogether, we intend to clarify the relevance of deep RL in production systems and to provide stimuli for potential applications.

4.3.2 Literature search

For conducting the review, we initially defined the search terms and determined the underlying databases. The found literature is then filtered to obtain the final subset for the later in-depth analysis.

4.3.2.1 Phase 1 - database and iterative keyword selection

The search databases utilized in our review are the Web of Science (all fields), ScienceDirect (title, abstract or author-specified keywords), and IEEE Xplore (journals), similar to Lohmer and Lasch (2020) or other scholars.

To ensure a representative coverage of the research literature, we defined the keywords in an interactive process and had a rather broad focus, which comprised an algorithmic, a general, and a more specific domain. Within the iterative process, besides *production* and *manufacturing*, we incorporated *assembly*, *automation*, and *industry* as general keywords. To avoid missing any sub-discipline, additional subsets were incorporated into the search and included *quality control*, *maintenance*, and others as listed in Table 4.2. Because the term of *deep RL* is not always mentioned, we also linked RL with *artificial intelligence*, *deep learning*, and *machine learning*.

Algorithmic keywords	General keywords	Specific keywords
$\left[\begin{array}{l} \text{Deep RL}^a \text{ OR} \\ \text{RL}^a \text{ AND } \left[\begin{array}{l} \text{Artificial intelligence OR} \\ \text{Deep learning OR} \\ \text{Machine learning} \end{array} \right] \end{array} \right]$	$\text{AND } \left[\begin{array}{l} \text{Assembly OR} \\ \text{Automation OR} \\ \text{Industry OR} \\ \text{Manufacturing OR} \\ \text{Production} \end{array} \right]$	$\text{OR } \left[\begin{array}{l} \text{Logistics OR} \\ \text{Maintenance OR} \\ \text{Process control OR} \\ \text{Quality control OR} \\ \text{Real-time control OR} \\ \text{Tool control} \end{array} \right]$

^a RL = Reinforcement Learning

Table 4.2 Defined keywords for the SLR

4.3.2.2 Phase 2 - defining inclusion and exclusion criteria

To systematically narrow the scope and ensure a high review quality, we defined several inclusion and exclusion criteria. For quality reasons, we only considered publications from peer-reviewed journals, proceedings, conference papers, and books (Light and Pillemer, 1984; Durach et al., 2017). We excluded workings papers, pre-prints and other non-peer reviewed publications. We also excluded publications that were not written in English and since significant successes of

deep RL were especially observed with the publication from Mnih et al. (2013), we only included papers that were published after 2010.

A thematic definition of the inclusion and exclusion criteria is ensured by the defined research questions and taxonomy framework. Based on our target to identify industrial deep RL applications, we excluded papers that focus primarily on the development of methodologies, theories, or algorithms without transferring the results to a production use case. Review papers were used as appropriate to identify potential additional studies of relevance. Given the focus of our study, we reviewed papers that address the direct application of deep RL in real or simulated production environments and seek to leverage system performances. Only papers, that apply deep RL methods for policy approximation were considered. In contrast, papers dealing with conventional RL methods (i.e. a Q-table) were not reviewed.

4.3.2.3 Phase 3 - conducting the literature search

The literature search was conducted from December 2020 and a final extract was retrieved from the mentioned databases on February 10, 2021. A summary of the whole process is given in Figure 4.3 and starts with the aggregation of the articles found in the three databases. In total, 1255 papers were collected based on the defined keywords. Duplicates were removed and years filtered before applying in-depth thematic criteria.

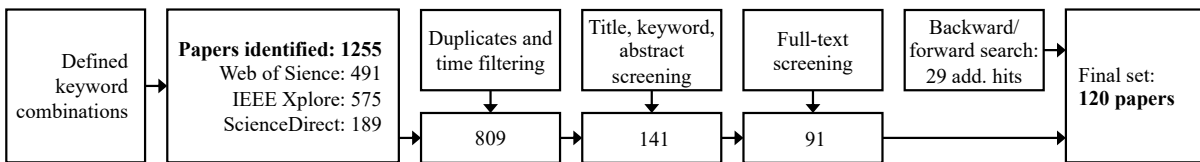


Figure 4.3 Conducted review process

According to Thomé et al. (2016), to ensure a high search quality, we examined the remaining 809 papers by their title, keywords, and abstract regarding the defined inclusion and exclusion criteria and the research questions. If possible, we already captured the applied algorithms, considered processes, and the application objective. In this step, many papers were excluded due to a missing production context or a non-deep RL implementation, which reduced the number to 141 papers. In the next step, we conducted a full-text review based on the same criteria. Besides capturing the first essential information for the later analysis, the full-text review provided the remaining 91 papers as a basis for the subsequent backward/forward search.

Following the review structure proposed by Webster and Watson (2002), the backward/forward search is an important extension to the previously conducted keyword-based search. Similarly, Greenhalgh and Peacock (2005) underlines the importance of this last literature search step to identify further interdisciplinary literature beyond the self-defined search scope. After this final

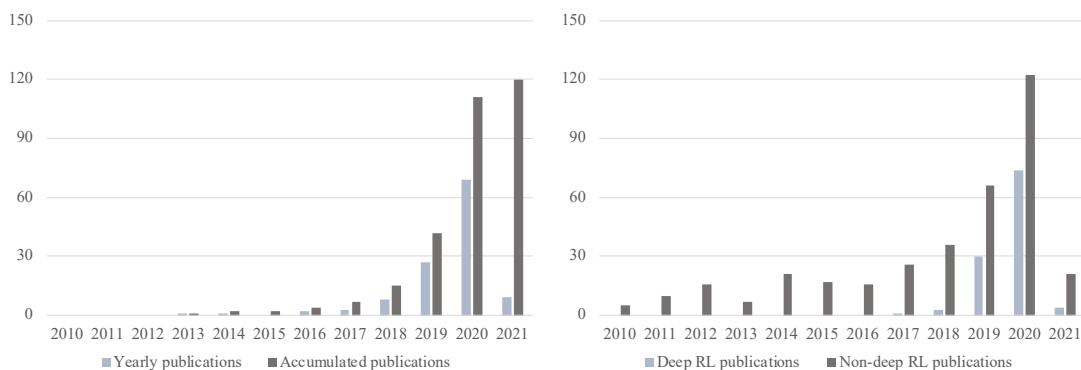
search, we identified 29 additional papers in scope, resulting in a total set of 120 papers.

4.3.2.4 Phase 4 - data gathering

To conduct the subsequent literature analysis, we developed a concept matrix with regard to Thomé et al. (2016) and Webster and Watson (2002), that focused on the initial research questions. The categorization and coding of the final data set was based on the production discipline, industry or process background including the specific application, optimization objective, applied deep RL algorithm and neural network, benchmark results, and the application in a simulated and/or real environment.

4.3.3 Analysis of yearly and outlet related contributions

An initial analysis based on publication years allows conclusions to be drawn about the general research development. Figure 4.4(a) indicates a strong increase of deep RL publications in a production context since 2018. While 3 papers were published in 2017, there were already 8 in 2018, 27 in 2019, and 69 in 2020. In 2021, 9 papers were published in January / February up to the time of the database query. This indicates the growing relevance of deep RL in a production context and its rising attention within the research community.



(a) Yearly and accumulated deep RL publications (b) Yearly deep and non-deep RL publications

Figure 4.4 Analysis of yearly deep RL publications, 2021 includes Jan./Feb.

One reason for this development could be due to Mnih et al. (2013) as described earlier, who laid a foundation for high performance deep RL in 2013. This also becomes evident in Figure 4.4(b) in which we compared deep and non-deep RL publications in the Web of Science database (with keywords from Table 4.2, non-reviewed). While in 2017 1 deep RL and 26 non-deep RL papers were published, there were 74 deep and 122 non-deep publications in 2020. While this suggests a significant increase in both fields, it highlights the ongoing focus on neural network based RL.

4 Publication 1 - Deep reinforcement learning based production

Figure 4.5 lists the most frequently cited outlets with more than three published papers from 2010 to 2021. Most papers were published in journals (92, 76%) followed by conference papers (14, 12%) and proceedings (14, 12%). In total, the papers were accessed from 54 journals, 16 conferences, and 4 proceedings. This not only indicates the high quality of the selected papers, but also reflects the broad application range of deep RL in various fields of production related systems.

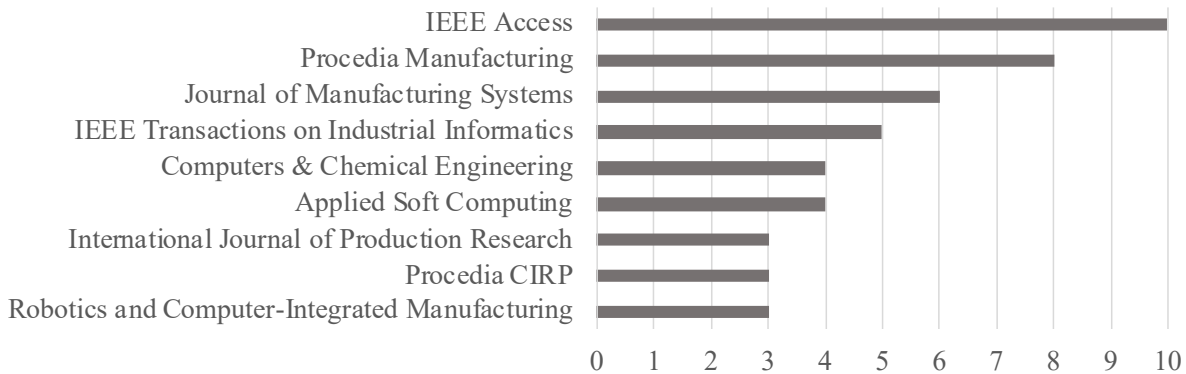


Figure 4.5 Number of publications per outlet; 2010-2021

4.4 Literature analysis

To address RQ1 we first outline existing domains of deep RL applications in production systems. Figure 4.6 contains the disciplines obtained after the final iterative review step and the respective number of publications.

Most of the reviewed papers were published in the field of process planning followed by scheduling, and assembly. The application landscape covers almost all relevant disciplines in a production system and confirms the ability of deep RL to address a variety of tasks. The further analysis is organized according to the structure indicated in Figure 4.6.

4.4.1 Process control

To circumvent a conventional model-based approach and an online adaption to continuous process modifications, Noel and Pandian (2014) initially developed a deep RL approach to control the liquid levels of multiple connected tanks. The controller minimized the target state difference and adjusted inlet flow rates between multiple tanks accordingly. Whereas conventional methods struggle to compensate for large changes in system parameters, the deep RL approach optimized control and simultaneously reduced process fluctuations and overshoot. Spielberg et al. (2017) and Spielberg et al. (2019) proposed a model-free controller design for single-/multiple-input and -output processes that was applied to various application scenarios. The controller reduced

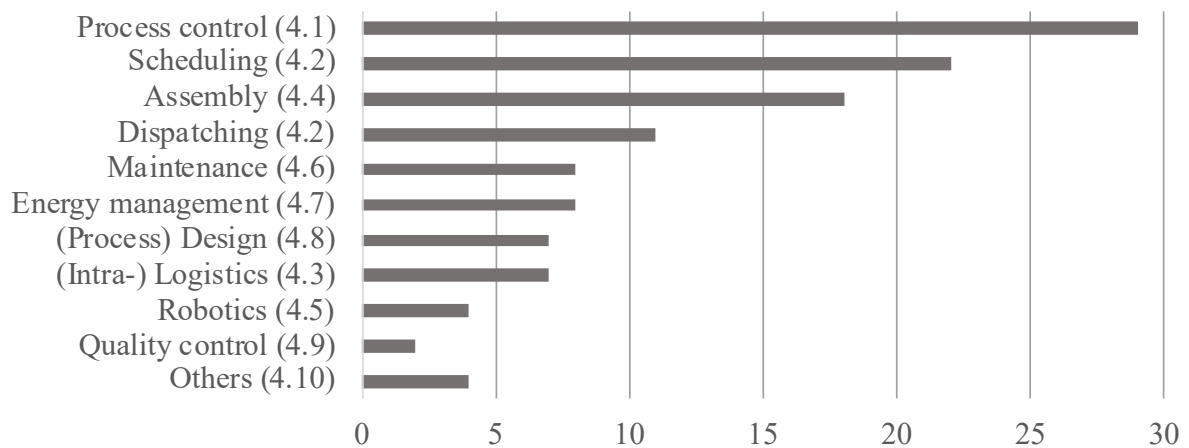


Figure 4.6 Number of publications allocated to the production disciplines

maintenance efforts and computation costs and was capable of regulating the desired states and set-points. Similarly, deep RL approaches in chemical-mechanical polishing (Yu and Guo, 2020) and microdroplet reactions (Zhou et al., 2017), outperformed conventional methods in minimizing process deviations and enabled an interactive and data-driven decision making and online process control which reduced temporal and monetary expenditures.

Deep RL outperformed 13 out of 16 conventional benchmarks and improved system performances. To reach such performance, a reward function is required, which transforms process targets into rewards, allowing to learn the optimal policy. The reward design can be based on different target variables, such as real-time profits (Powell et al., 2020), cost-per-time function (Quah et al., 2020), or similarity measures based on specified performance criteria (He et al., 2020). The individual goal-oriented design enables a broad application in further applications such as flotation processes to reduce non-dynamic drawbacks of model-based approaches (Jiang et al., 2018), in laser welding to increase process repeatabilities (Masinelli et al., 2020), and others, or in injection molding to broaden up narrow process windows of conventional methods in ultra-high precision processes (Guo et al., 2019). A detailed list of all process control applications and related publications can be found in Table 4.3. Besides, the table lists the applied algorithm and, if conducted, the performance result compared to conventional benchmarks. A significant portion of the papers conducted their testings in simulated environment and only 4 papers conducted real world testings. The implementation hurdles in the area of process control are large and require highest levels of process reliability, which prevents a rapid implementation for research purposes in real processes.

The majority of publications (79%) utilized policy-based or hybrid algorithms, which benefit from a continuous action space and do not require action discretization. Thus, process parameters can be set smoothly and do not require a step-wise control approach. Beyond that, the motivation

4 Publication 1 - Deep reinforcement learning based production

	Application	Algorithm	Superiority	Source
1	Batch process	DDPG	Superior	Xu et al. (2020)
2	Brine injection process	Actor-critic	Superior	Andersen et al. (2019)
3	Liquid moulding process	DQN	Superior	Szarski and Chauhan (2021)
4	Chemical microdroplet reactions	Actor-critic	Superior	Zhou et al. (2017)
5	Color fading	Actor-critic	Superior	He et al. (2020)
6	Continuously stirred tank reactor	Actor-critic	-	Pandian and Noel (2018)
7	Continuously stirred tank reactor	DQN	Comparable	Powell et al. (2020)
8	Continuously stirred tank reactor	Actor-critic	Comparable	Quah et al. (2020)
9	Interacting tank liquid level control	Actor-critic	Superior	Noel and Pandian (2014)
10	Double dome draping	Actor-critic	-	Zimmerling et al. (2020)
11	General discrete-time processes	Actor-critic	-	Spielberg et al. (2019)
12	General discrete-time processes	Actor-critic	-	Spielberg et al. (2017)
13	Goethite iron-removal process	DDPG	-	Chen et al. (2020)
14	Hematite iron ore processing	DQN/PG	-	Li et al. (2020)
15	Laser welding	DQN	-	Günther et al. (2016)
16	Laser Welding	Value based	-	Jin et al. (2019)
17	Laser welding	Value based	-	Masinelli et al. (2020)
18	Laser Welding	Actor-critic	Superior	Zou and Lan (2020)
19	Metal sheet deep drawing	ANN-PSO/PPO	-	Dornheim et al. (2020)
20	Non-lin. semi-batch polymerization	DQN/DPG/REI	Comparable	Ma et al. (2019)
21	One-stage mineral grinding	Actor-Critic	-	Lu et al. (2016)
22	Optical lens manufacturing	PPO	Superior	Guo et al. (2019)
23	Propylene oxide batch polymerization	DQN	Superior	Yoo et al. (2021)
24	Rot. chemical mechanical polishing	DDPG	Superior	Yu and Guo (2020)
25	Single-cell flotation process	DDPG	Superior	Jiang et al. (2018)
26	Single-cell flotation process	DDPG	-	Jiang et al. (2019)
27	Single-circuit ball mill grinding	DRO	Superior	Guo et al. (2019)
28	Tempered glass manufacturing	Actor-critic	-	Mazgualdi et al. (2021)
29	Well surveillance	Actor-critic	Superior	Tewari et al. (2020)

Table 4.3 Summary of deep RL applications in process control

for applying deep RL is often an inaccurate mapping of conventional methods that cannot adequately cope with non-linearities (Lu et al., 2016) or relies too much on error-prone expert knowledge (Mazgualdi et al., 2021). With their adaptive and non-discretized action space, deep RL can thus avoid waste, especially in sensitive processes, and keep processes stable, which might be problematic with static or human-based process modeling (Andersen et al., 2019).

4.4.2 Production scheduling and dispatching

Already in 1995, Zhang and Dietterich (1995) described a neural network based job-shop scheduling approach which demonstrated superior performance and reduced costs for manual system design. Followed by other approaches such as Riedmiller and Riedmiller (1999) or Gabel and Riedmiller (2007), the advantage of deep RL in production planning and control was emphasized early on, but could not prevail, among other reasons, due to the lack of computational resources.

4.4.2.1 Production scheduling

The complexity of production scheduling is caused by high uncertainties regarding customized products, shutdowns, or similar. To cope with the complexities and to reduce human-based

decisions, Lin et al. (2019) proposed a multi-class DQN approach that feeds local information to schedule job shops in semiconductor manufacturing. Based on the edge framework, the DQN demonstrated superior performance and reduced makespans, and average flow times. To reduce the high setup and computational costs of conventional solutions in job-shop scheduling, Liu et al. (2020) and Baer et al. (2019) adopted a self-learning multi-agent approach to meet local and global production objectives and to ensure an increased adaptation to prevent rescheduling cost. To train multiple agents Baer et al. (2019) employed a multi-stage learning strategy in which a single agent was trained locally first, while others applied chosen heuristics. Subsequently, all agents were trained individually and finally optimized together towards a global goal. Besides, Baer et al. (2020) demonstrated the agent's ability to adapt to new scenarios and proofed its scalability. The training of 700 scheduling topologies took only twice as long as the training of a single one. The deep RL policy learned basic task principles and modified its policy slightly concerning the new task specifics and thereby reduced re-configuration times and costs compared to conventional methods.

In total, 89% of the benchmarked deep RL implementations increased the scheduling performance and reached lower total tardiness, higher profits, or other problem-specific objectives as indicated in Table 4.4. Zhou et al. (2020) managed to minimize the completion time of all given tasks, for random incoming orders. Similarly, Wu et al. (2020) demonstrated that deep RL based rescheduling can operate faster and more efficiently than heuristics. The deep RL approach reduced CPU times remarkably for the high volatile medical mask production in times of Covid-19. Besides mask production, deep RL demonstrated superior performances in batch processing which reduced tardiness for repair scheduling operations (Palombarini and Martinez, 2018; Palombarini and Martínez, 2019), in chemical scheduling to increase profitability and deal with fluctuating prices, shifting demands, and stoppages (Hubbs et al., 2020), and in paint job scheduling to minimize costs of color changeovers within the automotive industry (Leng et al., 2020). Discipline-specific scheduling objectives were addressed by Lee et al. (2020), who increased sustainability and minimized tardiness in injection mold scheduling, or by Xie et al. (2019) who reduced total throughput time and lateness in single-machine processes.

From an industry perspective, the semiconductor industry is one of the most competitive and capital-intensive. Interconnected machines must operate at full capacity, and production schedules need to be continuously optimized online (Kang et al., 2020). Due to a large number of machines and process steps, the die attach and wire bonding process poses a major challenge that cannot optimally be solved by single heuristics. To cope with the complexities, Park et al. (2020) feed all relevant process information such as setup status continuously into the PPO neural network. It was able to outperform conventional heuristics such as shortest setup or processing time and reduced total makespan and computation times after training. A further increase in generalization was reached by Park et al. (2021) by applying a graph neural network (GNN).

The GNN learned the basic spatial structure of the problem in form of a graph that could be transferred to new problems and adapted its mapped policy. Thus, the GNN-PPO was not only able to adapt to novel job shop problems, but also outperformed algorithms that were configured scenario-specific.

Based on all reviewed papers in the field of production scheduling, 67% applied value-based algorithms. These assume a discrete action space, which must be determined beforehand. However, for scheduling-related problems, the action space can often be discretized according to possible transition actions such as *transfer* or *idle* (Shi et al., 2020). It is noticeable that in comparison to process control, even fewer approaches have been adopted in a real environment. In scheduling and additionally in the subsequent dispatching and logistics section, a fast implementation of the scheduling policies in an established production environment would be complex and increase research efforts significantly.

4.4.2.2 Production dispatching

Personalized production has an enormous impact on the complexity of production control due to individual product configuration options. Depending on the customer requirements, the products must be dispatched to where they can be processed, under consideration of several technical and logistic constraints and optimization variables (Waschneck et al., 2018).

To meet the requirements in wafer fabrication dispatching, Altenmüller et al. (2020) implemented a single-agent DQN that processed 210 data points as a single state input (such as machine loading status or machine setup). This enabled the DQN to meet strict time constraints better than competitive heuristics (TC, FIFO) while reaching predefined work-in-progress (WIP) targets as a secondary goal. Stricker et al. (2018) and Kuhnle et al. (2020) proposed a single-agent adaptive production control system that maximized machine utilization and reduced lead and throughput times compared to conventional methods that struggle partially known environments. Waschneck et al. (2018) proposed a multi-agent system to meet flexible objectives within wafer processing and enable higher flexibilities with fewer delays. Similar to Waschneck et al. (2018), the algorithms targeted plant-wide parameters to reduce the risk of a local operation optimization. Besides, the simulations considered complex job shop specifics such as re-entrant flows, sequence-dependent setups, and varying processing times, reaching comparable performances against multiple heuristics. For general production dispatching, Dittrich and Fohlmeister (2020) introduced a multi-agent system with global performance objectives to avoid local optimization tendencies. Although the agents received detailed local state information, they not only selected the fastest local dispatching actions but also improved the global logistics performance. Further, the distributed agents enabled real-time responses, a feature also emphasized by Kumar et al. (2020) for the short-term value stream adaptation in a copper mining complex. Based on the

current mining process and component data, the single-agent framework allowed to deliver continuous updates regarding the latest plant status and increased the expected net present value by 6.5%. Considering capital constraints in production, Kanban or Conwip cards are often employed to limit WIP levels. As an alternative to those conventional pull production controls and to optimize local and global production indices in parallel, Silva and Azevedo (2019) proposed a deep RL algorithm that balanced conflicting throughput and WIP level targets. Despite the trade-off between these, WIP levels were reduced by 43% compared to conventional methods through dynamic adjustments without affecting the total throughput.

A mixed-rule dispatching approach was proposed by Luo (2020) and Heger and Voß (2020) for general job shop systems to enable a dynamic dispatching adaptation to changing production conditions. Based on current state information, the algorithm determined which of the predefined rules (i.e. Heger and Voß: *SPT*, *EDD*, *FIFO*, *SIMSET*) should be activated in the current situation to reduce the mean and total tardiness. Table 4.4 briefly summarizes the reviewed literature and contains the implemented algorithms of the respective papers and their performance results compared to conventional methods.

4.4.3 (Intra-) Logistics

The review results for intralogistics are briefly summarized in Table 4.4. Beginning with Malus et al. (2020), an intralogistics-related dispatching solution was implemented to meet real-time requirements and handle a rapidly changing production by utilizing autonomous mobile robots (AMRs). Based on the observations of the individual agents, they could negotiate with each other and virtually raised bids for orders. Similarly, Feldkamp et al. (2020) simulated a self-regulating modular production system. Depending on current job information, station status, and others, the algorithm determined the optimal machine and reduced lead times compared to the benchmarked methods. In another approach, Hu et al. (2020) implemented a mixed rule dispatching approach that determines the dispatching rule (*FCFS*, *STD*, *EDD*, *LWT*, *NV*) for an automated guided vehicle (AGV) depending on its observed state which reduced the makespan and delay ratio by approximately 10% compared to the benchmarks.

Regarding conveyor systems, Kim et al. (2020) proposed a deep RL control to enable a faster product distribution for a 3-grid sorting system in which all of the 9 fields and corresponding inputs and outputs were controlled by respective agents. The pick and place of items from a conveyor belt into baskets was investigated by Hildebrand et al. (2020). To reach a pre-defined weight, the trays should still be filled quickly to prevent dead-locks. Without an initial parameter tuning, which would have been necessary for conventional probability-based methods, the PPO reached a remarkable success rate of 48% after training. A further collaborative task completion of two robots for adaptive stacking was considered in Xia et al. (2020) which highlighted the

4 Publication 1 - Deep reinforcement learning based production

flexible virtual commissioning abilities and demonstrated an above-human performance.

Scheduling				
Application	Algorithm	Superiority	Source	
30	Chemical scheduling	A2C	Superior	Hubbs et al. (2020)
31	Cloud manufacturing	DQN	Superior	Dong et al. (2020)
32	Cloud manufacturing	PG	Superior	Zhu et al. (2020)
33	Dynamic scheduling	DQN	-	Zhou et al. (2020)
34	Dynamic scheduling	DQN	Superior	Hu et al. (2020)
35	Flow shop scheduling	Reinforce	Superior	Wu et al. (2020)
36	Job-shop scheduling	DDPG	Superior	Liu et al. (2020)
37	Job-shop scheduling	PPO	Superior	Park et al. (2021)
38	Job-shop scheduling	DQN	-	Baer et al. (2020)
39	Job-shop scheduling	-	-	Baer et al. (2019)
40	Job-shop scheduling	DQN	Superior	Zhou et al. (2021)
41	Job-shop scheduling	DDDQN	Superior	Han and Yang (2020)
42	Job-shop scheduling	(M)DQN	Superior	Lin et al. (2019)
43	Lot scheduling	PPO	Superior	Rummukainen and Nurminen (2019)
44	Mold scheduling	DQN	Superior	Lee et al. (2020)
45	Multichip production	DQN	Superior	Park et al. (2020)
46	Packaging line scheduling	DQN	Superior	Chen et al. (2019)
47	Paint job scheduling	Double DQN	Superior	Leng et al. (2020)
48	Parallel, re-entrant production	DQN	Comparable	Shi et al. (2020)
49	Rescheduling	DQN	Superior	Palombarini and Martínez (2018)
50	Rescheduling	DQN	Superior	Palombarini and Martínez (2019)
51	Single machine scheduling	DQN	Comparable	Xie et al. (2019)

Dispatching				
Application	Algorithm	Superiority	Source	
52	General job-shop	DQN	Comparable	Dittrich and Fohlmeister (2020)
53	General job-shop	double DQN	Superior	Luo (2020)
54	General job-shop	DQN	Comparable	Heger and Voß (2020)
55	General job-shop	Reinforce	Superior	Zheng et al. (2020)
56	Mining materials flow	PG	Superior	Kumar et. al (2020)
57	Wafer fabrication	DQN	Comparable	Waschneck et al. (2018)
58	Wafer fabrication	TRPO	Comparable	Kuhnle et al. (2020)
59	Wafer fabrication	DQN	Comparable	Waschneck et al. (2018)
60	Wafer fabrication	DQN	Superior	Stricker et al. (2018)
61	Wafer fabrication	DQN	Superior	Altenmüller et al. (2020)
62	WIP bounding	DQN	Superior	Silva and Azevedo (2019)

(Intra-) Logistics				
Application	Algorithm	Superiority	Source	
63	AGV scheduling	DQN	Superior	Feldkamp et. al (2020)
64	AGV scheduling	DQN	Superior	Hu et al. (2020)
65	AMR dispatching	TD3	Superior	Malus et al. (2020)
66	Item batching into trays	PPO	-	Hildebrand et al. (2020)
67	QoS service composition model	duelingDQN	Superior	Liang et al. (2021)
68	Syringe filling process	doubleDQN	Superior	Xia et al. (2020)
69	Three-grid sortation system	DQN	-	Kim et al. (2020)

Table 4.4 Summary of deep RL applications in production scheduling, dispatching, and (intra-) logistics

4.4.4 Assembly

A significant share of the reviewed assembly-related papers focused on the peg-in-hole task (56%). It comprises the insertion of a specific object into a hole under defined assembly conditions, utilizing a robotic arm in most cases. To avoid large fluctuations in execution and to ensure a

high level of safety, most papers utilized a post-processing force controller that processes the neural network outputs (Kim et al., 2020)).

The deep RL implementation was often motivated by disadvantages of conventional algorithms such as limited adaptability (Li et al., 2019), complex online optimization processes (Inoue et al., 2017) or the need for re-programming in case of new tasks due to hand-engineered parameters (Luo et al., 2018).

Beginning with hole position uncertainties, Beltran-Hernandez et al. (2020) trained a transfer learning supported deep RL algorithm to fit a cuboid-shaped plug into a hole with 0.1mm tolerance and reached a 100% success rate. Also, the insertion of electronic connectors (success rate: 65%), Lan connectors (60%), and USB connectors (80%) was investigated but reached lower success rates.

For contact-rich tasks Kim et al. (2020), Lämmle et al. (2020), and Beltran-Hernandez et al. (2020) proposed a imitation learning supported force-regulated approach consisting of hole approach, alignment, and insertion for the square-shaped peg assembly (tolerance: 0.1mm). For smaller tolerances in high-precision assembly, Zhao et al. (2020) and Inoue et al. (2017) reached success rates of up to 86.7% and 100% with tolerances of 0.02mm and 0.01/0.02mm, respectively. Whereas Zhao et al. (2020) thereby minimized the number of required interactions, Inoue et al. (2017) was able to significantly reduce online parameter adjustment efforts that are required by conventional methods. The insertion of the peg into a deformable hole with a smaller diameter was investigated by Luo et al. (2018) who utilized a force-torque controller for task completion.

For the double peg-in-hole task and a tolerance of 0.04mm for each peg, Xu et al. (2019) reached a success rate of 100%. In case of a changed start position, the success rate was reduced and required re-training. Not only stiff but also dangling pegs have been investigated by Hoppe et al. (2019), that required a contact-rich assembly. Through a combined global state space exploration and learning by demonstration strategy, the DDPG reached a 100% success rate. The learning by demonstration was also investigated by Wang et al. (2020), taking into account bigger arm and fine hand motions. Assuming different peg objects with a tolerance of 4.2mm and a one-shot demonstration, a success rate of 67% was reached. The assembly of a circuit breaker housing was addressed by Li et al. (2019). Divided into free movement, movement under contact, and insertion phase, the two housings with four mounting spots were assembled with success rates of up to 88%.

Other deep RL applications included the vision-based insertion of a Misumi Model-E connector (success rate: up to 100%, as in Schöttler et al., 2020), a long/short-term memory supported shoe-tongue assembly (up to 97%, Tsai et al., 2020), and a space-force controller and force/torque information supported gear-set assembly (up to 100%, Luo et al. 2019).

A multi-component assembly sequence planning approach to increase human-robot collaboration efficiencies was proposed by Yu et al. (2020). Assuming an adjustable desk as an example, the scheduling process was transformed into a chessboard-shaped planning structure that was able to complete planning significantly faster than conventional methods. Besides, to increase planning efficiencies while obtaining better generalization, Zhao et al. (2019) combined a DQN with curriculum learning and parameter transfer techniques. Compared to a simple DQN, this increased the learning speed and adaptability to other environments.

Remarkably, assembly research conducted the highest number of real-world testings, and 15 out of 18 reviewed papers transferred results to reality under the prevailing conditions and on real hardware. Deep RL based assembly research benefits from low preconditions and well-scoped scenarios compared to the other domains, which reduces testing complexity and safety constraints.

The summary of applications in Table 4.5 differs from the previous ones due to the lack of compared algorithms. Only in 4 cases a benchmark was compared, which was outperformed by the deep RL algorithm in each case. Instead, the general task itself, as well as the specific use-case were referred for further classification.

4.4.5 Robotics

To obtain a significantly smoothed motion planning, Scheiderer et al. (2019) compensated disadvantages of existing RL planning approaches due to time discretization. If the robot exceeded a certain trajectory mark, the observation of the next step was triggered, and a Bézier curve was generated that aligned smoothly with the previous one. Similarly, Li et al. (2020) investigated the smoothing of CNC trajectories to enable high-speed machining. Based on a high-speed x-y motion platform, a real-time smoothing could be realized, which processed a pre-computed tool trajectory and smoothed out the path, calculated tool velocities, and emitted servo commands. An early image-based control of servos by deep RL was proposed in Miljković et al. (2013). The robot processed the captured images as states which were processed by a SARSA or DQN and ejected as spatial camera velocities. Thus, high robustness and accuracy of the control process were reached despite calibration errors and sensor noises. Following the same structure as the assembly domain, table 4.5 summarizes the main review results and includes the general application and the specific use case due to the lack of benchmarks.

4.4.6 Maintenance

The interaction of several linked machines in a serial production line was considered by Huang et al. (2020). Based on a large state space that contained buffer levels, operating inputs, and fault indicators for each machine, the algorithm made decisions about which individual machines

Assembly				
Application	Use-case	Algorithm	Source	
70	Sequence planning	Building block model	DQN	Watanabe and Inada (2020)
71	Sequence planning	Lift desk assembly	As AlphaGoZero	Yu et al. (2020)
72	Sequence planning	Seven parts assembly process	DQN	Zhao et al. (2019)
73	High precision insertion	Circuit breaker housing	DQN	Li et al. (2019)
74	High precision insertion	Gear set assembly	Model-based	Luo et al. (2019)
75	High precision insertion	Peg-in-hole	DQN	Inoue et al. (2017)
76	High precision insertion	Peg-in-hole	DQN, DDPG	Li et al. (2019)
77	High precision insertion	Peg-in-hole	SAC	Zhao et al. (2020)
78	Insertion task	Peg-in-hole (contact-rich, deform.)	AC	Luo et al. (2018)
79	Insertion task	Double peg-in-hole	DDPG	Xu et al. (2019)
80	Insertion task	Double peg-in-hole (contact-rich)	DDPG	Hoppe et al. (2019)
81	Insertion task	Peg-in-hole	DDPG	Wang et al. (2020)
82	Insertion task	Peg-in-hole	DDPG	Lämmle et al. (2020)
83	Insertion task	Peg-in-hole (cuboid)	SAC	Beltran-Hernandez et al. (2020)
84	Insertion task	Peg-in-hole (square)	DDPG	Kim et al. (2020)
85	Insertion task	Ring-insertion	SAC	Beltran-Hernandez et al. (2020)
86	Plug insertion tasks	Model-E connector	SAC, TD3	Schoettler et al. (2020)
87	Shoe tongue assembly	Soft fabric shoe tongues	DQN	Tsai et al. (2020)
Robotics				
Application	Use-case	Algorithm	Source	
88	Intelligent gripping	Find optimal grasp position	PPO	Park et al. (2020)
89	Motion planning	Real-time CNC traj. smoothing	DQN, DDPG	Li et al. (2020)
90	Motion planning	Bézier curve trajectory smoothing	DDPG	Scheiderer et. al (2019)
91	Visual control	Low-cost servo control	Sarsa, DQN	Miljković et al. (2013)

Table 4.5 Summary of deep RL applications in assembly and robotics

needed to be turned off at a time for service. Conventional methods often rely on the static recommendations of machine manufacturers and do not take system dependencies into account. In comparison, deep RL reduced the average maintenance costs by approximately 20% compared to a run-to-failure strategy, 7% compared to an age-dependant, and 5% compared to an opportunistic maintenance strategy. The same interdependencies between multiple components with competing failure probabilities were considered by Zhu et al. (2020) to avoid static and ineffective maintenance limits of conventional methods in large-scale systems. In several scenarios, the deep RL algorithm was able to reduce maintenance cost in multi-component systems without requiring experience-based or predefined thresholds.

The issue of limited resources to perform maintenance due to insufficient monetary, technical, and human capital was considered by Liu et al. (2020). Conventional methods only take the success of a single maintenance mission as a success factor, but neglect possible follow-up missions. Compared to benchmarks, deep RL thus demonstrated a 30% higher number of successful maintenance missions. Regarding, rotary machines fault diagnosis, Dai et al. (2020) and Ding et al. (2019) employed deep RL to detect faults from machine data at an early stage in real environments. Whereas Dai et al. (2020) focused on the detection of faulty components such as a cracked gear, Ding et al. (2019) focused on non-linear correlations between possible fault conditions by measuring raw sensor signals. Both times, errors could be detected at an early stage without the need for manual tuning efforts, expert experience, or pre-filtering of the data as

required by conventional methods.

Despite conducting only three benchmarks, the deep RL algorithms demonstrated superior maintenance-specific performance in all of these. Additional maintenance related publications of deep RL in recent years are listed in Table 4.6.

4.4.7 Energy management

In modern production, not only the maximum process performance but also the energy consumption and environmental impact become more crucial. To meet the challenge of greener production, Leng et al. (2021) addressed the order acceptance in the energy- and resource-intensive PCB fabrication under the assumption of resource constraints and environmental metrics. Compared to conventional methods (FIFO, random forest), the deep RL algorithm was able to increase profits and minimize carbon consumption, while optimizing lead time and cost. Considering a steel powder manufacturing process, Huang et al. (2019) proposed a model-free control design to optimize the energy consumption plan based on current energy costs and individual process components (i.e. atomizer, crusher). Compared to conventional methods, which often require a complex system model and neglect price fluctuations, the controller adjusted the production schedule to the electricity prices, which reduced energy costs by 24%. The same objective was addressed by Lu et al. (2020) for a lithium-ion battery assembly process which reduced electricity costs by 10%.

An approach to enable more energy-efficient and high reliable transmissions in low latent networks was proposed by Yang et al. (2020). Based on the channel status and other indicators, the algorithm selected radio frequency or visible light communication. It assigned an appropriate channel and performed the transmission power management. Thereby, energy efficiency, number of successful services, and latency were improved and a higher fulfillment of compulsory quality-of-service requirements was accomplished. Other applications included the single-machine energy optimization (Bakakeu et al., 2018), blast furnace gas tank energy scheduling for steel industry (Zhang and Si, 2020), and others as listed in Table 4.6.

A total of 4 benchmarks were carried out in the field of energy management, in which the deep RL algorithms again outperformed conventional ones. However, no real-world testing was conducted in the domain of energy management. Similar to previous categories, this would have entailed extraordinarily high expenses and would have caused a significantly increased implementation efforts at an early stage.

4.4.8 (Process) Design

Beginning with integrated circuit design, Liao et al. (2020) addressed the global routing process, which became a major challenge due to increased transistors densities and multiple design constraints. To cope with the complexity, Liao et al. (2020) modeled the circuit as a grid graph from which information was fed into the DQN router and outperformed the conventional A* approach.

Oh et al. (2020) proposed a deep RL algorithm for the design and fine-tuning of notch filters, which are commonly used in servo systems to suppress resonances. In complex cases, however, the filters not only need to be deployed in large numbers but also fine-tuned manually based on expert knowledge. The proposed notch tuning automatism avoided these and optimized several notch filters simultaneously and successfully stabilized a belt-drive servo system. Zhou et al. (2020) addressed the machining optimization of centrifugal impellers for a five-axis flank milling processing. By considering aerodynamic and machining parameters, an optimized path planning for the machine tool was developed, which reduces development time and cost.

Among the publications listed in Table 4.6, other design approaches included 2D-strip packing to improve space utilization in Zhu et al. (2020) which reduced average gaps by 20% compared to several benchmark algorithms, or the design of a SaaS architecture in Scheiderer et al. (2020) which significantly reduced optimization times in heavy-plate rolling compared to manual tuning.

4.4.9 Quality control

The field of quality control is affected by the increased product diversification and must adapt accordingly to carry out necessary component inspections. To support the workforce in quality related tasks, cobots can contribute to more stable processes. Brito et al. (2020) addressed the collaborative cooperation to combine the accuracy of the robot with the flexibility of the workforce. In case of an unforeseen inspection incident, the workforce taught the robot its new path, which was learned and reproduced by the DDPG. Unlike other methods that require an interruption of the production process, the DDPG enabled an online adaptation and significantly increased productivity and reduced stoppages.

Another approach for real-time quality monitoring of additive manufacturing processes was proposed by Wasmer et al. (2019). Conventional methods often rely on temperature data or high-resolution images, which have difficulties in reflecting the processes below the surface. To provide further process information, the implemented algorithm took acoustic emissions as an input for the process analysis and could thereby derive a pore concentration based quality categorization with an accuracy of up to 82% in real testings.

4 Publication 1 - Deep reinforcement learning based production

Maintenance				
	Application	Use-case	Algorithm	Source
92	Condition-based maintenance	Minimize cost rates in multiple stages	Dou.DQN	Zhang and Si (2020)
93	Machine fault diagnosis	Gear root crack analysis	DQN	Dai et al. (2020)
94	Machine fault diagnosis	Rolling bearing fault	DQN	Ding et al. (2019)
95	Oportunistic maintenance	Minimize prod./ maint. interference	PPO	Kuhnle et al. (2019)
96	Real-time prev. maintenance	Minimize long-run costs in serial prod.	Dou.DQN	Huang et. al (2020)
97	Selective maintenance	Maximize maint. mission success	AC	Liu et. al (2020)
98	Self-diagnosis and self-repair	Optimize self-repair in prod. lines	DQN	Epureanu et al. (2020)
99	Sensor-driven maintenance	Calc. remaining useful (turbofan)	DQN	Skordilis et al. (2020)
Energy management				
	Application	Use-case	Algorithm	Source
100	Energy system balancing	Tank level scheduling	DQN	Zhang et al. (2020)
101	Multi-agent energy optimization	CPS energy coordination	AC	Bakakeu et al. (2020)
102	Network resource management	Energy efficient RF/VLC network	DQN	Yang et al. (2020)
103	PCB order acceptance	Real-time order acceptance decisions	DQN	Leng et al. (2021)
104	Production-energy schedule opt.	Single machine optimization	DQN	Bakakeu et al. (2018)
105	Production-energy schedule opt.	Lithium-ion battery assembly	DDPG	Lu et al. (2020)
106	Production-energy schedule opt.	Steel powder manufacturing process	AC	Huang et al. (2019)
107	Sustainable joint energy control	Two machine, one buffer system	DQN	Hu et al. (2019)
(Process) Design				
	Application	Use-case	Algorithm	Source
108	Clamping position optimization	Milling machine	SAC	Samsonov et al. (2020)
109	Computer-aided process planning	Constrained machining	AC	Wu et al. (2021)
110	Integrated circuit design	Global IC routing	DQN	Liao et al. (2020)
111	Notch filter design	Industrial servo systems	DDPG	Oh et al. (2020)
112	Rectangular item placement	2D Strip Packing	DQN	Zhu et al. (2020)
113	SaaS remote training	Heavy plate rolling	SAC	Scheiderer et al. (2020)
114	Tool path design	Geometric impeller optimization	DDPG	Zhou et al. (2020)

Table 4.6 Summary of deep RL applications in maintenance, energy management, and (process) design

4.4.10 Further applications

Further categories with single publications are listed in Table 4.7. These include specific topics such as building an agent swapping framework to allow learning in a non-real-time environment and execution in a real-time environment (Schmidt et al., 2020) or the deep RL based selection of optimal prediction models in the semiconductor manufacturing domain to cope with demand fluctuations and avoid shortages and overstock (Chien et al., 2020).

Quality control				
	Application	Use-case	Algorithm	Source
115	In-situ quality monitoring	Subsurface dynamics analysis	sim. AlphaGO	Wasmer et al. (2019)
116	Quality inspection	Path teaching and adaption	AC	Brito et al. (2020)
Further applications				
	Application	Use-case	Algorithm	Source
117	Novel PLC learning/acting arch.	Real-time framework	-	Schmidt et al. (2020)
118	Select opt. demand forecast model	Semiconductor components	DQN	Chien et al. (2020)
119	Multi-task policy generalization	Mfg. system with various tasks	DQN	Wang et al. (2019)
120	Investigation of malicious behaviors	Function-/performance attacks	DQN	Liu et al. (2021)

Table 4.7 Summary of deep RL applications in quality control and further applications

4.5 Implementation challenges and research agenda

In the previous section, the broad application base and benefits associated with the deployment of the deep RL algorithms were highlighted. Nevertheless, there are some challenges and hurdles that must be overcome that prevent an extensive deployment (RQ2) and need to be addressed in future research (RQ3).

4.5.1 Implementation challenges and research gaps

The key insights of the review analysis are summarized in Table 4.8. The table is aligned in its sequence with the previous chapter and comprises the most frequently applied algorithms and neural networks as well as the simulation-only and superiority share (related to the conducted benchmarks).

Production domain	#Publications	Most frequent algorithm	Most frequent neural netw.	Simulation-only share	Superiority (#benchmarks)
Process control	29	AC	FFNN	86%	81% (16)
Scheduling	22	DQN	FFNN	100%	89% (19)
Dispatching	11	DQN	FFNN	100%	55% (11)
(Intra-) Logistics	7	DQN	FFNN	86%	100% (5)
Assembly	18	DQN/DDPG	FFNN	17%	100% (4)
Robotics	4	DQN/DDPG	FFNN	75%	- (0)
Maintenance	8	DQN	FFNN	75%	100% (3)
Energy Management	8	DQN	FFNN	100%	100% (4)
(Process) Design	7	(S)AC	FFNN	71%	100% (4)
Quality Control	2	AC/AlphaGo	FFNN	0%	- (0)
Others	4	DQN	FFNN	75%	100% (2)

Table 4.8 Summary of the key findings from the review analysis

Table 4.8 highlights some of the challenges we identified during the literature review. We categorized those into the following 4 major and subsequent minor application challenges and research gaps.

- **Algorithm selection:** After identification of a potential implementation, the question arises which algorithm and parameters should be used for the planned scenario. Although these have a significant impact on the resulting performance, there are no or only a few guidelines that can assist during the selection and parameter optimization process. As mentioned by Rummukainen and Nurminen (2019), Yoo et al. (2021), and others, this selection is a central issue that can worsen the resulting performance and hinder the full development of deep RL capabilities. Table 4.8 and Figure 4.7(a) demonstrate the reliance on standard algorithms that may result from missing guidelines. A majority of the reviewed papers implemented a DQN, although possible improvements like the doubleDQN can significantly improve performances (van Hasselt et al., 2016). As one of the few examples,

4 Publication 1 - Deep reinforcement learning based production

Li et al. (2019) thus improved success rates by 13% to 94% utilizing a DDPG for a robot assembly process compared to a DQN. Similarly, a significantly improved performance was reached through learning rate and batch size modifications in Baer et al. (2020).

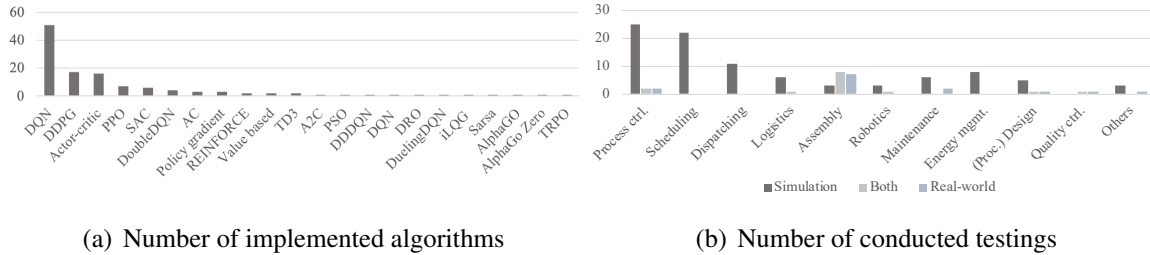


Figure 4.7 Quantitative analysis of applied algorithms and testing environments

- Further modifications:** A majority of the papers utilized a common (convolutional) neural network (85%). Only a small fraction utilized LSTM (6%) or recurrent neural networks (1%), which can better reflect long-term experiences. Besides, only four papers compared more than one network and six papers more than one RL algorithm, leading to a gap in benchmarks and performance correlations. Other extensions such as a prioritized experience replay were only applied occasionally although they might increase production performance significantly, similar to the testings within the Atari environment (Schaul et al., 2016).
- Transfer of results:** Another major challenge is the transfer of the simulation results to real-world scenarios. Overall, 76% of the papers have validated the proposed solution within simulations. Only 24% of the papers conducted real-world testings, half of which considered either a purely real-world scenario or both. The percentage in assembly was particularly high, as 83% of tests were conducted in real or simulation-based environments. Assembly benefits in particular from confined and segregated environments, which limit hurdles and mitigate risks. In contrast, no real testing was carried out in production scheduling and dispatching as indicated in Figure 4.7(b). Particularly high safety and reliability-related entry hurdles must be met, besides high system implementation efforts, that prevent large-scale and rapid testings in those fields. Besides, simulations only obtain a simplified representation of the real problem. Due to the considerable differences in complexity between the simulation and real applications, a reduced performance of the approaches after the real-world transfer is to be expected. In particular, the implementation into large real-world systems is rather challenging and has to cope with many unconsidered parameters and a non-preprocessed set of data.

- **Local optimization:** In addition to the aforementioned challenges, there is the risk that the algorithms only perform local optimization (Dittrich and Fohlmeister, 2020). As discussed by Rossit et al. (2019), smart scheduling, in particular, is composed of decentralized structures in which multiple deep RL agents can interact and perform their tasks in the defined task domain. This might result in potential small-scale control loops that are optimized intrinsically and exploit local information, but neglect larger interdependencies. Besides, a non-optimal problem-solving strategy may arise from a lack of exploration of the state and action space, resulting in the selection of non-optimal actions and a non-existent optimality guarantee (Spielberg et al., 2019; Guo et al., 2019).

In addition to the challenges mentioned above, others arise from exponentially growing action and state spaces in complex production systems that require high computational efforts or multitasking scenarios that can not be managed by a single agents (Wang et al., 2019; Beltran-Hernandez et al., 2020). Besides, non-smooth execution due to jumps in the policy decision can result in the inability to execute optimal actions and negatively impact process qualities (Noel and Pandian, 2014). Last, differences in training performance (Ma et al., 2019) or vibration during the training of complex tasks (Shi et al., 2020), can lead to less repeatable processes and lower predictability, resulting in low reliability and raising safety concerns.

4.5.2 Future research agenda

Although the hurdles and challenges described above do not yet enable a full-scale adoption of deep RL in production systems, further efforts can assist in accelerating the process towards industrial maturity. The bullet points below address the outlined challenges and provide research colleagues and practitioners incentives for future research.

- **Standardized implementation approach:** Future deep RL based production research can incorporate more insightful benchmarks by considering advanced algorithms, modifications, and parameter tuning within the same simulations. Similar to van Hasselt et al. (2016) in the Atari environment, this could yield a significant increase in performance without causing high adaption efforts. To assist future research, the benchmarks could additionally serve as a basis to derive further guidance for optimization and control problems with similar state and action spaces circumventing expert advice needed for a fast system adoption and applicability.

The generation of prototype evaluations can also benefit from the definition of model environments, similar to the Atari environment. Frameworks such as the *SimRLFab* for production dispatching (Kuhnle, 2020) can be integrated quickly and enable algorithm benchmarks without requiring large implementation efforts.

- **Accelerated simulation to real-world transfer:** To enable a faster integration to real production environments, the respective system requirements must be satisfied. This primarily involves the consideration of safety-relevant parameters to avoid critical actions and threats. In this context, a constraint-driven approach in non-deep RL was proposed by Ge et al. (2019), in which permitted actions were limited through preliminary filtering, or by Xiong and Diao (2021) who proposed a safety-based evaluation of policy robustness. Further studies should approximate the simulations and frameworks to real-world conditions even more, which includes consideration of hard real-time requirements, significant parameters, uncertainties, and indeterminacies. Thus, by establishing a digital twin that copies reality, hardware-in-the-loop environments (HiL), and separate training and testing sequences, the gap between research and practical testings is narrowed and the transfer of results and validation can be accelerated and performed with less risk. The HiL approach would enable a real-time use of machine data and also address the data quality issue. In this context, data pre-processing is essential and may be integrated in the simulation, but can only be matched to reality with great efforts. The same applies to the state-action-reward design, which must process the changed or even additional input variables and cope with unknown process variables. The algorithms could be thoroughly investigated under real conditions in the hybrid HiL environment, parameters optimized and the real system dynamics between input and output variables analyzed. Especially, domains that require large-scale implementations like production scheduling, might benefit from such a step-by-step HiL approach that anticipates transfer issues and identifies unknown disturbances at an early stage.
- **Generalizability:** The ability of the agents to adapt more effectively to changing production conditions should be considered to further optimize their learning stability and robustness. Even though this has already been considered by learning general behaviors instead of specific policies in Baer et al. (2020), it was also observed that small deviations of the starting conditions led to performance reductions (Beltran-Hernandez et al., 2020). Future research should therefore focus on methods that enable agents to adapt to different scenarios as quickly as possible. This not only includes a particularly fast re-training under changed conditions, but also an accelerated transfer of the adapted policy to the real agent. Such a swift transfer could be facilitated by applying a permanently trained agent within the digital twin and a subsequent policy transfer. Another approach to increase generalizability and performance under changing conditions could be addressed by implementations that go beyond the use of isolated deep RL solutions. Combining deep RL with classical approaches such as scenario analysis, combined rule decisions, or task decomposition could help circumvent common drawbacks such as low sample-efficiencies and reduce error-proneness.

- **Handling production complexity:** If the network receives too many state inputs and has to decide on a large number of possible actions, this increases problem complexity and significantly complicates optimal decision making. Thus, to keep large-scale production problems manageable, they must be reduced in their dimension and problem complexity to circumvent the curse of dimensionality. For this purpose, the complexity of whole production systems could be decomposed by decentralized structures and allocated to multiple agents. Having been trained to optimize specific parameters, these individual agents can be deployed situation-dependent. Through the associated orchestration and complexity break-down, a significantly improved scalability might be reached, since no individual agent has to cope with the entire complexity and the exponentially growing state and action space in large-scale applications. Local and global optimization loops could run in parallel and minimize the risk of a local optimization.

Although Wang et al. (2019) already demonstrated such an ability of deep RL to optimize multiple objectives utilizing generalized policies, further research should elaborate on multitasking and leverage the generalizability of deep RL algorithms.

Besides, research should focus on transfer learning to enable agents to learn and perform complex tasks faster and better. Thus, in multi-agent systems, single agents could benefit from the experience gained by others and cope better with unfamiliar situations. The development of such swarm intelligence could better exploit local and global information and enable a flexible response and adaptation of the production system to unforeseen incidents.

- **Coordinated optimization:** In distributed production systems, local optimization of individual agents must be opposed by adjusting input variables, reward functions, and training strategy. Agents must receive essential global and local information and should be evaluated on individual as well as multi-agent performance criteria. This can include maximizing the utilization of machines in the local agent environment while minimizing the total cycle time of the overall multi-agent process. Further research could scale this sensitivity towards multiple objectives which might be accomplished by staged training sequences in which individual agents first find optimal local solutions and subsequently target global objectives in a multi-agent training phase (Baer et al., 2019).

Besides, the exploration strategy of a single agent must be determined by appropriate parameters to avoid an intrinsic local optimization. This can be remedied by specific tuning and should be considered more in-depth in deep RL controlled multi-agent production systems.

4.6 Discussion

Today's production systems must cope with increasingly sophisticated customer requirements, shorter product and development cycles, and short-term fluctuations in demand. One approach to address these challenges in production is deep RL, which differs from other machine learning methods primarily through its online adaptability and real-time processing of sensor data. Although other technical domains have already emphasized the benefits of deep RL, a focused review in production systems has yet to be conducted. Our purpose was to provide a systematic literature review of current deep RL applications in production systems and to outline challenges and fields of future research to address these. Based on a taxonomy framework, 120 retrieved papers from three databases were reviewed and classified according to their manufacturing discipline, industry background, specific application, optimization objective, applied deep RL algorithm, and neural network, heuristic benchmark results, and its application in a simulated and/or real environment.

An application of deep RL was found in a wide range of production engineering disciplines. Although a large portion of the applications were implemented in simulations (76%), the superiority of deep RL driven production optimization was evident. In more than 85% of the total comparisons, deep RL algorithms outperformed the corresponding benchmarks and increased problem-specific performances.

4.6.1 Managerial implications

Future factories will be increasingly interconnected, products and processes will become more complex, and development cycles will be more accelerated. To cope with these, companies should challenge current practices and consider alternatives to minimize process risks and to fully exploit algorithmic performances and organizational capabilities. To give a first introduction, this literature review presents a variety of possible applications of deep RL in production systems and helps managers to identify potential internal use cases. As a reference, the surveyed papers can provide valuable guidance for own deep RL implementation approaches and assist in the further selection of algorithms and parameters.

In contrast to static methods that can react to changing conditions only to a limited extent, deep RL algorithms were able to increase productions robustness and adaptability. In most applications, it proofed its practical relevance and not only improved technical parameters, but in some applications increased cash flow and reduced (online) conversion costs. Through deep RL, companies can limit the dependency on increasingly scarce human capital and leverage data-driven operations proactively to reduce cost-intensive manual and expert-based processes.

4.6.2 Limitations

Although the work is based on a taxonomy and methodology framework, we would like to emphasize the existing limitations of our review. We conducted the literature search based on three selected databases and an iterative keyword search, in which we tried to determine essential domains, but may miss some that would have yielded relevant supplementary results. To compensate for this bias, we conducted a forward and backward search to aggregate correlated publications. To satisfy our claim of providing a representative review and to provide a broad foundation, we also included proceedings and conference papers, which may cause bias compared to other reviews. However, by ensuring peer review we sought to reduce this bias and to meet all quality requirements. Besides, a limitation arises from the definition of a restricted review scope. Publications from enterprise research or other domains that may have interfaces to production were not specifically considered. Specific reviews can provide insights for the application of deep RL in these production related environments, which we recommend and encourage.

4.7 Conclusion

It became evident that deep RL is widely used from process control to maintenance and other domains, outperforming conventional algorithms in most cases, demonstrating its ability to adapt to a variety of scenarios and deal with existing production uncertainties (RQ1). This not only reduced lead times and WIP levels, reached high accuracies in assembly, or developed robust scheduling policies, but also mitigated current drawbacks of conventional methods such as limited adaptation capabilities, cost intensive re-optimizations, or high dependencies on human-based decisions.

Nevertheless, some challenges still prevent widespread adoption in production systems (RQ2). Besides missing hands-on guidelines and limited use of the available algorithm base, only a few deep RL applications have been evaluated in reality and optimized in-depth, making further validation mandatory. In future research (RQ3), the simulations need to be further refined to incorporate additional uncertainties, reduce current transfer barriers, and enable real-world applications. Additional optimization alternatives such as more powerful deep RL algorithms that are currently less utilized, extensive elaboration on increased generalizability, alternative training strategies, and reduction of production task complexities can be further considered to realize more optimal performances.

The challenge remains of defining a thorough approach that will assist scholars and practitioners through the application and optimization process, providing guidelines for deployment, and accelerating the implementation in potential use-cases. Further research efforts on collaborative and hierarchical multi-agent architectures, as well as the use of fleet intelligence, can further

strengthen the application of deep RL in production systems and make it a widely applicable and robust edge and global optimization method.

Copyright notice

This is an accepted version of the article published in:

Panzer, M., and B. Bender (2022). Deep reinforcement learning in production systems: A Systematic Literature Review. *International Journal of Production Research* 60 (13), p. 4316–4341. <https://doi.org/10.1080/00207543.2021.1973138>

Clarification of the copyright adjusted according to the guidelines of the publisher.

Contributor roles

This paper is the result of collaborative efforts where specific responsibilities were allocated to ensure the effective completion of the research and the preparation of the manuscript:

- **Marcel Panzer:** Played a pivotal role in the majority of this publication's aspects. Responsibilities included conceptualizing the research, designing and implementing the literature analysis methodology, conducting the review, synthesizing and analyzing findings, and primarily drafting the manuscript. Additionally, contributions were made in compiling and refining the final manuscript during the review process.
- **Benedict Bender:** Played an essential role in the advancement of this review, offering important guidance. His input involved rigorous critiques, along with providing thoughtful feedback and recommendations. These contributions were key in refining the publication and upholding its integrity.

The *Declaration of the Co-Authors* is inserted at the end of this thesis.

Publication 1 - References

Altenmüller, T., T. Stüker, B. Waschneck, A. Kuhnle and G. Lanza (2020). Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints. *Production Engineering* 14(3), p. 319–328. doi: 10.1007/s11740-020-00967-8.

Andersen, R. E., S. Madsen, A. B. K. Barlo, S. B. Johansen, M. Nør, R. S. Andersen and S. Bøgh (2019). Self-learning Processes in Smart Factories: Deep Reinforcement Learning

- for Process Control of Robot Brine Injection. *Procedia Manufacturing* 38, p. 171 – 177. doi: 10.1016/j.promfg.2020.01.023.
- Arinez, J. F., Q. Chang, R. X. Gao, C. Xu and J. Zhang (2020). Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook. *Journal of Manufacturing Science and Engineering* 142(11), p. 110804. doi: 10.1115/1.4047855.
- Baer, S., J. Bakakeu, R. Meyes and T. Meisen (2019). Multi-Agent Reinforcement Learning for Job Shop Scheduling in Flexible Manufacturing Systems. In: *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*, Laguna Hills, CA, USA. doi: 10.1109/AI4I46381.2019.00014.
- Baer, S., D. Turner, P. Mohanty, V. Samsonov, R. Bakakeu and T. Meisen (2020). Multi Agent Deep Q-Network Approach for Online Job Shop Scheduling in Flexible Manufacturing. In: *2020 International Conference on Manufacturing System and Multiple Machines*, Tokyo, Japan.
- Bakakeu, J., S. Baer, J. Bauer, H.-H. Klos, J. Peschke, A. Fehrle, W. Eberlein, J. Bürner, M. Brossog, L. Jahn and J. Franke (2018). An Artificial Intelligence Approach for Online Optimization of Flexible Manufacturing Systems. *Applied Mechanics and Materials* 882, p. 96–108. doi: 10.4028/www.scientific.net/AMM.882.96.
- Bakakeu, J., D. Kisskalt, J. Franke, S. Baer, H.-H. Klos and J. Peschke (2020). Multi-Agent Reinforcement Learning for the Energy Optimization of Cyber-Physical Production Systems. In: *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, London, ON, Canada. doi: 10.1109/CCECE47787.2020.9255795.
- Bellman, R. (1957). *Dynamic programming*, Volume 1. Princeton, NJ, USA: Princeton University Press. OCLC: 830865530.
- Beltran-Hernandez, C. C., D. Petit, I. G. Ramirez-Alpizar and K. Harada (2020). Variable Compliance Control for Robotic Peg-in-Hole Assembly: A Deep-Reinforcement-Learning Approach. *Applied Sciences* 10(19), p. 6923. doi: 10.3390/app10196923.
- Beltran-Hernandez, C. C., D. Petit, I. G. Ramirez-Alpizar, T. Nishi, S. Kikuchi, T. Matsubara and K. Harada (2020). Learning Force Control for Contact-Rich Manipulation Tasks With Rigid Position-Controlled Robots. *IEEE Robotics and Automation Letters* 5(4), p. 5709–5716. doi: 10.1109/LRA.2020.3010739.
- Brito, T., J. Queiroz, L. Piardi, L. A. Fernandes, J. Lima and P. Leitão (2020). A Machine Learning Approach for Collaborative Robot Smart Manufacturing Inspection for Quality Control Systems. *Procedia Manufacturing* 51, p. 11 – 18. doi: 10.1016/j.promfg.2020.10.003.
- Brocke, J., A. Simons, B. Niehaves, K. Riemer, R. Plattfaut and A. Cleven (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process.

Proceedings of the 17th European Conference on Information Systems (ECIS).

- Cao, D., W. Hu, J. Zhao, G. Zhang, B. Zhang, Z. Liu, Z. Chen and F. Blaabjerg (2020). Reinforcement Learning and Its Applications in Modern Power and Energy Systems: A Review. *Journal of Modern Power Systems and Clean Energy* 8(6), p. 1029–1042. doi: 10.35833/MPCE.2020.000552.
- Chen, B., J. Wan, Y. Lan, M. Imran, D. Li and N. Guizani (2019). Improving Cognitive Ability of Edge Intelligent IIoT through Machine Learning. *IEEE Network* 33(5), p. 61–67. doi: 10.1109/MNET.001.1800505.
- Chen, N., S. Luo, J. Dai, B. Luo and W. Gui (2020). Optimal Control of Iron-Removal Systems Based on Off-Policy Reinforcement Learning. *IEEE Access* 8, p. 149730–149740. doi: 10.1109/ACCESS.2020.3015801.
- Chien, C.-F., Y.-S. Lin and S.-K. Lin (2020). Deep reinforcement learning for selecting demand forecast models to empower Industry 3.5 and an empirical study for a semiconductor component distributor. *International Journal of Production Research* 58(9), p. 2784–2804. doi: 10.1080/00207543.2020.1733125.
- Cooper, H. M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society* 1, p. 104–126.
- Dai, W., Z. Mo, C. Luo, J. Jiang, H. Zhang and Q. Miao (2020). Fault Diagnosis of Rotating Machinery Based on Deep Reinforcement Learning and Reciprocal of Smoothness Index. *IEEE Sensors Journal* 20(15), p. 8307–8315. doi: 10.1109/JSEN.2020.2970747.
- Ding, Y., L. Ma, J. Ma, M. Suo, L. Tao, Y. Cheng and C. Lu (2019). Intelligent fault diagnosis for rotating machinery using deep Q-network based health state classification: A deep reinforcement learning approach. *Advanced Engineering Informatics* 42, p. 100977. doi: 10.1016/j.aei.2019.100977.
- Dittrich, M.-A. and S. Fohlmeister (2020). Cooperative multi-agent system for production control using reinforcement learning. *CIRP Annals* 69(1), p. 389 – 392. doi: 10.1016/j.cirp.2020.04.005.
- Dong, T., F. Xue, C. Xiao and J. Li (2020). Task scheduling based on deep reinforcement learning in a cloud manufacturing environment. *Concurrency and Computation: Practice and Experience* 32(11). doi: 10.1002/cpe.5654.
- Dornheim, J., N. Link and P. Gumbsch (2020). Model-free Adaptive Optimal Control of Episodic Fixed-horizon Manufacturing Processes Using Reinforcement Learning. *International Journal of Control, Automation and Systems* 18(6), p. 1593–1604. doi: 10.1007/s12555-019-0120-7.
- Durach, C. F., J. Kembro and A. Wieland (2017). A New Paradigm for Systematic Literature

- Reviews in Supply Chain Management. *Journal of Supply Chain Management* 53(4), p. 67–85. doi: 10.1111/jscm.12145.
- Epureanu, B. I., X. Li, A. Nassehi and Y. Koren (2020). Self-repair of smart manufacturing systems by deep reinforcement learning. *CIRP Annals* 69(1), p. 421 – 424. doi: 10.1016/j.cirp.2020.04.008.
- Feldkamp, N., S. Bergmann and S. Strassburger (2020). Simulation-based deep reinforcement learning for modular production systems. In: *Proceedings of the 2020 Winter Simulation Conference*, p. 1596–1607.
- Gabel, T. and M. Riedmiller (2007). Adaptive Reactive Job-Shop Scheduling with Reinforcement Learning Agents. *International Journal of Information Technology and Intelligent Computing*.
- Ge, Y., F. Zhu, X. Ling and Q. Liu (2019). Safe Q-Learning Method Based on Constrained Markov Decision Processes. *IEEE Access* 7, p. 165007–165017. doi: 10.1109/ACCESS.2019.2952651.
- Greenhalgh, T. and R. Peacock (2005). Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources. *BMJ* 331(7524), p. 1064–1065. doi: 10.1136/bmj.38636.593461.68.
- Guo, F., X. Zhou, J. Liu, Y. Zhang, D. Li and H. Zhou (2019). A reinforcement learning decision model for online process parameters optimization from offline data in injection molding. *Applied Soft Computing* 85, p. 105828. doi: 10.1016/j.asoc.2019.105828.
- Guo, L., H. Wang and J. Zhang (2019). Data-Driven Grinding Control Using Reinforcement Learning. In: *2019 IEEE 21st International Conference on High Performance Computing and Communications*, Zhangjiajie, China. doi: 10.1109/HPCC/SmartCity/DSS.2019.00395.
- Günther, J., P. M. Pilarski, G. Helfrich, H. Shen and K. Diepold (2016). Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement learning. *Mechatronics* 34, p. 1 – 11. doi: 10.1016/j.mechatronics.2015.09.004.
- Han, B.-A. and J.-J. Yang (2020). Research on Adaptive Job Shop Scheduling Problems Based on Dueling Double DQN. *IEEE Access* 8, p. 186474–186495. doi: 10.1109/ACCESS.2020.3029868.
- He, Z., K.-P. Tran, S. Thomassey, X. Zeng, J. Xu and C. Yi (2020). A deep reinforcement learning based multi-criteria decision support system for optimizing textile chemical process. *Computers in Industry* 125, p. 103373. doi: 10.1016/j.compind.2020.103373.
- Heger, J. and T. Voß (2020). Dynamically changing sequencing rules with reinforcement learning in a job shop system with stochastic influences. *Proceedings of the 2020 Winter*

- Simulation Conference*, p. 1608–1618.
- Hildebrand, M., R. S. Andersen and S. Bøgh (2020). Deep Reinforcement Learning for Robot Batching Optimization and Flow Control. *Procedia Manufacturing* 51, p. 1462 – 1468. doi: 10.1016/j.promfg.2020.10.203.
- Hoppe, S., Z. Lou, D. Hennes and M. Toussaint (2019). Planning Approximate Exploration Trajectories for Model-Free Reinforcement Learning in Contact-Rich Manipulation. *IEEE Robotics and Automation Letters* 4(4), p. 4042–4047. doi: 10.1109/LRA.2019.2928212.
- Hu, H., X. Jia, Q. He, S. Fu and K. Liu (2020). Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0. *Computers & Industrial Engineering* 149, p. 106749. doi: 10.1016/j.cie.2020.106749.
- Hu, L., Z. Liu, W. Hu, Y. Wang, J. Tan and F. Wu (2020). Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. *Journal of Manufacturing Systems* 55, p. 1 – 14. doi: 10.1016/j.jmsy.2020.02.004.
- Hu, W., Z. Sun, Y. Zhang and Y. Li (2019). Joint Manufacturing and Onsite Microgrid System Control Using Markov Decision Process and Neural Network Integrated Reinforcement Learning. *Procedia Manufacturing* 39, p. 1242 – 1249. doi: 10.1016/j.promfg.2020.01.345.
- Huang, J., Q. Chang and J. Arinez (2020). Deep reinforcement learning based preventive maintenance policy for serial production lines. *Expert Systems with Applications* 160, p. 113701. doi: 10.1016/j.eswa.2020.113701.
- Huang, X., S. H. Hong, M. Yu, Y. Ding and J. Jiang (2019). Demand Response Management for Industrial Facilities: A Deep Reinforcement Learning Approach. *IEEE Access* 7, p. 82194–82205. doi: 10.1109/ACCESS.2019.2924030.
- Hubbs, C. D., C. Li, N. V. Sahinidis, I. E. Grossmann and J. M. Wassick (2020). A deep reinforcement learning approach for chemical production scheduling. *Computers & Chemical Engineering* 141, p. 106982. doi: 10.1016/j.compchemeng.2020.106982.
- Inoue, T., G. De Magistris, A. Munawar, T. Yokoya and R. Tachibana (2017). Deep reinforcement learning for high precision assembly tasks. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC. doi: 10.1109/IROS.2017.8202244.
- Jiang, Y., J. Fan, T. Chai and F. L. Lewis (2019). Dual-Rate Operational Optimal Control for Flotation Industrial Process With Unknown Operational Model. *IEEE Transactions on Industrial Electronics* 66(6), p. 4587–4599. doi: 10.1109/TIE.2018.2856198.
- Jiang, Y., J. Fan, T. Chai, J. Li and F. L. Lewis (2018). Data-Driven Flotation Industrial Process Operational Optimal Control Based on Reinforcement Learning. *IEEE Transactions on Industrial Informatics* 14(5), p. 1974–1989. doi: 10.1109/TII.2017.2761852.

- Jin, Z., H. Li and H. Gao (2019). An intelligent weld control strategy based on reinforcement learning approach. *The International Journal of Advanced Manufacturing Technology* 100(9-12), p. 2163–2175. doi: 10.1007/s00170-018-2864-2.
- Kagermann, H., W. Wahlster and J. Helbig (2013). *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0 – Securing the Future of German Manufacturing Industry*. Acatech - National Academy of Science and Engineering.
- Kang, Z., C. Catal and B. Tekinerdogan (2020). Machine learning applications in production lines: A systematic literature review. *Computers & Industrial Engineering* 149, p. 106773. doi: 10.1016/j.cie.2020.106773.
- Khan, A.-M., R. J. Khan, A. Tooshil, N. Sikder, M. A. P. Mahmud, A. Z. Kouzani and A.-A. Nahid (2020). A Systematic Review on Reinforcement Learning-Based Robotics Within the Last Decade. *IEEE Access* 8, p. 176598–176623. doi: 10.1109/ACCESS.2020.3027152.
- Kim, J.-B., H.-B. Choi, G.-Y. Hwang, K. Kim, Y.-G. Hong and Y.-H. Han (2020). Sortation Control Using Multi-Agent Deep Reinforcement Learning in N-Grid Sortation System. *Sensors* 20(12), p. 3401. doi: 10.3390/s20123401.
- Kim, Y.-L., K.-H. Ahn and J.-B. Song (2020). Reinforcement learning based on movement primitives for contact tasks. *Robotics and Computer-Integrated Manufacturing* 62, p. 101863. doi: 10.1016/j.rcim.2019.101863.
- Kuhnle, A. (2020). SimRLFab: Simulation and reinforcement learning framework for production planning and control of complex job shop manufacturing systems. *GitHub*. Accessed 20 March 2021. <https://github.com/AndreasKuhnle/SimRLFab>.
- Kuhnle, A., J.-P. Kaiser, F. Theiß, N. Stricker and G. Lanza (2020). Designing an adaptive production control system using reinforcement learning. *Journal of Intelligent Manufacturing* 32, p. 855–876. doi: 10.1007/s10845-020-01612-y.
- Kumar, A., R. Dimitrakopoulos and M. Maulen (2020). Adaptive self-learning mechanisms for updating short-term production decisions in an industrial mining complex. *Journal of Intelligent Manufacturing* 31(7), p. 1795–1811. doi: 10.1007/s10845-020-01562-5.
- Lange, S., M. Riedmiller and A. Voigtlander (2012). Autonomous reinforcement learning on raw visual input data in a real world application. In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*, Brisbane, Australia. doi: 10.1109/IJCNN.2012.6252823.
- Lee, J.-H. and H.-J. Kim (2021). Reinforcement learning for robotic flow shop scheduling with processing time variations. *International Journal of Production Research*. doi: 10.1080/00207543.2021.1887533.
- Lee, J. H., J. Shin and M. J. Realff (2018). Machine learning: Overview of the recent pro-

- gresses and implications for the process systems engineering field. *Computers & Chemical Engineering 114*, p. 111 – 121. doi: 10.1016/j.compchemeng.2017.10.008.
- Lee, S., Y. Cho and Y. H. Lee (2020). Injection Mold Production Sustainable Scheduling Using Deep Reinforcement Learning. *Sustainability 12*(20), p. 8718. doi: 10.3390/su12208718.
- Lei, L., Y. Tan, K. Zheng, S. Liu, K. Zhang and X. Shen (2020). Deep Reinforcement Learning for Autonomous Internet of Things: Model, Applications and Challenges. *IEEE Communications Surveys & Tutorials 22*(3), p. 1722–1760. doi: 10.1109/COMST.2020.2988367.
- Leng, J., C. Jin, A. Vogl and H. Liu (2020). Deep reinforcement learning for a color-batching resequencing problem. *Journal of Manufacturing Systems 56*, p. 175 – 187. doi: 10.1016/j.jmsy.2020.06.001.
- Leng, J., G. Ruan, Y. Song, Q. Liu, Y. Fu, K. Ding and X. Chen (2021). A loosely-coupled deep reinforcement learning approach for order acceptance decision of mass-individualized printed circuit board manufacturing in industry 4.0. *Journal of Cleaner Production 280*, p. 124405. doi: 10.1016/j.jclepro.2020.124405.
- Li, B., H. Zhang, P. Ye and J. Wang (2020). Trajectory smoothing method using reinforcement learning for computer numerical control machine tools. *Robotics and Computer-Integrated Manufacturing 61*, p. 101847. doi: 10.1016/j.rcim.2019.101847.
- Li, F., Q. Jiang, W. Quan, S. Cai, R. Song and Y. Li (2019). Manipulation Skill Acquisition for Robotic Assembly Based on Multi-Modal Information Description. *IEEE Access 8*, p. 6282–6294. doi: 10.1109/ACCESS.2019.2934174.
- Li, F., Q. Jiang, S. Zhang, M. Wei and R. Song (2019). Robot skill acquisition in assembly process using deep reinforcement learning. *Neurocomputing 345*, p. 92 – 102. doi: 10.1016/j.neucom.2019.01.087.
- Li, J., J. Ding, T. Chai and F. L. Lewis (2020). Nonzero-Sum Game Reinforcement Learning for Performance Optimization in Large-Scale Industrial Processes. *IEEE Transactions on Cybernetics 50*(9), p. 4132–4145. doi: 10.1109/TCYB.2019.2950262.
- Liang, H., X. Wen, Y. Liu, H. Zhang, L. Zhang and L. Wang (2021). Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning. *Robotics and Computer-Integrated Manufacturing 67*, p. 101991. doi: 10.1016/j.rcim.2020.101991.
- Liao, H., W. Zhang, X. Dong, B. Póczos, K. Shimada and L. Burak Kara (2020). A Deep Reinforcement Learning Approach for Global Routing. *Journal of Mechanical Design 142*(6), p. 061701. doi: 10.1115/1.4045044.
- Liao, Y., F. Deschamps, E. d. F. R. Loures and L. F. P. Ramos (2017). Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal. *International*

- Journal of Production Research* 55(12), p. 3609–3629. doi: 10.1080/00207543.2017.1308576.
- Light, R. J. and D. B. Pillemer (1984). *Summing up: the science of reviewing research*. Cambridge, Mass: Harvard University Press.
- Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra (2016). Continuous control with deep reinforcement learning. *Proceedings of 4th International Conference on Learning Representations*. arXiv: 1509.02971.
- Lin, C.-C., D.-J. Deng, Y.-L. Chih and H.-T. Chiu (2019). Smart Manufacturing Scheduling With Edge Computing Using Multiclass Deep Q Network. *IEEE Transactions on Industrial Informatics* 15(7), p. 4276–4284. doi: 10.1109/TII.2019.2908210.
- Liu, C.-L., C.-C. Chang and C.-J. Tseng (2020). Actor-Critic Deep Reinforcement Learning for Solving Job Shop Scheduling Problems. *IEEE Access* 8, p. 71752–71762. doi: 10.1109/ACCESS.2020.2987820.
- Liu, X., H. Xu, W. Liao and W. Yu (2019). Reinforcement Learning for Cyber-Physical Systems. In: *2019 IEEE International Conference on Industrial Internet (ICII)*, Orlando, FL, USA. doi: 10.1109/ICII.2019.00063.
- Liu, X., W. Yu, F. Liang, D. Griffith and N. Golmie (2021). On deep reinforcement learning security for Industrial Internet of Things. *Computer Communications* 168, p. 20 – 32. doi: 10.1016/j.comcom.2020.12.013.
- Liu, Y., Y. Chen and T. Jiang (2020). Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach. *European Journal of Operational Research* 283(1), p. 166 – 181. doi: 10.1016/j.ejor.2019.10.049.
- Lohmer, J. and R. Lasch (2020). Production planning and scheduling in multi-factory production networks: a systematic literature review. *International Journal of Production Research*, p. 1–27. doi: 10.1080/00207543.2020.1797207.
- Lu, R., Y.-C. Li, Y. Li, J. Jiang and Y. Ding (2020). Multi-agent deep reinforcement learning based demand response for discrete manufacturing systems energy management. *Applied Energy* 276, p. 115473. doi: 10.1016/j.apenergy.2020.115473.
- Lu, X., B. Kiumarsi, T. Chai and F. L. Lewis (2016). Data-driven optimal control of operational indices for a class of industrial processes. *IET Control Theory & Applications* 10(12), p. 1348–1356. doi: 10.1049/iet-cta.2015.0798.
- Luo, J., E. Solowjow, C. Wen, J. A. Ojea and A. M. Agogino (2018). Deep Reinforcement Learning for Robotic Assembly of Mixed Deformable and Rigid Objects. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain. doi: 10.1109/IROS.2018.8594353.

- Luo, J., E. Solowjow, C. Wen, J. A. Ojea, A. M. Agogino, A. Tamar and P. Abbeel (2019). Reinforcement Learning on Variable Impedance Controller for High-Precision Robotic Assembly. In: *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada. doi: 10.1109/ICRA.2019.8793506.
- Luo, S. (2020). Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Applied Soft Computing* 91, p. 106208. doi: 10.1016/j.asoc.2020.106208.
- Luong, N. C., D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang and D. I. Kim (2019). Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. *IEEE Communications Surveys Tutorials* 21(4), p. 3133–3174. doi: 10.1109/COMST.2019.2916583.
- Lämmle, A., T. König, M. El-Shamouty and M. F. Huber (2020). Skill-based Programming of Force-controlled Assembly Tasks using Deep Reinforcement Learning. *Procedia CIRP* 93, p. 1061 – 1066. doi: <https://doi.org/10.1016/j.procir.2020.04.153>.
- Ma, Y., W. Zhu, M. G. Benton and J. Romagnoli (2019). Continuous control of a polymerization system with deep reinforcement learning. *Journal of Process Control* 75, p. 40 – 47. doi: 10.1016/j.jprocont.2018.11.004.
- Mahadevan, S. and G. Theodorou (1998). Optimizing Production Manufacturing Using Reinforcement Learning. In: *Proceedings of the Eleventh International FLAIRS Conference*, p. 372–377.
- Malus, A., D. Kozjek and R. Vrabič (2020). Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. *CIRP Annals* 69(1), p. 397 – 400. doi: 10.1016/j.cirp.2020.04.001.
- Masinelli, G., T. Le-Quang, S. Zanolli, K. Wasmer and S. A. Shevchik (2020). Adaptive Laser Welding Control: A Reinforcement Learning Approach. *IEEE Access* 8, p. 103803–103814. doi: 10.1109/ACCESS.2020.2998052.
- Mazgualdi, C. E., T. Masrour, I. E. Hassani and A. Khoudi (2021). A Deep Reinforcement Learning (DRL) Decision Model for Heating Process Parameters Identification in Automotive Glass Manufacturing. In: *Artificial Intelligence and Industrial Applications*, Volume 1193, p. 77–87. Cham: Springer International Publishing.
- Miljković, Z., M. Mitić, M. Lazarević and B. Babić (2013). Neural network Reinforcement Learning for visual control of robot manipulators. *Expert Systems with Applications* 40(5), p. 1721–1736. doi: 10.1016/j.eswa.2012.09.010.
- Mishra, M., J. Nayak, B. Naik and A. Abraham (2020). Deep learning in electrical utility industry: A comprehensive review of a decade of research. *Engineering Applications of Artificial Intelligence* 96, p. 104000. doi: 10.1016/j.engappai.2020.104000.

- Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller (2013). Playing Atari with Deep Reinforcement Learning. p. arXiv:1312.5602.
- Mohammed, M. Q., K. L. Chung and S. C. Chua (2020). Review of Deep Reinforcement Learning-Based Object Grasping: Techniques, Open Challenges, and Recommendations. *IEEE Access* 8, p. 178450–178481. doi: 10.1109/ACCESS.2020.3027923.
- Mosavi, A., Y. Faghan, P. Ghamisi, P. Duan, S. F. Ardabili, E. Salwana and S. S. Band (2020). Comprehensive Review of Deep Reinforcement Learning Methods and Applications in Economics. *Mathematics* 8(10), p. 1640. doi: 10.3390/math8101640.
- Naeem, M., S. T. H. Rizvi and A. Coronato (2020). A Gentle Introduction to Reinforcement Learning and its Application in Different Fields. *IEEE Access* 8, p. 209320–209344. doi: 10.1109/ACCESS.2020.3038605.
- Nguyen, H. and H. La (2019). Review of Deep Reinforcement Learning for Robot Manipulation. In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*, Naples, Italy. doi: 10.1109/IRC.2019.00120.
- Noel, M. M. and B. J. Pandian (2014). Control of a nonlinear liquid level system using a new artificial neural network based reinforcement learning approach. *Applied Soft Computing* 23, p. 444 – 451. doi: 10.1016/j.asoc.2014.06.037.
- Oh, T.-H., J.-S. Han, Y.-S. Kim, D.-Y. Yang, S.-H. Lee and D.-I. D. Cho (2020). Deep RL Based Notch Filter Design Method for Complex Industrial Servo Systems. *International Journal of Control, Automation and Systems* 18(12), p. 2983–2992. doi: 10.1007/s12555-020-0153-y.
- Palombarini, J. A. and E. C. Martinez (2018). Automatic Generation of Rescheduling Knowledge in Socio-technical Manufacturing Systems using Deep Reinforcement Learning. In: *2018 IEEE Biennial Congress of Argentina (ARGENCON)*, San Miguel de Tucumán, Argentina. doi: 10.1109/ARGENCON.2018.8646172.
- Palombarini, J. A. and E. C. Martínez (2019). Closed-loop Rescheduling using Deep Reinforcement Learning. *IFAC-PapersOnLine* 52(1), p. 231 – 236. doi: 10.1016/j.ifacol.2019.06.067.
- Pandian, B. J. and M. M. Noel (2018). Tracking Control of a Continuous Stirred Tank Reactor Using Direct and Tuned Reinforcement Learning Based Controllers. *Chemical Product and Process Modeling* 13(3). doi: 10.1515/cppm-2017-0040.
- Park, I.-B., J. Huh, J. Kim and J. Park (2020). A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities. *IEEE Transactions on Automation Science and Engineering* 17(3), p. 1420–1431. doi: 10.1109/TASE.2019.2956762.
- Park, J., J. Chun, S. H. Kim, Y. Kim and J. Park (2021). Learning to schedule job-shop problems: representation and policy learning using graph neural network and reinforcement learning.

- International Journal of Production Research*, p. 1–18. doi: 10.1080/00207543.2020.1870013.
- Park, J., S. Lee, J. Lee and J. Um (2020). GadgetArm—Automatic Grasp Generation and Manipulation of 4-DOF Robot Arm for Arbitrary Objects Through Reinforcement Learning. *Sensors* 20(21), p. 6183. doi: 10.3390/s20216183.
- Peres, R. S., X. Jia, J. Lee, K. Sun, A. W. Colombo and J. Barata (2020). Industrial Artificial Intelligence in Industry 4.0 - Systematic Review, Challenges and Outlook. *IEEE Access* 8, p. 220121–220139. doi: 10.1109/ACCESS.2020.3042874.
- Petticrew, M. and H. Roberts (Hrsg.) (2006). *Systematic Reviews in the Social Sciences*. Oxford, UK: Blackwell Publishing Ltd. doi: 10.1002/9780470754887.
- Powell, B. K. M., D. Machalek and T. Quah (2020). Real-time optimization using reinforcement learning. *Computers & Chemical Engineering* 143, p. 107077. doi: 10.1016/j.compchemeng.2020.107077.
- Quah, T., D. Machalek and K. M. Powell (2020). Comparing Reinforcement Learning Methods for Real-Time Optimization of a Chemical Process. *Processes* 8(11), p. 1497. doi: 10.3390/pr8111497.
- Riedmiller, S. and M. Riedmiller (1999). A neural reinforcement learning approach to learn local dispatching policies in production scheduling. *Proceedings of the 16th international joint conference on Artificial intelligence* 2, p. 764–769.
- Rossit, D. A., F. Tohmé and M. Frutos (2019). Industry 4.0: Smart Scheduling. *International Journal of Production Research* 57(12), p. 3802–3813. doi: 10.1080/00207543.2018.1504248.
- Rummukainen, H. and J. K. Nurminen (2019). Practical Reinforcement Learning - Experiences in Lot Scheduling Application. *IFAC-PapersOnLine* 52(13), p. 1415–1420. doi: 10.1016/j.ifacol.2019.11.397.
- Samsonov, V., C. Enslin, H.-G. Köpken, S. Baer and D. Lütticke (2020). Using Reinforcement Learning for Optimization of a Workpiece Clamping Position in a Machine Tool:. *Proceedings of the 22nd International Conference on Enterprise Information Systems*, p. 506–514. doi: 10.5220/0009354105060514.
- Schaul, T., J. Quan, I. Antonoglou and D. Silver (2016). Prioritized Experience Replay. In: *International Conference on Learning Representations*, San Juan, Puerto Rico.
- Scheiderer, C., T. Thun, C. Idzik, A. F. Posada-Moreno, A. Krämer, J. Lohmar, G. Hirt and T. Meisen (2020). Simulation-as-a-Service for Reinforcement Learning Applications by Example of Heavy Plate Rolling Processes. *Procedia Manufacturing* 51, p. 897 – 903. doi: 10.1016/j.promfg.2020.10.126.
- Scheiderer, C., T. Thun and T. Meisen (2019). Bézier Curve Based Continuous and Smooth

- Motion Planning for Self-Learning Industrial Robots. *Procedia Manufacturing* 38, p. 423 – 430. doi: 10.1016/j.promfg.2020.01.054.
- Schmidt, A., F. Schellroth and O. Riedel (2020). Control architecture for embedding reinforcement learning frameworks on industrial control hardware. In: *Proceedings of the 3rd International Conference on Applications of Intelligent Systems*, Las Palmas de Gran Canaria Spain. doi: 10.1145/3378184.3378198.
- Schoettler, G., A. Nair, J. Luo, S. Bahl, J. A. Ojea, E. Solowjow and S. Levine (2020). Deep Reinforcement Learning for Industrial Insertion Tasks with Visual Inputs and Natural Rewards. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, Nevada, USA. doi: 10.1109/IROS45743.2020.9341714.
- Serin, G., B. Sener, M. Ozbayoglu and H. O. Unver (2020). Review of tool condition monitoring in machining and opportunities for deep learning. *The International Journal of Advanced Manufacturing Technology* 109(3-4), p. 953–974. doi: 10.1007/s00170-020-05449-w.
- Sewak, M. (2019a). *Deep Reinforcement Learning: Frontiers of Artificial Intelligence* (1st ed.). Singapore: Springer Singapore. doi: 10.1002/9780470754887.
- Sewak, M. (2019b). Policy-Based Reinforcement Learning Approaches: Stochastic Policy Gradient and the REINFORCE Algorithm. In: *Deep Reinforcement Learning*, p. 127–140. Singapore: Springer Singapore. doi: 10.1007/978-981-13-8285-7_10.
- Shi, D., W. Fan, Y. Xiao, T. Lin and C. Xing (2020). Intelligent scheduling of discrete automated production line via deep reinforcement learning. *International Journal of Production Research* 58(11), p. 3362–3380. doi: 10.1080/00207543.2020.1717008.
- Shyalika, C., T. Silva and A. Karunananda (2020). Reinforcement Learning in Dynamic Task Scheduling: A Review. *SN Computer Science* 1(6), p. 306. doi: 10.1007/s42979-020-00326-5.
- Silva, T. and A. Azevedo (2019). Production flow control through the use of reinforcement learning. *Procedia Manufacturing* 38, p. 194 – 202. doi: 10.1016/j.promfg.2020.01.026.
- Silver, D., T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan and D. Hassabis (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv:1712.01815 [cs]*.
- Spielberg, S., R. Gopaluni and P. Loewen (2017). Deep reinforcement learning approaches for process control. In: *2017 6th International Symposium on Advanced Control of Industrial Processes*, Taipei, Taiwan. doi: 10.1109/ADCONIP.2017.7983780.
- Spielberg, S., A. Tulshyan, N. P. Lawrence, P. D. Loewen and R. Bhushan Gopaluni (2019). Toward self-driving processes: A deep reinforcement learning approach to control. *AICHE*

- Journal* 65(10). doi: 10.1002/aic.16689.
- Stricker, N., A. Kuhnle, R. Sturm and S. Friess (2018). Reinforcement learning for adaptive order dispatching in the semiconductor industry. *CIRP Annals* 67(1), p. 511 – 514. doi: 10.1016/j.cirp.2018.04.041.
- Sutton, R. S. and A. G. Barto (2017). *Reinforcement learning: an introduction* (2nd ed.). Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press. ISBN: 978-0-262-03924-6.
- Szarski, M. and S. Chauhan (2021). Composite temperature profile and tooling optimization via Deep Reinforcement Learning. *Composites Part A: Applied Science and Manufacturing* 142, p. 106235. doi: 10.1016/j.compositesa.2020.106235.
- Tewari, A., K.-H. Liu and D. Papageorgiou (2020). Information-theoretic sensor planning for large-scale production surveillance via deep reinforcement learning. *Computers & Chemical Engineering* 141, p. 106988. doi: 10.1016/j.compchemeng.2020.106988.
- Thomé, A. M. T., L. F. Scavarda and A. J. Scavarda (2016). Conducting systematic literature review in operations management. *Production Planning & Control* 27(5), p. 408–420. doi: 10.1080/09537287.2015.1129464.
- Tranfield, D., D. Denyer and P. Smart (2003). Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. *British Journal of Management* 14(3), p. 207–222. doi: 10.1111/1467-8551.00375.
- Tsai, Y.-T., C.-H. Lee, T.-Y. Liu, T.-J. Chang, C.-S. Wang, S. J. Pawar, P.-H. Huang and J.-H. Huang (2020). Utilization of a reinforcement learning algorithm for the accurate alignment of a robotic arm in a complete soft fabric shoe tongues automation process. *Journal of Manufacturing Systems* 56, p. 501 – 513. doi: 10.1016/j.jmsy.2020.07.001.
- van Hasselt, H., A. Guez and D. Silver (2016). Deep Reinforcement Learning with Double Q-learning. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, p. 2094–2100. arXiv: 1509.06461.
- Wang, F., X. Zhou, J. Wang, X. Zhang, Z. He and B. Song (2020). Joining Force of Human Muscular Task Planning With Robot Robust and Delicate Manipulation for Programming by Demonstration. *IEEE/ASME Transactions on Mechatronics* 25(5), p. 2574–2584. doi: 10.1109/TMECH.2020.2997799.
- Wang, H., B. R. Sarker, J. Li and J. Li (2020). Adaptive scheduling for assembly job shop with uncertain assembly times based on dual Q-learning. *International Journal of Production Research*. doi: 10.1080/00207543.2020.1794075.
- Wang, H.-n., N. Liu, Y.-y. Zhang, D.-w. Feng, F. Huang, D.-s. Li and Y.-m. Zhang (2020).

- Deep reinforcement learning: a survey. *Frontiers of Information Technology & Electronic Engineering* 21(12), p. 1726–1744. doi: 10.1631/FITEE.1900533.
- Wang, J. P., Y. K. Shi, W. S. Zhang, I. Thomas and S. H. Duan (2019). Multitask Policy Adversarial Learning for Human-Level Control With Large State Spaces. *IEEE Transactions on Industrial Informatics* 15(4), p. 2395–2404. doi: 10.1109/TII.2018.2881266.
- Wang, K., B. Kang, J. Shao and J. Feng (2020). Improving Generalization in Reinforcement Learning with Mixture Regularization. In: *34th Conference on Neural Information Processing Systems*, Vancouver, Canada.
- Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp and A. Kyek (2018). Deep reinforcement learning for semiconductor production scheduling. In: *2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, Saratoga Springs, NY, USA. doi: 10.1109/ASMC.2018.8373191.
- Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp and A. Kyek (2018). Optimization of global production scheduling with deep reinforcement learning. *Procedia CIRP* 72, p. 1264 – 1269. doi: 10.1016/j.procir.2018.03.212.
- Wasmer, K., T. Le-Quang, B. Meylan and S. Shevchik (2019). In Situ Quality Monitoring in AM Using Acoustic Emission: A Reinforcement Learning Approach. *Journal of Materials Engineering and Performance* 28(2), p. 666–672. doi: 10.1007/s11665-018-3690-2.
- Watanabe, K. and S. Inada (2020). Search algorithm of the assembly sequence of products by using past learning results. *International Journal of Production Economics* 226, p. 107615. doi: 10.1016/j.ijpe.2020.107615.
- Watkins, C. J. C. H. and P. Dayan (1992). Q-learning. *Machine Learning* 8(3-4), p. 279–292. doi: 10.1007/BF00992698.
- Webster, J. and R. T. Watson (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly* 26(2), p. xiii–xxiii. Publisher: Management Information Systems Research Center, University of Minnesota.
- Wu, C.-X., M.-H. Liao, M. Karatas, S.-Y. Chen and Y.-J. Zheng (2020). Real-time neural network scheduling of emergency medical mask production during COVID-19. *Applied Soft Computing* 97, p. 106790. doi: 10.1016/j.asoc.2020.106790.
- Wu, W., Z. Huang, J. Zeng and K. Fan (2021). A fast decision-making method for process planning with dynamic machining resources via deep reinforcement learning. *Journal of Manufacturing Systems* 58, p. 392 – 411. doi: 10.1016/j.jmsy.2020.12.015.
- Xanthopoulos, A. S., A. Kiatipis, D. E. Koulouriotis and S. Stieger (2018). Reinforcement Learning-Based and Parametric Production-Maintenance Control Policies for a Deteriorating

- Manufacturing System. *IEEE Access* 6, p. 576–588.
- Xia, K., C. Sacco, M. Kirkpatrick, C. Saidu, L. Nguyen, A. Kircaliali and R. Harik (2020). A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence. *Journal of Manufacturing Systems* 58. doi: 10.1016/j.jmsy.2020.06.012.
- Xie, S., T. Zhang and O. Rose (2019). Online Single Machine Scheduling Based on Simulation and Reinforcement Learning. In: *18. ASIM Fachtagung Simulation in Produktion und Logistik*, Chemnitz.
- Xiong, H. and X. Diao (2021). Safety robustness of reinforcement learning policies: A view from robust control. *Neurocomputing* 422, p. 12–21. doi: 10.1016/j.neucom.2020.09.055.
- Xu, J., Z. Hou, W. Wang, B. Xu, K. Zhang and K. Chen (2019). Feedback Deep Deterministic Policy Gradient With Fuzzy Reward for Robotic Multiple Peg-in-Hole Assembly Tasks. *IEEE Transactions on Industrial Informatics* 15(3), p. 1658–1667. doi: 10.1109/TII.2018.2868859.
- Xu, L. D., E. L. Xu and L. Li (2018). Industry 4.0: state of the art and future trends. *International Journal of Production Research* 56(8), p. 2941–2962. doi: 10.1080/00207543.2018.1444806.
- Xu, X., H. Xie and J. Shi (2020). Iterative Learning Control (ILC) Guided Reinforcement Learning Control (RLC) Scheme for Batch Processes. In: *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, Liuzhou. doi: 10.1109/DDCLS49620.2020.9275065.
- Yang, H., A. Alphones, W.-D. Zhong, C. Chen and X. Xie (2020). Learning-Based Energy-Efficient Resource Management by Heterogeneous RF/VLC for Ultra-Reliable Low-Latency Industrial IoT Networks. *IEEE Transactions on Industrial Informatics* 16(8), p. 5565–5576. doi: 10.1109/TII.2019.2933867.
- Yoo, H., B. Kim, J. W. Kim and J. H. Lee (2021). Reinforcement learning based optimal control of batch processes using Monte-Carlo deep deterministic policy gradient with phase segmentation. *Computers & Chemical Engineering* 144, p. 107133. doi: 10.1016/j.compchemeng.2020.107133.
- Yu, J. and P. Guo (2020). Run-to-Run Control of Chemical Mechanical Polishing Process Based on Deep Reinforcement Learning. *IEEE Transactions on Semiconductor Manufacturing* 33(3), p. 454–465. doi: 10.1109/TSM.2020.3002896.
- Yu, T., J. Huang and Q. Chang (2020). Mastering the Working Sequence in Human-Robot Collaborative Assembly Based on Reinforcement Learning. *IEEE Access* 8, p. 163868–163877. doi: 10.1109/ACCESS.2020.3021904.
- Zhang, N. and W. Si (2020). Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks. *Reliability Engineering*

- & *System Safety* 203, p. 107094. doi: 10.1016/j.res.2020.107094.
- Zhang, T., F. Zhou, J. Zhao and W. Wang (2020). Deep Reinforcement Learning for Secondary Energy Scheduling in Steel Industry. In: *2020 2nd International Conference on Industrial Artificial Intelligence (IAI)*, Shenyang, China. doi: 10.1109/IAI50351.2020.9262196.
- Zhang, W. and T. G. Dietterich (1995). A Reinforcement Learning Approach to Job-Shop Scheduling. *Proceedings of the 14th International Joint Conference on Artificial Intelligence Volume 2*, p. 1114–1120.
- Zhao, M., X. Guo, X. Zhang, Y. Fang and Y. Ou (2019). ASPW-DRL: assembly sequence planning for workpieces via a deep reinforcement learning approach. *Assembly Automation* 40(1), p. 65–75. doi: 10.1108/AA-11-2018-0211.
- Zhao, X., H. Zhao, P. Chen and H. Ding (2020). Model accelerated reinforcement learning for high precision robotic assembly. *International Journal of Intelligent Robotics and Applications* 4(2), p. 202–216. doi: 10.1007/s41315-020-00138-z.
- Zheng, S., C. Gupta and S. Serita (2020). Manufacturing Dispatching Using Reinforcement and Transfer Learning. *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, p. 655–671.
- Zhou, L., L. Zhang and B. K. P. Horn (2020). Deep reinforcement learning-based dynamic scheduling in smart manufacturing. *Procedia CIRP* 93, p. 383 – 388. doi: 10.1016/j.procir.2020.05.163.
- Zhou, T., D. Tang, H. Zhu and L. Wang (2021). Reinforcement Learning With Composite Rewards for Production Scheduling in a Smart Factory. *IEEE Access* 9, p. 752–766. doi: 10.1109/ACCESS.2020.3046784.
- Zhou, Y., T. Xing, Y. Song, Y. Li, X. Zhu, G. Li and S. Ding (2020). Digital-twin-driven geometric optimization of centrifugal impeller with free-form blades for five-axis flank milling. *Journal of Manufacturing Systems* 58 (b), p. 22–35. doi: 10.1016/j.jmsy.2020.06.019.
- Zhou, Z., X. Li and R. N. Zare (2017). Optimizing Chemical Reactions with Deep Reinforcement Learning. *ACS Central Science* 3(12), p. 1337–1344. doi: 10.1021/acscentsci.7b00492.
- Zhu, H., M. Li, Y. Tang and Y. Sun (2020). A Deep-Reinforcement-Learning-Based Optimization Approach for Real-Time Scheduling in Cloud Manufacturing. *IEEE Access* 8, p. 9987–9997. doi: 10.1109/ACCESS.2020.2964955.
- Zhu, K., N. Ji and X. D. Li (2020). Hybrid Heuristic Algorithm Based On Improved Rules Reinforcement Learning for 2D Strip Packing Problem. *IEEE Access* 8, p. 226784–226796. doi: 10.1109/ACCESS.2020.3045905.
- Zimmerling, C., C. Poppe and L. Kärger (2020). Estimating Optimum Process Parameters in

Textile Draping of Variable Part Geometries - A Reinforcement Learning Approach. *Procedia Manufacturing* 47, p. 847 – 854. doi: 10.1016/j.promfg.2020.04.263.

Zou, Y. and R. Lan (2020). An End-to-End Calibration Method for Welding Robot Laser Vision Systems With Deep Reinforcement Learning. *IEEE Transactions on Instrumentation and Measurement* 69(7), p. 4270–4280. doi: 10.1109/TIM.2019.2942533.

5 Publication 2

Neural agent-based production planning and control: an architectural review

Marcel Panzer^{1a}, Benedict Bender^a, and Norbert Gronau^a

^a *Chair of Business Informatics, Processes and Systems, University of Potsdam,
Karl-Marx-Street 67, 14482 Potsdam, Germany*

ABSTRACT

Nowadays, production planning and control must cope with mass customization, increased fluctuations in demand, and high competition pressures. Despite prevailing market risks, planning accuracy and increased adaptability in the event of disruptions or failures must be ensured, while simultaneously optimizing key process indicators. To manage that complex task, neural networks that can process large quantities of high-dimensional data in real time have been widely adopted in recent years. Although these are already extensively deployed in production systems, a systematic review of applications and implemented agent embeddings and architectures has not yet been conducted. The main contribution of this paper is to provide researchers and practitioners with an overview of applications and applied embeddings and to motivate further research in neural agent-based production. Findings indicate that neural agents are not only deployed in diverse applications, but are also increasingly implemented in multi-agent environments or in combination with conventional methods - leveraging performances compared to benchmarks and reducing dependence on human experience. This not only implies a more sophisticated focus on distributed production resources, but also broadening the perspective from a local to a global scale. Nevertheless, future research must further increase scalability and reproducibility to guarantee a simplified transfer of results to reality.

Keywords

Production planning and control, Machine learning, Neural networks, Systematic literature review, Taxonomy

¹Corresponding author

Submitted to the Journal of Manufacturing Systems on 8 February 2022, accepted on 25 October 2022.

5.1 Introduction

Despite growing market uncertainties and increasingly complex product structures up to mass customization, production planning and control (PPC) must enable a robust production and meet internal and external customer requirements. Besides common key performance indicators (KPIs) such as product quality or lead time, these increasingly include aspects of sustainability and the ability to adapt quickly to new environmental conditions. The remarkable set of addressable capabilities, performance measures, and environmental factors that can be significantly leveraged through intelligent production planning and control has already been analyzed by Bueno et al. (2020), indicating a wide range of potentials for process optimization.

To reduce system complexity, besides single-agent (SA) systems, various multi-agent (MA) implementations have been proposed that imply collaborative, competitive, or mixed-agent interactions (Babiceanu and Chen, 2006; Gronauer and Diepold, 2021). In addition, machine learning (ML) is increasingly employed due to the growing capabilities of the given infrastructure in recent years (Kang et al., 2020). ML can assist in performing multi-criteria optimization involving local and global objectives, multiple resources, machines, and factories (Zhang et al., 2019) that demand a continuously optimized production control and schedule (Kang et al., 2020).

However, according to Cadavid et al. (2019), 75% of potential research domains in the field of ML-based PPC have not yet been sufficiently investigated. This also becomes apparent in the work of Liao et al. (2017), who state that while Big Data and other disciplines are increasingly focused on PPC-related research, ML lags behind these. This impression has already been countered in a previous review of ours in the field of deep reinforcement learning (RL)-based production (Panzer and Bender, 2021), but also by others such as Weichert et al. (2019), Kang et al. (2020), and Zhou et al. (2022), highlighting the versatility of ML algorithms in various production scenarios. Nevertheless, Weichert et al. (2019) emphasize that the integration of an ML model for the optimization of production processes must be carried out carefully to balance the increasing process and model complexities and ensure that appropriate decisions are made regarding the algorithmic structure and its interaction with the environment.

As one possible ML technique, (deep) neural networks (NN) in particular are increasingly utilized in production due to their ability to process large amounts of data in real time and map complex non-linear interdependencies, thus avoiding the need for complex models (Cadavid et al., 2019). Our review specifically addresses the application and embedding of NN-based algorithms in production as a data-driven online optimization approach and highlights their beneficial properties for production systems. Considering the flexible and scalable properties and high performance of NNs, our contribution aims to capture the current state of the art in real and simulated production systems. Furthermore, we want to identify existing challenges and derive future research directions.

Already in 1990, Rabelo et al. (1990) demonstrated the superior abilities of a hybrid scheduling approach to combine NN-based pattern identification and expert-based constraint refinement. By identifying scheduling task patterns, Zhou et al. (1990) adopted what Baker (1998) considered to be a heterarchical approach to job shop scheduling - mapping scheduling operations to the network structure. As a result of their successes, Zhang and Huang (1995) and Garetti and Taisch (1999) summarized previous efforts and highlighted the effectiveness of NNs for handling PPC problems. However, in recent reviews, NN-based PPC has only partially been considered in the context of flexible job shop scheduling (Zhang et al., 2019), or just in the context of other ML techniques (Kang et al., 2020), lacking a consolidation of NN-based PPC contributions. This also becomes apparent in the work of De Modesti et al. (2020), who described the increased relevance of NNs but, like Çaliş and Bulkan (2015) for job shop scheduling or Bertolini et al. (2021) for general industrial use cases, incorporated NNs into the context of general ML.

From an organizational perspective, the potential of hierarchical production processes was highlighted by Bitran et al. (1982). The benefits of holonic systems were further outlined, among others, in Babiceanu and Chen (2006) and recently in Derigent et al. (2021), and was featured as one of 6 enablers for smart manufacturing control in Rojas and Rauch (2019). Beyond that, Lee and Kim (2008) and Monostori et al. (2014) outlined how MA systems enable robust and flexible production, similar to Gronauer and Diepold (2021) or Herrera et al. (2020), who focused on deep reinforcement learning as a possible implementation of MA systems and general systems engineering. However, a focused review of the existing results of NN-based PPC and the applied architectures has yet to be conducted.

To the best of our knowledge, this is the first attempt to capture the main findings of NN-based applications and agent embeddings in PPC. The review should serve practitioners in identifying potential research directions and provide incentives for implementation. We intend to highlight performance potentials that might arise from applying NN-based PPC in practice, but also emphasize existing challenges. For this purpose, we attempt to answer the following research questions.

- RQ1: What are current neural network applications in PPC?
- RQ2: What are the predominant neural network-based PPC embeddings?
- RQ3: What are major challenges of the reviewed PPC implementations?
- RQ4: How can those challenges be addressed and what future fields of research emerge?

The paper is structured as follows: Section 2 describes the basics of NN-based PPC methods. Section 3 specifies the methodology and conceptual framework of the review. Section 4 answers RQ1 and RQ2 based on the conducted review. Section 5 outlines the corresponding taxonomy design followed by the predominant challenges (RQ3) and future research fields (RQ4) in Section

6. Section 7 discusses the results of the review, existing limitations, and managerial insights. A conclusion is provided in Section 5.8. Analysis tables with detailed review information can be found in the Appendix.

5.2 Neural network-based production planning and control

The goal of PPC is to maintain production and meet the desired technical, financial, and organizational objectives, even given uncertainties around the markets and production itself (Zipfel et al., 2019). Production planning refers to disciplines such as scheduling, which must cope with multi-product environments, limited resources, and rush orders to achieve high efficiency and cost-effectiveness. Production control, such as dispatching, on the other hand, must execute planned actions taking into account unsteady conditions such as machine status or varying processing times to compensate for unforeseen events and maintain stable and robust production (Ramsauer, 1997). Related to Industry 4.0, the adoption of technologically advanced techniques in PPC can be deployed to improve performance (Kagermann et al., 2013). In recent years, this has included NNs in particular, which have not only experienced great success with Google DeepMind (Silver et al., 2017) but are also increasingly implemented in production and can prevent extensive modeling or high dependence on human experience (as in Ding et al. (2019)).

5.2.1 Neural networks

NNs can learn (long-term) dependencies and exploit past experiences gained. The networks learn and store experiences by updating the strength of the neural connections, which enables real-time computations and adaptive behavior. Based on non-linear computations that mimic the nervous system, inputs are processed and outputs are derived in the form of direct action recommendations, classifications, or others. Besides feed-forward networks (FFNN), which process inputs in one direction, others such as recurrent NN, long-short-term memory networks (LSTM), or deep belief networks possess different forms of information processing and provide certain properties and strengths (Mehlig, 2021). NNs can help to increase the performance of ML algorithms such as (semi-)supervised, unsupervised, or reinforcement learning through their ability to process large and stochastic data sets while still exhibiting high generalizability (Mehlig, 2021; Arunraj and Ahrens, 2015).

5.2.2 ML-based PPC

As a data-driven optimization method, NN-based ML approaches can help not only to optimize production schedules and control, but also to maintain robust operation of production lines.

Whereas conventional decision rules often have problems coping with machine failures or other dynamic and stochastic events occurring such as new order entries, intelligent agents can help not only to reduce problem complexity by means of task decomposition but also to better deal with the above incidents due to their learning behavior (Csáji et al., 2006; Kádár et al., 2003). According to Baker (1998), an agent is a self-controlled software object that has its own values and communicates with other objects. Based on this, Patel et al. (2001) attributes intelligent properties to this agent, enabling it to interpret its perceptions and independently select actions to pursue its specific goals and, through its learning skills, to adapt its behavior to the changing environment (Ueda et al., 2001). Early approaches in 1995 such as Zhang and Dietterich (1995a) and Zhang and Dietterich (1995b) demonstrated the superiority of a reinforcement learning-based scheduling mechanism over an iterative repair-based scheduling. Having more agents available, early MA and NN-based approaches were proposed to optimize PPC problems (Riedmiller and Riedmiller, 1999; Monostori et al., 2004). Riedmiller and Riedmiller (1999) pursued an RL-based dispatching approach consisting of distributed machine agents that learn local dispatching rules and decide on which order to process, thereby outperforming heuristics while demonstrating good generalization behavior. Monostori et al. (2004) proposed a 3-level MA scheduling scheme consisting of order, mobile, and resource agents. Herein, mobile agents explore possible routes, and those with the best schedule yield the final one, which significantly reduces computational costs with increasing operation numbers compared to a branch and bound algorithm. Since a detailed introduction of algorithms and MA systems would go beyond the scope of this paper, we would like to refer to Aggarwal (2018) and Dorri et al. (2018), respectively.

5.2.3 MA system organization

A further differentiation of MA systems is established by classifying them as hierarchical, heterarchical, or holonic structures, depending on their agent collaboration (Beigi and Mozayani, 2016). Whereas a hierarchy is characterized by multiple master-slave relationships, a heterarchy predominantly consists of peer-level relationships with distributed privileges to satisfy global and local objectives (Baker, 1998). The intermediate step between both extremes is characterized as a holonic structure (Bongaerts et al., 2000). Agent interaction itself can be classified as either a collaborative way to achieve a common (global) goal or a competitive way, in which each agent tries to accomplish its own goal (Hoen et al., 2006). For further classification, we additionally differentiate between MA, incorporated embedded, and plain NN agent designs. Plain NN approaches employ one or more NNs using the same ML method, like target and value network in deep Q-learning, to solve a task. Embedded approaches can consist of multiple NN-based learning methods, but also combine NN approaches with heuristics in a construct consisting of multiple stages. Each stage can address a sub-problem that contributes to the solution of the whole task.

Unlike other algorithmic or ML-focused reviews in manufacturing, applications and embeddings of NNs in PPC have not yet been consolidated in a focused manner. To address this gap and illustrate the diversity of existing approaches, an overview of applications and NN embeddings can help practitioners and researchers to identify individual use-cases and highlight challenges and fields for future research.

5.3 Methodology

The following section specifies the review methodology that is used to identify relevant NN-based PPC publications. To ensure a comprehensive and transparent review and content analysis, we follow the guidelines provided by Tranfield et al. (2003) and Thomé et al. (2016). Thereby, we try to consolidate and analyze relevant research in the field at the time of the review and provide researchers with much faster access. This will help researchers and practitioners identify research gaps, incentivize research, and provide management insights (Petticrew and Roberts, 2006). Following Thomé et al. (2016), we have organized the systematic literature review (SLR) into 8 (iterative) steps, from planning to updating the review, which are addressed next.

5.3.1 Review focus

The research questions to be answered and current research needs were discussed in Section 1 above. The review team consisted of the three authors of this study, who performed each step separately and eventually merged their work. To specify the problem and scope of the review and facilitate the collection and evaluation of contributions, the review planning is based on Brocke et al. (2009) and follows the associated taxonomy framework of Cooper (1988), which is outlined in Table 5.1. Cells highlighted in gray represent the selection of underlying characteristics of the review and the associated objectives and focus areas.

Characteristic	Categories			
(1) Focus	Research outcomes	Research methods	Theories	Applications
(2) Goal	Integration	Criticism		Central issues
(3) Perspective	Neutral representation		Espousal of position	
(4) Coverage	Exhaustive	Exhaustive & selective	Representative	Central/pivotal
(5) Organisation	Historical	Conceptual		Methodological
(6) Audience	Specialized scholars	General scholars	Practitioners/politicians	General public

Table 5.1 Pursued taxonomy framework

Concerning the presented taxonomy, the review focuses on existing applications and obtained research outcomes and applications of NN-based PPC (1). The goal is to present existing research in an integrative and synthesizing manner, highlighting the benefits but also the prevailing key

challenges of the research field (2). It is intended to provide a neutral (3), representative (4), and conceptual (5) synthesis of the scope under consideration. Finally, the review should appeal to a broad audience (6). We refrain from in-depth algorithmic explanations or other technical details, which benefits general scholars and practitioners while attempting to provide specialized scholars with an overview of detailed research streams. We intend to highlight the broad application opportunities of the deployed NN structures as a promising optimization method in production and inspire further research and implementations.

5.3.2 Literature search

To conduct the review, we initially determined the search terms and underlying databases. The raw literature output was then screened based on pre-defined criteria to obtain the final dataset for the later in-depth analysis.

5.3.2.1 Phase 1 - database and iterative keyword selection

To conduct the review, we included the databases Web of Science (all fields), Scopus (article title, abstract, keywords), and IEEE Xplore (journals) to identify relevant publications. The keywords were defined in an iterative process and are listed in Table 5.2. Besides a keyword category that addresses deep learning algorithms, a domain-based category was included that covers relevant aspects of production planning and control. To obtain the intended scope of papers, organizational keywords were not included. Terms such as *Holonic* or *Heterarchic* were rarely mentioned and reduced the hit ratio in the search query.

Algorithmic keywords		Domain keywords		
Artificial intelligence OR	AND	Assembly OR	Control OR	
Deep learning OR			Dispatching OR	
Intelligent OR		Manufacturing OR	AND	Planning OR
Machine learning OR		Production	AND	Scheduling
Neural network				

Table 5.2 Keywords defined for the review

5.3.2.2 Phase 2 - defining inclusion and exclusion criteria

To define a clear review scope and systematically constrain the obtained literature set, we established several inclusion and exclusion criteria. To ensure high quality, we only considered publications from peer-reviewed journals, proceedings, conference papers, and books (as in Light and Pillemer (1984)). Working papers, pre-prints, and other non-peer-reviewed publications were not included. In addition, we considered only publications in English and, because of the significant improvements of NN performance in recent years, those that were published after

2010. For instance, it was not until the release of Mnih et al. (2013) in 2013 that the field of deep RL was enabled on a large scale and with high performance in various applications. Based on the defined research questions and taxonomy, we defined thematic inclusion and exclusion criteria. Due to the focus on NN-based PPC applications and the subsequent analysis of the employed organization and interactions, papers were excluded that primarily dealt with methodology development, theory generation, or algorithms without validating them for an explicit production use case. Other reviews were accessed to identify additional potentially relevant papers. Given the focus, we only considered papers that address a real or simulated NN implementation in PPC and attempt to leverage production performance. Papers that do not use NNs were not reviewed.

5.3.2.3 Phase 3 - conducting the literature search

The review process was conducted from October through December 2021, with a final data retrieval completed on 12/29/2021. A summary of the review is outlined in Figure 5.1. Beginning with the database extraction and the 1794 papers initially obtained, duplicates were first removed and years filtered before applying thematic criteria.

To ensure a high review quality, we screened the remaining 708 papers by title, keywords, and abstract according to Thomé et al. (2016) based on the inclusion and exclusion criteria and research questions. In the next step, many papers were excluded due to a lack of production context or missing application of NNs, reducing it to 185 papers. During the full-text review, the remaining set was reduced to 82 and, in addition to the initial essential coding, the groundwork was laid for the forward/backward search. Following the approach proposed by Webster and Watson (2002), the backward/forward search is an essential extension to identify papers beyond the initial search scope. In this last retrieval, an additional 47 papers were found, increasing the total number of papers to be considered to 129.

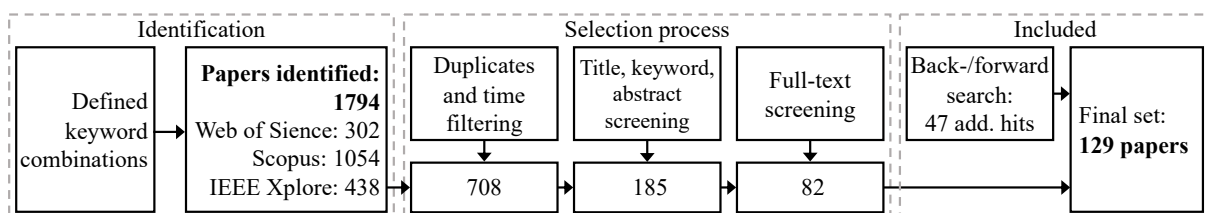


Figure 5.1 Consolidated review process

5.3.2.4 Phase 4 - data gathering

In accordance with Thomé et al. (2016) and Webster and Watson (2002), we developed a concept matrix for the subsequent analysis based on the objectives and research questions.

The categorization and coding of the resulting dataset was based on the PPC domain and agent

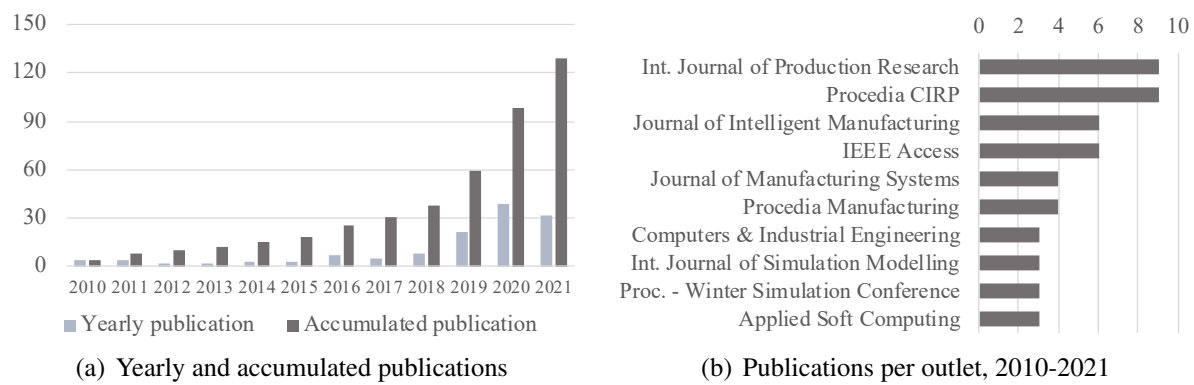


Figure 5.2 Analysis of yearly and outlet publications

configuration as the main criteria. In terms of configuration, the approaches were categorized into plain, embedded, and MA systems. Within these categories, a selective screening for configuration-unspecific and configuration-specific properties was conducted. For all approaches, configuration-unspecific properties included the particular application, the optimization objective, applied algorithms, and NNs, as well as possible benchmark results and deployment in a simulated and/or real environment. For the embedded approaches, the type of embedding and the supplementary implemented algorithms were further examined. In contrast, for MA systems, the type of agent interaction and training of the agent population were additionally considered.

5.3.3 Analysis of yearly and outlet-related contributions

A preliminary analysis of publication years outlines the increased research activity of NN-based PPC in Figure 5.2(a). Whereas a constant number of publications was observed until 2017, it has since increased from 5 in 2017 to 31 in 2021 (the time of the last retrieval), thus highlighting the increased relevance of NN-based PPC.

An analysis of the most frequently cited outlets with three or more published papers is given in Figure 5.2(b). Overall, most findings were published in journals (89, 69%), followed by proceedings (25, 19%) and conference papers (15, 12%). Altogether, contributions from 59 journals, 15 conferences, and 12 proceedings were accessed.

5.4 Analysis

To focus on fundamental developments within the defined scope, a summary of the research field is presented first. Subsequently, the individual categories defined during the iterative analysis are addressed to answer research questions RQ1 and RQ2. Finally, a general analysis is conducted in Section 5.4.4.

Besides the increasing tendency for total publication numbers from Figure 5.2(a), a shift of

research foci within the field becomes apparent in the system split shown on the left in Table 5.3. Whereas in the years 2010-2013 one MA paper was published (8%), they take up 14% of the publications in recent years. Especially in the last year, the field of MA approaches has grown rapidly (8 publications) and is already further along in 2021 than in previous years (3 at the last data retrieval). We may conclude that particularly difficulties in agent communication and collaboration are continuously addressed, and distributed learning is accessible for broader applications.

Although most of the publications within the categorical split on the right of Table 5.3 implemented plain approaches, especially in production control, one-fifth of the publications were based on MA approaches that benefit from a resulting complexity breakdown and increased scalability. The high embedded share within forecasting is also striking, indicating an increased utilization of the combined benefits of the individual methods, such as fuzzy C-Means job classification and subsequent NN cycle time prediction (as in Chen (2016)).

To ensure a consistent structure and address RQ1, the rest of the review is organized according to the section numbers given in Table 5.3 within the categorical split. In each category, we further classified the papers according to the agent organization, either MA, embedded, or plain approach. In addition to production planning and production control, forecasting was incorporated as a further subcategory following the final interactive review step as an increasingly important production planning support tool.

	Yearly split			Categorical split		
	avg. 2010-2013	avg. 2014-2017	avg. 2018-2021	Production planning (4.1)	Production forecasting (4.2)	Production control (4.3)
Plain agent	6 (50%)	11 (61%)	56 (57%)	27 (49%)	23 (64%)	23 (61%)
Embedded agent	5 (42%)	5 (28%)	29 (29%)	20 (36%)	12 (33%)	7 (18%)
Multi-agent	1 (8%)	2 (11%)	14 (14%)	8 (15%)	1 (3%)	8 (21%)
Sum	12	18	99	55	36	38

Table 5.3 System and categorical split of the reviewed literature

For the subsequent sections we have truncated some terms. However, in order to facilitate understanding of the topics addressed, please find below a list of the abbreviations and their meanings.

5.4.1 Production planning

The objective of production planning is to exploit production resources in such a way that the forecast is met and target parameters such as minimum cost are realized, which can comprise

optimal utilization of resources, lot sizing, scheduling, and others (Gelders and Van Wassenhove, 1981).

5.4.1.1 Plain NN planning approaches

The category of plain NN-based approaches employs a single ML method for optimization. As in the other planning categories, most of the papers (85%) were superior to the conventional approaches. Hereby, a common motivation for implementation was the high computational overheads of conventional methods, which were 1000 times smaller in flow shop scheduling when employing a combination of unsupervised RL for training and supervised learning while maintaining the same or better performance (Wu et al., 2020). Also in flow-shop scheduling, Marchesano et al. (2021) demonstrated how a DQN can optimize complex production by selecting dispatching rules as actions. By applying the same rule selection approach using a DDDQN with PER and a DQN, respectively, Han and Yang (2020) and Lin et al. (2019) outperformed heuristics such as FIFO or SPT in job-shop scheduling. For implementation, Lin et al. (2019) employed a multi-class DQN that contained structured indicators for all job-shop machines and corresponding rules for all edge devices summed up in its output layer. Other use cases were implemented by employing an A2C RL algorithm to increase profitability (Hubbs et al., 2020), a DQN RL approach to minimize makespan in dynamic scheduling (Hu et al., 2020), a double DQN in rescheduling color batches to minimize change-over cost (Leng et al., 2020), or a BP algorithm in lot sizing to minimize production, set-up, and inventory costs (Şenyiğit and Atici, 2013), among others listed in Table 5.9.

5.4.1.2 Embedded NN planning approaches

Apart from plain approaches, embedded systems leverage a combined optimization within the internal agent structure during training and operation. In flow-shop scheduling, whereas Kumar and Giri (2020) chose a hybrid fuzzy and NN-based approach to minimize the makespan and reduce time-consuming efforts, Ramanan et al. (2011) proposed two approaches in which a NN generates priorities of a scheduling action, which is subsequently optimized by a heuristic or genetic algorithm (GA). Other superior hybrid approaches were implemented by deploying a NN to prioritize orders and heuristics for resolving ties (Sim et al., 2020), or by splitting the scheduling problem into sub-problems and deploying a GA in training and a convolution two-dimensional transformation that elaborates scheduling features, thus providing a highly generalizable approach (Zang et al., 2019). To improve slow convergences and avoid the local optima trap in job-shop scheduling, Zhang et al. (2019) combined a particle swarm optimization (PSO) algorithm with a NN. Particle positions were associated with weights of the NN and performances were further leveraged by optimizing the sub-problem of machine selection and

scheduling through elite retention and neighborhood search. Another PSO approach was pursued by Lan et al. (2010), employing a NN to estimate the credibility objective, which is embedded in the PSO and takes significantly less time for planning compared to using conventional approximation methods. A virus PSO was implemented by Wen et al. (2015), using a NN to approximate the expectation function by converting an infinite into a finite dimensional optimization problem, thereby solving a 2-stage remanufacturing problem better than a PSO. Another approach was to combine a NN with GA in remanufacturing to prevent the slow convergence of the NN and calculate the target output of chromosomes (Wen et al., 2017; Zhang, 2019).

In a hybrid simulation, Sobottka et al. (2019) leveraged the NN as a meta-modeler of an industrial bakery for a GA, reducing global energy costs by 25%. In a real copper mining complex, Kumar and Dimitrakopoulos (2021) employed a self-play approach using a Kalman filter and Monte Carlo tree search to train a NN, all of which benefited from each other's interaction, improving self-play and increasing cumulative cash flow by 12%. Further approaches of embedded NNs are listed in Table 5.10.

5.4.1.3 Multi-agent planning approaches

To prevent the exploration problem of existing approaches in large state spaces and circumvent the problem of gaining knowledge from stochastic production systems, Hammami et al. (2015) introduced multiple decision agents in job-shop scheduling that chose dispatching rules to reduce mean tardiness. Depending on the performance criterion, each agent had different embedded NNs to choose from and might have received intervention from choice agents assisting with choosing the optimal policy. Other decision agents were contacted as acquaintances to collect information and were optimized concurrently with respect to a global objective.

To reduce high implementation costs of conventional approaches, Liu et al. (2020) and Baer et al. (2019) defined local and shared global rewards to meet production goals and ensure better process adaptation. To subsequently optimize the agents with respect to the global goal, Baer et al. (2019) employed a multi-stage learning strategy in which a single agent was trained locally first while others were controlled by heuristics. Furthermore, Baer et al. (2020) demonstrated the generalizability and scalability of the MA approach, in which each agent was controlled by a DQN RL. By learning the basic task principles and deploying a parameter-sharing training strategy among the agents, training 700 scheduling topologies took only twice as long as training a single one. In the case of a new scenario, the agent slightly modified its policy with respect to the new task specifics, thus reducing reconfiguration time and cost.

Similar to the work of Baer et al. (2020), in which agents did not communicate and only perceived each other's actions, Park et al. (2020) and Lee et al. (2020) pursued planning approaches in

semiconductor and mold scheduling. Both utilized a centralized DQN learning approach and let agents exploit the same NN while benefiting from each other's experiences. Although the agents did not communicate directly, both approaches outperformed conventional ones and did not need to be retrained for new scheduling scenarios. Based on Baer et al. (2020), Pol et al. (2021) integrated a re-training phase after local-only training, in which local rewards were multiplied by a global factor or by receiving sparse global rewards based on eligibility traces. Combined with a policy-sharing strategy among the production agents, the local-only optimization was outperformed.

The only real scenario of MA in production planning was implemented by Zhou et al. (2021) on a small scale and deployed RFID for collaboration among participants to prevent inefficiencies in centralized data processing. This enabled the participants to consider attributes from other machines and learn from the experiences of other agents through mutual updates of manufacturing value networks. Each participant in the scenario, such as a warehouse or drill machine, was provided with a NN, which was activated when a job needed to be scheduled or information was needed. The distributed NNs were trained using RL, and were superior to a central RL and a GA. Implementations in MA-based planning were carried out with flat hierarchies to a large extent in the reviewed papers, which have, however, outperformed conventional methods in all benchmarks as indicated in Table 5.11.

5.4.2 Forecasting

Production forecasting can be deployed, among other opportunities, as a support tool to increase the robustness of planning processes. Often, complex non-linear processes that require sophisticated modeling and cause high computational costs motivate the use of NN-based forecasting as in Worapradya and Thanakijkasem (2015). To avoid terminological conflicts, we categorized each paper according to the key variable addressed.

5.4.2.1 Plain NN forecasting approaches

Plain NN-based forecasting approaches were often adopted due to existing planning uncertainties or complex dependencies, including human factors. In garment production, Onaran and Yanik (2020) predicted cycle time significantly better than linear regression with feature extraction, despite high dependency on human capabilities. Likewise, in textile production, to cope with highly fluctuating process times of different products and avoid production imbalances and the inclusion of human estimates, Cao and Ji (2021) implemented a NN-based cycle-time prediction and obtained a maximum error of 5%. An approach to improve holistic production control and circumvent complex modeling due to non-linear interdependencies was proposed by Glavan et al. (2013), who employed three NNs as black-box models to calculate cost, production, and

quality metrics. To avoid rescheduling due to volatile electricity prices, Windler et al. (2019) proposed a superior approach to the monthly forecast and energy cost-oriented planning. To perform a simulative what-if analysis for production control, Huang et al. (2016) estimated the throughput based on scheduling information, constant work in progress, and mean-time-to-repair levels. In a real petrol mine scenario, by employing six NNs for six wells, Pham and Phan (2016) reached superior results predicting the production rates of liquid, oil, and gas flow to optimize the production back-allocation of each well. While the difference in throughput was only about 2% compared to simulation, the computational effort was reduced about 100 times. A superior flow-time prediction was implemented by Silva et al. (2017) based on job and shop status information to estimate the due date. Based on the flow time, Karaoglan and Karademir (2017) further estimated production costs to generate more precise price offers. Among other papers listed in Table 5.12, Kramer et al. (2020) predicted lead times assuming constantly changing environmental variables, which cannot be captured in regular models. Similarly, Göppert et al. (2021) predicted makespans, which is difficult to achieve through conventional methods in dynamic environments due to ever-changing variables such as remaining jobs states, gate queue lengths, process duration, and others.

5.4.2.2 Embedded NN forecasting approaches

To forecast cycle times in wafer fabrication, Chen (2016) deployed multiple NNs for jobs of different categories, which were determined by a fuzzy c-means classifier beforehand. Compared to conventional approaches, this reduced mean absolute forecasting error by more than 38%. A combined prediction of cycle, blockage, and starvation time in an assembly line was proposed by Lai et al. (2018) by applying a 2-stage LSTM framework, which increased prediction accuracy by 35% compared to conventional approaches. Based on the forecasted cycle time of the first LSTM and the historical cycle time, as well as blockage and starvation time, these two were forecasted in the second stage. In lead-time prediction, Schneckenreither et al. (2021) in a three-stage make-to-order flow shop and Mezzogori et al. (2019) in a 6-machine job shop outperformed conventional approaches (1) by integrating two FFNNs, one of which predicted flow time for bottleneck and non-bottleneck products, and (2) via NN-based LT prediction combined with workload control to determine delivery dates. Other approaches exploited a NN-generated gross demand forecast for subsequent scheduling algorithms to circumvent high computational cost and unknown system dynamics Sadiq et al. (2020) or proposed a combined analytical and LSTM approach Huang et al. (2020) to cope with arising production complexities. While the analytical model calculates the lower bound of the product completion time, the LSTM adds aggregates based on varying inputs in real time, thereby outperforming a plain LSTM-based approach and other conventional ones. Worapradya and Thanakijkasem (2015) predicted the mean and standard deviation of the system performance in a continuous steelmaking casting process by employing

one NN for each machine group. Based on a K-means clustering of machines with similar processes for complexity decomposition, extensive modeling could be avoided and non-linear relationships were reflected more accurately with a computational time that was approximately 30 times lower than a Monte-Carlo simulation. Besides a deep autoencoder and NN-based order remaining time prediction implemented by Fang et al. (2020), other approaches are listed in Table 5.13.

5.4.2.3 Multi-agent forecasting approaches

As the only MA forecasting approach, Morariu and Borangiu (2018) implemented multiple LSTMs to optimize production cost and subsequent scheduling according to pre-defined objectives. The LSTM networks for each resource operated based on a bidding mechanism, and a resource was subsequently assigned or not assigned to a job according to its bid or prediction of what a job would likely cost if produced with the resource.

5.4.3 Production control

Apart from planning and forecasting, production control in particular must be capable of coping with direct production complexities and solving optimization problems despite the inherent dynamics and non-linear interrelationships. Although production planning already tries to incorporate potential incidents and breakdowns on the job floor, production control must update schedules and direct production decisions in real time to keep processes stable and adjust decisions based on the current production state.

5.4.3.1 Plain NN control approaches

The semiconductor industry, as one of the fastest moving, was addressed with 5 publications to handle high cost pressures and complex processes. To circumvent missing methodological approaches, Kuhnle et al. (2021) implemented a TRPO-based RL approach to determine a dispatching agent's next move to minimize throughput and waiting time and maximize utilization rates. In wafer fabrication, Altenmüller et al. (2020) implemented an agent to choose the next operation destination with a shifting local to global reward function. While work-in-progress levels were optimized in both phases, the local utilization ratio was optimized in the first, followed by minimizing global time constraints in the second one.

Due to the limited capabilities of existing models to cope with dynamic system behavior, Bergmann and Stelzer (2011) and Bergmann et al. (2014) applied a control strategy approximation approach to increase the accuracy of system reproduction and minimize manual interventions. Luo (2020) adopted a double DQN RL to minimize total tardiness and avoid otherwise assumed

static conditions. The state-dependent selection of dispatching rules outperformed the respective individually applied rules. Following the same basic concept, Mouelhi-Chibani and Pierreval (2010) and Zhao and Zhang (2021) outperformed conventional approaches using NN-based rule selection depending on the flow- or job-shop system parameters. For this purpose, Zhao and Zhang (2021) employed a convolutional NN, which takes matrices of processing times, and two Boolean matrices of pending and completed operations as input to choose rules such as SPT and LPT, and outperformed a GA in terms of machine utilization, waiting times, etc. Similarly, in a job-store environment, deploying the production state representation as a 2-D matrix and a dispatching policy transfer, Zheng et al. (2020) not only proved strong performance but also increased generalizability using the transfer strategy.

Other publications listed in Table 5.14 considered, for example, short-term material flow control in a copper mining complex to reduce costly re-optimizations and avoid unsteady updates based on the quality and quantity of extracted materials Kumar et al. (2020), or implemented unit-cost minimization to mitigate the disadvantage of conventional methods' uncertain demands and long changeovers in a dishwasher wire-rack production system Wu et al. (2016).

5.4.3.2 Embedded NN control approaches

A pure NN-based approach for job allocation and operation sequence selection to minimize makespan and tardiness was proposed by Lang et al. (2020). Due to the generalization of the FFNN-based allocation and LSTM-based sequencing DQN RL agents, the prediction of new schedules was significantly faster. Another superior two-hierarchical DQN job-shop scheduling approach was implemented by Luo et al. (2021). The controller NN determined temporary goals for the lower DQN, which selected a dispatching rule depending on the indicated goal and production state. Goals were defined as different reward functions that aimed at optimizing a certain production indicator such as tardiness or machine utilization. A DQN as a hyper-heuristic to adjust parameters of a sequencing rule reduced mean tardiness up to 5% in Heger and Voss (2021). Kim et al. (2020) combined a NN with a heuristic to maximize machine utilization via supervised machine buffer selection and rule-based dispatching. An overview of the reviewed papers is given at the top of Table 5.15.

5.4.3.3 Multi-agent control approaches

To cope with the inherent dynamics in job-shop scheduling, Hammami et al. (2017) implemented an MA system based on simultaneous learning and inter-agent information exchange to reduce mean tardiness. Each resource was linked with a decisional agent that, to leverage decision making, involved a choice agent for NN selection. A central DQN module for training was used by Dittrich and Fohlmeister (2020) and Hofmann et al. (2020). In Dittrich and Fohlmeister

(2020), the central module is optimized based on the globally defined rewards and transferred to individual agents, which can request required local and global system information for decision making. Hofmann et al. (2020) provides agents with immediate rewards for selected actions and delayed rewards based on the total global cycle time achieved to increase the speed of learning. In comparison to a rule-based and a non-coordinated strategy, this strategy, which prevented the blocking of other agents and assigned global rewards, outperformed the previous strategies. Another training strategy was introduced by Waschneck et al. (2018) in a wafer fabrication job shop, which, for reasons of stability and learning speed, initially trained one NN at a time while the other work centers were controlled by heuristics. Subsequently, each work center was controlled by one NN respectively and the system was optimized cooperatively toward a maximum uptime utilization as a global goal. The same training strategy was applied to minimize cost in a car after-paint buffer control system (Gros et al., 2020). Gros et al. (2020) implemented one NN for each, inserting and discharging from the buffer, and the agents were trained with an iterative curriculum learning strategy in which only one agent was trained at a time to circumvent instabilities that arise from parallel training.

An order-bidding approach for dispatching was proposed by Malus et al. (2020) for 5 autonomous mobile robots with a common global reward to minimize tardiness. Based on the observed state, the agent that bids the most but does not handle more than 2 orders at the same time is assigned to the order. To decrease execution time and increase utilization efficiency, May et al. (2021) followed an economic bidding approach in which each participant in the production system should reach a maximum profit independently of other participants. Based on a deep RL PPO, the global utilization efficiency after part completion and locally accepted quotes, non-value-adding time, as well as consecutive failed quotations, could be optimized. Other MA production control papers, besides those introduced above, are listed at the bottom of Table 5.15.

5.4.4 General analysis

The general analysis is briefly summarized in Table 5.4. Out of the 129 reviewed papers, a total of 95% were implemented and validated in simulations. As a common outlook of the individual papers, the transfer to reality was mentioned as a further objective to incorporate other parameters and to be able to map complexity more accurately. Besides, with 89%, the high share of superior approaches is conspicuous, which does not contain similarly performing approaches. Especially the field of MA and embedded-based planning yielded impressive results and outperformed conventional approaches in all tests.

Moreover, algorithmic deductions can be drawn on the basis of the reviewed papers. A DQN or deep RL is mainly implemented in planning and control, regardless of the agent structure. The learning-by-doing behavior as well as the straightforward definition of rewards, likewise the

absence of necessity for an already existing set of data, constitute an advantage. In forecasting, NNs are primarily trained via BP algorithms, rather than with deep RL (one approach), and most employ an FFNN. Whereas 8 of all considered papers employed an LSTM architecture, 6 were employed in forecasting, thus profiting from their capability to map long-term dependencies. Nevertheless, the FFNN share (25) is decisively higher, similar to the other disciplines.

	Paper count	Simulation-only share	Superiority (#benchmarks)	Most freq. NN	Most freq. algorithm
Planning	55	95%	90% (41)	FFNN	DQN
Plain	27	100%	82% (22)	FFNN	DQN
Embedded	20	90%	100% (13)	FFNN	BP
Multi-agent	8	88%	100% (6)	FFNN	DQN
Forecasting	36	92%	95% (25)	FFNN	BP
Plain	23	91%	79% (14)	FFNN	BP
Embedded	12	92%	100% (11)	FFNN	BP
Multi-agent	1	100%	- (-)	RNN	Supervised
Control	38	100%	87% (30)	FFNN	DQN
Plain	23	100%	94% (16)	FFNN	DQN/TRPO
Embedded	7	100%	86% (7)	FFNN	DQN
Multi-agent	8	100%	71% (7)	FFNN	DQN
Total	129	95%	89% (96)	FFNN	DQN

Table 5.4 Key statistics from the review process

Referring to MA systems, most papers defined a global objective for agent-to-agent interaction (see Table 5.5). For this purpose, Pol et al. (2021) derived a reward factor based on the total makespan and multiplied it by the local rewards for each agent. Waschneck et al. (2018), on the other hand, considered the total sum of all due-date derivatives of all lots as a global minimizing objective. Another interaction type was the direct or active system information exchange between the agents, e.g. by information requests between machine and order agents in Dittrich and Fohlmeister (2020) or by partial activation of agents in Zhou et al. (2021), which subsequently provided real-time state information and became idle again when no scheduling task was pending. The provision of agent state information in a Boolean job-agent matrix was implemented in Liu et al. (2020). Such sharing of agent information was also referred to as indirect interaction (Pol et al., 2021) or sensing (Baer et al., 2020), indicating that agents must anticipate what the others might do next. A direct collaboration approach was also facilitated by bidding (as in Malus et al. (2020)) or negotiation mechanisms (as in Shin et al. (2012)).

Another analysis examined the training patterns of MA systems. 44% of the reviewed MA papers pursued a centralized learning approach, such as implementing a central intelligence as in Park et al. (2020) or Dittrich and Fohlmeister (2020), and executed it in a decentralized manner. Thus, aggregated experiences were leveraged through transfer learning or parameter sharing, making

the experience of individual agents available to others, which enabled an increased scalability (as in Lee et al. (2020)). Others, such as Morariu and Borangiu (2018) and Waschneck et al. (2018), adopted a decentralized or decentralized iterative training approach, respectively. While Morariu and Borangiu (2018) deployed LSTMs in parallel to learn machine cost patterns and generate bids, Waschneck et al. (2018) trained one agent first, while the others were controlled by heuristics before all were controlled by one DQN each. Additionally, reward designs were designed accordingly, such as in Pol et al. (2021). Where agents learned to meet local goals first in a decentralized manner, they were optimized to reach a global goal in a subsequent phase.

Interaction						Training		
Global objective	Agent exchange	Agent state information	Bidding mechanism	Market-based negotiation	None	Central	Decentral	n.a.
36%	18%	14%	9%	5%	18%	44%	37%	19%

Table 5.5 MA system interaction and training approaches

5.5 Taxonomy

We propose a taxonomy to classify the implementation of NN in production and general systems, following the taxonomy development method of Nickerson et al. (2013). Proceeding from the empirical-to-conceptual path, we first identified object subsets based on the employed NNs and agent structures through our review and then condensed them into a coherent framework. The clustering of these is integrated into Table 5.6 and describes the central dimensions for the taxonomy creation, independent of the specific production background.

Initially, the classification is driven by the assumption that a self-contained system is considered, which consists of clearly defined boundaries as well as input and output variables for the optimization of the problem. Segregated multi-production or factory systems that do not interact with each other are not included.


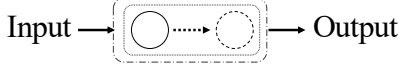
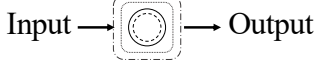
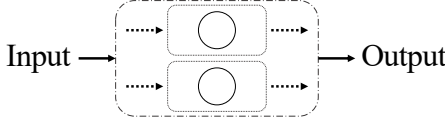
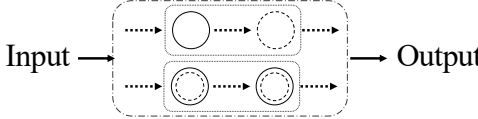
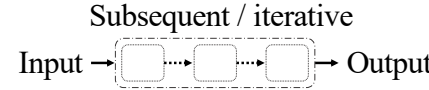
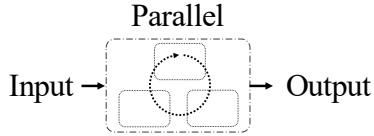
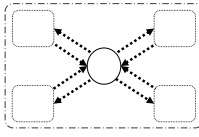
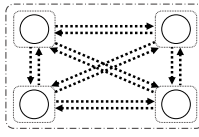
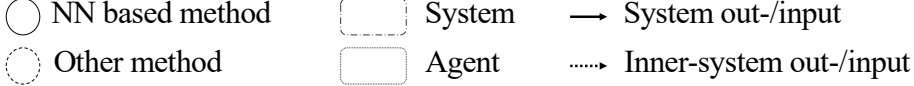
Methods Agents	= 1	> 1
Single-agent = 1	(1) Plain agent system 	(2) Embedded agent system; i.e.:  <hr/> 
Multi-agent > 1	(3) Multi-agent system 	(4) Multi embedded-agent system 
	Task completion: 	Parallel 
	Control or training model: 	Decentralized 
		

Table 5.6 Proposed taxonomy for single- and multi-agent system interaction

Starting with the top left and with one agent and method each (1), a classical optimization approach is described whose inputs provide parameters for optimization. In particular, its fast implementation and few inherent interdependencies reduce initial personnel and computational efforts. This enables prototypical use cases to be quickly evaluated for their benefits, and experience can be gathered to clarify necessary follow-up actions and potential serial deployment. In large systems, however, applying just one NN might imply low scalability and performance due to the curse of dimensionality (as in Bellman (1966)). The embedded approach (2) combines a NN-based optimization with other ML methods or heuristics. It describes an intrinsically structured approach that breaks down the overall optimization problem and complexity into sub-tasks. This approach can be carried out either in parallel or subsequently, e.g. by predetermining a baseline through an analytical model, to which the NN output is added to dynamically determine total product completion times (as in Huang et al. (2020)). Although the implementation is more extensive in terms of effort, the advantages of the respective methods can be exploited to leverage

the overall performance and cope with complex tasks. Additionally, available employee and system knowledge can be utilized to enable an optimal division of tasks and derive appropriate problem-solving strategies. While these two MA variants are more complex or costly than the others in terms of interaction design, system requirements, and computational effort, they are more suitable for large environments due to their improved scalability and straightforward adding of agents.

The bottom lines of Table 5.6 (cases 3/4) describe MA approaches within a confined system. Several plain or embedded agents of the upper line are combined and interact with each other. The agents act as independent autonomous entities according to the definition of Patel et al. (2001) and are provided with information from the same associated system. The respective system inputs can be distributed and received by an agent as a collective set but can also be specifically filtered and processed. Filtering and provisioning can be dependent on the linked entity (e.g., a machine), and can be independent of the overall system and global states if only local state variables are considered.

It is feasible to link several of the above forms of organization and interaction in a hybrid manner to benefit from the advantages of both. For instance, a plain or embedded agent in one subsystem can keep performing a specific task in some expert role without interaction. Neighboring subsystems, on the other hand, can be designed as multi-agent systems. As such, the system could exploit its strength in leveraging its group dynamics and take on logistical tasks where the agents act as autonomous planners bidding on transportation orders.

In addition to the above differentiation, a distinction can be made between parallel and iterative task completion and the applied control or training model. In parallel completion, agents are able to work concurrently on jobs of the same category, and each agent, such as a logistics robot, can be assigned to each job. In subsequent or iterative completion, agents can differ in their capabilities and thus influence process chains. Rather, a set of jobs is not allocated to different agents, for example, to increase the throughput in logistics with each additional agent, but segments of the process chain are distributed to the appropriate agents.

In the centralized control or training of agents (lower left row), the agent shares input data with a central intelligence instead of processing it individually. This can be facilitated by deploying a central NN that learns through the experiences of each individual agent, rather than having an independent NN for each agent. An intermediate step may be embodied by a parameter sharing strategy that collects and shares relevant experiences after specific time intervals. Compared to the other methods, initial efforts and adjustments for MA training and interaction between decentralized agents are significantly higher during training and control optimization. Nevertheless, this approach provides a high scalability and agents can be easily added and benefit from all experiences without the need for costly re-training. If the agent

is trained in a decentralized manner, it can better adapt to its specific role in the respective subsystem and act as a kind of dedicated expert. Thus, if a system consists of procedurally independent interacting possessors, the NN should not be shared in order to allow the specific shaping and development of unique skills to maximize the system performance. In a parameter-sharing or completely centralized strategy, on the other hand, agents with the same or similar tasks benefit from all experiences and thereby optimize global performances. However, this could suppress specific skills due to a progressive standardization of agent behavior.

As indicated in Table 5.5, the interaction between agents can adopt different forms. Based on this, Table 5.7 summarizes the possibilities related to the specific type of inter-agent interaction and exchanged information. The interaction between agents can be described as direct (agents directly interact with each other) or indirect (no direct data transfer). In addition, depending on the exchanged or mediated good, the type of exchange is considered in terms of the (processed) state information of an agent (such as the workload), or relevance criteria. The special case of sensing, i.e. no exchange at all, is not covered.

A direct interaction based on state information can be considered a direct form of communication. An indirect interaction, on the other hand, is not based on any direct information exchange but, for example, on a global goal or the sharing of global state information. The agent does not receive information from other agents but from the system as such. One step further, agents can exchange already processed information and negotiate with each other in a direct manner (Table 5.7, bottom left) or place bids that are not communicated directly with each other, but are submitted to an independent entity such as a machine or an order itself. In this context, the prior evaluation of an order in terms of the pre-negotiation measure or bid level is interpreted as an indicator of the order's relevance to an agent.

The exact interaction that should be chosen for a specific application depends on the exact task and environment. For a fast use case creation, but also the indirect communication of global information, a global objective can optimize the system as a whole. The direct communication of state information would rather serve the local optimization and only consider the closer environment. Advanced mechanisms for the processing of relevant information facilitate the joint processing of several agents' observations and impressions. In this case, it is not the individual agent that decides whether or not to do something, but rather other agents are involved in the decision-making process. Thereby, processes can be designed in a more interactive and balanced way to profit from group dynamics. Nevertheless, negotiation and bidding are more complex in their implementation and require a thoroughly elaborated design. It is further possible to combine the presented types of interaction. For instance, an agent can pursue a global objective based on a DRL, but still be in contact with other agents via negotiation.

		Type of agent interaction	
		Direct	Indirect
Exchange of ...	State information	Direct communication	Global objective, receive global states
	Relevance information	Market-based negotiation	Bidding mechanism

Table 5.7 Types of interaction in multi-agent systems

5.6 Implementation challenges and research agenda

In the previous section, the broad application base, embedding variants, and benefits of NN-based PPC were highlighted and properties were defined in a taxonomy. However, there are still some challenges that prevent widespread adoption and real-world deployment (RQ3) and that need to be addressed in future research (RQ4).

5.6.1 Implementation challenges

During the review, we identified some challenges and categorized them into the following subgroups, which are further reflected upon afterward by identifying respective research gaps.

- **Transferability:** Many of the above-mentioned papers examined the implemented approaches within a pre-defined simulation scope. The extent to which these are structurally rigid and require NN adjustments in the case of modified scenarios was hardly considered. Even though approaches like Baer et al. (2020) attempted to identify fundamental and transferable relationships in scheduling, small changes in scenarios can cause decreasing performances and demand large efforts of reconfiguration and retraining as well as deep process insights. Also, Lang et al. (2020) pointed out that in DQN-based scheduling, i.e. if a new machine or buffer location is added, it cannot be mapped directly by the prevailing NN structure that limits adaptability and reliability in dynamic processes, especially of plain agent approaches.
- **MA training and interaction:** Additional complexities in MA environments require deep consideration during implementation and increased evaluation of the learning behavior of each agent. Concurrent learning might lead to instabilities during training and cause the mutual dynamic and non-stationary behavior of the agents to negatively affect the individual, as mentioned in Malus et al. (2020). To avoid instabilities, Gros et al. (2020) and Waschneck et al. (2018) chose an iterative approach, which must be optimally adjusted in terms of frequency and transition to pure NN-based operation. To facilitate synergy effects between the agents and guarantee mutual optimization, it should further be clarified which form of interaction is selected depending on the specific scenario. Although an

indirect communication in Baer et al. (2020) proved to be functional without direct agent interaction, other papers integrated global states and rewards. However, advanced negotiation and bidding mechanisms were scarce and, in summary, represent an additional complexity dimension in addition to finding an appropriate training strategy, algorithm, and NN parameters, which potentially impede implementation efforts.

- **Handling (real) production complexity:** A total of 123 papers (or 95%) were implemented and evaluated solely in simulations. Although simulations are becoming more accurate due to the inclusion of failures, noise, etc., they do not capture the full complexity of a real system with its non-linear dependencies, human intervening factors, etc. Therefore, the results cannot directly be transferred to reality and no general conclusions can be drawn about the reliability and sustainability of the results in real environments where additional influences would affect the system and agent. Such effects can lead to unstable learning (Gros et al., 2020) or vibration during training (Shi et al., 2020). Particularly in the field of production control, no approach was implemented due to the high implementation and security efforts required in real operations.
- **Limited diversification of NNs and algorithms:** In summary, 80% of the papers employed an FFNN, which in most cases outperformed conventional approaches. Nevertheless, leveraged performances could be reached through the deployment of more advanced networks such as LSTMs for capturing long-term relationships or convolutional networks, i.e. for processing production state matrices. Furthermore, 47% of the papers employed a BP or DQN RL, both of which are basic algorithms in machine learning. In the case of the DQN, however, it was often inferior to a DoubleDQN, which was employed in only 2% of the papers, even though it exhibited outstanding performance in Hasselt et al. (2016).
- **Manual parameter optimization:** In addition to the algorithm and NN adaptation to the framework conditions, the search for fine-grained parameters represents a central hurdle during implementation (Rummukainen and Nurminen, 2019) and has a tremendous impact on the final performance (Bergmann et al., 2014; Zhou et al., 2017). In particular, Wu et al. (2020) visually demonstrated the effects of the optimizer setting, number of NN layers, and neurons, and their impact on performance. Still, there are no common guidelines or rules for setting NN parameters that must be set by hand, thereby enforcing a black-box model character. Consequently, parameter tuning not only consumes a lot of time and causes significant computational efforts, but also requires expert knowledge in parameter fitting, which is not always available among practitioners.

5.6.2 Future research agenda

Although the aforementioned challenges still prevent seamless real-world and large-scale applications, they did reveal some opportunities for further research during the course of the review. These are summarized in the following bullet points.

- **Scalability:** Most of the reviewed papers already revealed the capabilities of NN-based solutions in PPC and forecasting. However, a stronger focus on MA systems could help to cope with large-scale production environments, as indicated in Waschneck et al. (2018). The system would not have to rely on only a single NN as in a plain agent optimization, but could distribute the production complexity and data streams accordingly as introduced by Wang et al. (2022) in a resource preemption environment, or by Kim et al. (2020) in dynamic resource scheduling by deploying job weights and multi-agent sociability aspects. Assuming that machines or others are added to a production line, agents would not necessarily need to be retrained, but could instead rely on the same logic. Potentially resulting scale effects, the necessity of altered global objectives, and the question of what purposeful data provisioning should look like all still need to be investigated in further research. Also, novel dynamic and hybrid control approaches such as the non-NN-based hybrid hierarchical predictive and heterarchical reactive architecture, as in Pach et al. (2014), can lead to increased scalability due to the combination of respective organizational benefits.
- **Design of NN-based MA systems:** As a considered sub-area, especially research on MA systems is not yet exhausted. Previously mentioned as a hurdle, there are still a lot of potentials, especially in communication design, stable and reliable training methods, and the definition of guidelines for developing MA systems. The extent to which a centralized intelligence and parameter-sharing strategies are advantageous, or whether fully decentralized and co-learning swarm intelligence strategies should rather be applied, are conceptual questions that need to be clarified. The same accounts for the choice of interaction, such as collaborative, competitive, or hybrid approaches, and how bidding or negotiation mechanisms must be designed to enhance performance, adaptability, and resilience. In addition, the agent's interaction behavior in new environments, how the adaptability of a collective set differs from that of a single agent, and how interaction approaches can be exploited to maintain production stability are further research directions that could accelerate a broader implementation of MA systems.
- **Simplification through embedded approaches:** Increasingly large state spaces and, in general, the emergence of Big Data coupled with ever larger data streams caused by a growing amount of sensor data and system interdependencies lead to less-manageable problem spaces. The decomposition of tasks into sub-tasks, plain and embedded approaches can

better cope with and help to contribute to increasing algorithm performances. Further research could focus on how holistic NN-based approaches can be enabled and optimally deployed through sequential task sharing or parallelization of tasks. Parallelization can be problem-centric, but also location-, strategy-, or scenario-centric, such as the bottleneck and non-bottleneck flow time forecast in Schneckenreither et al. (2021), depending on the specific complexity allocation.

Another non-NN-based example was presented in Minguillon and Lanza (2019) by combining centralized and decentralized scheduling properties for the adjustment of degrees of freedom. As mentioned in Schwung et al. (2021), a NN-based system can also initially learn from established methods before applying them individually. This allows already recognized system knowledge to be transferred and expert knowledge to be leveraged in future applications. A collaborative application to mitigate exceptional events with the help of human operators might be explored in more detail and can be initially realised in learning factory environments, as discussed by Teichmann et al. (2021).

- **Generalizability:** The flexibility and adaptability of an approach to quickly fit to new environments could be deepened. This would not only mitigate exceptional situations such as machine breakdowns or large-scale events such as the Corona pandemic, but also increase system sustainability through significantly increased resource utilization and longer service lives, since not only would fewer machines or robots be needed, but also necessary manual and technological adjustments that cause constant effort would be minimized. Further research on how basic task patterns can be learned, as in Baer et al. (2020), or the implementation of a central intelligence that prevents local skill generation and exploitation would leverage generalizability. Such over-adaptation could be circumvented by adequate exploration of the broader problem space or increased context awareness to adapt more quickly and robustly to new environments and scenarios, as already done in computer vision (Athanasopoulou et al., 2020) or nuclear mass training (Zhao and Zhang, 2022). To achieve this within the relevant scope, Zang et al. (2019) developed a hybrid approach with a prior problem classification before being solved by the NN scheduler. Further investigation of NN-based optimization and adaption of advanced analytics or heuristics could exploit both methods' advantages. Once analytically accurate but static knowledge is available, the NN model can be added as a dynamic and adaptive component to generalize process knowledge. With a high accuracy and adaptability, appropriate trade-offs between conventional and ML-based optimization could be facilitated. Thus, by circumventing the vanishing applicability of simulations and hard-coded algorithms, generalizability could be optimized.
- **Simulation to reality transfer:** To take further steps toward the implementation of real

applications, simulations could be designed more realistically. By integrating dynamics and non-linear parameters, implemented approaches can already be evaluated for robustness at an early stage. A further step toward reality could be accelerated by hybrid hardware-in-the-loop (HiL) environments, in which real elements like control units are installed and the rest of the environment is simulated. Likewise indicated by Jones (2021), it is worthwhile to advance the approaches to higher cognitive levels in order to circumvent existing limitations of prevailing machine learning approaches and not only build a sophisticated digital twin, but benefit from the strong artificial intelligence paradigm. Also, small-scale implementations such as in Zhou et al. (2021) can help to collect initial insights before transferring the applied methods to larger scales. At this level, further tests can be carried out, and reliability as well as safety factors can be evaluated. Especially in forecasting, approaches can be pre-tested in parallel to already proven methods and assist, i.e., by conducting what-if analyses in Huang et al. (2016) for decision support.

5.7 Discussion

Today's PPC, as well as forecasting, must increasingly cope with dynamic processes, fast-paced product cycles, and sharp fluctuations in demand. To ensure robust and adaptive production, NNs have been increasingly deployed in recent years since they can process large amounts of data in real time and provide great flexibility. Although the potentials of NNs as an optimization tool have already been indicated in other reviews, a specific review of NN-based PPC was still missing until now. Based on a taxonomy framework, we retrieved 120 papers and subdivided them according to their PPC application, agent configuration, applied NN and algorithm, pursued objective, benchmark results, and other category-dependent criteria, such as interaction in MA systems.

Although 95% of the reviewed papers were assessed in simulations, we could identify a broad application range and superior performance in 89% of benchmarks. NN-based approaches demonstrated their ability to cope with external interference and unpredictable events while maintaining robust production and optimizing a variety of performance indicators. This not only reduced lead times, costs, and manual effort, but also increased overall flexibility and adaptability. Additionally, based on the review results, a taxonomy was defined which enables the classification of the implemented NN approach based on the agent and method count. In this regard, implementations are categorized into plain, embedded, and multi-(embedded) agent systems, which differ particularly in terms of scalability, implementation effort, and prevailing task breakdown.

5.7.1 Managerial implications

Companies must be able to generate profits and meet customer expectations despite the challenging market conditions and increasingly complex production processes. To counteract the disadvantages of conventional methods such as high manual effort, companies should leverage the increasingly available machine and process data to enable data-driven analysis and optimization. This review is intended to demonstrate the potential of NN-based PPC to increase production efficiencies and minimize process risks.

The review revealed the practical relevance and superior performance of NN-based PPC, which not only saved costs and increased production throughputs but also optimized production flexibility and robustness. The reviewed papers and defined taxonomy can serve managers as guidance for the identification and prototypical design of company-specific implementations. A plain approach, with minimized trade-offs, can help with rapid integration, whereas embedded and multi-agent approaches can solve more complex and larger-scale problems, but also entail higher implementation effort and development complexity. Through the integration of NNs in PPC and forecasting, dependency on human experience can be reduced, and data-driven production optimization, as well as real-time process adaptation, can be facilitated.

5.7.2 Limitations

Although the review is based on a fundamental methodology for conducting the review, as well as for creating the taxonomy, existing limitations should be mentioned. First, the review originates from iteratively defined keywords, which were optimized in the course of the review. Also, the retrieved database was supplemented by a forward and backward search. Yet, despite our best endeavors, some papers may not have been identified. Further, some supplementary articles may not have been included by the databases, although Scopus, WoS, and IEEE Xplore should cover the most accessible articles. Lastly, we integrated proceedings and conference papers in addition to journal articles to obtain a comprehensive literature set, which, however, may cause bias to similar reviews.

5.8 Conclusion

This review intends to provide an outline of existing NN approaches in PPC and forecasting and establishes a taxonomy to classify the implementations based on the number of employed agents and intrinsically combined methods. The broad application base and superior performance of the approaches were highlighted in a variety of different scenarios (RQ1). A multitude of process and economic parameters could be improved, and process accuracy and flexibility were optimized. Drawbacks of conventional methods, such as costly re-training or high dependency

on human experience, were thereby significantly reduced.

The different types of embeddings (RQ2) were incorporated into the basic review structure and the developed taxonomy framework. Whereas most papers employed one NN for plain optimization, particularly since 2018 a significant increase can be observed in intrinsically embedded approaches that combine multiple methods, including non-NN-based ones, and MA approaches that split the task among multiple agents through complexity partitioning and appropriate communication.

Although the combined benefits of the respective methods in embedded approaches and the scalability and robustness of the MA approaches became apparent, the lack of guidelines still poses a major challenge (RQ3) that leads to sophisticated design processes and manual efforts in framework and parameter selection, as well as extensive procedures for training and interaction design. In addition, only a limited number of different algorithms and NN types were deployed and trials were primarily conducted in simulations.

Future research (RQ4) could focus on optimizing the generalizability and transferability of trained agents with limited additional effort, e.g. through non-specific scenario training and learning general tasks patterns, as well as adopting a broader range of algorithms and NNs. To further mitigate the gap to real-world testing, simulations can be designed more realistically by incorporating additional input and disturbance parameters and deploying hybrid environments.

Advancing embedded and collaborative MA approaches can contribute to the ability to cope with the ever-increasing process complexity and significantly optimize production efficiency. Although few approaches have been tested in reality, NN-based PPC provides an opportunity to create robust and sustainable production processes and has already demonstrated its superior capabilities. Further research and a shift to large-scale and hybrid environments can further drive NN-based PPC solutions in manufacturing in order to benefit from simultaneous global and local optimization opportunities in times of on-going automation and an increasing importance of data-driven decisions in the sense of Big Data.

Copyright Notice

This is an accepted version of the article published in:

Panzer, M., B. Bender and N. Gronau (2022). Neural agent-based production planning and control: An architectural review. *Journal of Manufacturing Systems* 65, p. 743-766.

<https://doi.org/10.1016/j.jmsy.2022.10.019>

Clarification of the copyright adjusted according to the guidelines of the publisher.

Contributor roles

This paper is the result of collaborative efforts where specific responsibilities were allocated to ensure the effective completion of the research and the preparation of the manuscript:

- **Marcel Panzer:** Played a pivotal role in the majority of this publication's aspects. Responsibilities included conceptualizing the research, designing and implementing the literature analysis methodology, conducting the review, synthesizing and analyzing findings, and primarily drafting the manuscript. Additionally, contributions were made in compiling and refining the final manuscript during the review process.
- **Norbert Gronau and Benedict Bender:** Both were instrumental in the progression of this review, providing valuable guidance. Their involvement included thorough evaluations and the provision of insightful feedback and suggestions. Such contributions were crucial in enhancing the quality of the publication and maintaining its overall integrity.

The *Declaration of the Co-Authors* is inserted at the end of this thesis.

Publication 2 - References

- Aggarwal, C. C. (2018). *Neural networks and deep learning: a textbook*. Cham, Switzerland: Springer.
- Altenmüller, T., T. Stüker, B. Waschneck, A. Kuhnle and G. Lanza (2020). Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints. *Production Engineering* 14(3), p. 319–328. doi: 10.1007/s11740-020-00967-8.
- Arunraj, N. S. and D. Ahrens (2015). A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics* 170, p. 321–335. doi: 10.1016/j.ijpe.2015.09.039.
- Athanasopoulou, L., A. Papacharalampopoulos and P. Stavropoulos (2020). Context awareness system in the use phase of a smart mobility platform: A vision system for a light-weight approach. *Procedia CIRP*. doi: 10.1002/9780470754887.
- Azab, E., M. Nafea, L. A. Shihata and M. Mashaly (2021). A Machine-Learning-Assisted Simulation Approach for Incorporating Predictive Maintenance in Dynamic Flow-Shop Scheduling. *Applied Sciences* 11(24), p. 11725. doi: 10.3390/app112411725.
- Babiceanu, R. F. and F. F. Chen (2006). Development and Applications of Holonic Manufacturing Systems: A Survey. *Journal of Intelligent Manufacturing* 17(1), p. 111–131. doi: 10.1007/s10845-005-5516-y.
- Baer, S., J. Bakakeu, R. Meyes and T. Meisen (2019). Multi-Agent Reinforcement Learning for Job Shop Scheduling in Flexible Manufacturing Systems. In: *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*, Laguna Hills, CA, USA, p. 22–25. IEEE.
- Baer, S., D. Turner, P. K. Mohanty, V. Samsonov, R. Bakakeu and T. Meisen (2020). Multi Agent Deep Q-Network Approach for Online Job Shop Scheduling in Flexible Manufacturing. In: *International Conference on Manufacturing System and Multiple Machines*, Tokyo, Japan, p. 1–9.
- Baeza Serrato, R. (2018). Stochastic plans in SMEs: A novel multidimensional fuzzy logic system (mFLS) approach. *Ingeniería e Investigación* 38(2), p. 70–78. doi: 10.15446/ing.investig.v38n2.65357.
- Baker, A. D. (1998). A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull. *Journal of Manufacturing Systems* 17(4), p. 297–320. doi: 10.1016/S0278-6125(98)80077-0.
- Beigi, A. and N. Mozayani (2016). Dialogue strategy for horizontal communication in MAS

- organization. *Computational and Mathematical Organization Theory* 22(2), p. 161–183. doi: 10.1007/s10588-015-9201-1.
- Bellman, R. (1966). Dynamic programming. *Science* 153(3731), p. 34–37.
- Bergmann, S. and S. Stelzer (2011). Approximation of Dispatching Rules in Manufacturing Control Using Artificial Neural Networks. In: *2011 IEEE Workshop on Principles of Advanced and Distributed Simulation*, Nice, France, p. 1–8. IEEE.
- Bergmann, S., S. Stelzer and S. Strassburger (2014). On the use of artificial neural networks in simulation-based manufacturing control. *Journal of Simulation* 8(1), p. 76–90. doi: 10.1057/jos.2013.6.
- Bertolini, M., D. Mezzogori, M. Neroni and F. Zammori (2021). Machine Learning for industrial applications: A comprehensive literature review. *Expert Systems with Applications* 175, p. 114820. doi: 10.1016/j.eswa.2021.114820.
- Bitran, G. R., E. A. Haas and A. C. Hax (1982). Hierarchical Production Planning: A Two-Stage System. *Operations Research* 30(2), p. 232–251. doi: 10.1287/opre.30.2.232.
- Bongaerts, L., L. Monostori, D. McFarlane and B. Kádár (2000). Hierarchy in distributed shop floor control. *Computers in Industry* 43(2), p. 123–137. doi: 10.1016/S0166-3615(00)00062-2.
- Both, C. and R. Dimitrakopoulos (2021). Applied Machine Learning for Geometallurgical Throughput Prediction—A Case Study Using Production Data at the Tropicana Gold Mining Complex. *Minerals* 11(11), p. 1257. doi: 10.3390/min11111257.
- Brocke, J., A. Simons, B. Niehaves, K. Riemer, R. Plattfaut and A. Cleven (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process. *Proceedings of the 17th European Conference on Information Systems (ECIS)*.
- Bueno, A., M. Godinho Filho and A. G. Frank (2020). Smart production planning and control in the Industry 4.0 context: A systematic literature review. *Computers & Industrial Engineering* 149, p. 106774. doi: 10.1016/j.cie.2020.106774.
- Burduk, A., T. Chlebus and R. Waszkowski (2018). Assessment of the Feasibility of a Production Plan with the Use of an Artificial Neural Network Model. In: *Intelligent Systems in Production Engineering and Maintenance – ISPEM 2017*, Volume 637, p. 179–188. Cham: Springer International Publishing. doi: 10.1007/978-3-319-64465-3_18.
- Cadavid, J. P. U., S. Lamouri, B. Grabot and A. Fortin (2019). Machine Learning in Production Planning and Control: A Review of Empirical Literature. *IFAC-PapersOnLine* 52(13), p. 385–390. doi: 10.1016/j.ifacol.2019.11.155.
- Cao, H. and X. Ji (2021). Prediction of Garment Production Cycle Time Based on a Neural Network. *Fibres & Textiles in Eastern Europe* 29, p. 8–12. doi: 10.5604/01.3001.0014.5036.

- Chakravorty, S. and N. N. Nagarur (2020). An Artificial Neural Network Based Algorithm For Real Time Dispatching Decisions. In: *2020 31st Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, Saratoga Springs, NY, USA, p. 1–5.
- Chen, B., J. Wan, Y. Lan, M. Imran, D. Li and N. Guizani (2019). Improving Cognitive Ability of Edge Intelligent IIoT through Machine Learning. *IEEE Network* 33(5), p. 61–67. doi: 10.1109/MNET.001.1800505.
- Chen, T. (2016). Embedding a back propagation network into fuzzy c-means for estimating job cycle time: wafer fabrication as an example. *Journal of Ambient Intelligence and Humanized Computing* 7(6), p. 789–800. doi: 10.1007/s12652-015-0336-1.
- Cooper, H. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society* 1, p. 104–126.
- Csáji, B. C., L. Monostori and B. Kádár (2006). Reinforcement learning in a distributed market-based production control system. *Advanced Engineering Informatics* 20(3), p. 279–288. doi: 10.1016/j.aei.2006.01.001.
- Danishvar, M., S. Danishvar, E. Katsou, S. A. Mansouri and A. Mousavi (2021). Energy-Aware Flowshop Scheduling: A Case for AI-Driven Sustainable Manufacturing. *IEEE Access* 9, p. 141678–141692. doi: 10.1109/ACCESS.2021.3120126.
- De Modesti, P. H., E. Carvalhar Fernandes and M. Borsato (2020). Production Planning and Scheduling Using Machine Learning and Data Science Processes. In: K. Säfsten and F. Elgh (Hrsg.), *Advances in Transdisciplinary Engineering*. IOS Press. doi: 10.3233/ATDE200153.
- Derigent, W., O. Cardin and D. Trentesaux (2021). Industry 4.0: contributions of holonic manufacturing control architectures and future challenges. *Journal of Intelligent Manufacturing* 32(7), p. 1797–1818. doi: 10.1007/s10845-020-01532-x.
- Ding, Y., L. Ma, J. Ma, M. Suo, L. Tao, Y. Cheng and C. Lu (2019). Intelligent fault diagnosis for rotating machinery using deep Q-network based health state classification: A deep reinforcement learning approach. *Advanced Engineering Informatics* 42, p. 100977. doi: 10.1016/j.aei.2019.100977.
- Dittrich, M.-A. and S. Fohlmeister (2020). Cooperative multi-agent system for production control using reinforcement learning. *CIRP Annals* 69(1), p. 389–392. doi: 10.1016/j.cirp.2020.04.005.
- Dong, T., F. Xue, C. Xiao and J. Li (2020). Task scheduling based on deep reinforcement learning in a cloud manufacturing environment. *Concurrency and Computation: Practice and Experience* 32(11). doi: 10.1002/cpe.5654.
- Dorri, A., S. S. Kanhere and R. Jurdak (2018). Multi-Agent Systems: A Survey. *IEEE Access* 6,

- p. 28573–28593. doi: 10.1109/ACCESS.2018.2831228.
- Fang, W., Y. Guo, W. Liao, K. Ramani and S. Huang (2020). Big data driven jobs remaining time prediction in discrete manufacturing system: a deep learning-based approach. *International Journal of Production Research* 58(9), p. 2751–2766. doi: 10.1080/00207543.2019.1602744.
- Feng, X. and G. Yuan (2011). Optimizing Two-Stage Fuzzy Multi-Product Multi-Period Production Planning Problem. *International Journal on Information* 6(6), p. 1879–1893.
- Fnaiech, N., H. Hammami, A. Yahyaoui, C. Varnier, F. Fnaiech and N. Zerhouni (2012). New Hopfield Neural Network for joint Job Shop Scheduling of production and maintenance. In: *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, Canada, p. 5535–5541. IEEE.
- Gahm, C., A. Uzunoglu, S. Wahl, C. Ganschietz and A. Tuma (2022). Applying machine learning for the anticipation of complex nesting solutions in hierarchical production planning. *European Journal of Operational Research* 296(3), p. 819–836. doi: 10.1016/j.ejor.2021.04.006.
- Gallina, V., L. Lingitz, J. Breitschopf, E. Zudor and W. Sihm (2021). Work in Progress Level Prediction with Long Short-Term Memory Recurrent Neural Network. *Procedia Manufacturing* 54, p. 136–141. doi: 10.1016/j.promfg.2021.07.047.
- Gannouni, A., V. Samsonov, M. Behery, T. Meisen and G. Lakemeyer (2020). Neural Combinatorial Optimization for Production Scheduling with Sequence-Dependent Setup Waste. In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Toronto, ON, Canada, p. 2640–2647. IEEE. doi: 10.1109/SMC42975.2020.9282869.
- Garetti, M. and M. Taisch (1999). Neural networks in production planning and control. *Production Planning & Control* 10(4), p. 324–339. doi: 10.1080/095372899233082.
- Gelders, L. F. and L. N. Van Wassenhove (1981). Production planning: a review. *European Journal of Operational Research* 7(2), p. 101–110. doi: 10.1016/0377-2217(81)90271-X.
- Glavan, M., D. Gradišar, S. Strmčnik and G. Mušič (2013). Production modelling for holistic production control. *Simulation Modelling Practice and Theory* 30, p. 1–20. doi: 10.1016/j.simpat.2012.07.010.
- Gronauer, S. and K. Diepold (2021). Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*. doi: 10.1007/s10462-021-09996-w.
- Gros, T. P., J. Gros and V. Wolf (2020). Real-Time Decision Making for a Car Manufacturing Process Using Deep Reinforcement Learning. In: *2020 Winter Simulation Conference (WSC)*, Orlando, FL, USA, p. 3032–3044. IEEE. doi: 10.1109/WSC48552.2020.9383884.
- Groth, M., P. Freier and M. Schumann (2021). Using self-play within deep Q learning to improve real-time production scheduling. In: *27th Annual Americas Conference on Information*

- Systems (AMCIS 2021)*, Montreal, Canada.
- Göppert, A., L. Mohring and R. H. Schmitt (2021). Predicting performance indicators with ANNs for AI-based online scheduling in dynamically interconnected assembly systems. *Production Engineering* 15(5), p. 619–633. doi: 10.1007/s11740-021-01057-z.
- Hammami, Z., W. Mouelhi and L. Ben Said (2017). On-line self-adaptive framework for tailoring a neural-agent learning model addressing dynamic real-time scheduling problems. *Journal of Manufacturing Systems* 45, p. 97–108. doi: 10.1016/j.jmsy.2017.08.003.
- Hammami, Z., W. Mouelhi and L. B. Said (2015). A Self Adaptive Neural Agent Based Decision Support System for Solving Dynamic Real Time Scheduling Problems. In: *2015 10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Taipei, Taiwan, p. 494–501. IEEE. doi: 10.1109/ISKE.2015.79.
- Han, B.-A. and J.-J. Yang (2020). Research on Adaptive Job Shop Scheduling Problems Based on Dueling Double DQN. *IEEE Access* 8, p. 186474–186495. doi: 10.1109/ACCESS.2020.3029868.
- Han, Z., Q. Zhang, Y. Jiang and B. Duan (2019). Research on the Production Scheduling Method of a Semiconductor Packaging Test Based With the Clustering Method:. *International Journal of Information Systems and Supply Chain Management* 12(2), p. 36–56. doi: 10.4018/IJISSCM.2019040103.
- Hasselt, H. v., A. Guez and D. Silver (2016). Deep Reinforcement Learning with Double Q-Learning. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16*, p. 2094–2100. AAAI Press.
- Heger, J. and T. Voss (2021). Dynamically adjusting the k -values of the ATCS rule in a flexible flow shop scenario with reinforcement learning. *International Journal of Production Research*, p. 1–15. doi: 10.1080/00207543.2021.1943762.
- Herrera, M., M. Pérez-Hernández, A. Kumar Parlikad and J. Izquierdo (2020). Multi-Agent Systems and Complex Networks: Review and Applications in Systems Engineering. *Processes* 8(3), p. 312. doi: 10.3390/pr8030312.
- Hoen, P. J., K. Tuyls, L. Panait, S. Luke and J. A. La Poutré (2006). An Overview of Cooperative and Competitive Multiagent Learning. In: K. Tuyls, P. J. Hoen, K. Verbeeck, and S. Sen (Hrsg.), *Learning and Adaption in Multi-Agent Systems*, Volume 3898, p. 1–46. Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/11691839_1.
- Hofmann, C., C. Krahe, N. Stricker and G. Lanza (2020). Autonomous production control for matrix production based on deep Q-learning. *Procedia CIRP* 88, p. 25–30. doi: 10.1016/j.procir.2020.05.005.

- Hu, L., Z. Liu, W. Hu, Y. Wang, J. Tan and F. Wu (2020). Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. *Journal of Manufacturing Systems* 55, p. 1–14. doi: 10.1016/j.jmsy.2020.02.004.
- Hu, S. and L. Zhou (2020). Prediction of order completion time based on the BP neural network optimized by GASA. In: *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Harbin, China, p. 1111–1114. IEEE.
- Huang, J., Q. Chang and J. Arinez (2020). Product Completion Time Prediction Using A Hybrid Approach Combining Deep Learning and System Model. *Journal of Manufacturing Systems* 57, p. 311–322. doi: 10.1016/j.jmsy.2020.10.006.
- Huang, P. B., J. C. Chen, C.-W. Yu and C.-C. Chen (2016). Integration of simulation and neural network in forecasting the throughput for TFT-LCD colour filter fabs. *International Journal of Computer Integrated Manufacturing* 29(3), p. 298–308. doi: 10.1080/0951192X.2015.1032356.
- Huang, S., Y. Guo, D. Liu, S. Zha and W. Fang (2019). A Two-Stage Transfer Learning-Based Deep Learning Approach for Production Progress Prediction in IoT-Enabled Manufacturing. *IEEE Internet of Things Journal* 6(6), p. 10627–10638. doi: 10.1109/JIOT.2019.2940131.
- Hubbs, C. D., C. Li, N. V. Sahinidis, I. E. Grossmann and J. M. Wassick (2020). A deep reinforcement learning approach for chemical production scheduling. *Computers & Chemical Engineering* 141, p. 106982. doi: 10.1016/j.compchemeng.2020.106982.
- Jain, S., D. Lechevalier and A. Narayanan (2017). Towards smart manufacturing with virtual factory and data analytics. In: *2017 Winter Simulation Conference (WSC)*, Las Vegas, NV, p. 3018–3029. IEEE.
- Jones, D. (2021). Artificial cognitive systems: the next generation of the digital twin. An opinion. *Digital Twin* 1, p. 3. doi: 10.12688/digitaltwin.17440.2.
- Kagermann, H., W. Wahlster and J. Helbig (2013). *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0 – Securing the Future of German Manufacturing Industry*. Acatech - National Academy of Science and Engineering.
- Kang, Z., C. Catal and B. Tekinerdogan (2020). Machine learning applications in production lines: A systematic literature review. *Computers & Industrial Engineering* 149, p. 106773. doi: 10.1016/j.cie.2020.106773.
- Karaoglan, A. D. and O. Karademir (2017). Flow time and product cost estimation by using an artificial neural network (ANN): A case study for transformer orders. *The Engineering Economist* 62(3), p. 272–292. doi: 10.1080/0013791X.2016.1185808.
- Kardos, C., C. Laflamme, V. Gallina and W. Sihn (2021). Dynamic scheduling in a job-

- shop production system with reinforcement learning. *Procedia CIRP* 97, p. 104–109. doi: 10.1016/j.procir.2020.05.210.
- Kim, H., D.-E. Lim and S. Lee (2020). Deep Learning-Based Dynamic Scheduling for Semiconductor Manufacturing With High Uncertainty of Automated Material Handling System Capability. *IEEE Transactions on Semiconductor Manufacturing* 33(1), p. 13–22. doi: 10.1109/TSM.2020.2965293.
- Kim, Y., S. Lee, J. Son, H. Bae and B. D. Chung (2020). Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system. *Journal of Manufacturing Systems* 57, p. 440–450. doi: 10.1016/j.jmsy.2020.11.004.
- Kramer, K. J., C. Wagner and M. Schmidt (2020). Machine Learning-Supported Planning of Lead Times in Job Shop Manufacturing. In: B. Lalic, V. Majstorovic, U. Marjanovic, G. von Cieminski, and D. Romero (Hrsg.), *Advances in Production Management Systems. The Path to Digital Transformation and Innovation of Production Management Systems*, Volume 591, p. 363–370. Cham: Springer International Publishing. doi: 10.1007/978-3-030-57993-7_41.
- Kuhnle, A., J.-P. Kaiser, F. Theiß, N. Stricker and G. Lanza (2021). Designing an adaptive production control system using reinforcement learning. *Journal of Intelligent Manufacturing* 32(3), p. 855–876. doi: 10.1007/s10845-020-01612-y.
- Kuhnle, A., M. C. May, L. Schäfer and G. Lanza (2021). Explainable reinforcement learning in production control of job shop manufacturing system. *International Journal of Production Research*, p. 1–23. doi: 10.1080/00207543.2021.1972179.
- Kuhnle, A., N. Röhrig and G. Lanza (2019). Autonomous order dispatching in the semiconductor industry using reinforcement learning. *Procedia CIRP* 79, p. 391–396. doi: 10.1016/j.procir.2019.02.101.
- Kuhnle, A., L. Schäfer, N. Stricker and G. Lanza (2019). Design, Implementation and Evaluation of Reinforcement Learning for an Adaptive Order Dispatching in Job Shop Manufacturing Systems. *Procedia CIRP* 81, p. 234–239. doi: 10.1016/j.procir.2019.03.041.
- Kumar, A. and R. Dimitrakopoulos (2021). Production scheduling in industrial mining complexes with incoming new information using tree search and deep reinforcement learning. *Applied Soft Computing* 110, p. 107644. doi: 10.1016/j.asoc.2021.107644.
- Kumar, A., R. Dimitrakopoulos and M. Maulen (2020). Adaptive self-learning mechanisms for updating short-term production decisions in an industrial mining complex. *Journal of Intelligent Manufacturing* 31(7), p. 1795–1811. doi: 10.1007/s10845-020-01562-5.
- Kumar, H. and S. Giri (2020). A neural network-based algorithm for flow shop scheduling problems under fuzzy environment. *International Journal of Process Management and Benchmarking* 10(2), p. 282. doi: 10.1504/IJPMB.2020.106144.

- Kádár, B., L. Monostori and B. Csáji (2003). Adaptive approaches to increase the performance of production control systems. *Proceedings of 36th CIRP ISMS*, p. 305–312.
- Lai, X., H. Shui and J. Ni (2018). A Two-Layer Long Short-Term Memory Network for Bottleneck Prediction in Multi-Job Manufacturing Systems. In: *Volume 3: Manufacturing Equipment and Systems*, Texas, USA. American Society of Mechanical Engineers. doi: 10.1115/MSEC2018-6678.
- Lan, Y., Y. Liu and G. Sun (2010). An approximation-based approach for fuzzy multi-period production planning problem with credibility objective. *Applied Mathematical Modelling* 34(11), p. 3202–3215. doi: 10.1016/j.apm.2010.02.013.
- Lan, Y., R. Zhao and W. Tang (2011). Minimum risk criterion for uncertain production planning problems. *Computers & Industrial Engineering* 61(3), p. 591–599. doi: 10.1016/j.cie.2011.04.014.
- Lang, S., F. Behrendt, N. Lanzerath, T. Reggelin and M. Muller (2020). Integration of Deep Reinforcement Learning and Discrete-Event Simulation for Real-Time Scheduling of a Flexible Job Shop Production. In: *2020 Winter Simulation Conference (WSC)*, FL, USA, p. 3057–3068. doi: 10.1109/WSC48552.2020.9383997.
- Lang, S., T. Reggelin, F. Behrendt and A. Nahhas (2020). Evolving Neural Networks to Solve a Two-Stage Hybrid Flow Shop Scheduling Problem with Family Setup Times. In: *53rd Hawaii International Conference on System Sciences*, Hawaii, USA, p. 1298–1307.
- Lee, J.-H. and C.-O. Kim (2008). Multi-agent systems applications in manufacturing systems and supply chain management: a review paper. *International Journal of Production Research* 46(1), p. 233–265. doi: 10.1080/00207540701441921.
- Lee, S., Y. Cho and Y. H. Lee (2020). Injection Mold Production Sustainable Scheduling Using Deep Reinforcement Learning. *Sustainability* 12(20), p. 8718. doi: 10.3390/su12208718.
- Leng, J., C. Jin, A. Vogl and H. Liu (2020). Deep reinforcement learning for a color-batching resequencing problem. *Journal of Manufacturing Systems* 56, p. 175–187. doi: 10.1016/j.jmsy.2020.06.001.
- Liao, Y., F. Deschamps, E. d. F. R. Loures and L. F. P. Ramos (2017). Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal. *International Journal of Production Research* 55(12), p. 3609–3629. doi: 10.1080/00207543.2017.1308576.
- Light, R. J. and D. B. Pillemer (1984). *Summing up: the science of reviewing research*. Cambridge, Mass: Harvard University Press.
- Lin, C.-C., D.-J. Deng, Y.-L. Chih and H.-T. Chiu (2019). Smart Manufacturing Scheduling With Edge Computing Using Multiclass Deep Q Network. *IEEE Transactions on Industrial*

- Informatics 15*(7), p. 4276–4284. doi: 10.1109/TII.2019.2908210.
- Liu, C.-L., C.-C. Chang and C.-J. Tseng (2020). Actor-Critic Deep Reinforcement Learning for Solving Job Shop Scheduling Problems. *IEEE Access* 8, p. 71752–71762. doi: 10.1109/ACCESS.2020.2987820.
- Liu, Y., R. Zhang, M. Wang and X. Zhu (2015). A Decomposition-Based Two-Stage Optimization Algorithm for Single Machine Scheduling Problems with Deteriorating Jobs. *Mathematical Problems in Engineering 2015*, p. 1–8. doi: 10.1155/2015/340769.
- Luo, S. (2020). Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Applied Soft Computing 91*, p. 106208. doi: 10.1016/j.asoc.2020.106208.
- Luo, S., L. Zhang and Y. Fan (2021). Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. *Computers & Industrial Engineering 159*, p. 107489. doi: 10.1016/j.cie.2021.107489.
- Madureira, A., J. M. Santos, S. Gomes, B. Cunha, J. Pereira and I. Pereira (2014). Manufacturing rush orders rescheduling: a supervised learning approach. In: *2014 Sixth World Congress on Nature and Biologically Inspired Computing (NaBIC 2014)*, Porto, Portugal, p. 299–304. IEEE.
- Malus, A., D. Kozjek and R. Vrabič (2020). Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. *CIRP Annals 69*(1), p. 397–400. doi: 10.1016/j.cirp.2020.04.001.
- Marchesano, M. G., G. Guizzi, L. C. Santillo and S. Vespoli (2021). Dynamic Scheduling in a Flow Shop Using Deep Reinforcement Learning. In: A. Dolgui, A. Bernard, D. Lemoine, G. von Cieminski, and D. Romero (Hrsg.), *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, Volume 630, p. 152–160. Cham: Springer International Publishing. doi: 10.1007/978-3-030-85874-2_16.
- Mashhadi, F. and S. A. Salinas Monroy (2020). Deep Learning for Optimal Resource Allocation in IoT-enabled Additive Manufacturing. In: *2020 IEEE 6th World Forum on Internet of Things*, LA, USA, p. 1–6. IEEE. doi: 10.1109/WF-IoT48130.2020.9221038.
- May, M. C., L. Behnen, A. Holzer, A. Kuhnle and G. Lanza (2021). Multi-variate time-series for time constraint adherence prediction in complex job shops. *Procedia CIRP 103*, p. 55–60. doi: 10.1016/j.procir.2021.10.008.
- May, M. C., L. Kiefer, A. Kuhnle, N. Stricker and G. Lanza (2021). Decentralized Multi-Agent Production Control through Economic Model Bidding for Matrix Production Systems. *Procedia CIRP 96*, p. 3–8. doi: 10.1016/j.procir.2021.01.043.
- Mayer, S., T. Classen and C. Endisch (2021). Modular production control using deep reinforce-

- ment learning: proximal policy optimization. *Journal of Intelligent Manufacturing* 32(8), p. 2335–2351. doi: 10.1007/s10845-021-01778-z.
- Mehlig, B. (2021). *Machine Learning with Neural Networks: An Introduction for Scientists and Engineers* (1 ed.). Cambridge University Press. doi: 10.1017/9781108860604.
- Mezzogori, D., G. Romagnoli and F. Zammori (2019). Deep learning and WLC: how to set realistic delivery dates in high variety manufacturing systems. *IFAC-PapersOnLine* 52(13), p. 2092–2097. doi: 10.1016/j.ifacol.2019.11.514.
- Minguillon, F. E. and G. Lanza (2019). Coupling of centralized and decentralized scheduling for robust production in agile production systems. *Procedia CIRP* 79, p. 385–390. doi: 10.1016/j.procir.2019.02.099.
- Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller (2013). Playing Atari with Deep Reinforcement Learning. p. 1–9.
- Monostori, L., B. Csáji and B. Kádár (2004). Adaptation and Learning in Distributed Production Control. *CIRP Annals* 53(1), p. 349–352. doi: 10.1016/S0007-8506(07)60714-8.
- Monostori, L., P. Valckenaers, A. Dolgui, H. Panetto, M. Brdys and B. C. Csáji (2014). Cooperative Control in Production and Logistics. *IFAC Proceedings Volumes* 47(3), p. 4246–4265. doi: 10.3182/20140824-6-ZA-1003.01026.
- Moon, J. and J. Jeong (2021). Smart Manufacturing Scheduling System: DQN based on Cooperative Edge Computing. In: *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Seoul, Korea (South), p. 1–8. IEEE. doi: 10.1109/IMCOM51814.2021.9377434.
- Morariu, C. and T. Borangiu (2018). Time series forecasting for dynamic scheduling of manufacturing processes. In: *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, Cluj-Napoca, p. 1–6. IEEE.
- Morariu, C., O. Morariu, S. Răileanu and T. Borangiu (2020). Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. *Computers in Industry* 120, p. 103244. doi: 10.1016/j.compind.2020.103244.
- Mouelhi-Chibani, W. and H. Pierreval (2010). Training a neural network to select dispatching rules in real time. *Computers & Industrial Engineering* 58(2), p. 249–256. doi: 10.1016/j.cie.2009.03.008.
- Nickerson, R., U. Varshney and J. Muntermann (2013). A Method for Taxonomy Development and its Application in Information Systems. *European Journal of Information Systems* 22. doi: 10.1057/ejis.2012.26.
- Németh, P., T. Ladinig and B. Ferenczi (2016). Use of Artificial Neural Networks in the

- Production Control of Small Batch Production. *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, p. 237–240.
- Onaran, E. and S. Yanık (2020). Predicting Cycle Times in Textile Manufacturing Using Artificial Neural Network. In: C. Kahraman, S. Cebi, S. Cevik Onar, B. Oztaysi, A. C. Tolga, and I. U. Sari (Hrsg.), *Intelligent and Fuzzy Techniques in Big Data Analytics and Decision Making*, Volume 1029, p. 305–312. Cham: Springer International Publishing. doi: 10.1007/978-3-030-23756-1_38.
- Overbeck, L., A. Hugues, M. C. May, A. Kuhnle and G. Lanza (2021). Reinforcement Learning Based Production Control of Semi-automated Manufacturing Systems. *Procedia CIRP 103*, p. 170–175. doi: 10.1016/j.procir.2021.10.027.
- Pach, C., T. Berger, T. Bonte and D. Trentesaux (2014). ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. *Computers in Industry 65*(4), p. 706–720. doi: 10.1016/j.compind.2014.02.005.
- Panzer, M. and B. Bender (2021). Deep reinforcement learning in production systems: a systematic literature review. *International Journal of Production Research*, p. 1–26. doi: 10.1080/00207543.2021.1973138.
- Park, I.-B., J. Huh, J. Kim and J. Park (2020). A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities. *IEEE Transactions on Automation Science and Engineering*, p. 1–12. doi: 10.1109/TASE.2019.2956762.
- Park, J., J. Chun, S. H. Kim, Y. Kim and J. Park (2021). Learning to schedule job-shop problems: representation and policy learning using graph neural network and reinforcement learning. *International Journal of Production Research 59*(11), p. 3360–3377. doi: 10.1080/00207543.2020.1870013.
- Patel, M., V. Honavar and K. Balakrishnan (Hrsg.) (2001). *Advances in the evolutionary synthesis of intelligent agents*. Cambridge, Mass: MIT Press.
- Petticrew, M. and H. Roberts (2006). *Why Do We Need Systematic Reviews?* Oxford, UK: Blackwell Publishing Ltd. doi: 10.1002/9780470754887.ch1.
- Pham, Q. T. and T. K. D. Phan (2016). Apply neural network for improving production planning at Samarang petrol mine. *International Journal of Intelligent Computing and Cybernetics 9*(2), p. 126–143. doi: 10.1108/IJICC-09-2015-0032.
- Pol, S., S. Baer, D. Turner, V. Samsonov and T. Meisen (2021). Global Reward Design for Cooperative Agents to Achieve Flexible Production Control under Real-time Constraints:. In: *Proceedings of the 23rd International Conference on Enterprise Information Systems*, Online, p. 515–526. doi: 10.5220/0010455805150526.

- Pusnik, M., B. Sucic, A. Podgornik, F. AL-Mansour and T. Vuk (2014). Net fitting based production planning and decision support system for energy intensive industries. In: *2014 IEEE International Energy Conference*, Cavtat, Croatia, p. 1236–1242. IEEE.
- Rabelo, L., S. Alptekin and A. Kiran (1990). Synergy of artificial neural networks and knowledge-based expert systems for intelligent FMS scheduling. In: *1990 IJCNN International Joint Conference on Neural Networks*, San Diego, CA, USA, p. 359–366 vol.1. IEEE.
- Ramanan, T. R., R. Sridharan, K. S. Shashikant and A. N. Haq (2011). An artificial neural network based heuristic for flow shop scheduling problems. *Journal of Intelligent Manufacturing* 22(2), p. 279–288. doi: 10.1007/s10845-009-0287-5.
- Ramos, D., P. Faria, Z. Vale and R. Correia (2021). Short Time Electricity Consumption Forecast in an Industry Facility. *IEEE Transactions on Industry Applications*, p. 1–1. doi: 10.1109/TIA.2021.3123103.
- Ramsauer, C. (1997). *Dezentrale PPS-Systeme*. Wiesbaden: Gabler Verlag. doi: 10.1007/978-3-322-89084-9.
- Riedmiller, S. and M. Riedmiller (1999). A neural reinforcement learning approach to learn local dispatching policies in production scheduling. *Proceedings of the 16th international joint conference on Artificial intelligence* 2, p. 764–769.
- Rojas, R. A. and E. Rauch (2019). From a literature review to a conceptual framework of enablers for smart manufacturing control. *The International Journal of Advanced Manufacturing Technology* 104(1-4), p. 517–533. doi: 10.1007/s00170-019-03854-4.
- Rouhani, S., M. Fathian, M. Jafari and P. Akhavan (2010). Solving the Problem of Flow Shop Scheduling by Neural Network Approach. In: F. Zavoral, J. Yaghob, P. Pichappan, and E. El-Qawasmeh (Hrsg.), *Networked Digital Technologies*, Volume 88, p. 172–183. Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-14306-9_18.
- Rummukainen, H. and J. K. Nurminen (2019). Practical Reinforcement Learning - Experiences in Lot Scheduling Application. *IFAC-PapersOnLine* 52(13), p. 1415–1420. doi: 10.1016/j.ifacol.2019.11.397.
- Sadiq, S. S., A. M. Abdulazeez and H. Haron (2020). Solving Multi-Objective Master Production Scheduling Model of Kalak Refinery System Using Hybrid Evolutionary Imperialist Competitive Algorithm. *Journal of Computer Science* 16(2), p. 137–149. doi: 10.3844/jc-ssp.2020.137.149.
- Sajko, N., S. Kovacic, M. Ficko, I. Palcic and S. Klancnik (2020). Management and Production Engineering Review. doi: 10.24425/MPER.2020.134931.
- Schneckenreither, M. and S. Haeussler (2019). Reinforcement Learning Methods for Operations

- Research Applications: The Order Release Problem. In: G. Nicosia, P. Pardalos, G. Giuffrida, R. Umeton, and V. Sciacca (Hrsg.), *Machine Learning, Optimization, and Data Science*, Volume 11331, p. 545–559. Cham: Springer International Publishing. doi: 10.1007/978-3-030-13709-0_46.
- Schneckenreither, M., S. Haeussler and C. Gerhold (2021). Order release planning with predictive lead times: a machine learning approach. *International Journal of Production Research* 59(11), p. 3285–3303. doi: 10.1080/00207543.2020.1859634.
- Schwung, D., S. Yuwono, A. Schwung and S. X. Ding (2021). Decentralized learning of energy optimal production policies using PLC-informed reinforcement learning. *Computers & Chemical Engineering* 152, p. 107382. doi: 10.1016/j.compchemeng.2021.107382.
- Seito, T. and S. Munakata (2020). Production Scheduling based on Deep Reinforcement Learning using Graph Convolutional Neural Network:. In: *Proceedings of the 12th International Conference on Agents and Artificial Intelligence*, Valletta, Malta, p. 766–772. doi: 10.5220/0009095207660772.
- Shi, D., W. Fan, Y. Xiao, T. Lin and C. Xing (2020). Intelligent scheduling of discrete automated production line via deep reinforcement learning. *International Journal of Production Research* 58(11), p. 3362–3380. doi: 10.1080/00207543.2020.1717008.
- Shin, M., K. Ryu and M. Jung (2012). Reinforcement learning approach to goal-regulation in a self-evolutionary manufacturing system. *Expert Systems with Applications* 39(10), p. 8736–8743. doi: 10.1016/j.eswa.2012.01.207.
- Silva, C., V. Ribeiro, P. Coelho, V. Magalhães and P. Neto (2017). Job Shop Flow Time Prediction using Neural Networks. *Procedia Manufacturing* 11, p. 1767–1773. doi: 10.1016/j.promfg.2017.07.309.
- Silva, T. and A. Azevedo (2019). Production flow control through the use of reinforcement learning. *Procedia Manufacturing* 38, p. 194–202. doi: 10.1016/j.promfg.2020.01.026.
- Silver, D., T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan and D. Hassabis (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm.
- Sim, M. H., M. Y. H. Low, C. S. Chong and M. Shakeri (2020). Job Shop Scheduling Problem Neural Network Solver with Dispatching Rules. In: *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, Singapore, p. 514–518. IEEE.
- Sobottka, T., F. Kamhuber, M. Faezirad and W. Sihn (2019). Potential for Machine Learning in Optimized Production Planning with Hybrid Simulation. *Procedia Manufacturing* 39, p. 1844–1853. doi: 10.1016/j.promfg.2020.01.254.

- Stauder, M. and N. Kühl (2021). AI for in-line vehicle sequence controlling: development and evaluation of an adaptive machine learning artifact to predict sequence deviations in a mixed-model production line. *Flexible Services and Manufacturing Journal*. doi: 10.1007/s10696-021-09430-x.
- Stricker, N., A. Kuhnle, R. Sturm and S. Friess (2018). Reinforcement learning for adaptive order dispatching in the semiconductor industry. *CIRP Annals* 67(1), p. 511–514. doi: 10.1016/j.cirp.2018.04.041.
- Tang, J. and K. Salonitis (2021). A Deep Reinforcement Learning Based Scheduling Policy for Reconfigurable Manufacturing Systems. *Procedia CIRP* 103, p. 1–7. doi: 10.1016/j.procir.2021.09.089.
- Teichmann, M., A. Ullrich, D. Kotarski and N. Gronau (2021). Facing the Demographic Change – Recommendations for Designing Learning Factories as Age-Appropriate Teaching-Learning Environments for Older Blue-Collar Workers. *SSRN Electronic Journal*. doi: 10.2139/ssrn.3858716.
- Thomé, A. M. T., L. F. Scavarda and A. J. Scavarda (2016). Conducting systematic literature review in operations management. *Production Planning & Control* 27(5), p. 408–420. doi: 10.1080/09537287.2015.1129464.
- Tranfield, D., D. Denyer and P. Smart (2003). Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. *British Journal of Management* 14(3), p. 207–222. doi: 10.1111/1467-8551.00375.
- Ueda, K., A. Markus, L. Monostori, H. Kals and T. Arai (2001). Emergent Synthesis Methodologies for Manufacturing. *CIRP Annals* 50(2), p. 535–551. doi: 10.1016/S0007-8506(07)62994-1.
- Wang, C. and P. Jiang (2018). Manifold learning based rescheduling decision mechanism for recessive disturbances in RFID-driven job shops. *Journal of Intelligent Manufacturing* 29(7), p. 1485–1500. doi: 10.1007/s10845-016-1194-1.
- Wang, C. and P. Jiang (2019). Deep neural networks based order completion time prediction by using real-time job shop RFID data. *Journal of Intelligent Manufacturing* 30(3), p. 1303–1318. doi: 10.1007/s10845-017-1325-3.
- Wang, L., X. Hu, Y. Wang, S. Xu, S. Ma, K. Yang, Z. Liu and W. Wang (2021). Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning. *Computer Networks* 190, p. 107969. doi: 10.1016/j.comnet.2021.107969.
- Wang, X., L. Zhang, T. Lin, C. Zhao, K. Wang and Z. Chen (2022). Solving job scheduling problems in a resource preemption environment with multi-agent reinforcement learning. *Robotics and Computer-Integrated Manufacturing* 77, p. 102324. doi: 10.1016/j.rcim.2022.102324.

- Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmuller, T. Bauernhansl, A. Knapp and A. Kyek (2018). Deep reinforcement learning for semiconductor production scheduling. In: *2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, NY, USA, p. 301–306. IEEE. doi: 10.1109/ASMC.2018.8373191.
- Webster, J. and R. T. Watson (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly* 26(2), p. 13–23.
- Weichert, D., P. Link, A. Stoll, S. Rüping, S. Ihlenfeldt and S. Wrobel (2019). A review of machine learning for the optimization of production processes. *The International Journal of Advanced Manufacturing Technology* 104(5-8), p. 1889–1902. doi: 10.1007/s00170-019-03988-5.
- Wen, H., S. Hou, Z. Liu and Y. Liu (2017). An optimization algorithm for integrated remanufacturing production planning and scheduling system. *Chaos, Solitons & Fractals* 105, p. 69–76. doi: 10.1016/j.chaos.2017.10.012.
- Wen, H., M. Liu, C. Liu and C. Liu (2015). Remanufacturing production planning with compensation function approximation method. *Applied Mathematics and Computation* 256, p. 742–753. doi: 10.1016/j.amc.2015.01.070.
- Windler, T., J. Busse and J. Rieck (2019). One month-ahead electricity price forecasting in the context of production planning. *Journal of Cleaner Production* 238, p. 117910. doi: 10.1016/j.jclepro.2019.117910.
- Worapradya, K. and P. Thanakijkasem (2015). Proactive Scheduling for Steelmaking-Continuous Casting Plant with Uncertain Machine Breakdown Using Distribution-Based Robustness and Decomposed Artificial Neural Network. *Asia-Pacific Journal of Operational Research* 32(02), p. 1550010. doi: 10.1142/S0217595915500104.
- Wu, C.-H., F.-Y. Zhou, C.-K. Tsai, C.-J. Yu and S. Dauzère-Pérès (2020). A deep learning approach for the dynamic dispatching of unreliable machines in re-entrant production systems. *International Journal of Production Research* 58(9), p. 2822–2840. doi: 10.1080/00207543.2020.1727041.
- Wu, C.-X., M.-H. Liao, M. Karatas, S.-Y. Chen and Y.-J. Zheng (2020). Real-time neural network scheduling of emergency medical mask production during COVID-19. *Applied Soft Computing* 97, p. 106790. doi: 10.1016/j.asoc.2020.106790.
- Wu, H., G. Evans and K.-H. Bae (2016). Production control in a complex production system using approximate dynamic programming. *International Journal of Production Research* 54(8), p. 2419–2432. doi: 10.1080/00207543.2015.1086035.
- Xie, S., T. Zhang and O. Rose (2019). Online Single Machine Scheduling Based on Simulation and Reinforcement Learning. In: *18. ASIM Fachtagung Simulation in Produktion und Logistik*,

- Chemnitz, p. 1–10.
- Yamashiro, H. and H. Nonaka (2021). Estimation of processing time using machine learning and real factory data for optimization of parallel machine scheduling problem. *Operations Research Perspectives* 8, p. 100196. doi: 10.1016/j.orp.2021.100196.
- Yang, S. and Z. Xu (2021). Intelligent scheduling and reconfiguration via deep reinforcement learning in smart manufacturing. *International Journal of Production Research*, p. 1–18. doi: 10.1080/00207543.2021.1943037.
- Zang, Z., W. Wang, Y. Song, L. Lu, W. Li, Y. Wang and Y. Zhao (2019). Hybrid Deep Neural Network Scheduler for Job-Shop Problem Based on Convolution Two-Dimensional Transformation. *Computational Intelligence and Neuroscience* 2019, p. 1–19. doi: 10.1155/2019/7172842.
- Zhang, H.-C. and S. H. Huang (1995). Applications of neural networks in manufacturing: a state-of-the-art survey. *International Journal of Production Research* 33(3), p. 705–728. doi: 10.1080/00207549508930175.
- Zhang, H. P. (2019). Optimization of Remanufacturing Production Scheduling Considering Uncertain Factors. *International Journal of Simulation Modelling* 18(2), p. 344–354. doi: 10.2507/IJSIMM18(2)CO8.
- Zhang, J., G. Ding, Y. Zou, S. Qin and J. Fu (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing* 30(4), p. 1809–1830. doi: 10.1007/s10845-017-1350-2.
- Zhang, W. and T. G. Dietterich (1995a). High-Performance Job-Shop Scheduling With A Time-Delay TD Network. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, p. 1024–1030.
- Zhang, W. and T. G. Dietterich (1995b). A Reinforcement Learning Approach to Job-Shop Scheduling. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, p. 1114–1120.
- Zhang, Z., Z. L. Guan, J. Zhang and X. Xie (2019). A Novel Job-Shop Scheduling Strategy Based on Particle Swarm Optimization and Neural Network. *International Journal of Simulation Modelling* 18(4), p. 699–707. doi: 10.2507/IJSIMM18(4)CO18.
- Zhao, F. Q., J. H. Zou and Y. H. Yang (2010). A Hybrid Approach Based on Artificial Neural Network (ANN) and Differential Evolution (DE) for Job-Shop Scheduling Problem. *Applied Mechanics and Materials* 26-28, p. 754–757. doi: 10.4028/www.scientific.net/AMM.26-28.754.
- Zhao, T. and H. Zhang (2022). A new method to improve the generalization ability of neural

- networks: A case study of nuclear mass training. *Nuclear Physics A 1021*, p. 122420. doi: 10.1016/j.nuclphysa.2022.122420.
- Zhao, Y., Y. Wang, Y. Tan, J. Zhang and H. Yu (2021). Dynamic Jobshop Scheduling Algorithm Based on Deep Q Network. *IEEE Access 9*, p. 122995–123011. doi: 10.1109/ACCESS.2021.3110242.
- Zhao, Y. and H. Zhang (2021). Application of Machine Learning and Rule Scheduling in a Job-Shop Production Control System. *International Journal of Simulation Modelling 20(2)*, p. 410–421. doi: 10.2507/IJSIMM20-2-CO10.
- Zheng, S., C. Gupta and S. Serita (2020). Manufacturing Dispatching Using Reinforcement and Transfer Learning. *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, p. 655–671. doi: 10.1007/978-3-030-46133-1_39.
- Zhou, D., V. Cherkassky, T. Baldwin and D. Hong (1990). Scaling neural network for job-shop scheduling. In: *1990 IJCNN International Joint Conference on Neural Networks*, San Diego, CA, USA, p. 889–894 vol.3. IEEE. doi: 10.1109/IJCNN.1990.137947.
- Zhou, F.-Y., C.-H. Wu and C.-J. Yu (2017). Dynamic dispatching for re-entrant production lines — A deep learning approach. In: *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, Xi'an, p. 1026–1031. IEEE. doi: 10.1109/COASE.2017.8256238.
- Zhou, L., Z. Jiang, N. Geng, Y. Niu, F. Cui, K. Liu and N. Qi (2022). Production and operations management for intelligent manufacturing: a systematic literature review. *International Journal of Production Research 60(2)*, p. 808–846. doi: 10.1080/00207543.2021.2017055.
- Zhou, L., L. Zhang and B. K. Horn (2020). Deep reinforcement learning-based dynamic scheduling in smart manufacturing. *Procedia CIRP 93*, p. 383–388. doi: 10.1016/j.procir.2020.05.163.
- Zhou, T., D. Tang, H. Zhu and L. Wang (2021). Reinforcement Learning With Composite Rewards for Production Scheduling in a Smart Factory. *IEEE Access 9*, p. 752–766. doi: 10.1109/ACCESS.2020.3046784.
- Zhou, T., D. Tang, H. Zhu and Z. Zhang (2021). Multi-agent reinforcement learning for online scheduling in smart factories. *Robotics and Computer-Integrated Manufacturing 72*, p. 102202. doi: 10.1016/j.rcim.2021.102202.
- Zhu, H., M. Li, Y. Tang and Y. Sun (2020). A Deep-Reinforcement-Learning-Based Optimization Approach for Real-Time Scheduling in Cloud Manufacturing. *IEEE Access 8*, p. 9987–9997. doi: 10.1109/ACCESS.2020.2964955.
- Zipfel, A., S. Braunreuther and G. Reinhart (2019). Approach for a Production Planning and Control System in Value-Adding Networks. *Procedia CIRP 81*, p. 1195–1200. doi: 10.1016/j.procir.2019.03.291.

- Çalış, B. and S. Bulkan (2015). A research survey: review of AI solution strategies of job shop scheduling problem. *Journal of Intelligent Manufacturing* 26(5), p. 961–973. doi: 10.1007/s10845-013-0837-8.
- Şenyiğit, E. and U. Atici (2013). Artificial neural network models for lot-sizing problem: a case study. *Neural Computing and Applications* 22(6), p. 1039–1047. doi: 10.1007/s00521-012-0863-z.

5.10 Supplements and detailed review tables

AC	Actor-critic algorithm	Mfg.	Manufacturing
A2C	Advantage actor critic algorithm	ML	Machine learning
A3C	Asynchronous advantage actor critic	NEAT	Neuro evolution of augmenting topologies
ADP	Approximate dynamic programming	NN	Neural network
BP	Backpropagation algorithm	PER	Prioritized experience replay
Conv.	Convolutional neural network	PPC	Production planning and control
DBN	Deep belief network	PPO	Proximal policy optimization
DDDQN	Dueling double DQN	PSO	Particle swarm optimization
DDPG	Deep deterministic policy gradient	RBFN	Radial basis function network
DP	Dynamic programming	RL	Reinforcement learning
DQN	Deep Q-learning	RM	Regression model
DRL	Deep reinforcement learning	RNN	Recurrent neural network
GA	Genetic algorithm	SA	Simulated annealing
GCNN	Graph convolutional neural network	TD3	Twin delayed DDPG
GNN	Graph neural network	TRPO	Trust region policy optimization
HNN	Hopfield network	VPSO	Virus particle swarm optimization
LSTM	Long-short-term memory	WIP	Work in progress
MDP	Markov-decision process		

Table 5.8 List of abbreviations

Publication 2 - References

Plain planning approaches								
	Subtopic	Algo.	NN	Objective	Superior	Application	Simulation	Source
1	Dynamic scheduling	DQN	GCNN	Minimize makespan	Superior	Flexible manufacturing	Simulation	Hu et al. (2020)
2	Dynamic scheduling	A2C	FFNN	Max. profitability	Superior	Continuous chemical process	Simulation	Hubbs et al. (2020)
3	Dynamic scheduling	DQN	FFNN	Minimize completion time	-	General tasks, services	Simulation	Zhou et al. (2020)
4	Dynamic scheduling	Policy gradient	FFNN	Maximize resource utilization	Similar	Cloud manufacturing	Simulation	Zhu et al. (2020)
5	Flow-shop scheduling	DQN	FFNN	Maximize throughput	-	Flow-shop	Simulation	Marchesano et al. (2021)
6	Flow-shop scheduling	Levenberg-Marquardt	FFNN	Minimize overall completion time	-	Flow-shop	Simulation	Rouhani et al. (2010)
7	Flow-shop scheduling	REINFORCE	LSTM	Negative total tardiness	Superior	Medical mask production	Simulation	Wu et al. (2020)
8	Job-shop scheduling	AC	RNN	Min. setup waste	Similar	Blown film extrusion	Simulation	Gannouni et al. (2020)
9	Job-shop scheduling	HNN	HNN	Min. makespan	-	Job-shop	Simulation	Fnaiech et al. (2012)
10	Job-shop Scheduling	DQN	FFNN	Minimize makespan	Superior	Job-shop	Simulation	Groth et al. (2021)
11	Job-shop scheduling	DDDQN with PER	Conv.	Minimize makespan	Superior	Job-shop	Simulation	Han and Yang (2020)
12	Job-shop scheduling	DQN	FFNN	Minimize lead-time	Superior	Job-shop	Simulation	Kardos et al. (2021)
13	Job-shop scheduling	PPO	GNN	Minimize makespan	Superior	Job-shop	Simulation	Park et al. (2021)
14	Job-shop scheduling	PPO	FFNN	Optimize exec. time, minimize makespan	Superior	Job-shop	Simulation	Wang et al. (2021)
15	Job-shop scheduling	DQN	FFNN	Minimize makespan, costs, balance workloads	Superior	Job-shop	Simulation	Zhou et al. (2021)
16	Job-shop scheduling	DQN	FFNN	Completion time, energy con., utilization	Superior	Reconfigurable production	Simulation	Chen et al. (2019)
17	Job-shop scheduling	DQN	FFNN	Minimize makespan	Superior	Semiconductor	Simulation	Lin et al. (2019)
18	Job-shop scheduling	DQN	FFNN	Minimize completion time, lateness	Superior	Single machine job-shop	Simulation	Xie et al. (2019)
19	Job-shop scheduling	DRL	GCNN	Maximize fill rate	-	swv11 in OR library	Simulation	Seito et al. (2020)
20	Lot scheduling	PPO	FFNN	Min. waiting times, amount, cost	Superior	Single machine	Simulation	Rummukainen et al. (2019)
21	Lot-sizing	BP	FFNN	Minimize production, set-up, and inventory costs	Superior	Air supply and maintenance centre	Simulation	Şenyiğit and Atici (2013)
22	Re-entrant production	DQN	FFNN	Robustness	Similar	Single-product production	Simulation	Shi et al. (2020)
23	Rescheduling	DQN	Conv.	Minimize tardiness	Superior	Semi-continuous extruders	Simulation	Palombarini et al.
24	Rescheduling	DQN	Conv.	Minimize tardiness	Superior	Semi-continuous extruders	Simulation	Palombarini et al.
25	Rescheduling	Double DQN	Conv.	Minimize changeover costs	Superior	Color batching	Simulation	Leng et al. (2020)
26	Rush-order rescheduling	Supervised/ BP	FFNN	Precision and accuracy	Similar	Job-shop	Simulation	Madureira et al. (2014)
27	Task scheduling	DQN	FFNN	Minimize makespan	Superior	Cloud manufacturing	Simulation	Dong et al. (2020)

Table 5.9 Plain NN based approaches in production planning

5.10 Supplements and detailed review tables

Embedded planning approaches									
Subtopic	Algo.	NN	Objective	Super.	Embedding	Application	Simulation	Source	
28	Batch scheduling	RM	FFNN	Feasibility accuracy	Superior	NN anticipates batch feasibility for top batch scheduler. If feasible, instructions go to base model for final complex nesting	Metal processing	Simulation	Gahm et al. (2022)
29	Flow-shop scheduling	Supervised	FFNN	Minimize makespan	Superior	Hybrid fuzzy and NN based concept	Three echelon supply chain	Simulation	Kumar and Giri (2020)
30	Flow-shop scheduling	BP	FFNN	Minimize makespan	Superior	NN optimized by Suliman heuristic (1) and NN with GA (2)	Benchmark flow-shops	Simulation	Ramanan et al. (2011)
31	Job-shop scheduling	BP	FFNN	Minimize makespan	Superior	Hybrid algorithm, stand-alone heuristic combined with NN operation prioritizing with dispatching rules	Job-shop	Simulation	Sim et al. (2020)
32	Job-shop scheduling	BP	FFNN/Conv. NN	Minimize makespan	Superior	Hybrid scheduler, GA for training, then generate subproblems and scheduling transformation for NN scheduler.	Job-shop	Simulation	Zang et al. (2019)
33	Job-shop scheduling	Gradient search	Conv. NN	Minimize completion time	-	Conv. NN for scheduling, differential evolution for sequence optimization	Job-shop	Simulation	Zhao et al. (2010)
34	Job-shop scheduling	BP	FFNN	Minimize max. makespan	Superior	PSO-based NN optimization	Job-shop	Simulation	Zhang et al. (2019)
35	Modelling	BP	FFNN	Accuracy	Superior	NN as meta-modeller for GA tuning	Industrial bakery	Hybrid	Sobottka et al. (2019)
36	Order allocation	Lagrangian	FFNN	Utility of AM Cloud	Superior	Allocation and payment network	Additive mfg. order allocation	Simulation	Mashhadi et al. (2020)
37	Production planning	BP	FFNN	Production, inventory cost	-	NN approximates credibility objective and is embedded into PSO	6 sources/period production	Simulation	Lan et al. (2010)
38	Production planning	BP	FFNN	Production cost	-	Hybrid monkey algorithm, stochastic simulation, NN	Fuel production	Simulation	Lan et al. (2011)
39	Production planning	SA	FFNN	Optimal credibility	-	Combined NN and SA algorithm approximation for multi-product multi-period scheduling	Furniture manufacturing	Simulation	Feng and Yuan (2011)
40	Remanufacturing scheduling	BP	FFNN	Minimum completion time	-	Double fuzzy algorithm with GA to prevent local optimality and slow convergence of BP algorithm.	Crankshafts remanufacturing	Simulation	Zhang (2019)
41	Remanufacturing scheduling	RBFN	FFNN	Minimum total mfg. costs	Superior	NN for approximating the expectation function which converts infinite to finite problems for VPSO	Camshaft remanufacturing	Simulation	Wen et al. (2015)
42	Remanufacturing scheduling	BP	FFNN	Accuracy	-	FFNN into GA to calculate chromosome output	Cam-/crankshaft remanufacturing	Simulation	Wen et al. (2017)
43	Rescheduling	-	FFNN	Minimize response time	Superior	Supervised dimensionality reduction, GRNN mapping, SVM rescheduling	Job-shop	Simulation	Wang and Jiang (2018)
44	Scheduling / reconfiguration	A2C	FFNN	Minimum total tardiness cost	Superior	DRL (1) scheduling for job processing and (2) reconfiguration for production mode	Test instances	Simulation	Yang and Xu (2021)
45	Short-term scheduling	DRL (sim. AlphaGo)	FFNN	Short-term profit	Superior	MC tree search to train a NN to adapts short-term production. NN improves tree search strength for better experiences	Ore production	Reality	Kumar et al. (2021)
46	Single machine scheduling	BP	FFNN	Minimum total weighted tardiness	Superior	Two-stage approach with NN problem downscaling and metaheuristics solution	Single machine	Simulation	Liu et al. (2015)
47	Task scheduling	BP	FFNN	Optimal evaluation	-	3-module system with stochastic classification, training/validation NN, and interactive validation	Knitting processes	Simulation	Baeza Serrato (2018)

Table 5.10 Embedded NN based approaches in production planning

Publication 2 - References

Multi-agent planning approaches										
Subtopic	Algo.	NN	Objective	Superiority	Interaction	Training	Application	Simulation	Source	
48	Job-shop Scheduling	-	-	Minimize process time	-	Global objective	Iterative training of local NN, other agents crt. by heuristics	Job-shop	Simulation	Baer et al. (2019)
49	Job-shop scheduling	DQN	FFNN	Minimize makespan	-	None (sensing)	Joint-action learning	Job-shop	Simulation	Baer et al. (2020)
50	Job-Shop scheduling	Asyn. DDPG	Conv.	Minimize makespan	Superior	Agent state information	Central and parallel training	Job-shop	Simulation	Liu et al. (2020)
51	Job-shop scheduling	DQN	FFNN	Minimize makespan	Superior	Agent state information, global objective	Single NN instance	Job-shop	Simulation	Pol et al. (2021)
52	Job-shop scheduling	Modified DQN	FFNN	Minimize make-span	Superior	Agent information exchange	Central Q-value/ decentral scheduling network	Job-shop	Reality	Zhou et al. (2021)
53	Real-time scheduling	Simulated annealing	FFNN	Minimize tardiness	Superior	Agent information exchange, global objective	-	Job-shop	Simulation	Hammami et al. (2015)
54	Robust scheduling	DQN	FFNN	Minimize makespan	Superior	None	Central training	Semicond. scheduling	Simulation	Park et al. (2020)
55	Sustainable scheduling	DQN	FFNN	Minimize process time	Superior	None	Central training	Mold scheduling	Simulation	Lee et al. (2020)

Table 5.11 Multi-agent NN based approaches in production planning

5.10 Supplements and detailed review tables

Plain forecasting approaches								
Forecast	Subtopic	Algo.	NN	Super.	Application	Simulation	Source	
56	Cycle-time	Dispatching	BP	FFNN	-	Semiconductor	Simulation	Chakravorty and Nagarur (2020)
57	Cycle-time	Flexible mfg.	BP	FFNN	Superior	Textile mfg.	Simulation	Onaran and Yanik (2020)
58	Cycle-time	Production planning	BP	FFNN	-	Textile mfg.	Simulation	Cao and Ji (2021)
59	Cycle-time	Virtual machine prototype	BP	FFNN	-	Job-shop	Simulation	Jain et al. (2017)
60	Electricity price	Energy cost oriented planning	BP	FFNN	Superior	Electricity price forecast	Simulation	Windler et al. (2019)
61	Energy cost / consumption	Production planning	Levenberg-Marquardt	FFNN	-	Rotary clinker furnace	Simulation	Pusnik et al. (2014)
62	Energy consumption	Production planning	-	FFNN	Superior	Industrial facility	Simulation	Ramos et al. (2021)
63	Failure occurrence time	Dynamic scheduling	-	FFNN	Superior	Pharmaceutical factory	Simulation	Azab et al. (2021)
64	Flow-time	Cost estimation	BP	FFNN	Superior	Oil-dry-type cast resin transformers	Simulation	Karaoglan and Karademir (2017)
65	Flow-time	Job-shop scheduling	Levenberg-Marquardt	FFNN	Superior	Job-shop	Simulation	Silva et al. (2017)
66	Lead-time	Job-shop scheduling	Supervised	FFNN	Superior	Job-shop	Simulation	Kramer et al. (2020)
67	Lead-time	Job-shop scheduling	-	FFNN	-	Aluminium extrusion	Simulation	Sajko et al. (2020)
68	Make-span	Online scheduling	AlphaZero	Conv. NN	-	Interconnected assembly	Simulation	Göppert et al. (2021)
69	Number of mfg. products	Feasibility assessment	-	FFNN	-	Flywheel production	Simulation	Burduk et al. (2018)
70	Liquid, oil, gas flow	Production back allocation	BP	FFNN	Superior	Samarang petrol mine	Reality	Pham and Phan (2016)
71	Order compl. time	Job-shop control	BP	Deep belief network	Superior	RFID-driven job-shop	Simulation	Wang and Jiang (2019)
72	Costs, output, quality	Black-box modelling	Levenberg-Marquardt	FFNN	-	Tennessee Eastman proc.	Simulation	Glavan et al. (2013)
73	Processing times	Offline scheduling	BP	RNN	Inferior	Parallel machine sched.	Simulation	Yamashiro and Nonaka (2021)
74	Sequence deviation	Sequencing	BP	FFNN	Inferior	Automotive	Simulation	Stauder and Kühl (2021)
75	Time constraint violations	Production planning	BP	RNN/LSTM	Similar	Job-shop	Simulation	May et al. (2021)
76	Through-put	Process ctrl.	Supervised	FFNN	Superior	Geo-metallurgy	Simulation	Both and Dimitrakopoulos (2021)
77	Through-put	Process ctrl./ order release	BP	FFNN	-	Colour filter fabrication	Reality	Huang et al. (2016)
78	WIP	Production planning	-	LSTM	Superior	Bottleneck machine	Simulation	Gallina et al. (2021)

Table 5.12 Plain NN based approaches in production forecasting

Publication 2 - References

Embedded forecasting approaches									
Forecast	Subtopic	Algo.	NN	Super.	Embedding	Application	Simulation	Source	
79	Cycle-time	Multi-job production	BP	FFNN	Superior	Fuzzy c-means job classifying and NN based prediction for each class	Semiconductor	Simulation	Chen (2016)
80	Cycle/ blockage/ starvation time	Bottleneck prediction	Levenberg–Marquardt	LSTM	Superior	2-staged cycle and starvation time prediction	Underbody assembly	Simulation	Lai et al. (2018)
81	Energy con. patterns	Predictive planning	Unsupervised	LSTM	Superior	LSTM with prior classification and clustering	Job-shop	Simulation	Morariu et al. (2020)
82	Gross demand	Master prod. scheduling	BP	FFNN	Superior	NN forecast for subsequent scheduling algorithms	Kalak Refinery System	Simulation	Sadiq et al. (2020)
83	Job remaining time	Rescheduling	BP	FFNN	Superior	Deep autoencoder extracts features, NN predicts jobs remaining time forecast	Aeroengine production	Reality	Fang et al. (2020)
84	Lead-time	Make-to-order manufacturing	BP	FFNN	Superior	Non-/Bottleneck forecasting separation	Three-stage flow-shop	Simulation	Schneckenreither et al. (2021)
85	Lead-time	Workload control	-	FFNN	Superior	WLC based control with NN prediction to define delivery dates	Job-shop	Simulation	Mezzogori et al. (2019)
86	Load-value	Production scheduling	BP	FFNN	-	Affinity propagation operations clustering with FFNN forecasting	Semiconductor	Simulation	Han et al. (2019)
87	Order completion time	Production scheduling	BP	FFNN	Superior	NN for prediction, GA/SA for global/ local tuning	Job-shop	Simulation	Hu and Zhou (2020)
88	Performance mean/standard deviation	Proactive scheduling	BP	FFNN	Superior	K-means clustering for decomposition and NN based perf. measures	Steelmaking contin. casting	Simulation	Worapradya et al. (2015)
89	Product completion time	Production scheduling	-	LSTM	Superior	NN prediction w.analytical model as baseline	Multi-product serial production	Simulation	Huang et al. (2020)
90	Production progress	Make-to-order manufacturing	BP	DBN	Superior	2-staged DBN based encoding and progress prediction	Job-shop	Simulation	Huang et al. (2019)
Multi-agent forecasting approaches									
Forecast	Subtopic	Algo./ NN	Super.	Interaction	Training	Application	Simulation	Source	
91	Manufacturing cost	Production scheduling	Supervised/ RNN; LSTM	-	Bidding mechanism	Decentral training	General inter-connected assembly	Simulation	Morariu and Borangiu (2018)

Table 5.13 Embedded and multi-agent NN approaches in production forecasting

5.10 Supplements and detailed review tables

Plain control approaches								
	Subtopic	Algo.	NN	Objective	Superiority	Application	Simulation	Source
92	Accuracy control	Bacterial memetic	FFNN	Opt. performance measurement	-	Small-batch assembly	Simulation	Németh et al. (2016)
93	Dispatching	DQN	FFNN	WIP; util. ratio (1.); min. global time constraints (2.)	Superior	Semiconductor	Simulation	Altenmüller et al. (2020)
94	Dispatching	TRPO	FFNN	Min. throughput time	Superior	Semiconductor	Simulation	Kuhnle et al. (2019)
95	Dispatching	TRPO	FFNN	Max. utilization, min. lead time	Superior	Semiconductor	Simulation	Kuhnle et al. (2019)
96	Dispatching	TRPO	FFNN	Max. utilization, min.throughput/ waiting time	Superior/similar	Semiconductor	Simulation	Kuhnle et al. (2021)
97	Dispatching	DQN	FFNN	Max. utilization, min. lead times	Superior	Semiconductor	Simulation	Stricker et al. (2018)
98	Dispatching	DDDQN	FFNN	Min. reconfiguration, min. makespan	Superior	Reconfigurable mfg. system	Simulation	Tang and Saloniitis (2021)
99	Dispatching	MDP	FFNN	Min. average cycle time	-	Re-entrant production	Simulation	Wu et al. (2020)
100	Dispatching	DP	FFNN	Minimize total production cost	-	Re-entrant production	Simulation	Zhou et al. (2017)
101	Flow control	-	FFNN	Min. makespan, cost, energy consumption	Superior	WIP bounding	Simulation	Danishvar et al. (2021)
102	Flow control	DQN	FFNN	High throughput, min. WIP	Superior	WIP bounding	Simulation	Silva and Azevedo (2019)
103	Flow-shop scheduling	-	FFNN	Minimize mean tardiness	Superior	Flow-shop	Simulation	Mouelhi-Chibani et al. (2010)
104	Job-shop scheduling	BP algorithm	FFNN	Speed up modeling process, raise accuracy	-	Job-shop	Simulation	Bergmann and Stelzer (2011)
105	Job-shop scheduling	BP algorithm	FFNN	Imitation of dispatching rule	-	Job-shop	Simulation	Bergmann et al. (2014)
106	Job-shop scheduling	DoubleDQN	FFNN	Minimize total tardiness	Superior	Job-shop	Simulation	Luo (2020)
107	Job-shop scheduling	DQN	FFNN	Minimize makespan	Superior	Job-shop	Simulation	Moon and Jeong (2021)
108	Job-shop scheduling	PPO	FFNN	Max. productivity	-	Job-shop	Simulation	Overbeck et al. (2021)
109	Job-shop scheduling	AC	Cong.	Min. makespan and total delay	Superior	Job-shop	Simulation	Zhao and Zhang (2021)
110	Job-shop scheduling	REINFORCE	FFNN	Min. mean lateness, tardiness	Superior	Job-shop	Simulation	Zheng et al. (2020)
111	Material flow control	Policy gradient	FFNN	Max. profit, min. cost, target deviation	Superior	Copper mining complex	Simulation	Kumar et al. (2020)
112	Modular control	PPO	FFNN	Max. throughput	-	Modular production	Simulation	Mayer et al. (2021)
113	Order release	A3C, Q-learning	FFNN	Min. tardiness, throughput time	Superior	Two-stage flow-shop	Simulation	Scheckenreither et al. (2019)
114	Production and inventory control	ADP	FFNN	Min. total cost per unit	Superior	Dishwasher wire rack production system	Simulation	Wu et al. (2016)

Table 5.14 Plain NN based approaches in production control

Publication 2 - References

Embedded control approaches									
Subtopic	Algo.	NN	Objective	Superiority	Embedding	Application	Simulation	Source	
115	Dynamic scheduling	Supervised	FFNN	Maximize machine utilization	Superior	NN machine buffer targeting and rule based lot dispatching	Semicond.	Simulation	Kim et al. (2020)
116	Flow-shop scheduling	DQN	FFNN	Minimize mean tardiness	Superior	RL dynamically adjust scheduling k1/ k2 values	Flexible flow-shop	Simulation	Heger and Voss (2021)
117	Flow-shop scheduling	NEAT	FFNN	Min. total tardiness and makespan	Superior	GA sets NN topology/ hyper-parameters	Flow-shop	Simulation	Lang et al. (2020)
118	Job-shop scheduling	DQN	FFNN/LSTM	Minimize makespan and total tardiness	Superior	Job allocation and operation sequence agent	Job-shop	Simulation	Lang et al. (2020)
119	Job-shop scheduling	DoubleDQN	FFNN	Min. total weighted tardiness and max. machine utilization	Superior	Two-hierarchy, higher DQN determines temp. goal for lower DQN	Job-shop	Simulation	Luo et al. (2021)
120	Job-shop scheduling	DQN	FFNN	Optimize average slack time	Superior	3-staged release/order, DQN scheduling and allocation structure	Job-shop	Simulation	Zhao et al. (2021)
121	Job-shop scheduling	TRPO	FFNN	Explainability	Similar	RL scheduling and decision tree based control abstraction	Semicond.	Simulation	Kuhnle et al. (2021)
Multi-agent control approaches									
Subtopic	Algo./ NN	Objective	Superiority	Interaction	Training	Application	Simulation	Source	
122	Goal formulation	AC/ FFNN	Maximize profit / utilization	Superior	Market-based negotiation	GARIC framework	Self-evol. mfg. system	Simulation	Shin et al. (2012)
123	Job-shop scheduling	DQN/ FFNN	Min. mean cycle time	Similar	Agent information exchange, global objective	Central DQN module for approximator transfer	Job-shop	Simulation	Dittrich et al. (2020)
124	Job-shop scheduling	SA/ FFNN	Min. mean tardiness	-	Agent information exchange	Simultaneous learning with simulated annealing	Job-shop	Simulation	Hammami et al. (2017)
125	Job-shop scheduling	DQN/ FFNN	Min. throughput time	Superior	Agent state info., global objective	Central DQN module	Matrix production	Simulation	Hofmann et al. (2020)
126	Job-shop scheduling	DQN/ FFNN	Min. WIP, max. util.	Similar	Global objective	While one DQN is trained, others are controlled by heuristics	Semicond.	Simulation	Waschneck et al. (2018)
127	Order dispatching	TD3/ FFNN	Minimum tardiness	Superior	Order bidding mechanism, global objective	Concurrent learning	Job-shop	Simulation	Malus et al. (2020)
128	Re-ordering	DQN/ FFNN	Min. cost and decision time	Superior	None	Iterative curriculum learning	Car after paint buffer	Simulation	Gros et al. (2020)
129	Routing, dispatching, scheduling	PPPO/ Conv. NN	Min. execution time, max. util. efficiency	Superior	Economic bidding/ global objective	-	Matrix production system	Simulation	May et al. (2021)

Table 5.15 Embedded and multi-agent NN approaches in production control

6 Transition - from research gap definition to prototype design

In this chapter, based on the reviews and the refined taxonomy for deep learning based multi-agent systems outlined in Chapters 4 and 5, the detailed research scope of this thesis is established. This addresses the *DSRM* design and development step as proposed by Peffers et al. (2007). The scope and subsequent objectives define the primary design criteria for the artifact. These criteria will direct the development of the prototype to fulfill the established objectives.

In the previous two publications, nearly 2000 publications on deep learning based production systems are examined. From this analysis, a detailed selection is made to categorize the subject matter and identify distinct research streams and gaps. These identified gaps are further detailed in Section 6.1. Subsequently, Section 6.2 delineates the requirements criteria and associated artifact design components, which will be summarized in Section 6.3. The main focus is on developing an adaptive deep learning based control framework and its integration into an adaptive production simulation. Section 6.4 outlines the fundamental requirements for technical implementation.

6.1 Identification of the research gap and structuring research requirements

To address sub-research question S-RQ1, which concentrates on the requirements imposed by complex production systems for control optimization approaches, three significant research gaps were identified in the fields of deep learning and deep reinforcement learning. First, a predominant reliance on the standard DQN algorithm was observed in most approaches, which frequently served as the central decision-making mechanism, representing a singular optimization stage both task-wise and environment-wise. Second, the approaches exhibited limited transferability and generalizability to diverse scenarios, and third, there was an inadequate adaptation of these methods to real-world environments, especially in production control.

In production control, particularly in dispatching, swift decision-making is crucial due to its direct impact on the production workflow, which in turn affects process flows and performance metrics (Lödding, 2019). The challenge is to ensure thorough evaluation of all essential information for optimal decision-making, tailored to both process and order specifics, despite the increased urgency and shorter response times compared to planning activities. This thesis, with its central focus on production control and a special emphasis on order dispatching, addresses this control complexity. To systematically define the research gap and comprehensively address the various aspects of production control and dispatching, the identified gaps are categorized into three key perspectives, structural (1), organizational (2), and algorithmic (3). These perspectives, along with their inherent research gaps, are detailed below and will be further examined during the design specification phase in Section 6.2.

6.1.1 Structural perspective

From a structural perspective (1), the first publication pinpointed a predominant focus within production planning and control in job-shops, particularly in semiconductor manufacturing, as listed in Table 6.1. In addition to comparable deployments in chaku-chaku production, a U-shaped job-shop, and buffer optimization, three approaches were implemented in a matrix setup. The matrix methodologies in Hofmann et al. (2020), Gankin et al. (2021), and May et al. (2021) share similarities, integrating principles of fully flexible line planning and fluid process flows. Nonetheless, May et al. (2021) employed a 3x3 layout with an auxiliary machine and 15 transporters. Each of these transporters possess distinct economic bidding mechanisms, resulting predominantly in short-term or local optimization at the prevailing job-floor level. To proactively counteract missteps, action masking is introduced for process logic learning. To sidestep this issue, May et al. (2021) indicates the feasibility of hybrid control mechanisms. Gankin et al. (2021) integrates 25 workstations, with agents populating orders through the machine input buffer. The action space is formulated based on the count of existing machine and system inputs/ outputs, with action masking once again utilized. During training, all agents undergo simultaneous learning but utilize a central neural network to facilitate knowledge sharing. This raises the research gap concerning adapting the efficacy of the given strategy to alternative environments. Lastly, in Hofmann et al. (2020), a unified neural network is designated for all product agents, enabling each agent to select from a variety of process-machine pairings, determined by the pre-established layout.

Application Count	Production planning		Production control			
	Job-shop	Injection molding	Job-shop	Matrix	Line-buffer	Chaku-chaku line
	7	1	5	3	1	1

Table 6.1 Deep learning and multi-agent based approaches (Panzer et al., 2022; updated 2023)

None of the approaches ventured beyond the foundational concepts of job-shops or matrix production. A distinct research gap emerges from the absence of bundling and consolidating resources and capabilities to foster process synergies and to represent production structures like tool clusters and grouped manufacturing resources.

6.1.2 Organizational perspective

The gap between research and applied practice described above can be further complemented by the organizational perspective (2), particularly regarding agent orchestration and structuring. In the realm of deep learning based optimization, single- and centralized-agents are the most commonly utilized approaches for control and planning, as primarily discussed in the second publication. This encompasses methodologies that have executed distributed decision-making

but rely on a uniform shared policy for controlling all agents. Such an approach facilitates experience sharing but restricts the evolution of specialized attributes and knowledge bases, namely, potential expert statuses and joint process pathways. The predominantly employed centralized decision-making entity obtains comprehensive production data, however, with every alteration in input and output parameters, it necessitates retraining.

This causes an exponentially escalating control complexity for large scale manufacturing systems (Duffie, 1982; Duffie and Piper, 1986), resulting in the *Curse of Dimensionality* (Bellman, 1957). For illustrative purposes, consider a matrix production setup with 10x10 machines. A single transportation agent is assigned the task of dispatching orders, specifically to identify an order and its destination for the next processing step. Ignoring additional machine and order attributes like machine status, current usage, current queue length in the machine's input buffer, estimated processing time, order tardiness, throughput time, priority, and data from other transportation units, the theoretical state input size reaches 10,000 numbers. This number arises from combining 100 machines with 100 potential destinations for each order at one of the machines. However, segmenting this layout into four modules, each with 5x5 machines, plus a top-level module, results in an input of 625 for each agent, which is the results out of 25 machines times 25 potential order destinations. This translates to a theoretical complexity reduction of nearly 94%. This static complexity, a time-independent, structural variable, must be distinguished from dynamic process complexity (Deshmukh et al., 1998; Orfi et al., 2011). The latter results from time-dependent, often unexpected state changes and is therefore also considered in the algorithmic perspective (Wu et al., 2007; Herrera Vidal and Coronado Hernández, 2021). Wiendahl and Scholtissek (1994) also indicate that complexities arise not only from production itself, but also from the manufactured products and are determined by several factors such as the industry, product type and operations. The focus of further consideration is on production complexity, assuming that the products to be manufactured comprise different types and several process steps.

Integrating a production organization that reduces static and dynamic complexity is crucial for the artifact in development. This organization aims to divide the production system into autonomous decision-making units, promoting horizontal and vertical autonomy. These terms draw from the concept of integrating horizontal networks with vertical manufacturing systems (Pérez-Lara et al., 2020). Horizontally, this autonomy allows different product lines and manufacturing areas to operate and decide independently, enabling flexible responses to uncertainties and reducing complexity by managing only relevant information and processes. Vertically, it focuses on structuring the production into distinct layers, from the shop floor to the enterprise level, ensuring their independence. The challenge is to achieve seamless integration among these autonomous units to sustain efficient production flow. However, integrating vertical production layers remains

a significant gap in deep learning based production control research.

Beyond the top organizational model, it is crucial to define the numerous autonomous units in terms of both quantity and interaction dynamics. Recent research still underscores the need for more focused research on decentralized and autonomous decision-making, as noted by Zhou et al. (2021). Additionally, the integration of autonomous agents in multi-layered production structures is underexplored, leaving potentials in agent optimization untapped. This gap impedes the development of optimization drivers that align with different production layers, as indicated in the various variants of the automation pyramid (Meudt et al., 2017). A multi-level organizational structure could emphasize domain-specific experts, who might operate more resiliently and efficiently than generalists. These experts could be adaptable across different scenarios, such as intra-logistics or material dispatching, enhancing system adaptability. However, this concept of role-specific agents employed flexibly within their domains remains an inadequately researched area.

6.1.3 Algorithmic perspective

Prior efforts evaluated production through two perspectives, a structural one emphasizing production layout and resource amalgamation, and an organizational one for multi-layered and multi-agent systems. From an algorithmic perspective, deep learning based production approaches have a tendency towards standard algorithms, especially standard backpropagation (BP) and DQN, as illustrated in Figure 6.1. As indicated in the second publication, more advanced algorithms, such as genetic algorithms (GA) and simulated annealing (SA), capable of evaluating diverse solution alternatives, found application in only 5% of cases, with none in the production control domain. However, these evolutionary algorithms demand substantial training duration and runtime, also referring to the review of optimization methods in Bansal (2005), making them less suitable for real-time applications, such as dispatching tasks. Predominant deep reinforcement learning approaches such as the DQN encounter limitations in large-scale problems. Given the different organization layers, multiple machine and resource groups, and multiple optimization objectives, not only computational efforts increase, but the overall problem solvability decreases. With growing number of sub-tasks and conflicting objectives, decreasing performance values can be forecasted. Consequently, a notable research gap is evident. It is essential that deep reinforcement learning surpasses its prevailing limitations, integrating its strengths with other established methods to discover and harness previously not addressed synergies.

In summary, the confluence of structural, organizational, and algorithmic perspectives is expected to not only reduce control complexity but also to increase its adaptability. Consequently, attributes like system scalability, generalizability, and robustness should be considered in the

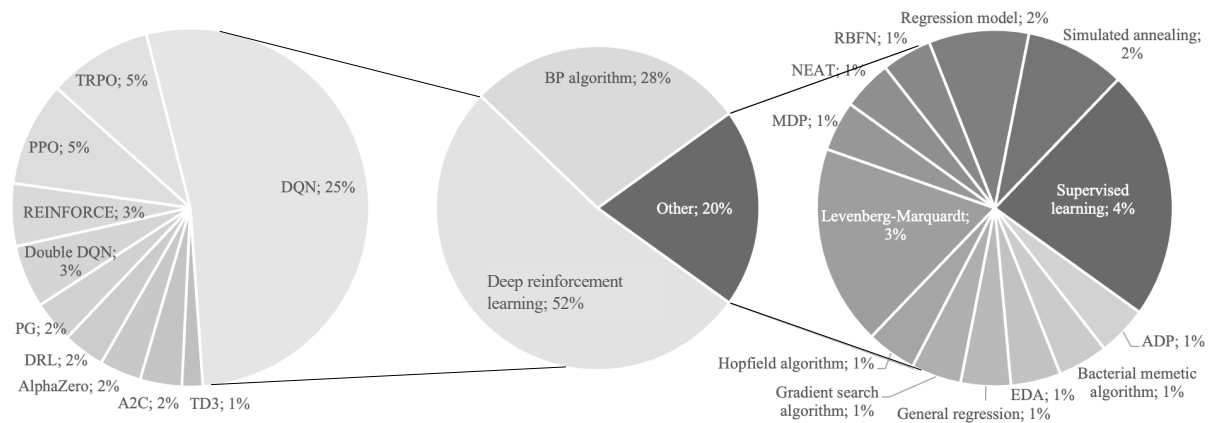


Figure 6.1 Algorithms in deep learning based production, extracted from the first publication

formulation of the design specifications. The following additional research gaps, previously not emphasized, relate to broader control aspects and should be differentiated from the aforementioned perspectives. These should be distinguished from the previously mentioned perspectives. The subsequent bullets briefly introduce these additional pillars that the developing artifact must address.

- Real-world application:** The application of deep learning based control strategies in real-world scenarios presents a foundational challenge that remains largely unexplored. As such, several studies, as evidenced in Baer et al. (2019); Dittrich and Fohlmeister (2020); Gros et al. (2020); Lee et al. (2020), have emphasized the augmentation of simulation complexity as a prospective avenue for research, aiming to bridge this existing disparity. It underscores the importance of understanding and simplifying the real dynamics and constraints of real-world applications when attempting to implement deep learning based control strategies.
- Multi-objective optimization:** Most methodologies concentrate on optimizing a single objective, such as order throughput time or tardiness. Fewer methodologies considered two objectives, especially if they may be in conflict, as demonstrated in May et al. (2021). Yet, a comprehensive approach that assimilates more than two or three production objectives is pending. Similarly, the incorporation of direct customer metrics, such as order priorities and urgency ratios, has yet to be realized. This highlights the potential for advancing the deep learning based production control by adopting a more holistic perspective.
- Multi-agent framework:** The integration of an autonomous mobile robot fleet was previously discussed by Malus et al. (2020), highlighting forthcoming organizational and orchestration challenges. There remains a need for a deep learning based framework that facilitates the integration and coordination of multiple agents, as discussed in earlier sections. This framework should promote concurrent learning, ensure high performance

and process stability, and foster broad dissemination. It should also be versatile across industries, being adaptable for applications beyond production control that are grounded in discrete event principles.

- **Re-use of trained policies:** A primary research focus is minimizing computational effort during neural network training. The aforementioned organizational strategy can, in theory, decrease network dimensions, leading to substantial economies of scale relative to central intelligence. However, methods to simplify the transfer of trained networks to different scenarios are still required. This would circumvent the expensive and potentially risky process of entirely re-training of networks, preventing the omission of present process logic.

The collective research gaps and requirements identified should be methodically converted into design specifications for the proposed artifact. According to the *DSRM* approach, undergoing object-centric iterations is crucial to ensure an optimal alignment between research requirements and artifact design.

6.2 Definition of design specifications and artifact construction

As indicated in Antons and Arlinghaus (2022), centralized control mechanisms face constraints with an expanding problem scope. This is evident in large-scale manufacturing networks as presented in Wang et al. (2016) and in discrete-time production planning as discussed in Pantke et al. (2016). A range of requirements, not exclusively within the domain of *Industry 4.0*, such as just-in-time production or same-day deliveries, have intensified the dynamic decision-making complexity. Managing such complexities is increasingly challenging for a single decision entity (Scholz-Reiter et al., 2011; Mourtzis et al., 2019). Therefore, as also recognized in the review by Antons and Arlinghaus (2022) on decentralized decision-making, there's a need for innovative control models.

To implement an autonomous control framework, it's essential to first define preliminary construction requirements. The artifact aims to explore the central research question of how a data-driven and autonomous production control can be effectively implemented in adaptive production systems. In pursuit of this objective, numerous approaches have underscored the interactive and continuous learning features of deep reinforcement learning, marking it as a notably adaptive technology within the machine learning domain. However, as highlighted in the previous section, there remain unexplored areas requiring in-depth research. These gaps are addressed through the combined insights of the structural (see Section 6.2.1), organizational (see Section 6.2.2), and algorithmic (see Section 6.2.3) design perspectives. Within the framework to be developed, a key focus is placed on reducing optimization complexity and enhancing scenario generalization.

6.2.1 Structural perspective

Earlier studies frequently employed job-shop setups, combined with single-layer and single-agent organizations, and a conventional DQN algorithm. Within such a framework, any modification, based on the adaptability and flexibility criteria, referring to Figure 1.3 (VDI, 2017), necessitates a thorough retraining of the core control model subsequent to structural adjustments. Consequently, it is postulated that a suitable production structure should possess robust re-configuration capacities to mirror structural modifications.

As delineated in the *VDI5201* (see Figure 6.2), the aforementioned re-configuration capabilities encompass system modifications like the addition, removal, or alteration of machine attributes. These elements are pivotal to adaptability strategies and must be incorporated into the deep learning control model. Over time, it becomes evident that shifts in layout also necessitate procedural modifications in the value chain. Although prevailing job-shop or matrix models consider such re-configurations, they frequently entail considerable coordination and routing complexities as well as re-training endeavors, offering less flexibility for structural synergies between shop-floor resources and products.

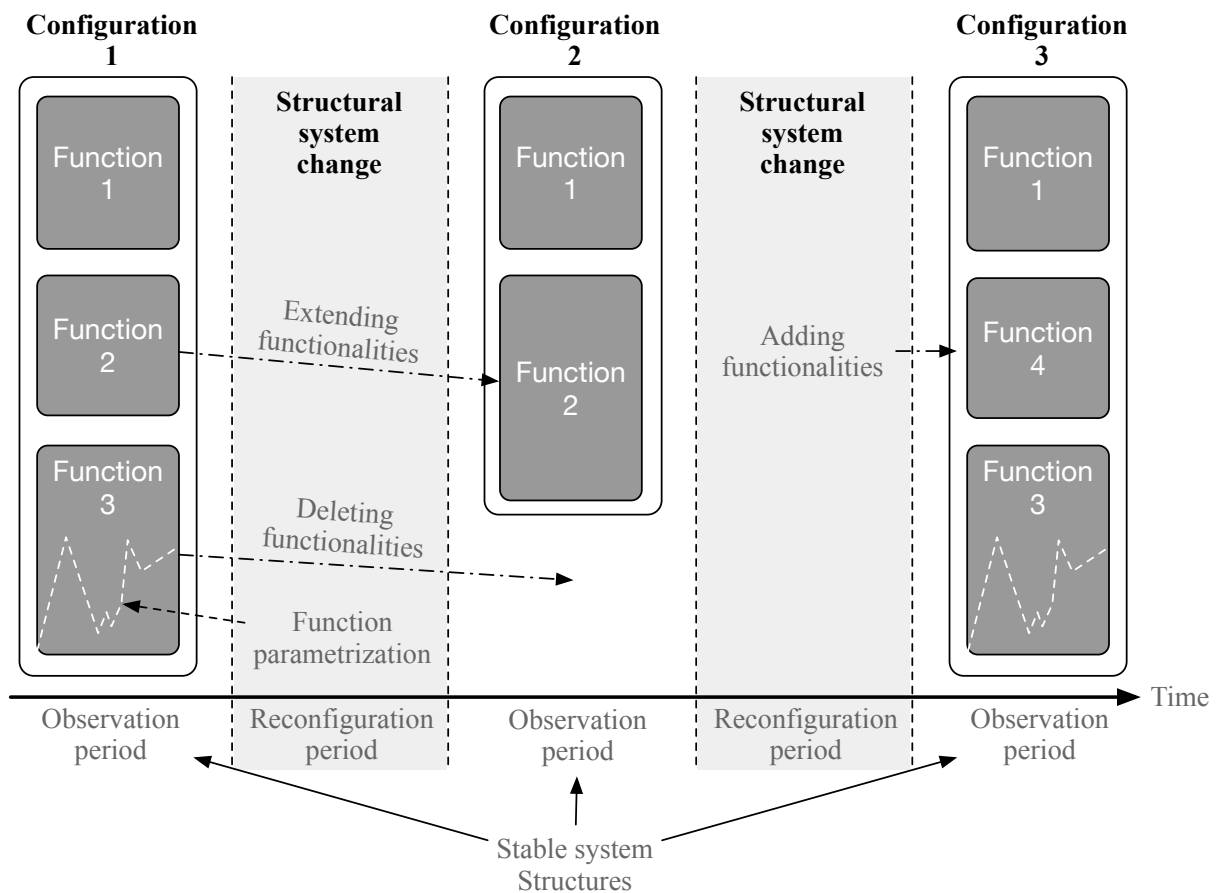


Figure 6.2 System re-configuration capabilities according to *VDI5201* (VDI, 2017)

In contemporary production systems, structural synergies can be achieved through modular resource bundling, tailored to specific product groups or variants. This object-oriented organization, integrating concepts such as center, group, or cell production, enables efficient workshop and flow production integration, a concept detailed in Reichwald and Dietel (1991) and Zäpfel (2000).

This thesis adopts an object-oriented perspective within a flexible manufacturing framework. Here, various groups or islands are interconnected through a standardized modules, facilitated by autonomous logistics systems, including logistics robots, as outlined by Kellner et al. (2020). Historically, modularity is considered from a product- or machine-centric viewpoint, emphasizing maintenance and complexity reduction advantages, as noted by Brunoe et al. (2021). Koren et al. (1999) introduced an early model of re-configurable manufacturing systems utilizing modular machines like CNC machines, aimed at producing entire part families, not just individual components.

This methodology is in line with a variant mix orientation, a concept also illustrated by Nyhuis et al. (2021) through the modular assembly, i.e. in the assembly of the Audi A8. This modular assembly process, characterized by its interconnections via autonomous vehicles, facilitates a discontinuous, non-cycled production flow. For such resource and layout based modularization, already Erixon et al. (1996) and Rogers and Bottaci (1997) highlighted its potential for optimizing the time-to-market metric. Modrak and Soltysova (2023) further highlight the positive impact of process modularity on lead times and process complexity. Meanwhile, Zäpfel (2000) categorizes the concept of manufacturing cells as having higher productivity than job-shops, albeit with reduced flexibility.

Therefore, this thesis progresses to considers the modular concept in the context of a flexible manufacturing system. This includes organizing machine resources through a standardized layout and interface strategy to efficiently represent product variants in production modules and minimize process complexity. Emphasis is placed on the essential nature of flexible module design to ensure production adaptability to varying demands. Enhanced by modular components and plug-and-play features, this flexibility is further amplified in the simulation artifact.

When compared with job-shop and matrix manufacturing, which are favored in deep learning based production, modularization offers distinct advantages. These include higher throughput than job-shop production and enhanced scalability, as summarized in Table 6.2. The modular approach effectively balances process and demand fluctuations by allowing resource and module adjustments. In contrast to the more rigid matrix production, where material flows are less restricted and harder to coordinate, modularization simplifies the object-oriented material flow and path planning within modules. Incorporating object-oriented specifications enables proactive bottleneck analysis, aiding in early detection and prevention of operational issues, like deadlocks.

Concept	Job-Shop Production	Matrix Production	Modular Production
Basic idea	Function-oriented	Matrix shaped layout	Bundled resources
Flexibility	High	Moderate	Flexibility within modules
Scalability	High	High	High
Layout	Functional	Functional and product-centered	Product-centered
Volume/ variety	Low volume, high variety	Moderate volume, moderate variety	Higher volume, variety based on modules
Control complexity	Very high due to order variability	Complex, more predictable	Simplified due to standardized modules
Applications	Custom orders	Standardized orders with customization	Standardized orders with customization

Table 6.2 Comparison of job-shop, matrix and modular production specifications, based on Reichwald and Dietel (1991); Zäpfel (2000); Greschke et al. (2014); Kellner et al. (2020)

Still, in scenarios where production of a standard product are considered, a serial assembly line still the most efficient approach (Reichwald and Dietel, 1991). However, this thesis primarily addresses the need for varying products. These products not only modify manufacturing parameters but also require distinct resources, leading to the necessity for re-configurations.

The modular configuration of production units also facilitates a clear definition of the optimization scope for the algorithmic control methodology. In this context, while information from neighboring modules might be available, it's deemed subsidiary for decision-making within a specific module. This selective integration of information significantly reduces the required data input for neural networks and ensures a more efficient process optimization.

6.2.2 Organizational perspective

The development of an artifact requires a flexible and modular production structure, supported by an equally adaptable organization. This necessitates a restructured organization for clear coordination among and within individual modules. The organization is therefore two-fold. First, inter-modular, which involves coordination among modules, and also intra-modular, focusing on participant arrangement within modules. Both aim to fulfill the objectives of this thesis, to increase performance and adaptability and to reduce the system complexity.

Prior deep learning production approaches lack in strategies for advanced organizations. For example, Mayer et al. (2021) approach considered up to 25 machines but in a single layer. This research gap necessitates a distinct approach to structure and optimize control optimization problems, leading to the relevance of an advanced organization. Therefore, this thesis proposes a semi-heterarchical simulated control framework, as conceptualized by Grassi et al. (2020). The control framework divides the system into a manufacturing layer at the shop-floor level and

multiple distribution layers above (see Figure 6.3). This structure accommodates both vertical and horizontal module configuration, integrating modules across various layers. This reduces complexity in decision making by only integrating relevant information from respective layers or modules, thereby fostering a distinct optimization scope. Moreover, layer-specific optimization strategies can be developed within modules. These can target i.e. the reduction of tool changes at the manufacturing layer, while distribution layers could aim to decrease average moving distances. The designation of the exact terms in Figure 6.3 for the various layers were tailored to the specific requirements of this thesis. However, these terms can alternatively be grounded in established structuring levels, as delineated in Wiendahl et al. (2007) for resource or space view, depending on the use-case. In the following the two-fold differentiation of manufacturing and distribution layers is introduced.

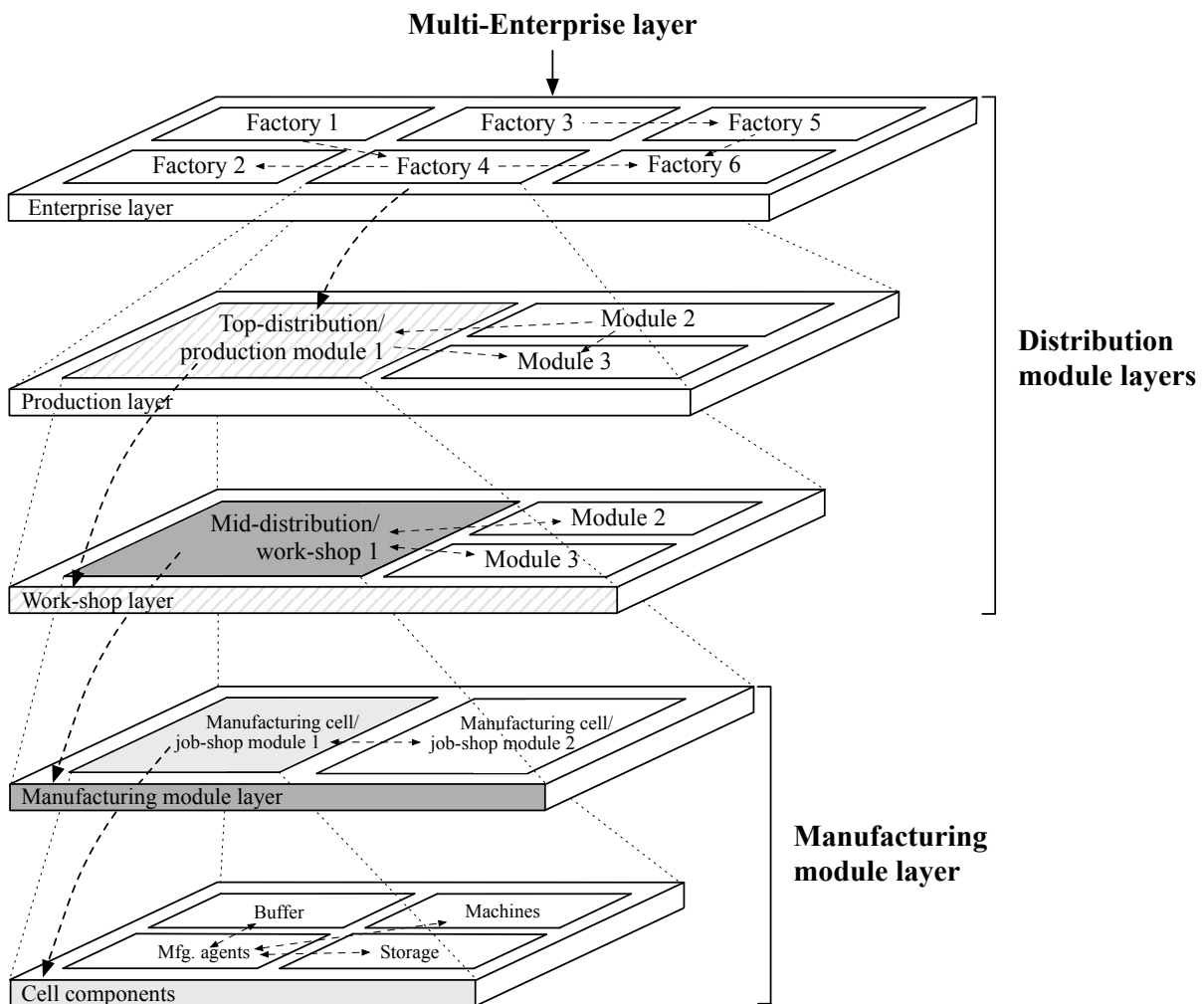


Figure 6.3 Semi-heterarchical control framework, left figure and layers adapted from Henn and Kühnle (1999); Wiendahl et al. (2007); Sallez et al. (2010)

Distribution module layers: In these layers, efficient route planning is crucial to eliminate unnecessary runs and minimize the distance to subsequent orders. Global objectives are more

significant, with systematic balancing needed to distribute orders and avoid high work-in-progress inventories and increased throughput times due to bottlenecks. Lower-level modules can provide aggregated information to simplify data flows and decision-making.

Manufacturing module layers: In contrast, the manufacturing module layers prioritize optimizing short-term indicators at the shop floor level. Priority is given to orders based on multi-criteria relevance, considering factors like customer importance or waiting time. Tactical decisions in process optimization include choosing between loading a machine first or using a product from the output buffer for module transfer.

Besides the layers, the initial publications, while introducing the concept of semi-heterarchical organization, also highlight gaps in our understanding of multi-agent systems and their management of complexity, an area not as extensively explored as single-agent systems. These publications highlight the necessity of addressing instabilities in training arising from the dynamic interactions of multiple agents. Additionally, they highlight the prevalent use of centralized intelligence in 44% of deep learning based multi-agent applications, noting its tendency to require extensive retraining and its difficulty in simultaneously managing objectives, constraints, and interconnections. In contrast, the proposed control framework shall emphasize structural consistency thanks to its high module locality. This design doesn't necessitate retraining for every minor change, and only agents impacted by direct structural modifications need retraining. This not only simplifies decision-making but also reduces the complexity and effort required in training. Effectively countering the *Curse of Dimensionality*, this method reduces overall complexity through its organizational structure.

In summary, this thesis integrates a flexible, semi-heterarchical control framework that integrates both vertical and horizontal structuring for optimal module coordination. This approach balances the complexity of individual layers with the overall system efficiency, enabling differing optimization strategies for each layer to enhance the entire production and distribution process.

6.2.3 Algorithmic perspective

This sub-section begins by outlining general algorithmic specifications, followed by defining the hyper-heuristic design, the reward function, and the training procedure, which are crucial for the performance of the deep learning agents. As outlined in the previous sections, the production system is divided into modules with multiple agents, categorized into semi-heterarchical layers to simplify the control optimization. This structure aids the algorithm during both policy training and operational phases, focusing on a generalized solution for the control task rather than specific scenarios.

The adaptability of the organisation must be supported by the integrated algorithmic control logic. Deep reinforcement learning, as highlighted in the first publication, excels in providing

adaptive decision-making suitable for dynamic environments and enables real-time operations in complex settings. However, deep reinforcement learning can face challenges in complex production environments, such as difficulty in recognizing process relationships and adequately mapping constraints, increasing risks during training. For example, Mayer et al. (2019) note the occurrence of deadlocks in simulations, leading to increased throughput times. This is why a differentiation must be met between the following two pivotal questions for the control and dispatching process.

1. What are suitable control actions?
2. What are optimal control actions?

In the realm of production control, a two-fold approach that combines embedded methodologies and deep learning proves advantageous. Chang et al. (2022) and Nachum et al. (2018) underscore the efficacy of this dual-stage approach, particularly when integrating multiple machine learning algorithms. This method surpasses conventional heuristics in performance but requires training both deep learning components, which could be a limitation if transferability and generalizability is reduced by the necessity to relearn process logics for each application. In contrast, conventional algorithms, like dispatching rules, are more straightforward, targeting specific, well-structured problems with predefined principles. In simple variants, these typically focus on optimizing a single parameter in a set sequence, without the complexities of time-based learning or costly adjustments.

Therefore, this thesis proposes a synergy of deep learning and conventional algorithms. The deep learning component should focus on dynamic control optimization, while conventional algorithms handle static objectives. Specifically, deep reinforcement learning, paired with dispatching rules, can combine adaptability and optimization, maintaining the efficiency of conventional methods. Deep learning orchestrates pre-defined objectives, and conventional algorithms adhere to the given process structures. In essence, deep reinforcement learning is a top-level heuristic for both global and local objectives, supported by low-level dispatching rules for operational aspects.

However, the application of deep reinforcement learning based hyper-heuristics in production control remains under explored, as Dokeroglu et al. (2024) notes. In contrast, non-deep learning hyper-heuristics are efficiently used in scheduling, with earlier publications like Hershauer and Ebert (1975) and Alexander (1987) providing evidence. Current research is more focused on deep learning based heuristics in areas such as the traveling salesman problem (Dantas et al., 2021), vehicle routing (Qin et al., 2021), and combinatorial optimization (Zhang et al., 2022b).

Recent production advancements have applied hyper-heuristics in job-shop and flow shop scheduling, with notable contributions from Zhang and Roy (2019), Lin (2019), Luo et al. (2020), and Song and Lin (2021). Q-learning approaches have also been successful in specific domains

like aero-engine blade manufacturing and semiconductor scheduling (Zhou et al., 2019; Lin et al., 2022). The growing interest in hyper-heuristics, especially within deep learning and deep reinforcement learning, is discussed by Wassim (2023), indicating their benefits for real-time decision-making and efficiency.

To ensure broad generalizability in this thesis, implementing hyper-heuristic operation in production control, as elaborated in later publications, requires standardized input data and a fixed action space for robust learning and operation. In contrast, advanced problem-centric heuristics demand precise calibration for each unique application. Additionally, selecting appropriate dispatching rules for the hyper-heuristic is crucial for consistent performance, limiting the deep learning agent to at most sub-optimal choices. The choice of dispatching rules as an action space also sets the optimization scope and can be tailored individually. As Kuhnle (2020) suggests, the state space should correspond with the optimization parameters, including the correlated dispatching rules. Furthermore, configuring the reward function significantly impacts the development of the control policy, and the training approach for the neural network must be carefully planned, as briefly outlined in the following sections.

Reward function design: In the field of production control, addressing multi-objective optimization spaces remains a substantial challenge. Most existing research concentrates on one or two objective variables, revealing untapped potential, such as integrating financial parameters through deep learning. Conventional methods often limit themselves to a narrow set of variables. However, current industry demands require a comprehensive evaluation of various performance metrics, including process efficiency, customer interactions, and business strategies. Simply optimizing specific technical indicators for individual cases is inadequate in the current customer-centric market landscape. A more effective approach is to develop a holistic and adaptable reward system. This system should be capable of assessing and improving multiple indicators concurrently, adhering to hyper-heuristic principles. Such a system would enable a more nuanced and flexible approach to optimizing production processes, aligning them more effectively with broader business objectives and market demands.

Moon and Jeong (2021) highlight the need for such an innovative and flexible reward function that can adapt to changing production indicators. This function should not only be flexible enough to incorporate specific goals but also dynamic enough to adjust to evolving production conditions. It involves pinpointing key technical, job-related, and financial parameters and integrating them effectively. Differentiating between global and local objectives and establishing various reward dimensions is essential. Furthermore, as Wiendahl (1997) pointed out, which will be explored more in Section 6.4, subsequent process-related evaluations will rely on standard indicators such as throughput times, on-time delivery, and inventory levels.

Training procedure: The standardization and preservation of neural networks within a network stack for reuse remains an unexplored area in production control. However, in a modular design production system, there may be modules that allow agents to utilize experiences from other modules. We can categorize the reuse of process knowledge into three types, within identical environments or modules (1), to similar modules (2), and restricted transfer due to vastly different structures (3). Agents can thereby benefit from faster re-training and adaptability to their specific roles by continuing training with transferred neural networks. This enables agents to develop specific skills and share experiences, improving performance across system modules by minimizing training efforts.

In the later training phases, all agents are trained concurrently, with the scoped module structure and decentralized decision-making process inherently preventing instabilities. For the general neural network design, similar to Mayer et al. (2021), it was recognized that the neural parameters were less relevant than the defined state input and reward functions. Still, a grid parameter search was conducted, exploring chosen combinations systematically. Due to the achieved complexity and therefore also state input and output dimensions, this grid search is advantageous compared to a random search, because it can be iteratively improved by previous findings (Paul et al., 2019).

The training process utilizes a modified ϵ -greedy strategy, effectively balancing exploration and exploitation across different modules. This approach differs from certain methods as it does not require mimic learning. Instead, it concentrates on optimization rather than process learning. In the initial stages, a significant number of random actions are chosen to facilitate a balanced action exploration and steady learning progression. As illustrated in Figure 6.4, the frequency of these random actions is progressively reduced over time. This strategy is adapted from other successful deep reinforcement learning approaches, particularly from Mnih et al. (2015).

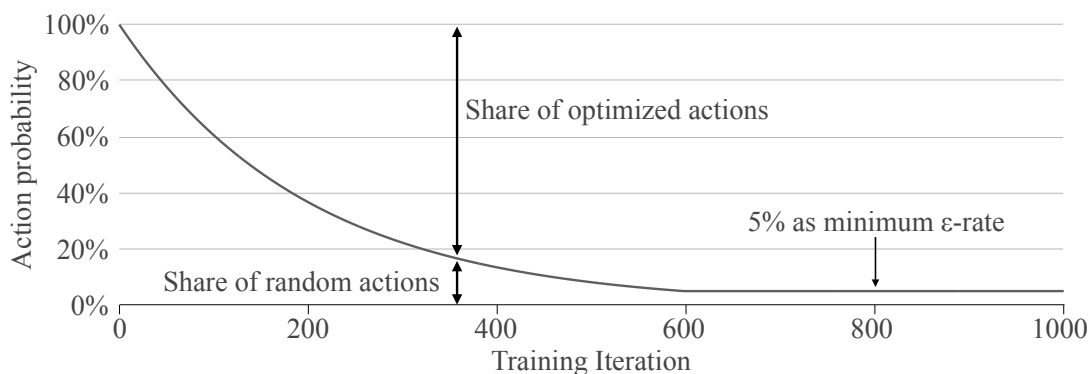


Figure 6.4 Decreasing heuristics share during initial learning process, $\epsilon = 0,995$

6.3 Artifact design summary

As outlined before, the artifact design is explored through three perspectives, structural, organizational, and algorithmic, each contributing to system performance and adaptability, as summarized in Figure 6.5. The structural and organizational aspects focus on systemic adaptability and scalability through a modular, semi-heterarchical multi-level, and multi-agent structure. This approach enables the system to adapt and scale efficiently across various resources and organizational layers. The algorithmic perspective, on the other hand, concentrates on enhancing intra-agent policies and adaptability, improving local system performance and optimization. Global variables further optimize overall system performance. These perspectives collectively offer a comprehensive framework for addressing the prevalent challenges and the formulated research questions effectively.

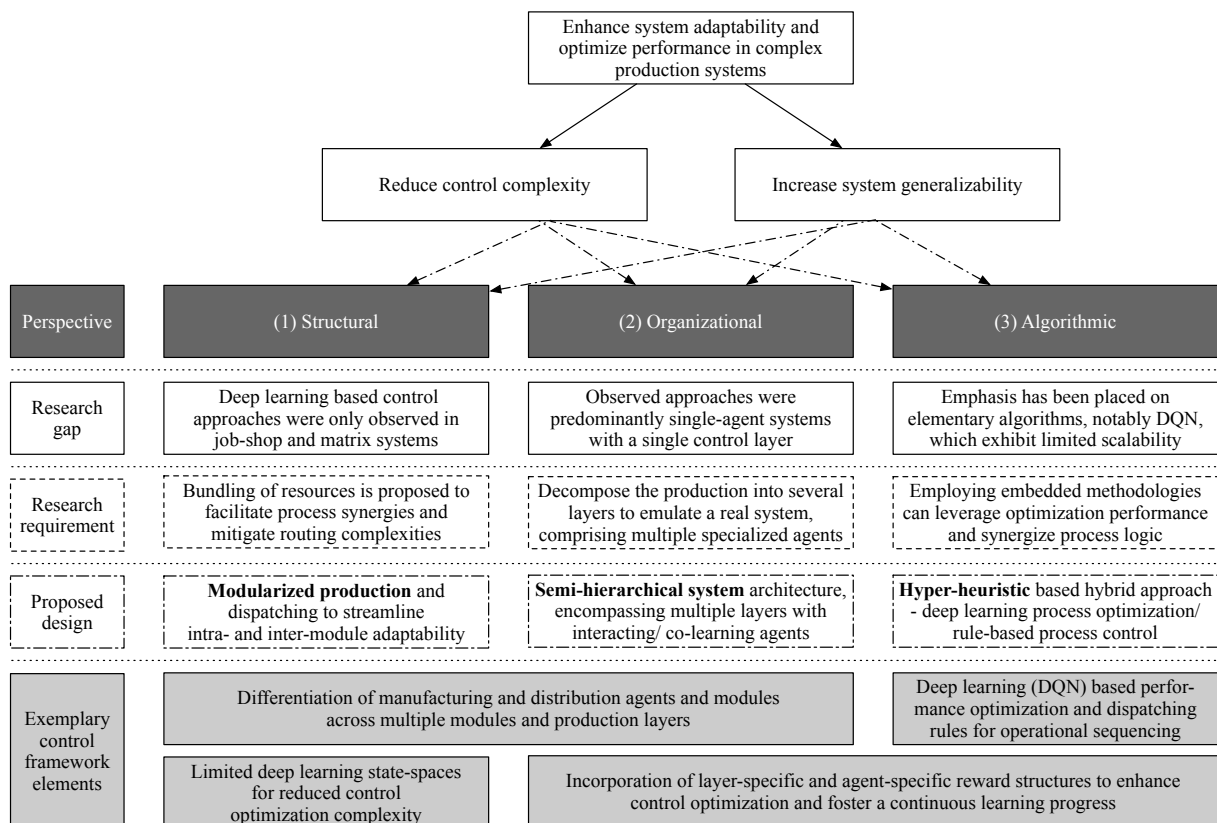


Figure 6.5 Summarized research gap, requirements, and specifications

In defining the research artifact, key aspects for its development were identified, aligning with the planning and control design framework by Bendul and Blunck (2019), which includes elements illustrated in Figure 6.6 on the right. Thereby, the artifact comprises two main components: the *SimPy* simulation and the *Control Brain (CoBra)* framework. These elements embody the technical methodologies and form the core of the artifact's structure.

SimPy simulation A previously developed *SimPy* simulation corresponds to the first two levels of system planning, as shown in Figure 6.6. It creates a model of the production environment, simulating internal operations, conditions, and agent states. This simulation allows for flexible plant design with arbitrary paths within modules. The operational complexity (see number 1 in Figure 6.6) can be customized, featuring multiple redundant resources both across and within modules, enabling diverse production paths. A matrix layout without module borders offers greater flexibility, though modules help reduce short-sighted behaviors by providing clear process boundaries. Structural complexity (2), while inherently limited, depends on the number of intersecting process paths, which modularization can manage and restrict.

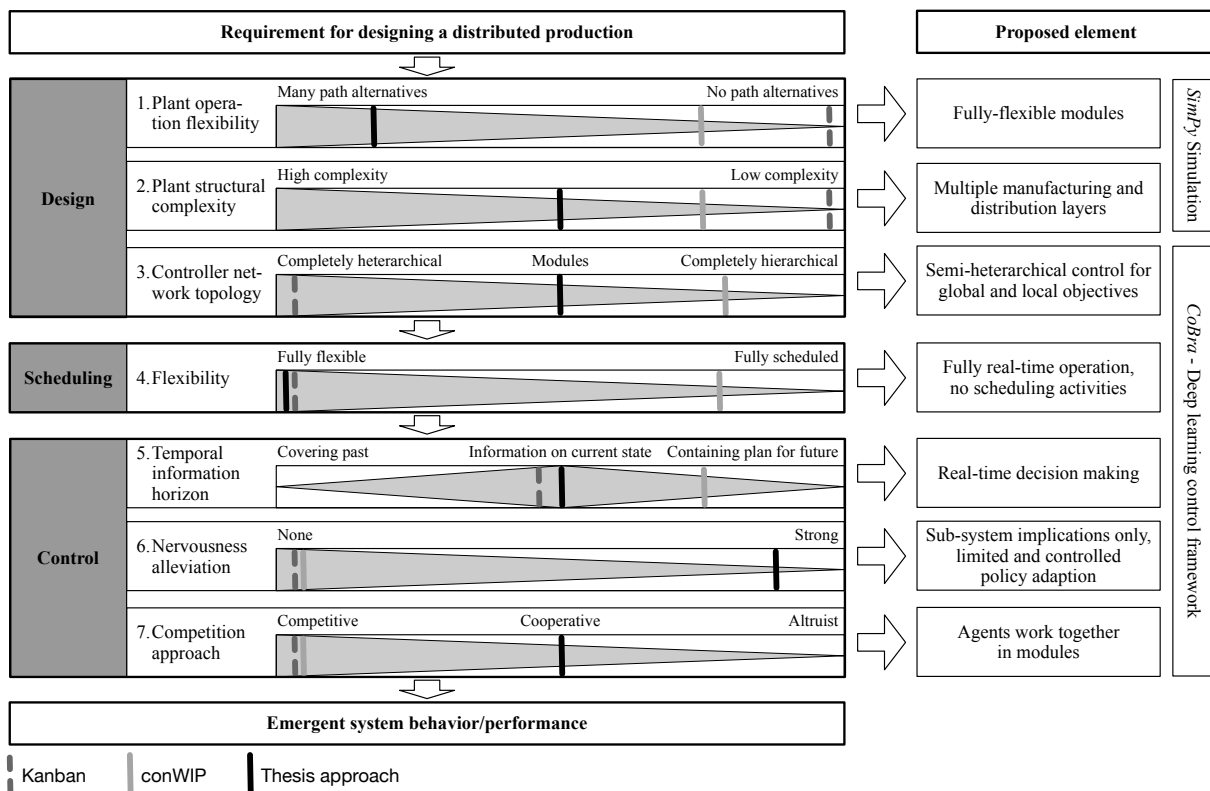


Figure 6.6 Design and control framework with proposed artifact elements, Bendul and Blunck (2019)

CoBra control framework The *CoBra* control framework, developed in the next publications, plays a central role in integrating deep learning based control capabilities. It plays a pivotal role in exchanging data and executing optimized dispatching actions with the simulation. This interaction is crucial for the deep reinforcement learning cycle, significantly influencing the development of the discrete event based simulation. *CoBra* covers rows three to seven in Figure 6.6, highlighting its deep learning focus. It features a semi-heterarchical controller typology, merging hierarchical and heterarchical methods (3). The framework is designed for flexibility and real-time control responses (4), while leveraging past made experiences without depending

on future data (5). It manages nervousness by allocating it only within modules and in response to changes in system states, thus avoiding effects across boundaries under limiting policy adaptation after each step (6). In terms of competition, the agents collaborate without competing, focusing on iterative joint processing (7).

In summary, the simulation generates process data for training, operating, and evaluating the *CoBra* framework, aligning with the DSRM to facilitate iterative design and development. The *CoBra* control logic is central to this structure, acting as a key decision-making entity. It not only addresses research questions but also serves as an interdisciplinary interface, vital for the later implementation of deep learning algorithms.

6.4 Technical implementation - low-barrier artifact design

The previous sections outlined the research foundation of the artifact, but there is the technical aspect yet to be discussed. To effectively share this research and enable the application of its findings, a simulation design is needed that is both accessible and applicable. To this end, the *CoBra* framework, as the central deep learning based control artifact of this thesis, must be integrated into the python based *SimPy* environment, previously introduced in Section 3.3.1. The *SimPy* framework is used for this integration, known for its processes, events, and resources. Processes in *SimPy* simulate activities like logistic robots moving from one point to another. Events represent system changes, such as completing a task, which trigger other processes. Resources are finite elements, like machines in use. In the following paragraphs, the basics of *SimPy*, the machine learning library *Keras*, and simulated technical indicators for performance evaluation are presented.

SimPy is chosen for its capabilities in handling processes, events, and resources. It effectively simulates activities such as logistic robots moving between points (processes), system changes like task completion (events), and finite elements like machines (resources). The ability of *SimPy* to model complex production systems is well-documented, as demonstrated by Mönch et al. (2013) or Kuhnle et al. (2019). *SimPy* accommodates variables like order times, setup durations, overdue orders, varying processing times, and machine breakdowns, as well as planning factors like adjustable order release levels for specific orders, which can lead to temporary system congestion and work-in-progress increases. These elements can be dynamically adjusted or set as static, for instance, through a fixed order release schedule.

Keras. The simulation is linked to the deep learning model underpinning it. To ensure this alignment, *Keras*, a versatile and open-source Python library for developing and training deep learning models, is utilized. *Keras* is distinguished by its compatibility with established deep

learning libraries like TensorFlow and its efficiency on both CPUs and GPUs, meeting diverse user requirements. It features pre-configured neural network layers, such as dense layers, and offers a variety of optimizers like *SGD* and *Adam*. These can be extensively customized through iterative optimization loops, providing users with the flexibility to add or modify layers and fine-tune parameters (Chollet, 2015). The adaptability and ease of use of *Keras* enhance the implementation and testing processes, enabling comprehensive customization of the simulation model.

Quantitative performance indicator evaluation The evaluation of a developed artifact in simulation research is a critical process. It involves assessing various performance indicators to ensure that the research objectives are satisfactorily met. This empirical evaluation is not merely a final step but a continuous, iterative process integral to the design cycle, as noted by Hevner (2007). Key logistics-oriented performance criteria include utilization rates, delivery times, schedule adherence, and inventory levels, as outlined by Wiendahl (1997) and Lödging (2016). Particularly critical is the evaluation of tardiness and adherence to customer-related schedules, a major criterion for delivery reliability (Mayer et al., 2016). For the later calculations, the throughput is considered within the system scope and is calculated according to Nyhuis and Wiendahl (2012), as illustrated in Figure 6.7.

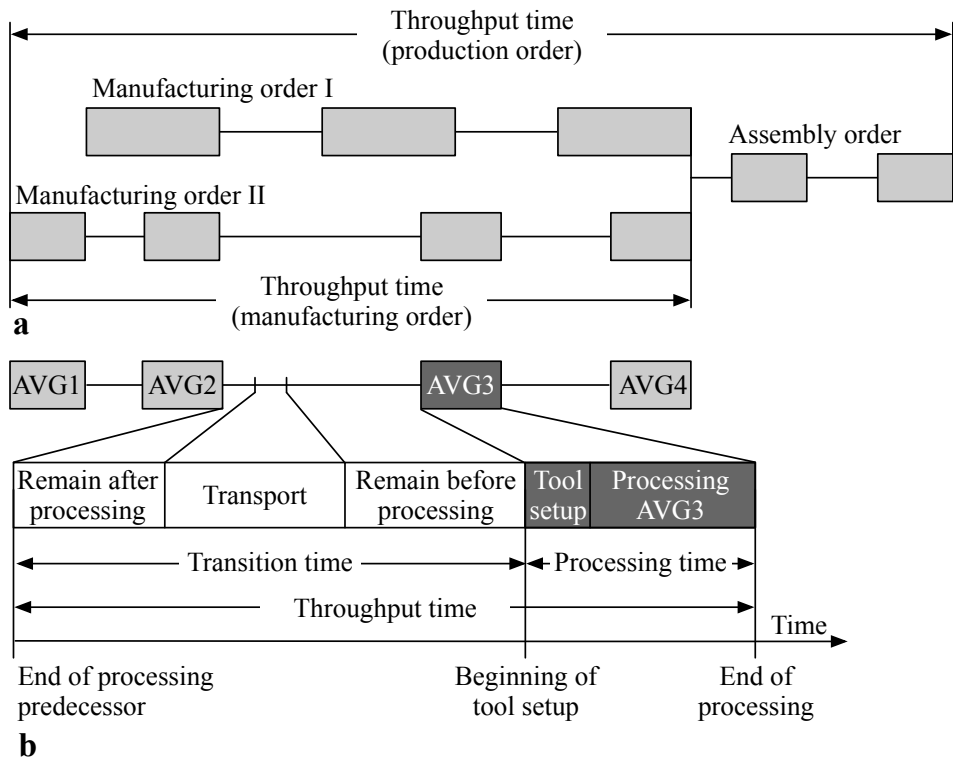


Figure 6.7 Breakdown of throughput time components as per Nyhuis and Wiendahl (2012): (a) order- and (b) process step-related processing times

In the simulation, throughput time is standardized, starting from order release and ending when the order exits the simulated environment. It includes tool setups as necessary, with processing times varying according to materials and specifications. Non-processing time is allocated to machine and module buffers, as well as storage slots, while transportation time covers all movements. Tardiness is calculated by comparing the deviation between the scheduled finishing time, d_i , a random value within a predefined range, and the actual completion time, C_i . Each order is weighted equally in this calculation, as detailed in Equation 6.1 (Brucker, 2007).

$$T_{td,mean} = \frac{1}{n} \sum_{i=1}^n \max(0, C_i - d_i) \quad (6.1)$$

Mean tardiness, a metric representing the average delay in fulfilling orders, serves as a pivotal benchmark for evaluation. This indicator, in conjunction with throughput time, is frequently utilized to evaluate system performance in deep learning based production publications. The duration of production processes and the precision of schedule adherence are critical factors that substantially influence the total delivery time. The latter, in turn, exerts a significant impact on the interplay between pricing and delivery timelines, which is especially relevant for the fifth publication in Chapter 9. Thereby, Lödging (2016) demonstrates the benefits of shorter delivery times in enhancing customer attractiveness and facilitating express manufacturing services for a printed circuit board manufacturing company. Specifically, in the production of rigid-flex printed circuit boards, decreasing the turnaround time from 15 days to 5 days can result in a prices increase of up to 200% . Notably, even a modest reduction in production time, from 15 to 13 days, can lead to an increase of roughly 40% in product costs. This example elucidates the profound influence on financial potentials that optimized production and delivery schedules can facilitate for manufacturing companies. Certainly, it's important to recognize that production is just one aspect of a broader operational chain. Effective management of procurement and development processes, and other order process stages must also be integrated, as these stages are crucial components in the overall workflow (Lödging, 2016) .

The criteria, adaptable to specific use-cases, focus primarily on technical and quantitative parameters, but still require a thorough analysis of qualitative factors pertinent to the research objectives. For instance, production flexibility will be assessed by simulating fluctuations in incoming orders to reflect unpredictable production volumes and customer demands (Zhang et al., 2003). This involves the capability to manage varying throughput volumes and diverse production shares within the product mix (Sethi and Sethi, 1990; Boyer, 2000). Beyond flexibility, adaptability is defined as the system's ability to re-configure and thereby alter its properties (Bordoloi et al., 2009; VDI, 2017). Additionally, generalizability and scalability are key sub-objectives. Generalizability refers to the potential to apply gathered insights and transfer the developed artifact to different, untested environments and scenarios (Lee and Baskerville, 2003).

Scalability involves the system's capacity to handle increased workloads through the addition of process-related resources (Bondi, 2000), addressing both the expansion and efficiency of the system. For this purpose, the following publications will clarify how the aforementioned qualitative and quantitative objectives can be effectively captured using auxiliary variables and thus assessed through quantitative measures.

7 Publication 3

Designing an adaptive and deep learning based production control framework for modular production systems

Marcel Panzer^{1a} and Norbert Gronau^a

^a *Chair of Business Informatics, Processes and Systems, University of Potsdam,
Karl-Marx-Street 67, 14482 Potsdam, Germany*

ABSTRACT

In today's rapidly changing production landscape with increasingly complex manufacturing processes and shortening product life cycles, a company's competitiveness depends on its ability to design flexible and robust production processes. On the shop-floor, in particular the production control plays a crucial role to cope with disruptions and maintain system stability and resilience. To address challenges arising from volatile sales markets or other factors, deep learning algorithms were increasingly applied in production to facilitate fast-paced operations. In particular deep reinforcement learning frequently surpassed conventional and intelligent approaches in terms of performance and computational efficiency and revealed high levels of control adaptability. However, existing approaches were often limited in scope and scenario-specific, which hinders a seamless transition to other control optimization problems. In this paper, we propose a flexible framework that integrates a deep learning based hyper-heuristic into modular production to optimize pre-defined performance indicators. The framework deploys a module recognition and agent experience sharing, enabling a fast initiation of multi-level production systems as well as robust control strategies. To minimize computational and re-training efforts, a stack of trained policies is utilized to facilitate an efficient re-use of previously trained agents. Benchmark results reveal that our approach outperforms conventional rules in terms of multi-objective optimization. The simulation framework further encourages research in deep-learning-based control approaches to leverage explainability.

Keywords

Modular production, Production control, Deep learning, Reinforcement Learning, Simulation framework, Explainability

¹Corresponding author

Submitted to the Journal of Intelligent Manufacturing on 12 May 2023, accepted on 12 October 2023.

7.1 Introduction

Nowadays, companies must respond quickly to both, internal and external disruptions and adapt their processes to remain competitive and maintain operational profitability. In this context, the trend towards mass customization and shortening development cycles pose significant challenges for today's production systems. By the same measure, they must be capable of operating in highly uncertain market conditions while satisfying many (conflicting) customer and process related objectives, in the shortest possible time (Schmidt and Nyhuis, 2021). In this regard, the use of advanced Industry 4.0 technologies, including the *Internet of Things* and artificial intelligence, is crucial to enable a data-driven process optimization and to cope with the increasingly complex requirements (Kang et al., 2020; Parente et al., 2020; Kapoor et al., 2021).

In recent years, simulation-based and combined hardware-in-the-loop approaches were implemented to facilitate a seamless transfer of research artifacts into practice in a low-risk environment. Especially in production planning and control, single- and multi-agent approaches were implemented to manage production complexity, each with different pre-defined agent-environment interactions (Babiceanu and Chen, 2006; Gronauer and Diepold, 2021). Regarding the production organization, modular systems demonstrated particular benefits, as they allocate the overall optimization task to accessible and reactive groups of agents (Sallez et al., 2010; Groover, 2019). Modular production systems are noted for their flexibility, scalability and adaptability. Unlike conventional production systems, which are often linear and inflexible, modules can be easily inserted, removed or re-positioned, enabling a swift response to market changes or adaptation requirements. Through task decomposition and distributing complexity across foundational modules, we can expedite the implementation of intelligent control methods, as demonstrated in Rojas and Rauch (2019); Zhou et al. (2022); Tao et al. (2023). The distributed modules possess pre-defined process capabilities, thus ensuring a high density of coordination within and between modules which increases responsiveness and robustness of the system through parallel processing and intelligent agent orchestration (Groover, 2019; Herrera et al., 2020; Sallez et al., 2010; Buckhorst et al., 2022). However, the multitude of interactions and parallel operational activities in multi-agent systems still pose a significant challenge for a coordinated control of shop-floor activities. The production control must handle a constantly growing number of data sources and information flows, to make situation-specific, optimal decisions and leverage process potentials.

To handle such complex optimization problems, machine learning techniques, particularly deep learning algorithms, were increasingly applied in production research (Kang et al., 2020; Samsonov et al., 2021; Oluyisola et al., 2022). Given their ability to capture complex non-linear relationships and to process large amounts of data in real-time for multi-objective optimization, the need for complex and rigid models is prevented. This enables the targeting of both local and

global process variables and facilitates a continuous improvement process by leveraging both, machine and human-related system resources (Cadavid et al., 2019; Zhang et al., 2019; Kang et al., 2020). However, despite the potential benefits, the exploitation of machine learning in production control is not yet fully addressed, as its adoption is rather concentrated on the field of Big Data or other related disciplines (Liao et al., 2017; Cadavid et al., 2019). Nevertheless, it becomes clear, notably in Weichert et al. (2019); Zhou et al. (2022), that due to the versatility of deep learning approaches, a multitude of practical control optimization problems can be addressed, in which fast decision-making contributes significantly to maintain process stability (Bueno et al., 2020; Zhang and Huang, 1995; Garetti and Taisch, 1999). However, the practical integration of a machine learning algorithm must be conducted in an objective-specific manner and requires a dedicated deployment to balance the increasing process and model complexities and to ensure appropriate decisions and a high process reliability (Weichert et al., 2019).

In recent years, in particular, deep reinforcement learning (RL) algorithms demonstrated superior efficacy against other conventional or machine learning based benchmarks (Zhou et al., 2022). In contrast to meta-heuristics, which serve as search process optimizers, deep RL offers significantly improved real-time capabilities, performance metrics, and a higher interpretability (Zhang et al., 2022; Grumbach et al., 2022; Kallestad et al., 2023). Based on collected sensor information, deep RL is capable to make online data-driven decisions and enables a responsive and adaptive control design that addresses the challenges of volatile manufacturing environments. Due to the direct agent-environment interaction, deep RL can generalize and leverage the obtained process knowledge to enhance production stability and performance (Arunraj and Ahrens, 2015; Mehlig, 2021). Even though the application of deep RL demonstrated outstanding performances in various production fields, multi-agent based production control approaches were less considered, especially in matrix- or modular-shaped production systems, as reviewed and analyzed in Panzer and Bender (2022) and Panzer et al. (2022). Although control approaches of Gankin et al. (2021), Mayer et al. (2021), and May et al. (2021) already indicated robust and performant multi-agent control policies, current research lacks an adaptive approach that can address various production scenarios and offers a high transferability to similar practical problems.

To harness the benefits of deep learning and a multi-agent-based production organization, this paper introduces a novel control framework that facilitates the flexible adaptation of modular production systems. By employing a hyper-heuristic control concept for varying production objectives, our approach seeks to improve production performance and adaptability. Owing to the hyper-heuristic based algorithmic approach, the deep RL based top-level decision entity focuses on selecting low-level heuristics, thereby avoiding the adoption of deficient system policies or erroneous actions. The proposed control framework is incorporated into a flexible simulation, which accommodates a wide range of production scenarios and enables the optimization of individual performance metrics. The simulation adheres to a modular principle, which

decomposes the overall production task complexity into manageable fragments, resembling the production system in its modular structure. Additionally, we distinguish between manufacturing and distribution modules, that are responsible for shop-floor and intra-logistics activities, respectively. By synergistically combining the concepts of modular production and hyper-heuristics, we harness the strengths of both domains. This fusion allows us to achieve a dual-fold reduction, both systemically and algorithmically, in optimization complexity.

The embedded deep learning based decision-making process leverages a module recognition and agent experience-sharing method that facilitates the rapid creation and initiation of multi-level production systems. The framework further aspires to progressively reduce computational efforts for neural network training through the integration of a batch of pre-trained policies.

The remainder of the paper is organized as follows. In the next section, the basics of prevailing simulation frameworks, deep RL, and multi-agent based production control are outlined and the research objective is specified. Then, the conceptual design and artifact requirements are defined and simulation results are presented and evaluated. Finally, the paper concludes with a discussion of the framework and a conclusion that synthesizes the main findings.

7.2 Related work

This section specifically focuses on the basics of discrete-event simulations (DES) and deep learning methods, which were increasingly applied for a wide range of production planning and control tasks over recent years. A DES constitutes an essential link between the theoretical concepts of adaptive and deep learning based production control concepts and their simulated and practical implementation in modular production systems. Without a robust foundation in DES, a framework would lack to emulate the dynamics and complexity of modular production systems. Therefore, the following DES subsection provides an in-depth analysis of the simulation foundations that are essential for operationalizing our approach.

Subsequently, the key concepts of deep RL as well as hyper-heuristics are introduced, which serve as core elements of the later developed artifact. These concepts are expected to provide significant performance improvements in production optimization through their continuous learning behavior and adaptability, enabling automated and data-driven optimization of production decisions. The discussion continues with a review of the current state of research, specifically in the context of integrating production control and deep RL.

Building upon the dual research gap from a DES and algorithmic perspectives, we present the problem formulation in which we state the specific problem of our research approach. Thereby, we outline the objectives of our approach for an adaptive and deep learning based modular production control framework.

7.2.1 Discrete-event based production simulation

A DES describes the development of a system based on pre-defined events and their chronological sequencing as discrete occurrences that affect the system state (Law, 2007). In DES, events are captured at discrete time points, and system variables are modified accordingly, allowing for incremental and traceable progression of the simulated system over time. By incorporating operational resources, such as machines or labor resources, system states, and process flows, a production system can be replicated, enabling the analysis of key performance metrics and identification of operational optimization potentials (Fowler et al., 2015; Jeon and Kim, 2016; Mayer et al., 2021). Such analysis may include bottleneck resource evaluation, optimal machine arrangement, or work efficiency assessment for specific system resources. Notably, this approach facilitates the uncritical testing of prototype solutions, which can be further examined in an intermediate hardware-in-the-loop approach until reaching satisfying real-world results.

However, simulation techniques are often applicable only for limited periods of time due to their difficulty and specificity of implementation (Neto et al., 2020). Thereby, Mourtzis (2020) further emphasizes the challenges of integrating artificial intelligence into these simulations. Although the DES approaches can be manifold, the number and type of information sources necessitate dedicated control implementation for a data-driven and optimal decision-making. To address these hurdles, the following simulation frameworks aim to bridge the gap between advanced planning and control theory and its practical application. These frameworks are also listed in Table 7.1.

Apart from production planning and control practices or similar production disciplines, other simulation approaches already delved into the creation of intelligent planning and control frameworks. Notable sectors and problems include vehicle routing (Nazari et al., 2018), energy supply chain management (Chen et al., 2021), or computational fluid dynamics (Pawar and Maulik, 2021).

In the realm of production planning and control, current research is primarily focused on simulation frameworks designed for planning purposes. A mixed-integer linear programming (MILP) framework for the scheduling of mining operations was proposed by Manriquez et al. (2020). An intelligent multi-agent *SwarmFabSim* framework was proposed by Umlauf et al. (2022), that deploys a swarm intelligence algorithm. Other DES scheduling approaches adopted quantum annealing (Venturelli et al., 2015), cuckoo search optimization (Phanden et al., 2019), or genetic algorithms (Fumagalli et al., 2018) to increase the applicability of the respective framework. A recurring feature of such approaches is potentially extended computation times, often attributed to meta-heuristic solution methods. Other established frameworks that use deep RL for production scheduling, like the *JSSEnv* (Tassel et al., 2021) or *Schlably* framework (Waubert De Puiseau et al., 2023), primarily focus on job-shop scheduling or order release and

sequencing (Samsonov et al., 2022).

Application	Specific application	Algorithm	Author
Other applications	Vehicle routing	RL	Nazari et al. (2018)
	Modular System design	-	Farsi et al. (2019)
	Energy supply chain	Genetic algorithm	Chen et al. (2021)
	Computational fluid dynamics	Deep RL	Pawar and Maulik (2021)
	Algorithmic trading	Deep RL	Shavandi and Khedmati (2022)
	Predictive maintenance	Deep RL	Rodríguez et al. (2022)
Job-shop scheduling (non-RL-based)	Predictive maintenance	Deep RL	Su et al. (2022)
	General scheduling	Cuckoo search	Phanden et al. (2019)
	Mining scheduling	MILP	Manriquez et al. (2020)
	General scheduling	Quantum annealing	Venturelli et al. (2015)
	General scheduling	Genetic algorithms	Fumagalli et al. (2018)
Job-shop scheduling (RL-based)	Semiconductor scheduling	Swarm intelligence	Umlauf et al. (2022)
	General <i>JSEm</i> framework	Deep RL	Tassel et al. (2021)
	General scheduling	Deep RL	Samsonov et al. (2022)
Job-shop control (single-agent)	General <i>Schlably</i> framework	Deep RL	Waubert De Puiseau et al. (2023)
	Multi-agent job-shop	Deep RL	Liu et al. (2022)
	<i>SimPyRLFab</i> semiconductor dispatching	Deep RL	Kuhnle et al. (2019)
Job-shop control (multi-agent)	General <i>L2D</i> framework	Deep RL	Zhang et al. (2020)
	Modular dispatching	Deep RL	Our Framework

Table 7.1 Overview of prevailing simulation frameworks

For dedicated production control problems, two approaches dealt with specific single-agent DES implementations, which analyze the impact of stochastic and unpredictable variables. Zhang et al. (2020) implemented the single-agent *L2D* framework by deploying a combined deep RL and disjunctive graph representation to learn priority dispatching rules in a 3x3 job-shop. Kuhnle et al. (2019) implemented the deep learning based production control framework *SimPyRLFab*, thereby considering the prevailing semiconductor process pre-requisites. In such DES frameworks, predefined and product-specific process sequences, machine failures, or other non-deterministic events can be triggered, and their effects on production participants, such as degrading production resources, warehouse inventories, or line effects, can be investigated (Law, 2007). Additionally, systemic and agent-centered relationships can be analyzed based on their organization and interaction.

Whereas Liu et al. (2022) implemented an advanced hierarchical multi-agent scheduling framework, distinguishing tasks between routing and inter-machine scheduling, other (DES) frameworks predominantly focused on plain scheduling or single-agent production control. Notably, these simulations did not account for multiple production layers and consistently operated on a singular level.

Conclusively, from the DES viewpoint, there is a need for a framework that facilitates the conceptualization and simulation of a multi-layered modular production system. Within this framework, the optimization task is governed by a semi-heterarchical framework, facilitating the attainment of both global and local objectives by multiple agents. The modular structure allows for a versatile modification and change of system properties by adding or removing modules

to meet current requirements or to cope with dynamic processes (Buckhorst et al., 2022). The semi-heterarchical backbone provides a high integration capability of potential scenarios through user-defined modules within a hierarchical organization. In parallel, the system is more robust due to the structured allocation of competencies and parallel processing, as in heterarchical systems (Valckenaers et al., 1994; Groover, 2019; Derigent et al., 2021).

A significant challenge is that current heuristics have limitations in optimizing multiple performance measures (Grabot and Geneste, 1994) and often exhibit minimal global coordination when processing information within local entities (Uzsoy et al., 1993; Holthaus and Rajendran, 1997). Yet, in operational production settings that demand rapid, potentially real-time, decision-making, approaches like meta-heuristics often underperform when compared to traditional heuristics. These methods, due to mathematical optimization techniques, may not provide real-time decisions, especially as the problem's scope expands, leading to substantial performance declines (Nasiri et al., 2017). Moreover, meta-heuristics necessitate profound expertise and pose challenges during initialization and modification (Rauf et al., 2020; Zhou et al., 2020). Given these constraints, RL methods, known for swift and interactive decision-making, have gained traction in operations and control tasks (Samsonov et al., 2021; Bahrpeyma and Reichelt, 2022; Panzer et al., 2022).

7.2.2 Basics of (deep) reinforcement learning and hyper-heuristics

RL constitutes an interactive paradigm of machine learning, wherein a decision-making agent selects actions for execution and thereby iteratively refines its policy to develop the process logic. The leap to the widespread adoption of RL was primarily reached through its successful implementation in the Atari environment, making it attractive for complex optimization problems (Mnih et al., 2013). In particular, deep RL, with the additional integration of a deep neural network that allows it to process large state variables, was adapted to a variety of data-centric online applications. A fundamental constraint for integration is the requirement for the optimization task or problem to adhere to the Markov property and for the decision or control process to align with a Markov Decision Process (MDP). This is accompanied by the Markov assumption, which states that all future production states depend only on the current state, but do not imply any influences from the past which reflects the basic assumption of our later DES approach (Sutton and Barto, 2017). The simulation employs a model-free, off-policy Q-learning algorithm, as implemented by other successful benchmarks in production control (such as Estes et al., 2022; Panzer and Bender, 2022). Q-learning does not require a model of the environment and estimates the value of a Q-function (Equation (1)), which assesses a potential action of an agent based on

the Bellman equation and total accumulated expected rewards G_t .

$$Q(s_t, a_t) = r(s, a) + \gamma \max(Q(s, a)) \quad (1)$$

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad (2)$$

In this context, s represents the current state, a is the selected action, and $r(s, a)$ summarizes the obtained reward after executing action a in state s . γ defines the discount factor ($\gamma \in [0, 1]$) that determines the relative weighting of future rewards with respect to the current reward across steps. s' is the subsequent state following the execution of action a , with $\max(Q(s, a))$ being the maximal Q-value across all feasible actions a' in the subsequent state s' . The primary difference between conventional deep Q-learning and its deep learning based counterpart is the latter's use of a neural network to approximate the Q-function. Therefore, the objective is to minimize the loss between the estimated Q-function and the target value. The loss can be defined as the mean squared error $L(\theta) = E[(Q(s, a; \theta) - y)^2]$ between the estimated value $Q(s, a; \theta)$ and the target value $y = r(s, a) + \gamma \max(Q(s, a; \theta))$. Minimizing this loss allows for the updating of neural network parameters θ to better approximate the Q function. This procedure is reiterated until the performance converges against a defined level or a certain number of training steps is reached. Using these, the formula for the DQN can be derived to approximate the Q-values by minimizing the loss function using the Bellman equation as summarized in Formula (3). To further stabilize learning and increase performance, a target network with weights θ^- is introduced to compute $Q(s', a')$ for the next states (Mnih et al., 2013, 2015; Sutton and Barto, 2017).

$$Q(s_t, a_t, \theta) \leftarrow Q(s_t, a_t, \theta) + \alpha [r + \gamma \max Q(s', a', \theta^-) - Q(s_t, a_t, \theta)] \quad (3)$$

7.2.3 Deep RL based production dispatching

The limited capabilities of existing models in coping with dynamic system behavior have led to the application of various deep learning based control approaches to increase the reproduction accuracy and to minimize manual intervention, i.e. by a control strategy approximation in Bergmann and Stelzer (2011) or Bergmann et al. (2014). Luo et al. (2021) relied on a double DQN RL to minimize total delays and avoided otherwise assumed static conditions. Similarly, Mouelhi-Chibani and Pierreval (2010) and Zhao and Zhang (2021) outperformed conventional approaches with neural network based rule selection depending on flow or job-shop parameters. The latter used a convolutional neural network that takes matrices of processing times and two Boolean matrices of pending and completed operations as input to select rules such as SPT and LPT and outperformed a GA in terms of machine utilization and waiting times. In a job-shop environment, using the production state representation as a 2-D matrix and applying transfer

learning, the scheduling policy demonstrated strong performance and increased generalizability (Zheng et al., 2020). However, these and other approaches, such as that of Altenmüller et al. (2020) or Kuhnle et al. (2020), were implemented in a single-agent environment.

To facilitate a decentralized decision-making, multi-agent approaches are of particular importance for the decomposition and allocation of the total optimization process to multiple agents and to maximize the exploitation of individual skill sets as listed in Table 7.2. Malus et al. (2020) suggested an order dispatch mechanism based on joint global rewards for autonomous mobile robots to minimize delays. Hammami et al. (2017) proposed a multi-agent system based on simultaneous learning and information sharing between agents to reduce average delays. Dittrich and Fohlmeister (2020) and Hofmann et al. (2020) applied a centralized DQN decision module for training. Waschneck et al. (2018) introduced a training strategy in a wafer fabrication facility to optimize maximum uptime as a global goal. In a recent study by Sakr et al. (2021), a DQN was utilized to minimize queue waiting and lead times in wafer production. Specifically, they compared their approach to a prevailing heuristics strategy and found significant improvements. Gros et al. (2020) minimized costs in a system to control a car buffer after painting operations. Overbeck et al. (2021), on the other hand, leverage a PPO to find the best action in an automated manufacturing system, that was designed according to the chaku-chaku principles.

However, the previous research on deep RL and multi-agent based production control primarily focused on job-shop environments. There are some approaches in matrix and modular based production systems as proposed by Hofmann et al. (2020), that provides agents with immediate rewards for selected actions and delayed rewards based on the total global cycle time. This strategy outperformed a rule-based and a non-coordinated strategy by preventing the blocking of other agents and allocating global rewards. The simulated system comprised 10 workstations and several AGVs that executed multiple process steps and are fully inter-connected. May et al. (2021) followed an economic bidding approach to reduce execution time and increase utilization efficiency. This involved two system configurations, each with 15 agents and 10 stations arranged in a matrix structure, with different buffer sizes. Based on a PPO, the global utilization rate after part completion and locally accepted bids, and non-value added time as well as consecutive failed bids could be optimized. Gankin et al. (2021) implemented a first large-scale plant consisting of 25 machines arranged in a five-by-five layout, based on the approach of Mayer et al. (2021). In this approach, an action masking mechanism was used to reduce the decision complexity of all 20 DQN based transportation resources that were being trained in parallel. The agents used the same neural network and buffer as the decision instance for experience sharing.

In summary, Table 7.2 indicates that always one organizational layer was integrated in previous approaches. There is no approach, that incorporates multiple layers and a semi-heterarchical organization within a modular production system. Furthermore, the presented approaches

predominantly rely on single dedicated algorithms, such as the DQN. However, there is a need for an approach that leverages the advantages of deep learning techniques with conventional methods, as discussed in Panzer et al. (2022). Another research gap concerns the predominantly technical optimization objectives, which are rather limited in scope. Customer-centric objectives like the processing of urgent and prioritized orders, which hold particular importance in today’s economic landscape, were inadequately addressed.

Application	Algorithm	Training strategy	Control strategy	Agent interaction	Objective parameter	Orga. levels	Source
Car buffer	DQN	Iterative learning	Decentral	-	Cost/ decision time	1	Gros et al. (2020)
Chaku-chaku line	PPO	Shared PPO module	Central	-	Utilization/ throughput	1	Overbeck et al. (2021)
	SA	Concurrent learning	Decentral	Agent information exchange	Mean tardiness	1	Hammami et al. (2017)
	DQN	Iterative DQN/ heuristics learning	Decentral	Global rewards	WIP/ uptime utilization	1	Waschneck et al. (2018)
Job-shop	DQN	Shared DQN module	Central	Agent information exchange	Mean cycle time	1	Dittrich et al. (2020)
	DQN	Concurrent learning	Decentral	Agent state information	Utilization, queue waiting/ lead times	1	Sakr et al. (2023)
	TD3	Concurrent learning	Decentral	Order bidding mechanism	Tardiness	1	Malus et al. (2020)
	DQN	Shared DQN module	-	Agent state information	Throughput time	1	Hofmann et al. (2020)
Matrix production	DQN	Shared DQN module	Central	-	Throughput	1	Gankin et al. (2021)
	PPO	-	Decentral	Economic bidding	Execution time/ utilization eff.	1	May et al. (2021)
Modular production	DQN-based hyper-heuristic	Concurrent learning	Decentral	Agent and cell/ order state exchange	Process- and customer related parameters	> 1	Our framework

Table 7.2 Summary of deep RL based control approaches in multi-agent production systems

7.2.4 Research highlights and key contributions

In this paper, we propose a customizable simulation framework for modular production systems that deploys multiple dispatching agents to address customer- and process-specific objectives that can be adapted to individual scenarios. As indicated in Table 7.2, our framework uniquely supports structuring across multiple and arbitrary organizational levels and modules. This allows for the definition and generation of module-specific control policies, depending on the process related requirements and optimization parameters within the differing production modules. The modularity should allow a specific generation of local process and control knowledge that still keeps track of multiple local and global objectives.

To control the agents, we utilize a deep learning based hyper-heuristic, that combines deep learning with heuristics, which enables a rapid scenario generation and increases key production indicators in terms of performance, resilience, and adaptability. For the deep learning based decision making, the top-level heuristic does not have to learn intrinsic constraints, but can focus on the optimization task. As evidenced in numerous studies, e.g., Liu and Dong (1996), Kashfi and Javadi (2015), Heger et al. (2015), Shiue et al. (2018), and Zhang and Roy (2019), the situation-dependent selection of dispatching rules can significantly reduce computation

costs and provide an efficient tool for process optimization (Grumbach et al., 2022). The deep learning framework is the first to facilitate an automated initialization of neural networks for the distributed agents within the modular entities. To further increase learning performance and operational efficiency, we deploy a module recognition and transfer learning strategy.

7.3 Simulation design

To ensure a systematic approach for reaching the defined research objectives, we adhered to the design science research methodology as proposed by Peffers et al. (2007, referring to Figure 7.1). As the problem identification and objective definition were dealt with in the previous section, we proceed with constructing the research artifact. To accommodate dynamic requirements and provide an adaptable simulation approach, it is essential to select a suitable and scalable simulation foundation. This should seamlessly incorporate the hyper-heuristic control approach, and facilitate a decentralized and parallel decision-making.

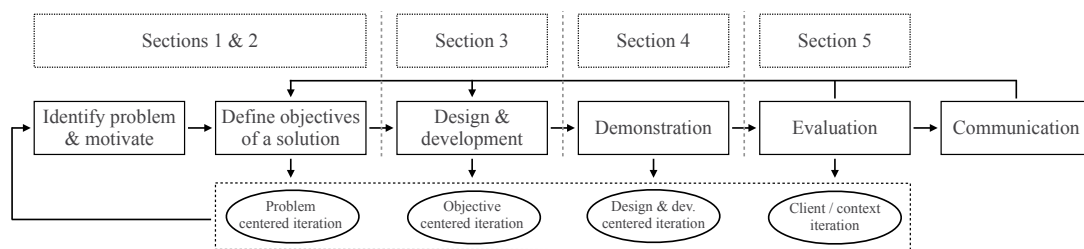


Figure 7.1 Pursued *DSRM* methodology, Peffers et al. (2007)

7.3.1 Simulation framework design

The deep learning based DES is built on a python-based simulation, developed at our chair and the *Centre for Industry 4.0*. The framework enables the rapid creation of a modular production layout with corresponding system organizations and control regulations. Incorporating the production within the DES enables the emulation of dynamic and stochastic process parameters, delineated in Table 7.3. Each parameter can be retrieved in its respective unit (pertaining to distance or time) or a discrete/categorical value, such as 0 or 1, i.e. indicative for the processed order type A/ B, respectively. Subsequently, these values undergo processing to be confined within predefined ranges, mitigating potential outliers detrimental to neural network efficacy. The simulation is based on the wide-spread SimPy simulation library, which is frequently applied in the field of DES in production control research, as it was demonstrated in previous studies (Kuhnle et al., 2020, 2021; Sakr et al., 2021).

For the control of the agents and the exploitation of deep learning mechanisms as well as conventional approaches, a hyper-heuristic is applied. The term hyper-heuristic was first defined

Entity	Attribute	Value	Attribute description
Order	Time related	[min]	Order start time, due date, time in cell
	Type related	[0,1]	Order type, complexity, priority
	Process chain	[0,1,...]	Count of remaining tasks, next/ finished tasks, position
	Process related	[0,1]	Processing, locked, picked up, in input/ same cell
Buffer/ storage	Position related	[m]	Order distance n
	Type related	[n]	Count of input / output buffer slots
Machine	Process related	[n]	Count of free slots
	Tool related	[0,1,...]	Machine type, current tool setup (tool change costs time), next setup
	Process related	[min]	Remaining setup time, currently manufacturing, remaining mfg. time
Agent	Failure related	[0,1]	Current failure, failure fixed in
	Position related	[0,1,...]	Current position, next position
	Process related	[0,1]	In movement, remaining moving time, has task, locked item

Table 7.3 Available state parameters within the simulation framework

by Cowling et al. (2001), and initially implemented using a machine learning algorithm to find an optimal order of a sales summit problem. In contrast to meta-heuristics, hyper-heuristics utilize a predetermined set of low-level heuristics, rather than searching through problem solution spaces, as illustrated in Figure 7.2. It tries to find an optimal operational sequence of the low-level heuristics that optimally solves an optimization task within the given solution space. In recent research, especially machine and more specifically, deep learning algorithms were proposed to flexibly adapt to optimization tasks as top-level heuristics and exploit the capabilities of underlying heuristics in a case-specific manner. This allows for the automation of the design process and the utilization of the knowledge of an on- or offline machine learning algorithm as an optimizer to derive near-optimal scheduling and dispatching policies based on the established and comprehensible low-level heuristics (Burke et al., 2010, 2019; Drake et al., 2020).

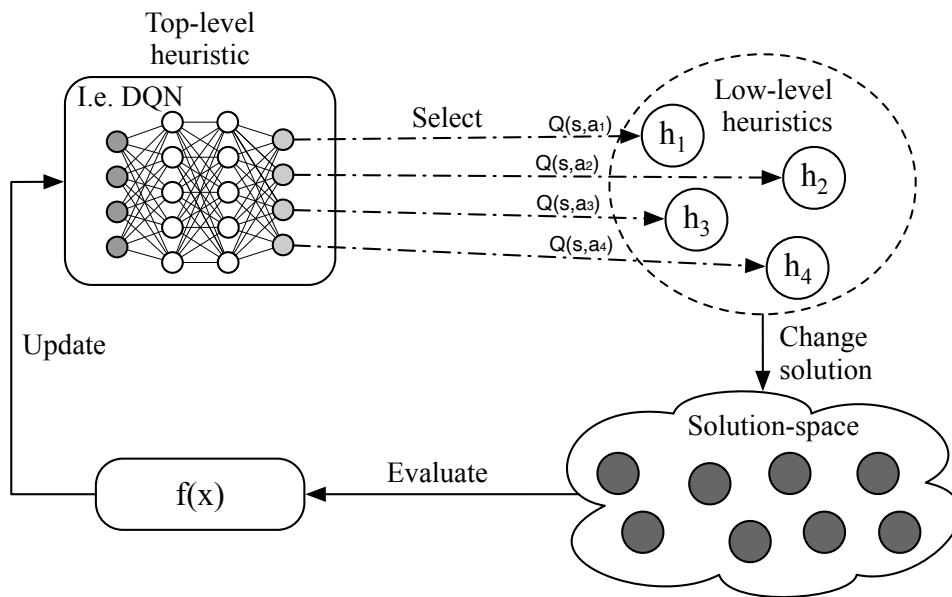


Figure 7.2 Hyper-heuristics and DQN based optimization, inspired by Goos et al. (2001), Lorente et al. (2001)

The simulation framework integrates the hyper-heuristic control concept through a multi-agent organization, where the deep learning driven agents communicate with the simulation through an order and state information exchange. Unlike a conventional single-agent approach, as illustrated on the left in Figure 7.3, the agents have individual state vectors and can execute independent actions that dynamically affect the processes. Within the modules, the agents can receive information about the other agents, such as their position or order status. To prevent local optimization tendencies, global objective variables, such as the global start time, can be received through the order parameters. Furthermore, additional variables can be derived from the available data set from Table 7.3, e.g. allowing the tracking of WIP (work in progress) levels for the individual cells, based on storage, machines, and buffers occupancies. The WIP level can in turn be utilized for operational decision-making, particularly at the upper distribution levels.

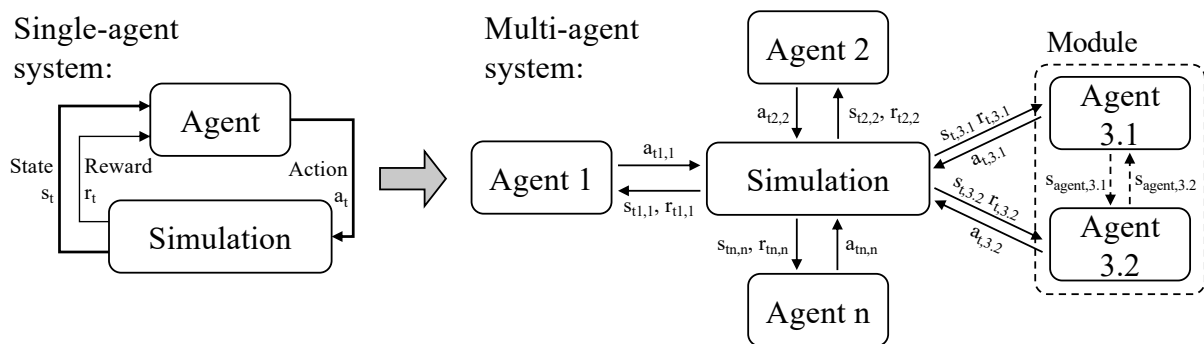


Figure 7.3 Distributed decision-making and parallel processing of the dispatching agents; left: Sutton and Barto (2017)

7.3.2 Simulation components

The given components of the base simulation are displayed in Figure 7.4, which allow for a wide range of potential optimization tasks. In this context, the individual module elements and module relationships can be flexibly defined. The number of elements within the cells and intermediate buffers can be adjusted according to the scenario specifications. Regarding the organization, the distinction between distribution and manufacturing cells is crucial, as they entail divergent intrinsic process optimization, especially in the design of the subsequent reward functions of the deep learning agents.

To clearly delineate the simulation boundaries, we established foundational criteria that our approach must adhere to. For ensuring a resilient modular design, each component functions as a unique entity, equipped with the autonomy to determine its operations. We also adopt fully observable states that are fed to the agents, thus facilitating a decision-making via deep learning. This underpins the ongoing learning and continuous improvement process and allows for a plug-and-play simulation design. During a run, we assume that system parameters such as

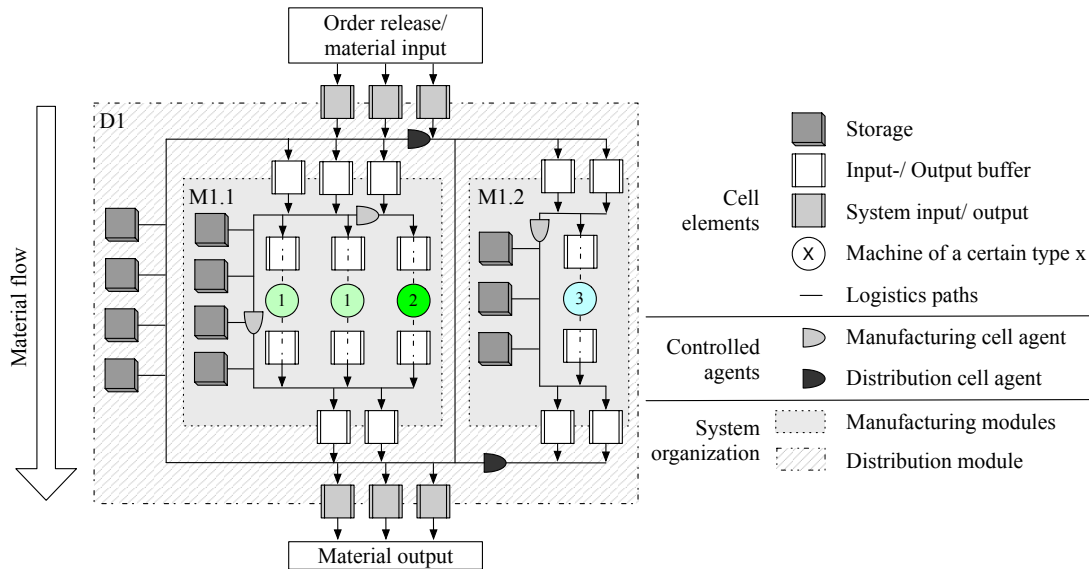


Figure 7.4 Descriptive manufacturing and distribution modules within the simulation

module layout and resource numbers remain unchanged throughout the simulated system.

However, our model does not presuppose a static system behavior. Instead, the system considers external factors such as fluctuations in order volumes, variations in system parameters due to machine malfunctions, and alterations in maintenance times. This adaptability is reflected in its capability to manage a wide array of stochastic parameters across both process-oriented and product-centric operations.

This also supports to maintain a realistic simulation design. The intention is to align the behavior of the simulated agents with real-world conditions and minimize the transfer gap. The consideration of stochastic parameters and the comprehensive set of system states, in coherence with the flexible module and deep learning based agents, contributes to a sophisticated production control simulation framework. This primarily supports the analysis of the different system parameters and can help to eliminate bottlenecks. Furthermore, it evaluates the system resilience and its ability to respond to machine failures or other unexpected occurrences.

7.4 Hyper-heuristics based control framework

The individual modules discussed in the previous section are operated by distinct dispatching agents that take decisions based on currently received system information. For this purpose, a set of ten heuristics is provided by the base simulation that take specific parameters into account for decision-making and perform rather static and straightforward operations. Based on the simulation parameters in Table 7.3, other (combined) rules can be quickly designed and implemented. Nonetheless, in dynamic environments, the operational model must adapt flexibly to external conditions to enable a resilient and indicator-oriented decision-making and

optimization.

Prior to the simulation and optimization process, the distributed and intelligent agents have to be initiated. Based on the defined requirements (as shown in Figure 7.5, on the left), a simulation is designed that meets the scenario specifications. Using the specified manufacturing and distribution modules, an appropriate neural network is sought for an agent, using a standardized unique network identifier, which aligns with the cell's structural conditions. These identifiers' properties include the cell type (distribution, manufacturing), (intermediate) buffer and storage capacities, and the number of machine resources. When no neural network fulfills the requirements, a new one is generated and tailored to the module and its associated state vector, to match the desired properties.

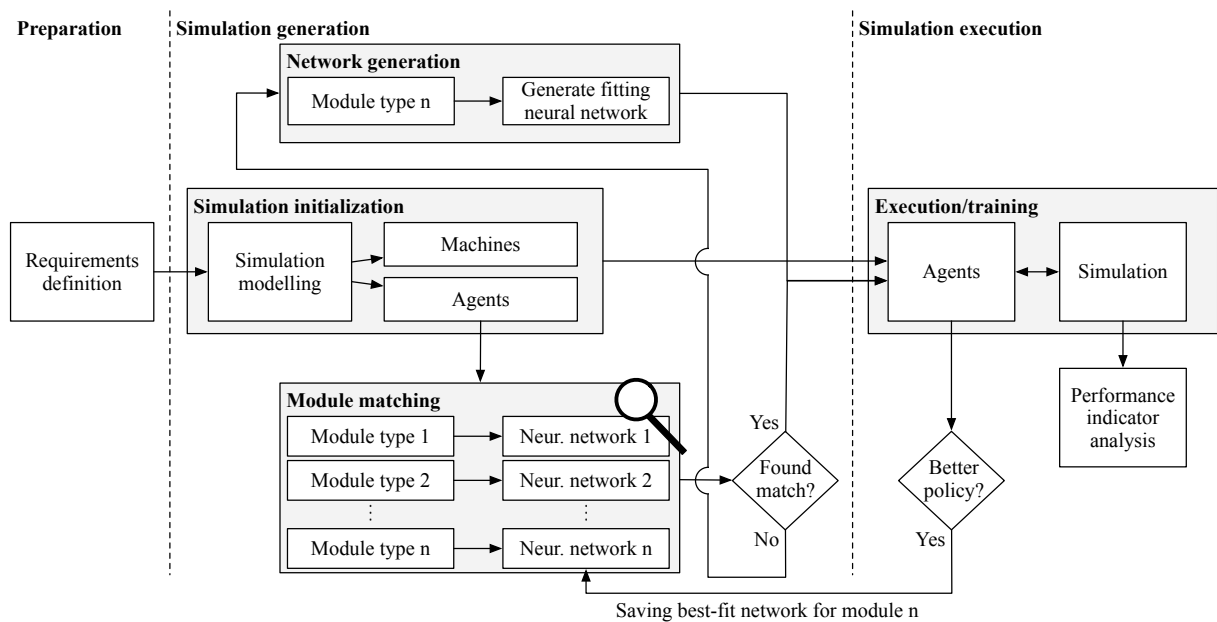


Figure 7.5 Simulation framework with flexible module recognition

For initializing the simulation and for the use of the already trained neural networks, different operating modes are available. On the one hand, during module detection, if a suitable neural network is identified, it can be embedded into the respective agent and serve as the basis for generating an optimized control policy (see case 1, Figure 7.6). In contrast, the system can be completely re-trained to avoid bias. However, this increases the computational load and is only necessitated when establishing a new simulation scenario (case 2). Alternatively, a purely operational application of the neural network is possible, in which the network is not trained and adapted (case 3). This case offers a distinct advantage in rapidly identifying suitable actions, which is particularly important in real-world applications.

All trained networks are stored within a neural network stack, and, during training, the best-performing networks are compared to determine the optimal control policy for each case (see

Figure 7.5, bottom). In addition, after each training, commencing from a pre-determined minimum number of training steps to avoid initial instability, the moving reward average is calculated and compared to the previous best performance. This method aims to facilitate continuous tracking of training progress without encouraging an overestimation of performance due to statistical anomalies.

An additional approach for transferring pre-existing knowledge involves freezing and transferring hidden layers, that substitute the initial weights of newly generated networks during initialization. In this case, the weights of a comparable manufacturing or distribution network are taken to provide the policy to a network in a new layout with collective knowledge that was gained in past simulations. This systematic storage and retrieval mechanism, enabled by the standardized neural network stack in the center of Figure 7.6, ensures that we leverage lessons-learned from the past, effectively optimizing and scaling the simulation process.

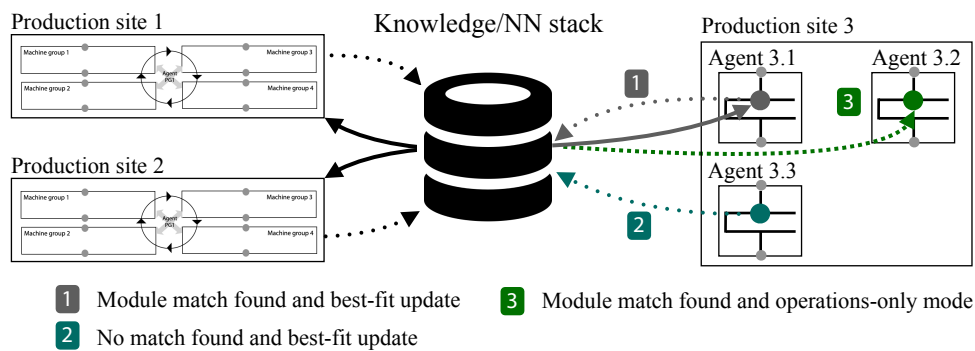


Figure 7.6 Enhancing simulation scalability through a standardized neural network stack

7.4.1 Hyper-heuristics control mechanism

In prevailing approaches, varying layouts or problem scenarios, resulting in structural changes, are associated with the creation of a new policy. Furthermore, during the initial phase of training without action specification, also wrong actions are chosen, which might be avoided through action masking. Often the actions were either an assignment of position (Gankin et al., 2021), or a combined instruction of which action is to be executed on which machine (Overbeck et al., 2021). Especially in large layouts, this quickly results in a large action space and elevated high task complexities. Conversely, the deep RL based top-level heuristic selects a low-level heuristic that is already possessing the process logic. As a result, the hyper-heuristic facilitates a complexity reduction by splitting the task into a high-level optimization and a low-level operational execution.

To implement the deep learning functions and maintain the accessibility of the *SimPy* simulation framework, *TensorFlow* was used to enable adaptive decision-making. However, prior to utilizing the *TensorFlow* framework, the deep RL control mechanism must be clarified. For this purpose,

the following section quantifies the optimization objectives through a reward function and subsequently defines states and action spaces.

7.4.2 Reward function design

The reward function serves to capture the degree of fulfillment of the defined objectives to track the training success and to refine the neural network accordingly. Initially, the objectives have to be defined first, which are then consolidated into a reward signal. In our approach, we seek to incorporate customer-related services, particularly considering order priority and urgency as known from *Prime* and other express services. In response to current trends in prime services and evolving customer expectations, we include novel customer-centric parameters that differentiate between priority/standard and rush/non-rush orders. In addition, we incorporate technical standard variables like WIP levels, throughput times, and tardiness, consistent with several other studies (Hammami et al., 2017; Waschneck et al., 2018; Hofmann et al., 2020; Malus et al., 2020).

To account for the specifications of the individual production layers, differing total rewards are designed for the distribution and manufacturing agents. As listed in Table 7.4, the order distance and the global order start time are included as the individual and process-related reward R_i fraction for the distribution agents (rewards *Dist.1.1/2*). Conversely, for the manufacturing cells, the local processing start times are included as correspondence for the throughput time to avoid dissipating WIP effects (reward *Mfg.1*). In addition to R_i , the common rewards R_c aggregate general order metrics of priority, urgency, and due dates that are considered at all levels to satisfy customer-related services, in addition to minimizing overall tardiness through a time related reward $R_{due\ to}$. All individual and general rewards are constrained within a range of -200 to 200.

Reward type	Formula
Individual rewards R_i	[<i>Mfg.1</i>] $R_{ltp} = \left(1 - 2 \frac{t_{ltp, max} - t_{ltp, n}}{t_{ltp, max} - t_{ltp, min}}\right)^5 * R_{ltp}$
	[<i>Dist.1.1</i>] $R_{gtp} = \left(1 - 2 \frac{t_{gtp, max} - t_{gtp, n}}{t_{gtp, max} - t_{gtp, min}}\right)^5 * R_{gtp}$
	[<i>Dist.1.2</i>] $R_{dist} = \left(2 \frac{t_{dist, max} - t_{dist, n}}{t_{dist, max} - t_{dist, min}} - 1\right)^5 * R_{dist}$
	[3] $R_{due\ to} = \left(2 \frac{t_{dt, max} - t_{dt, n}}{t_{dt, max} - t_{dt, min}} - 1\right)^5 * R_{dt}$
Common rewards R_c	[4] $R_{prio} = \begin{cases} 200 & \text{if status of order } i \text{ is prioritized} \\ 0 & \text{non-priority order, no priority order available} \\ -200 & \text{non-priority order, priority order available} \end{cases}$
	[5] $R_{urg} = \begin{cases} 200 & \text{if order } i \text{ is urgent} \\ 0 & \text{if order } i \text{ is non-urgent, no urgent order available} \\ -200 & \text{if order } i \text{ is non-urgent, urgent order available} \end{cases}$

Table 7.4 Summarized reward elements for individual and common rewards

7.4.3 State and action space design

The selection of a suitable state space design is of great importance for an efficient production control and should be in accordance with the previously defined objectives. The state vector should contain all essential information, which includes order due dates and urgency, local and global processing start time, distance, and job priority. It can further contain information about the machine's operating status or other process information. It further includes buffer or storage information, such as their availability or occupancy, including all necessary job information. To ensure stable training gradients, faster training, and correct weight initialization, a min-max-normalization is applied to the time and distance-related values to scale the state input to the predefined and limited range between $[-1, 1]$. Furthermore, for discrete state spaces such as order priorities and urgencies, state inputs are normalized to $[0, 1]$, implying an input of $s_{i,prio} = 0$ for normal/non-rush orders, and $s_{i,prio} = 1$ for prioritized/rush orders.

In our study, the state vector encompasses information regarding local and global start times, distances, due dates, and order priorities for each individual agent. When a change occurs in the state of a module, the corresponding agent is triggered to select and execute an optimal action based on the respective metric values for all possible positions within its module. In scenarios involving multiple production and order metrics, the framework constructs the state vector by concatenating the pre-defined set of metrics.

The action space design refers to the definition of potential actions that an agent can execute in each state to determine the processing sequence. In Kanervisto et al. (2020) generic optimization approach, the action space is discretized and only necessary actions are selected, with dispatching rules as control heuristics that are linked to corresponding deep RL outputs. The selection of low-level dispatching rules is a crucial step before the training and optimization procedure and results in a representative rule set derived from benchmarks and related approaches. One advantage is, that the action space does not grow even with large layouts, as the logic is mapped intrinsically. This also prevents adapting the state space for new product types, because it only affects process-related specifications at the low-level heuristics level. However, standard and generic variables are not affected such as processing length or optimization, and customer-related parameters such as due times or order priorities. Subsequently, the highest priority first (HP), local and global first-in-first-out (FiFo), earliest due date (EDD), and lowest-distance-first (LDF) rules are applied as the low-level rule set. These widely deployed dispatching rules enable a fast order selection, which reduces the overall processing time and increases production efficiency.

7.5 Demonstration and transfer of results

For the demonstration of the results and to facilitate an iterative optimization approach of the simulation framework in accordance with the DSRM (Peffer et al., 2007), a case study for the fabrication of two product groups is presented. Subsequently, the outcomes from the training processes and operational application will enable the evaluation of performance and other indicators, such as resilience, adaptability, and explainability.

7.5.1 Simulated case-study

For the analysis, the specification of a case-study is crucial to attain a specific benefit and to allow the deduction of product- and process-related performance indicators. For this purpose, we defined a three-stage system as presented in Figure 7.7, which consists of two mid-layer distribution modules D1.1 and D1.2, each comprising two production modules.

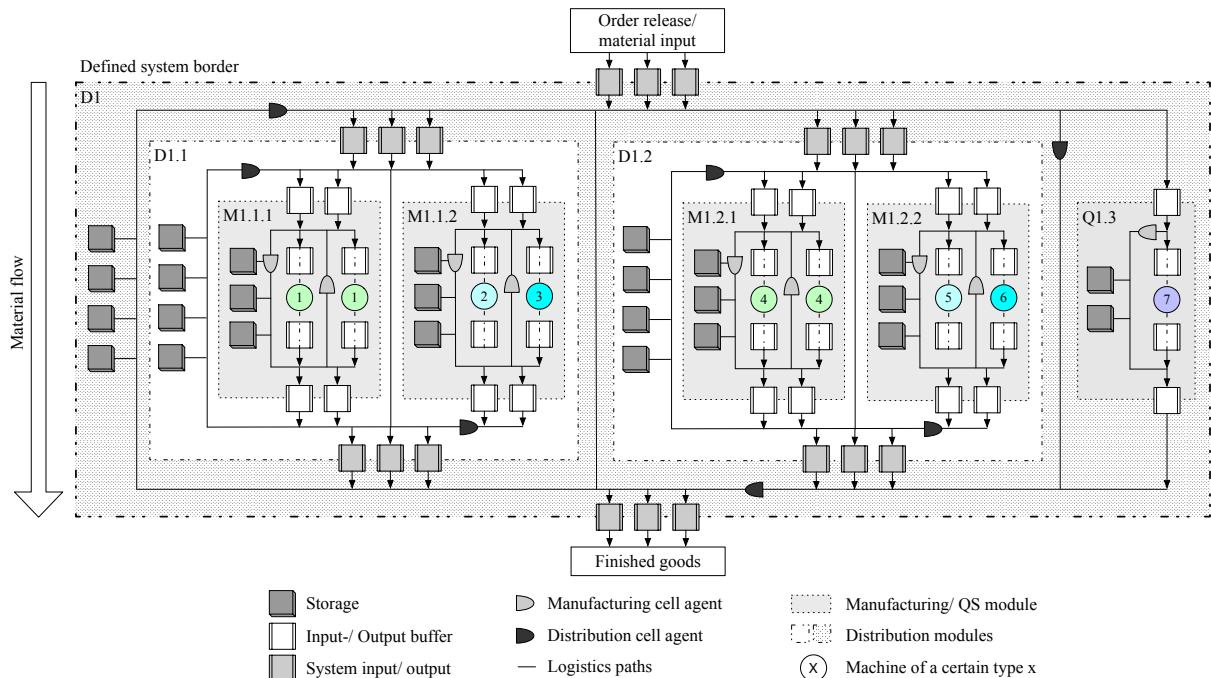


Figure 7.7 Simulated 3-staged modular production system for PCB and electric drive fabrication

A quality control module Q1.3 is provided in an independent additional module within the top layer D1. The modules D1.1 and D1.2 represent specifically defined production groups in which two types of goods are produced. This may include the production of two different kinds of printed circuit boards (PCB). At the machines of type 1, the PCBs undergo exposure and etching processes to establish circuit patterns and interconnections, followed by drilling procedures to create apertures for electronic components and interconnections (machine type 2). Subsequently, electrical interconnections are developed and electronic components are soldered to the PCBs

at machines of type 3. For descriptive purposes, it is assumed that two distinct PCB product groups are processed in parallel, with the first one in the D1.1, and the second one within the D1.2 module. This process serves as a demonstration and can be arbitrarily defined for other processes within the simulation framework. At this stage, we focus on the preliminary definition of the process chain to ensure the accurate adoption of fabrication procedures.

The assumed processing times are listed in Table 7.5. Although the described components have a seamless flow of material, the processes for manufacturing the varying PCBs are considered to be segregated operations. The incoming orders are classified as priority and/or rush orders with a 20% probability for each indicator. The orders are also subject to a 20% probability of being run through a quality assurance (QA) in module Q1.3, which takes another 2 minutes. The orders are released at the system input and transferred to the lower production levels by the distribution agents. Within the manufacturing cells, the orders are fed to the appropriate machines and then forwarded back to the higher levels after all scheduled operations are completed.

	Product group A PCB - type 1			Product group B PCB - type 2			
Processing step	Exposure	Drilling	Assembly	Exposure II	Drilling II	Assembly II	Quality check
Processing time [min.]	7	4	3	8	4	4	2
Average expected	21.4 (w.o. QA)			22.0 (w.o. QA)			
throughput time [min.]	26.4 (with QA)			28.1 (with QA)			

Table 7.5 Summarized processing times of both product groups; in [min.]

Table 7.6 outlines the configuration of the neural network model, detailing the number of neurons in both input and output layers, as well as the number and size of the hidden layers, and other additional parameters. The learning model is also specified, including the ϵ values for the beginning of the training phase, and a minimum ϵ value of 0.01, which determines the rate of random actions. A batch size of 128 was chosen to achieve a balance between learning speed and performance. The initial parameters were retrieved from established research, particularly the contributions of Mnih et al. (2015); Gankin et al. (2021), and subsequently refined through iterative optimization.

Parameter	Value	Parameter	Value
Input layer size	Cell dependent (i.e. 162 for D1)	Drop out ratio	0.01
Output layer size	5 (dispatch rules)	Learning rate α	0.005
# hidden layers	2	Discount factor γ	0.99
# neurons in hidden layers	128, 128	Learning batch size	128
Target update step	5	Minimum ϵ	0.01
Activation function/ optimizer	ReLU/ Adam	ϵ -decay	0.997

Table 7.6 Iteratively defined deep RL and training parameter settings

7.5.2 Exemplary simulation results

All of the following calculations were carried out on an Intel Core i9-12900k CPU and 64 GB RAM. If not otherwise mentioned, three simulations with varying order sequences were conducted for each metric and analysis, to provide a representative benchmark. For the introduced operational scenarios and benchmarks, all agents (despite Q1.3) were fitted with trained neural networks through a unique cell identifier. For the system described in Figure 7.7, 30 neural networks (target and online) were used for a total of 15 agents respectively.

7.5.3 Training process analysis

In order to assess the training outcome, the progress of the different objectives of handling prioritized and urgent orders over the course of the training was evaluated with regard to the through-put times and order tardiness. As illustrated in Figures 7.8(a) and 7.8(b), all processed orders were considered (dotted line), but also the combination of the different priorities and urgencies. Thereby, the considerably decreasing through-put times and tardiness rates of the higher-priority and more urgent orders, as well as their combination, becomes particularly clear. Compared to the later benchmarks, the training was conducted under a substantially elevated workload, as the focus was on maximizing learning outcomes rather than achieving a production equilibrium.

It becomes evident that throughput times and tardiness rates reach their global minimum at 1600 steps, after which the throughput times experiences an upward trend again. While this increase correlates with a surge in order quantity and diversity, the consistent tardiness observed for combined prioritized and urgent orders underscores the system's capacity to handle critical orders, relegating low-priority tasks to a waiting status.

As a result, after 2000 training steps, both priority and urgent orders exhibit a tardiness of about 10 seconds (Figure 7.8(b), right). Furthermore, it becomes evident that despite receiving the same rewards, priority orders are favored over urgent ones. This preference can be attributed to the implemented policy. If a prioritized order is selected, it is further sorted based on the due date if there are multiple orders that share the same priority. This process leads to an increased reward signal and, consequently, provokes a higher agent sensitization. Conversely, for urgent orders, no further sorting is conducted as there is often a most urgent one, resulting in no additional reward signal. This interdependence must be considered when balancing and calibrating the production objectives, which highlights the relevance of a proper reward function design and weighted rewards.

In a subsequent step, the decision-making of two agents in the D1 module was analyzed to trace the action selection back to the specific order type and to identify behavioral patterns. This enhances the optimization explainability and facilitates the comprehension of an agent's

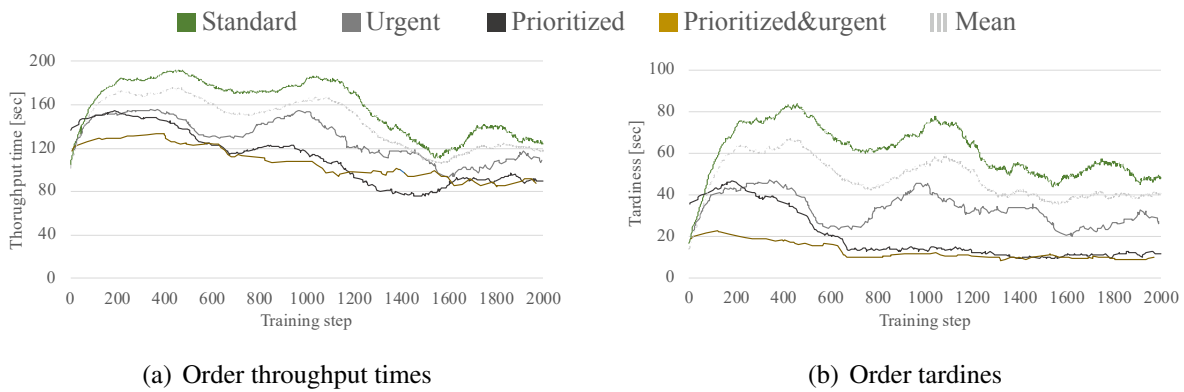


Figure 7.8 Development order throughput times and tardiness during the training process

decision-making process. Figures 7.9 and 7.10 depict the chosen actions or dispatching rules as a function of the affected order throughout the first 1000 training steps. While the Figures in 7.9 indicate a rather balanced progression, the Figures in 7.10 exhibit a much more pronounced action tendency. Both have in common that in the case of prioritized orders (see Figures 7.9(c), 7.9(d), 7.10(c), 7.10(d)), the high priority first rule is noticeably and comprehensible dominant. For the second agent in 7.10(c) and 7.10(d), it quickly displays a 100% rate of choosing the HP rule from the 400th training step onward.

In the case of urgent orders depicted in Figures 7.9(b) and 7.10(b), the earliest due date rule is employed in approximately 80% of taken actions. However, in Figure 7.9(b), this is partially offset by the low distance first rule, resulting in enhanced routing efficiency, especially at the upper levels. An indifferent behavior is observed for the standard orders in Figures 7.9(a) and 7.10(a). In Figure 7.9(a), standard orders adopt a more discernible distance and due-to time-based processing, whereas in Figure 7.10(a), no preferred action choice is evident for the distribution agent aside from the due-to date rule. Concurrently, a clear increase in the reward signal is observed, jumping from a moving average of 35 at the outset to 160 after 1000 training steps, suggesting a more likely increase in the achievement of the combined rewards and objectives, which contributes to the declining throughput times and tardiness as previously depicted in Figure 7.8.

A further training analysis examined the different training strategies. In particular, novel modules that were integrated into a system could initially satisfy a reasonable degree of optimization requirements and exhibit a sufficient degree of stability despite the unavailability of a clear control policy. Since the transfer learning strategy is intended to accelerate the learning process, the parameters of a module similar to the D1 module were used with an increased size for the input buffers of D1.1/D1.2 and an extended storage space. In addition, different ϵ_{start} values were considered for the transfer learning-based training. Noticeably, the learning rates for the agents with $\epsilon_{start} = 0.5$ are significantly faster. Although they still encounter a 50% chance for a

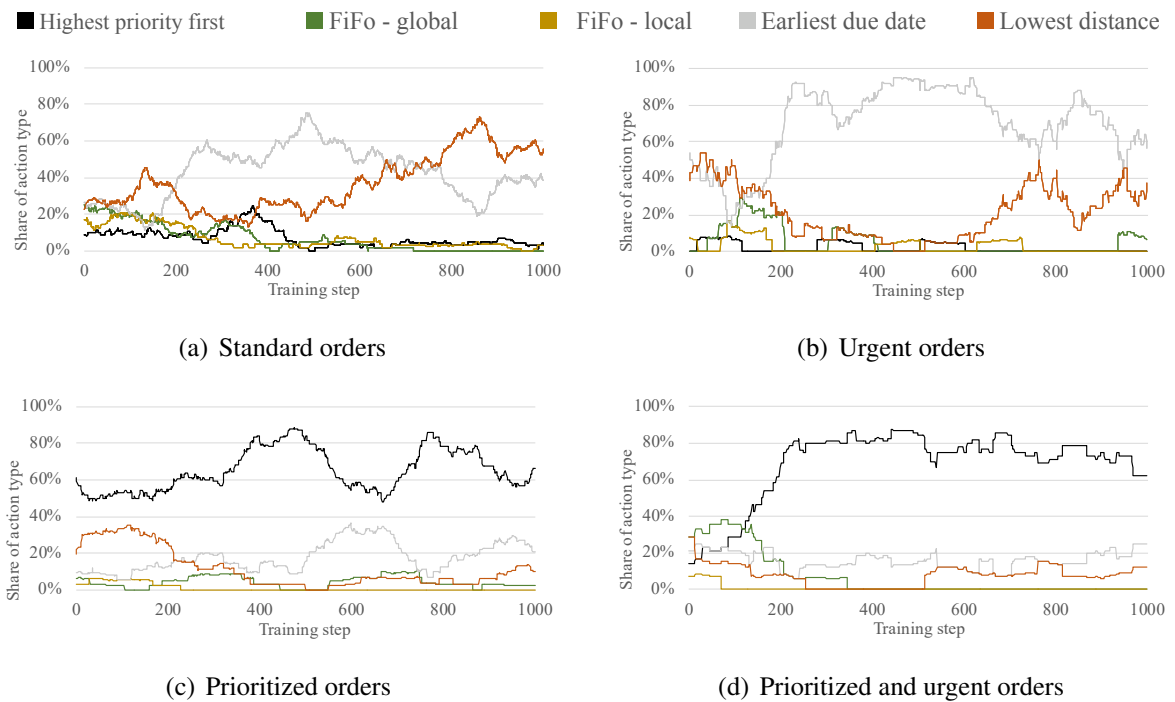


Figure 7.9 Moving average of chosen actions for a D1 module agent throughout the training process

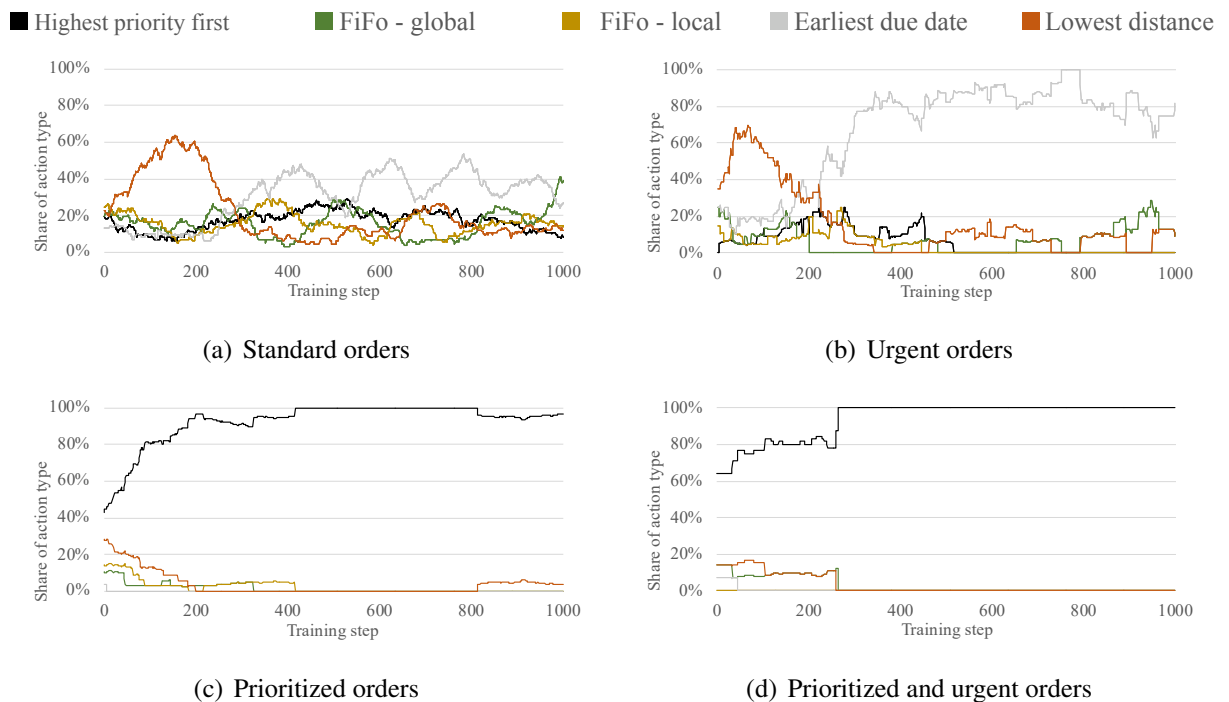


Figure 7.10 Moving average of a another agent in D1, emphasizing a clear trend towards explainable action selection

random action at the beginning, they exploit prior knowledge and reach higher rewards more quickly. Apparently, part of the existing control policy from the other agent could be used to make training decisions more effectively, despite the change in state inputs.

7.5.4 Analysis of customer related indicator benchmarks

To establish a comparative benchmark against prevalent dispatching rules, the consolidated results are listed in Table 7.7. For these benchmarks, a cumulative duration of 7200 minutes with 2700 scheduled orders was simulated. Each time with varying order sets and the objective to assess the modular system's adaptability and efficacy in response to fluctuating demands. For the analysis, the combined throughput times and the tardiness are condensed as the central evaluation indicators for the satisfaction of customer-related objectives.

The application of deep RL-based agents demonstrates a measurable improvement in order tardiness and throughput times for both prioritized and rush orders relative to standard orders and conventional dispatching rules. This suggests a heightened alignment with customer-centric parameters. Specifically, there was an observed enhancement in the processing efficiency of intricate orders. Combined prioritized and urgent orders had a direct impact on the reduction of tardiness by nearly -100% and throughput times of -52% , in contrast to standard orders. When assessing standard orders within the benchmark, the hyper-heuristic exhibited increased throughput time and tardiness. Nonetheless, these factors were considered of lesser importance due to their relative insignificance.

The deep learning approach also offers the ability to optimize the allocation of resources effectively. It facilitates the development of individual control schemes for each order backlog within a module, ensuring optimal resource utilization. The inherent self-learning mechanisms of the deep RL offer two primary advantages, they support the ongoing refinement of production control and objective realization, and they enhance the performance of production processes within a dynamic environment. The adaptability of the deep RL approach underscores its potential to efficiently address diverse operational scenarios in order processing.

7.5.5 Evaluation of adaptability and resilience

To conduct a thorough evaluation of the framework's adaptability, we analyzed the learned control policy, first, from a structural-related perspective, and second, from an order-related perspective. The former refers to the responsiveness to a changing production environment through additional manufacturing modules, processes, and technologies or products. By adopting the hyper-heuristic approach, the deep learning component is able to strip down arising changes such as new products to a straight-forward process level that did not affect the top-level decision-making logic and process optimization. Similarly, the response to a malfunctioning machine was compensated by the rule-based decision-making process while the optimization process continued within the redefined context. Only the structural change of a system or re-scaling system components in scope, which goes along with a changed state vector, requires a re-training

Total	Hyper-heuristic		Earliest due date		FiFo global		FiFo local	
Processed orders [#]	2576		2566		2567		2581	
Throughput time [min]	131,2		131,7		132,1		133,0	
Tardiness [min]	37,8		37,2		38,7		37,5	
WIP [#]	40,1		38,8		42,0		39,0	
Throughput time order type split	Standard order	Rush Order	Standard order	Rush Order	Standard order	Rush Order	Standard order	Rush Order
Standard [min]	149,7	115,3	140,7	98,8	132,6	131,4	133,5	132,2
Priority [min]	87,7	71,5	140,3	98,6	131,6	129,1	132,4	130,3
Standard	1	-23%	1	-30%	1	-1%	1	-1%
Priority	-41%	-52%	0%	-30%	-1%	-3%	-1%	-2%
Tardiness order type split	Standard order	Rush Order	Standard order	Rush Order	Standard order	Rush Order	Standard order	Rush Order
Standard [min]	51,7	24,1	43,1	15,8	39,2	37,4	38,1	36,5
Priority [min]	5,0	0,7	42,2	16,1	38,2	37,6	36,8	36,5
Standard	1	-53%	1	-63%	1	-5%	1	-4%
Priority	-97%	-100%	-2%	-63%	-2%	-4%	-3%	-4%
WIP order type split [#]	Standard order	Rush Order	Standard order	Rush Order	Standard order	Rush Order	Standard order	Rush Order
Standard	28,1	7,4	25,5	6,7	27,7	7,3	25,8	6,8
Priority	3,7	1,0	5,2	1,4	5,6	1,5	5,2	1,4

Table 7.7 Control optimization and benchmark against conventional heuristics

of the control policy, but only for the directly impacted and neighboring/upstream sub-systems. Due to the decentralized control paradigm, alterations such as the addition of a machine have a limited impact on other sub-systems. Only for the initial training, in which all agents are trained concurrently, the training takes significantly longer compared to a re-training of individual parts. For our case study, this resulted in a re-training time span of approximately 16 hours against 3 hours for training the M1.1.1 module. Another strategy allowed all but one of the agents in a module to be controlled by a heuristic. As a result, the training time for a single agent was notably reduced to 1.5 hours, allowing for an efficient transfer of acquired knowledge to the remaining agents after completing the training process.

With respect to the structural modifications, the control design effectively stabilized the system loads and improved resilience, as detailed in Table 7.8. Observations from the lower section reveal that the control approach consistently enhanced the handling of prioritized orders, even with an increase in system load from 2400 to 2800 orders.

The WIP numbers in Table 7.8 should be contextualized with order quantities since the scheduled order entries lead to the entry of significantly fewer prioritized and urgent orders (20% for each order property). Statistically, in the case of 2800 orders with a 45.6 WIP, 1.8 combined prioritized and urgent orders should be released (4%). Yet, the data indicates a WIP of only 1.3 orders, indicating a 28% reduction. In comparison to the tardiness observed for the 2700 episodes,

7 Publication 3 - A deep learning based production control framework

Scheduled orders	High-load scenario						Mid-load scenario			Low-load scenario					
	2800			2700			2600			2500			2400		
Split analysis:	WIP	Tardi.	TPT	WIP	Tardi.	TPT	WIP	Tardi.	TPT	WIP	Tardi.	TPT	WIP	Tardi.	TPT
Total	45,6	43,9	144,3	37,5	33,1	124,0	31,5	27,6	103,9	18,6	9,5	73,8	11,8	0,4	34,9
Non-prio/-urgent	31,3	56,9	159,3	26,0	43,4	138,1	21,9	35,9	114,5	12,7	12,9	81,7	7,8	0,6	35,5
Non-prio, urgent	8,3	30,5	129,5	6,9	23,1	110,9	5,8	18,1	91,1	3,4	5,5	63,5	2,1	0,0	34,5
Prio, non-urgent	4,8	14,2	111,2	3,7	8,9	91,2	3,0	8,9	81,7	2,0	1,8	57,8	1,5	0,0	33,2
Prio & urgent	1,3	5,3	94,3	1,0	5,5	82,8	0,8	3,1	68,7	0,6	0,2	46,5	0,4	0,0	32,6
Relative:															
Non-prio/-urgent	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Non-prio, urgent	-74%	-46%	-19%	-74%	-47%	-20%	-74%	-50%	-20%	-73%	-57%	-22%	-73%	-94%	-3%
Prio, non-urgent	-85%	-75%	-30%	-86%	-79%	-34%	-86%	-75%	-29%	-85%	-86%	-29%	-81%	-94%	-7%
Prio & urgent	-96%	-91%	-41%	-96%	-87%	-40%	-96%	-91%	-40%	-96%	-98%	-43%	-95%	-100%	-8%

Table 7.8 Assessment of system resilience against fluctuating order loads

the tardiness decreased from 5.5 seconds to 5.3 seconds at the 2800-episode mark. While this decrease could be incidental, it also underscores the control design's efficiency in processing orders of higher importance. In essence, the analysis underscores that under conditions of high system load (e.g., with 2700/2800 scheduled orders), the increased scope of operational action enlarges the optimization range for deep learning agents, thereby compensating for the elevated WIP and resulting throughput times.

7.6 Framework discussion

In contemporary markets, characterized by fluctuating sales and supply conditions, it is essential to pursue ongoing process adaptation and optimization to maintain a competitive edge. For this purpose, simulations are increasingly recognized as instrumental to evaluate the effectiveness of (intelligent) production control strategies, including those that leverage system intelligence, and for preemptively evaluating potential real-world scenarios. In the present study, we attempted to synergize the comprehensive flexibility inherent to a simulation framework – suited for diverse production scenarios – with the resilient performance and adaptability characteristic of a deep RL-based hyper-heuristic.

Our results demonstrate that by defining a simple reward function paired with a defined action space, we were able to optimize pre-defined objectives and outperform widely applied dispatching rules. The definition of differing distribution and manufacturing levels facilitated the simulation of large-scale systems, segmenting them into modules, to decompose and manage the overall system complexity. The adoption of a decentralized agent control and the modularization of the entire production system also offer great potential for re-using trained agents, since changes in the production system do not affect the entire system, but only individual sections.

The developed framework aims to minimize the transfer gap through the automated initiation of

the production system combined with the integration of various operation modes. This integration not only supports comprehensive re-training but also promotes selective and efficient utilization of individual policies, potentially resulting in faster and improved simulation outcomes. The incorporation of the modularization concept from the foundational simulation framework into the intelligent control strategy suggests enhanced transferability to diverse practical applications.

7.7 Conclusion

In this paper, we introduced a flexible simulation framework for modular production systems that is based on a novel module and neural network recognition mechanism and a stack of trained neural networks to increase simulation efficiency and adaptability. By integrating a deep RL based top-level heuristic and process constraint mapping through low-level dispatching rules, the framework enables the optimization of various target parameters within a multi-agent production system. The hyper-heuristic control mechanism facilitated the primary utilization of deep RL for optimization purposes, thereby promoting resilient and stable processes, even during the initial training. Both, distribution and manufacturing/shopfloor levels, were implemented and optimized regarding key performance indicators by using a concurrent learning paradigm. By leveraging the synergy between the flexible simulation framework and the adaptive control concept, requirements of customer-centric production services and process target indicators can be freely defined.

In a representative evaluation, we demonstrated the multi-objective optimization performance of the control framework. Prioritized and urgent orders were processed with reduced throughput times and tardiness than standard orders, leading to accelerated response times to external disruptions, such as increased order loads. Particularly in today's demanding market environments, this contributes to maintaining a company's competitiveness. Furthermore, the framework reached a more balanced optimization, as parameters were dynamically assessed, allowing a scenario-specific emphasis on individual objectives and an assessment of explainability for the chosen action.

The presented framework leverages a structural and process-related adaptability, thus providing a flexible response to order fluctuations and facilitating the targeted processing of arbitrary order types and quantities. Further, the influence of incoming orders and machine bottlenecks on WIP can be systematically examined. Within the field of complex manufacturing, our framework employs both, a structural modularization and an algorithmic hyper-heuristic, for system complexity decomposition. The decentralized decision-making further helped to reduce optimization complexity and enabled coping with the surging information volumes and ever-increasing customer requirements. This not only streamlines operational processes but also ensures a high data management efficiency. Given its inherent adaptability, the framework

remains efficient for a wide range of potential scenarios and motivates further research of intelligent control strategies in modular production systems.

Such future research endeavors might focus on examining the quantification of reward components and their weighted correlation to the attainment of desired objectives. We also foresee a focus on the practical transfer of these methodologies to real-world settings and the integration of advanced agent collaboration techniques. Additionally, weaving in a techno-economical analysis will be pivotal to ensure cost-effectiveness and operational efficiency in modular production systems

Copyright notice

This is an accepted version of this article published in:

Panzer, M. and N. Gronau (2023). Designing an adaptive and deep learning based control framework for modular production systems. *Journal of Intelligent Manufacturing*, p. 1-24.

<https://doi.org/10.1007/s10845-023-02249-3>

Clarification of the copyright adjusted according to the guidelines of the publisher.

Contributor roles

This paper is the result of collaborative efforts where specific responsibilities were allocated to ensure the effective completion of the research and the preparation of the manuscript:

- Marcel Panzer: Led the major portion of the work for this publication and created the research artifact and control framework. His involvement included the conceptualization of the research, the design and implementation of the methodology, conducting simulations and evaluations, data collection and analysis, and primarily drafting the original manuscript. He also contributed to the compilation and editing of the final manuscript during the review stages.
- Norbert Gronau: Contributed to the development of this publication through thorough review and guidance. His involvement was marked by critical analysis, offering valuable feedback and recommendations. His direction played a important role in refining the publication, ensuring scholarly thoroughness and adherence to the planned research goals.

The *Declaration of the Co-Authors* is inserted at the end of this thesis.

Publication 3 - References

- Altenmüller, T., T. Stüker, B. Waschneck, A. Kuhnle and G. Lanza (2020). Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints. *Production Engineering* 14(3), p. 319–328. doi: 10.1007/s11740-020-00967-8.
- Arunraj, N. S. and D. Ahrens (2015). A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics* 170, p. 321–335. doi: 10.1016/j.ijpe.2015.09.039.
- Babiceanu, R. F. and F. F. Chen (2006). Development and Applications of Holonic Manufacturing Systems: A Survey. *Journal of Intelligent Manufacturing* 17(1), p. 111–131. doi: 10.1007/s10845-005-5516-y.
- Bahrpeyma, F. and D. Reichelt (2022). A review of the applications of multi-agent reinforcement learning in smart factories. *Frontiers in Robotics and AI* 9, p. 1027340. doi: 10.3389/frobt.2022.1027340.
- Bergmann, S. and S. Stelzer (2011). Approximation of Dispatching Rules in Manufacturing Control Using Artificial Neural Networks. In: *2011 IEEE Workshop on Principles of Advanced and Distributed Simulation*, Nice, France, p. 1–8. IEEE. doi: 10.1109/PADS.2011.5936774.
- Bergmann, S., S. Stelzer and S. Strassburger (2014). On the use of artificial neural networks in simulation-based manufacturing control. *Journal of Simulation* 8(1), p. 76–90. doi: 10.1057/jos.2013.6.
- Buckhorst, A. F., L. Grahn and R. H. Schmitt (2022). Decentralized Holonic Control System Model for Line-less Mobile Assembly Systems. *Robotics and Computer-Integrated Manufacturing* 75, p. 102301. doi: 10.1016/j.rcim.2021.102301.
- Bueno, A., M. Godinho Filho and A. G. Frank (2020). Smart production planning and control in the Industry 4.0 context: A systematic literature review. *Computers & Industrial Engineering* 149, p. 106774. doi: 10.1016/j.cie.2020.106774.
- Burke, E. K., M. R. Hyde, G. Kendall, G. Ochoa, E. Özcan and J. R. Woodward (2010). A Classification of Hyper-Heuristic Approaches. In: M. Gendreau and J.-Y. Potvin (Hrsg.), *Handbook of Metaheuristics*, Volume 272, p. 453–477. Cham: Springer International Publishing. ISBN: 978-3-319-91085-7.
- Burke, E. K., M. R. Hyde, G. Kendall, G. Ochoa, E. Özcan and J. R. Woodward (2019). A Classification of Hyper-Heuristic Approaches: Revisited. In: M. Gendreau and J.-Y. Potvin (Hrsg.), *Handbook of Metaheuristics*, Volume 272, p. 453–477. Cham: Springer International Publishing. ISBN: 978-3-319-91085-7.

- Cadavid, J. P. U., S. Lamouri, B. Grabot and A. Fortin (2019). Machine Learning in Production Planning and Control: A Review of Empirical Literature. *IFAC-PapersOnLine* 52(13), p. 385–390. doi: 10.1016/j.ifacol.2019.11.155.
- Chen, S., W. Wang and E. Zio (2021). A Simulation-Based Multi-Objective Optimization Framework for the Production Planning in Energy Supply Chains. *Energies* 14(9), p. 2684. doi: 10.3390/en14092684.
- Cowling, P., G. Kendall and E. Soubeiga (2001). A Hyperheuristic Approach to Scheduling a Sales Summit. In: G. Goos, J. Hartmanis, J. van Leeuwen, E. Burke, and W. Erben (Hrsg.), *Practice and Theory of Automated Timetabling III*, Volume 2079, p. 176–190. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-42421-5.
- Derigent, W., O. Cardin and D. Trentesaux (2021). Industry 4.0: contributions of holonic manufacturing control architectures and future challenges. *Journal of Intelligent Manufacturing* 32(7), p. 1797–1818. doi: 10.1007/s10845-020-01532-x.
- Dittrich, M.-A. and S. Fohlmeister (2020). Cooperative multi-agent system for production control using reinforcement learning. *CIRP Annals* 69(1), p. 389 – 392. doi: 10.1016/j.cirp.2020.04.005.
- Drake, J. H., A. Kheiri, E. Özcan and E. K. Burke (2020). Recent advances in selection hyper-heuristics. *European Journal of Operational Research* 285(2), p. 405–428. doi: 10.1016/j.ejor.2019.07.073.
- Farsi, M., J. A. Erkoyuncu, D. Steenstra and R. Roy (2019). A modular hybrid simulation framework for complex manufacturing system design. *Simulation Modelling Practice and Theory* 94, p. 14–30. doi: 10.1016/j.simpat.2019.02.002.
- Fowler, J. W., L. Mönch and T. Ponsignon (2015). Discrete-event simulation for semiconductor wafer fabrication facilities: a tutorial. *International Journal of Industrial Engineering* 22(5). doi: 10.23055/IJMETAP.2015.22.5.2276.
- Fumagalli, L., E. Negri, E. Sottoriva, A. Polenghi and M. Macchi (2018). A novel scheduling framework: integrating genetic algorithms and discrete event simulation. *International Journal of Management and Decision Making* 17(4), p. 371. doi: 10.1504/IJMDM.2018.095738.
- Gankin, D., S. Mayer, J. Zinn, B. Vogel-Heuser and C. Endisch (2021). Modular Production Control with Multi-Agent Deep Q-Learning. In: *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vasteras, Sweden, p. 1–8. IEEE. doi: 10.1109/ETFA45728.2021.9613177.
- Garetti, M. and M. Taisch (1999). Neural networks in production planning and control. *Production Planning & Control* 10(4), p. 324–339. doi: 10.1080/095372899233082.

- Grabot, B. and L. Geneste (1994). Dispatching rules in scheduling Dispatching rules in scheduling: a fuzzy approach. *International Journal of Production Research* 32(4), p. 903–915. doi: 10.1080/00207549408956978.
- Gronauer, S. and K. Diepold (2021). Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*. doi: 10.1007/s10462-021-09996-w.
- Groover, M. P. (2019). *Automation, production systems, and computer-integrated manufacturing* (Fifth edition ed.). Hudson Street, New York: Pearson Education.
- Gros, T. P., J. Gros and V. Wolf (2020). Real-Time Decision Making for a Car Manufacturing Process Using Deep Reinforcement Learning. In: *2020 Winter Simulation Conference (WSC)*, Orlando, FL, USA, p. 3032–3044. IEEE. doi: 10.1109/WSC48552.2020.9383884.
- Grumbach, F., A. Müller, P. Reusch and S. Trojahn (2022). Robust-stable scheduling in dynamic flow shops based on deep reinforcement learning. *Journal of Intelligent Manufacturing*. doi: 10.1007/s10845-022-02069-x.
- Hammami, Z., W. Mouelhi and L. Ben Said (2017). On-line self-adaptive framework for tailoring a neural-agent learning model addressing dynamic real-time scheduling problems. *Journal of Manufacturing Systems* 45, p. 97–108. doi: 10.1016/j.jmsy.2017.08.003.
- Heger, J., T. Hildebrandt and B. Scholz-Reiter (2015). Dispatching rule selection with Gaussian processes. *Central European Journal of Operations Research* 23(1), p. 235–249. doi: 10.1007/s10100-013-0322-7.
- Herrera, M., M. Pérez-Hernández, A. Kumar Parlikad and J. Izquierdo (2020). Multi-Agent Systems and Complex Networks: Review and Applications in Systems Engineering. *Processes* 8(3), p. 312. doi: 10.3390/pr8030312.
- Hofmann, C., C. Krahe, N. Stricker and G. Lanza (2020). Autonomous production control for matrix production based on deep Q-learning. *Procedia CIRP* 88, p. 25–30. doi: 10.1016/j.procir.2020.05.005.
- Holthaus, O. and C. Rajendran (1997). Efficient dispatching rules for scheduling in a job shop. *International Journal of Production Economics* 48(1), p. 87–105. doi: 10.1016/S0925-5273(96)00068-0.
- Jeon, S. M. and G. Kim (2016). A survey of simulation modeling techniques in production planning and control (PPC). *Production Planning & Control* 27(5), p. 360–377. doi: 10.1080/09537287.2015.1128010.
- Kallestad, J., R. Hasibi, A. Hemmati and K. Sørensen (2023). A General Deep Reinforcement Learning Hyperheuristic Framework for Solving Combinatorial Optimization Problems. *European Journal of Operational Research*, p. S037722172300036X. doi:

10.1016/j.ejor.2023.01.017.

- Kanervisto, A., C. Scheller and V. Hautamaki (2020). Action Space Shaping in Deep Reinforcement Learning. In: *2020 IEEE Conference on Games (CoG)*, Osaka, Japan, p. 479–486. IEEE. doi: 10.1109/CoG47356.2020.9231687.
- Kang, Z., C. Catal and B. Tekinerdogan (2020). Machine learning applications in production lines: A systematic literature review. *Computers & Industrial Engineering* 149, p. 106773. doi: 10.1016/j.cie.2020.106773.
- Kapoor, K., A. Z. Bigdeli, Y. K. Dwivedi and R. Raman (2021). How is COVID-19 altering the manufacturing landscape? A literature review of imminent challenges and management interventions. *Annals of Operations Research*. doi: 10.1007/s10479-021-04397-2.
- Kashfi, M. A. and M. Javadi (2015). A model for selecting suitable dispatching rule in FMS based on fuzzy multi attribute group decision making. *Production Engineering* 9(2), p. 237–246. doi: 10.1007/s11740-015-0603-1.
- Kuhnle, A., J.-P. Kaiser, F. Theiß, N. Stricker and G. Lanza (2020). Designing an adaptive production control system using reinforcement learning. *Journal of Intelligent Manufacturing* 32, p. 855–876. doi: 10.1007/s10845-020-01612-y.
- Kuhnle, A., M. C. May, L. Schäfer and G. Lanza (2021). Explainable reinforcement learning in production control of job shop manufacturing system. *International Journal of Production Research*, p. 1–23. doi: 10.1080/00207543.2021.1972179.
- Kuhnle, A., N. Röhrig and G. Lanza (2019). Autonomous order dispatching in the semiconductor industry using reinforcement learning. *Procedia CIRP* 79, p. 391–396. doi: 10.1016/j.procir.2019.02.101.
- Kuhnle, A., L. Schäfer, N. Stricker and G. Lanza (2019). Design, Implementation and Evaluation of Reinforcement Learning for an Adaptive Order Dispatching in Job Shop Manufacturing Systems. *Procedia CIRP* 81, p. 234–239. doi: 10.1016/j.procir.2019.03.041.
- Law, A. M. (2007). *Simulation modeling and analysis* (4. ed ed.). McGraw-Hill series in industrial engineering and management science. Boston: McGraw-Hill.
- Liao, Y., F. Deschamps, E. d. F. R. Loures and L. F. P. Ramos (2017). Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal. *International Journal of Production Research* 55(12), p. 3609–3629. doi: 10.1080/00207543.2017.1308576.
- Liu, H. and J. J. Dong (1996). Dispatching rule selection using artificial neural networks for dynamic planning and scheduling. *Journal of Intelligent Manufacturing* 7(3), p. 243–250. doi: 10.1007/BF00118083.
- Liu, R., R. Piplani and C. Toro (2022). Deep reinforcement learning for dynamic scheduling of a

- flexible job shop. *International Journal of Production Research* 60(13), p. 4049–4069. doi: 10.1080/00207543.2022.2058432.
- Luo, S., L. Zhang and Y. Fan (2021). Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. *Computers & Industrial Engineering* 159, p. 107489. doi: 10.1016/j.cie.2021.107489.
- Malus, A., D. Kozjek and R. Vrabič (2020). Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. *CIRP Annals* 69(1), p. 397 – 400. doi: 10.1016/j.cirp.2020.04.001.
- Manriquez, F., J. Pérez and N. Morales (2020). A simulation–optimization framework for short-term underground mine production scheduling. *Optimization and Engineering* 21(3), p. 939–971. doi: 10.1007/s11081-020-09496-w.
- May, M. C., L. Kiefer, A. Kuhnle, N. Stricker and G. Lanza (2021). Decentralized Multi-Agent Production Control through Economic Model Bidding for Matrix Production Systems. *Procedia CIRP* 96, p. 3–8. doi: 10.1016/j.procir.2021.01.043.
- Mayer, S., T. Classen and C. Endisch (2021). Modular production control using deep reinforcement learning: proximal policy optimization. *Journal of Intelligent Manufacturing* 32(8), p. 2335–2351. doi: 10.1007/s10845-021-01778-z.
- Mehlig, B. (2021). *Machine Learning with Neural Networks: An Introduction for Scientists and Engineers* (1 ed.). Cambridge University Press. doi: 10.1017/9781108860604.
- Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller (2013). Playing Atari with Deep Reinforcement Learning. p. arXiv:1312.5602.
- Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg and D. Hassabis (2015). Human-level control through deep reinforcement learning. *Nature* 518(7540), p. 529–533. doi: 10.1038/nature14236.
- Mouelhi-Chibani, W. and H. Pierreval (2010). Training a neural network to select dispatching rules in real time. *Computers & Industrial Engineering* 58(2), p. 249–256. doi: 10.1016/j.cie.2009.03.008.
- Mourtzis, D. (2020). Simulation in the design and operation of manufacturing systems: state of the art and new trends. *International Journal of Production Research* 58(7), p. 1927–1949. doi: 10.1080/00207543.2019.1636321.
- Nasiri, M. M., R. Yazdanparast and F. Jolai (2017). A simulation optimisation approach for real-time scheduling in an open shop environment using a composite dispatching rule. *International Journal of Computer Integrated Manufacturing* 30(12), p. 1239–1252. doi:

10.1080/0951192X.2017.1307452.

- Nazari, M., A. Oroojlooy, L. V. Snyder and M. Takáč (2018). Reinforcement Learning for Solving the Vehicle Routing Problem. doi: 10.48550/ARXIV.1802.04240.
- Neto, A. A., F. Deschamps, E. R. Da Silva and E. P. De Lima (2020). Digital twins in manufacturing: an assessment of drivers, enablers and barriers to implementation. *Procedia CIRP* 93, p. 210–215. doi: 10.1016/j.procir.2020.04.131.
- Oluyisola, O. E., S. Bhalla, F. Sgarbossa and J. O. Strandhagen (2022). Designing and developing smart production planning and control systems in the industry 4.0 era: a methodology and case study. *Journal of Intelligent Manufacturing* 33(1), p. 311–332. doi: 10.1007/s10845-021-01808-w.
- Overbeck, L., A. Hugues, M. C. May, A. Kuhnle and G. Lanza (2021). Reinforcement Learning Based Production Control of Semi-automated Manufacturing Systems. *Procedia CIRP* 103, p. 170–175. doi: 10.1016/j.procir.2021.10.027.
- Panzer, M. and B. Bender (2022). Deep reinforcement learning in production systems: a systematic literature review. *International Journal of Production Research* 60(13), p. 4316–4341. doi: 10.1080/00207543.2021.1973138.
- Panzer, M., B. Bender and N. Gronau (2022). Neural agent-based production planning and control: An architectural review. *Journal of Manufacturing Systems* 65, p. 743–766. doi: 10.1016/j.jmsy.2022.10.019.
- Parente, M., G. Figueira, P. Amorim and A. Marques (2020). Production scheduling in the context of Industry 4.0: review and trends. *International Journal of Production Research* 58(17), p. 5401–5431. doi: 10.1080/00207543.2020.1718794.
- Pawar, S. and R. Maulik (2021). Distributed deep reinforcement learning for simulation control. *Machine Learning: Science and Technology* 2(2), p. 025029. doi: 10.1088/2632-2153/abdaf8.
- Peffer, K., T. Tuunanen, M. A. Rothenberger and S. Chatterjee (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24(3), p. 45–77. doi: 10.2753/MIS0742-1222240302.
- Phanden, R. K., Z. Palková and R. Sindhvani (2019). A Framework for Flexible Job Shop Scheduling Problem Using Simulation-Based Cuckoo Search Optimization. In: K. Shanker, R. Shankar, and R. Sindhvani (Hrsg.), *Advances in Industrial and Production Engineering*, p. 247–262. Singapore: Springer Singapore. ISBN: 978-981-13-6412-9.
- Rauf, M., Z. Guan, S. Sarfraz, J. Mumtaz, E. Shehab, M. Jahanzaib and M. Hanif (2020). A smart algorithm for multi-criteria optimization of model sequencing problem in assembly lines. *Robotics and Computer-Integrated Manufacturing* 61, p. 101844. doi:

- 10.1016/j.rcim.2019.101844.
- Rodríguez, M. L. R., S. Kubler, A. De Giorgio, M. Cordy, J. Robert and Y. Le Traon (2022). Multi-agent deep reinforcement learning based Predictive Maintenance on parallel machines. *Robotics and Computer-Integrated Manufacturing* 78, p. 102406. doi: 10.1016/j.rcim.2022.102406.
- Rojas, R. A. and E. Rauch (2019). From a literature review to a conceptual framework of enablers for smart manufacturing control. *The International Journal of Advanced Manufacturing Technology* 104(1-4), p. 517–533. doi: 10.1007/s00170-019-03854-4.
- Sakr, A. H., A. Aboelhasan, S. Yacout and S. Bassetto (2021). Simulation and deep reinforcement learning for adaptive dispatching in semiconductor manufacturing systems. *Journal of Intelligent Manufacturing* 34(3), p. 1311–1324. doi: 10.1007/s10845-021-01851-7.
- Sallez, Y., T. Berger, S. Raileanu, S. Chaabane and D. Trentesaux (2010). Semi-heterarchical control of FMS: From theory to application. *Engineering Applications of Artificial Intelligence* 23(8), p. 1314–1326. doi: 10.1016/j.engappai.2010.06.013.
- Samsonov, V., K. Ben Hicham and T. Meisen (2022). Reinforcement Learning in Manufacturing Control: Baselines, challenges and ways forward. *Engineering Applications of Artificial Intelligence* 112, p. 104868. doi: 10.1016/j.engappai.2022.104868.
- Samsonov, V., M. Kemmerling, M. Paegert, D. Lütticke, F. Sauermann, A. Gützlaff, G. Schuh and T. Meisen (2021). Manufacturing Control in Job Shop Environments with Reinforcement Learning:. In: *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*, Online Streaming, — Select a Country —, p. 589–597. SCITEPRESS - Science and Technology Publications. doi: 10.5220/0010202405890597.
- Schmidt, M. and P. Nyhuis (2021). *Produktionsplanung und -steuerung im Hannoveraner Lieferkettenmodell: innerbetrieblicher Abgleich logistischer Zielgrößen*. Berlin [Heidelberg]: Springer Vieweg. ISBN: 978-3-662-63896-5.
- Shavandi, A. and M. Khedmati (2022). A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. *Expert Systems with Applications* 208, p. 118124. doi: 10.1016/j.eswa.2022.118124.
- Shiue, Y.-R., K.-C. Lee and C.-T. Su (2018). Real-time scheduling for a smart factory using a reinforcement learning approach. *Computers & Industrial Engineering* 125, p. 604–614. doi: 10.1016/j.cie.2018.03.039.
- Su, J., J. Huang, S. Adams, Q. Chang and P. A. Beling (2022). Deep multi-agent reinforcement learning for multi-level preventive maintenance in manufacturing systems. *Expert Systems with Applications* 192, p. 116323. doi: 10.1016/j.eswa.2021.116323.

- Sutton, R. S. and A. G. Barto (2017). *Reinforcement learning: an introduction* (2nd ed.). Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press. ISBN: 978-0-262-03924-6.
- Tao, H., J. Qiu, Y. Chen, V. Stojanovic and L. Cheng (2023). Unsupervised cross-domain rolling bearing fault diagnosis based on time-frequency information fusion. *Journal of the Franklin Institute* 360(2), p. 1454–1477. doi: 10.1016/j.jfranklin.2022.11.004.
- Tassel, P., M. Gebser and K. Schekotihin (2021). A Reinforcement Learning Environment For Job-Shop Scheduling. doi: 10.48550/ARXIV.2104.03760.
- Umlauf, M., M. Schranz and W. Elmenreich (2022). SwarmFabSim: A Simulation Framework for Bottom-up Optimization in Flexible Job-Shop Scheduling using NetLogo:. In: *Proceedings of the 12th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, Lisbon, Portugal, p. 271–279. SCITEPRESS - Science and Technology Publications. doi: 10.5220/0011274700003274.
- Uzsoy, R., L. K. Church, I. M. Ovacik and J. Hinchman (1993). Performance evaluation of dispatching rules for semiconductor testing operations. *Journal of Electronics Manufacturing* 03(02), p. 95–105. doi: 10.1142/S0960313193000115.
- Valckenaers, P., F. Bonneville, H. Van Brussel, L. Bongaerts and J. Wyns (1994). Results of the holonic control system benchmark at KU Leuven. In: *Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, Troy, NY, USA, p. 128–133. IEEE Comput. Soc. Press. doi: 10.1109/CIMAT.1994.389083.
- Venturelli, D., D. J. J. Marchand and G. Rojo (2015). Quantum Annealing Implementation of Job-Shop Scheduling. doi: 10.48550/ARXIV.1506.08479.
- Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmuller, T. Bauernhansl, A. Knapp and A. Kyek (2018). Deep reinforcement learning for semiconductor production scheduling. In: *2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, Saratoga Springs, NY, USA. doi: 10.1109/ASMC.2018.8373191.
- Waubert De Puiseau, C., J. Peters, C. Dörpelkus, H. Tercan and T. Meisen (2023). schlably: A Python framework for deep reinforcement learning based scheduling experiments. *SoftwareX* 22, p. 101383. doi: 10.1016/j.softx.2023.101383.
- Weichert, D., P. Link, A. Stoll, S. Rüping, S. Ihlenfeldt and S. Wrobel (2019). A review of machine learning for the optimization of production processes. *The International Journal of Advanced Manufacturing Technology* 104(5-8), p. 1889–1902. doi: 10.1007/s00170-019-03988-5.
- Zhang, C., W. Song, Z. Cao, J. Zhang, P. S. Tan and C. Xu (2020). Learning to Dispatch for Job Shop Scheduling via Deep Reinforcement Learning. doi: 10.48550/ARXIV.2010.12367.

- Zhang, H. and U. Roy (2019). A semantics-based dispatching rule selection approach for job shop scheduling. *Journal of Intelligent Manufacturing* 30(7), p. 2759–2779. doi: 10.1007/s10845-018-1421-z.
- Zhang, H.-C. and S. H. Huang (1995). Applications of neural networks in manufacturing: a state-of-the-art survey. *International Journal of Production Research* 33(3), p. 705–728. doi: 10.1080/00207549508930175.
- Zhang, J., G. Ding, Y. Zou, S. Qin and J. Fu (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing* 30(4), p. 1809–1830. doi: 10.1007/s10845-017-1350-2.
- Zhang, Y., R. Bai, R. Qu, C. Tu and J. Jin (2022). A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties. *European Journal of Operational Research* 300(2), p. 418–427. doi: 10.1016/j.ejor.2021.10.032.
- Zhao, Y. and H. Zhang (2021). Application of Machine Learning and Rule Scheduling in a Job-Shop Production Control System. *International Journal of Simulation Modelling* 20(2), p. 410–421. doi: 10.2507/IJSIMM20-2-CO10.
- Zheng, S., C. Gupta and S. Serita (2020). Manufacturing Dispatching Using Reinforcement and Transfer Learning. *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, p. 655–671. doi: 10.1007/978-3-030-46133-1_39.
- Zhou, C., H. Tao, Y. Chen, V. Stojanovic and W. Paszke (2022). Robust point-to-point iterative learning control for constrained systems: A minimum energy approach. *International Journal of Robust and Nonlinear Control* 32(18), p. 10139–10161. doi: 10.1002/rnc.6354.
- Zhou, L., Z. Jiang, N. Geng, Y. Niu, F. Cui, K. Liu and N. Qi (2022). Production and operations management for intelligent manufacturing: a systematic literature review. *International Journal of Production Research* 60(2), p. 808–846. doi: 10.1080/00207543.2021.2017055.
- Zhou, Y., J.-j. Yang and Z. Huang (2020). Automatic design of scheduling policies for dynamic flexible job shop scheduling via surrogate-assisted cooperative co-evolution genetic programming. *International Journal of Production Research* 58(9), p. 2561–2580. doi: 10.1080/00207543.2019.1620362.

8 Publication 4

A deep reinforcement learning based hyper-heuristic for modular production control

Marcel Panzer^{1a}, Benedict Bender^a, and Norbert Gronau^a

^a *Chair of Business Informatics, Processes and Systems, University of Potsdam,
Karl-Marx-Street 67, 14482 Potsdam, Germany*

ABSTRACT

In nowadays production, fluctuations in demand, shortening product life-cycles, and highly configurable products require an adaptive and robust control approach to maintain competitiveness. This approach must not only optimize desired production objectives but also cope with unforeseen machine failures, rush orders, and changes in short-term demand. Previous control approaches were often implemented using a single operations layer and a standalone deep learning approach, which may not adequately address the complex organizational demands of modern manufacturing systems. To address this challenge, we propose a hyper-heuristics control model within a semi-heterarchical production system, in which multiple manufacturing and distribution agents are spread across pre-defined modules. The agents employ a deep reinforcement learning algorithm to learn a policy for selecting low-level heuristics in a situation-specific manner, thereby leveraging system performance and adaptability. We tested our approach in simulation and transferred it to a hybrid production environment. By that we were able to demonstrate its multi-objective optimization capabilities compared to conventional approaches in terms of mean throughput time, tardiness, and processing of prioritized orders in a multi-layered production system. The modular design is promising in reducing the overall system complexity and facilitates a quick and seamless integration into other scenarios.

Keywords

Production control; modular production; multi-agent system; deep reinforcement learning; deep learning; multi-objective optimization

¹Corresponding author

Submitted to the International Journal of Production Research on 15 March 2023, accepted on 22 June 2023.

8.1 Introduction

With growing challenges of fluctuating demand, market volatility, and increasingly complex manufacturing processes, there is a strong need for a resilient and adaptable production control (Kapoor et al., 2021). The production control must not only cope with unforeseen machine failures while managing high product individualization levels and dynamic processes, but also handle large amounts of data under sustainable matters which requires a sensible data collection and processing (Tao et al., 2018; Bueno et al., 2020). To cope with these challenges, companies must seize the opportunity to implement control approaches that can cope with varying production conditions and facilitate ongoing optimization of performance indicators to increase competitiveness (Lee et al., 2018; Grassi et al., 2020; Parente et al., 2020).

Recent advancements in cyber-physical systems, the industrial internet of things, and other related technologies already facilitated a widespread data collection and processing in production systems (Lee et al., 2015, 2016; Lass and Gronau, 2020; Ritterbusch and Teichmann, 2023). By leveraging the *Industry 4.0* principles, these technologies can unlock significant process potentials and competitive advantages (Parente et al., 2020). In production control practice, however, conventional algorithms are often applied, such as the *First-in-First-out* rule (*FiFo*), which do not guarantee global optimality while others are hard-coded and layout specific (Mönch et al., 2013; Kuhnle et al., 2021), which do not meet recent demands regarding flexibility.

A recent approach to process large amounts of input data, deep reinforcement learning (RL), was increasingly applied in production control in recent years (Sutton and Barto, 2017; Samsonov et al., 2021; Panzer and Bender, 2022). Deep RL is characterized by its interactive, trial-and-error learning principle and often demonstrated superior performance compared to conventional production control approaches. Its online optimization and direct data processing capabilities make it particularly well-suited for real-time decision making in fast-paced applications, setting it apart from other AI-based methods that may require longer computation times (Chang et al., 2022). Despite the considerable attention paid to deep RL-based single-agent systems, multi-agent-based systems have received comparatively less attention due to the significant challenges associated with agent orchestration and communication design (Panzer and Bender, 2022). Yet, they can assist in achieving both, local and global, performance objectives and develop robust control policies (Tampuu et al., 2017).

To combine the advantages of deep RL and multi-agent-based systems to cope with recent demands, this paper proposes a novel hyper-heuristics based control approach for modular multi-agent production systems that utilizes both, distributed resources and deep RL. The hyper heuristic is applied for control optimization that utilizes deep neural networks for selection of low-level heuristics. Each agent deploys its own neural networks, tailored to its specific modular production environment which is transferable to similar systems. To leverage adaptability and

scalability, our further motivation is to implement the approach within a semi-heterarchical production to cope with the prevailing organizational challenges of multi-layered production systems. The key contribution is an adaptive control approach that combines a deep learning based control with a adaptive and scalable production organization that optimizes pre-defined production performance indicators. The approach is designed to handle spontaneous events, such as machine failures and rush orders, while ensuring stability and facilitating a seamless transition to real-world production scenarios.

The remainder of the paper is organized as follows: In Section 8.2, basics of deep RL and multi-agent based production control are outlined and the research objective is specified. The conceptual design and artifact requirements are defined in Section 8.3. Results are outlined and evaluated in Section 8.4, and transferred to a real test-bed in Section 8.5. A discussion is outlined in Section 8.6 and a conclusion is given in Section 8.7.

8.2 Problem statement

This section first discusses the basics and organizations of modular and matrix production systems and elaborates on the principles of (deep) RL. Finally, results of a systematic literature review of the combination of these in decentralized and multi-agent based production control is conducted for defining the specific research objectives.

8.2.1 Modular and semi-heterarchical production systems

Nowadays, adaptability is vital in production systems to handle machine failures, rush orders, and other disruptions. To cope with such internal and external disruptions, modular production systems were designed and often validated in simulated approaches (May et al., 2021). Such modular systems allow individualized production processes through line-less control and the ability to define arbitrary production flows by using automated guided vehicles, which enable a detached process execution (Bányai, 2021; May et al., 2021; Mayer et al., 2021). This flexibility leads to higher use of resources through sharing strategies, shorter transportation routes, and reduced buffer stocks, as in Schenk et al. (2010) or Greschke et al. (2014). However, despite its advantages, modular production approaches suffer from an increased control complexity due to the highly flexible operation of manufacturing modules and the large control solution space (Schenk et al., 2010; Schmidtke et al., 2021). The increased number of potential process paths leads to large actions spaces that raises optimization complexity and a proper extraction of relevant information through a neural network gets more complex.

With this regard, previous research emphasized the importance of decentralizing decision-making among production agents, allowing them to make decisions based on their specific task and

available resources to leverage their reasoning, perception, and action capabilities (Parunak et al., 1986; Balaji and Srinivasan, 2010). Weiss (2001) particularly emphasizes the flexible and re-configurable properties of multi-agent structures as conventional decentralized control approaches. In a more recent review, Herrera et al. (2020) further emphasizes the relevance of multi-agent systems for existing and planned real-world applications. A specific differentiation of such multi-agent systems is established by sub-dividing them into organizational forms, depending on the allocation, grouping, and interaction of the agents. While a hierarchy is characterized by a multitude of fixed master-slave relationships, a heterarchy consists primarily of peer-level relationships with distributed privileges to fulfill global and local objectives (Baker, 1998; Bongaerts et al., 2000). Hierarchical systems are rather static, whereas heterarchical organizations suffer from local optimization tendencies and myopic behavior due to the lack of master-slave relationships (Sallez et al., 2010).

A semi-heterarchical production system seeks to combine the advantages of both hierarchical and heterarchical concepts. It achieves a high integrity of the sub-components through its hierarchical structure (Valckenaers et al., 1994) while maintaining a high reactivity and robustness through the distribution principle within the heterarchical systems (Groover, 2019). The semi-heterarchical concept simultaneously enables both, long-term and short-term objectives to be reached, and allows the corresponding parameters to be optimized (Sallez et al., 2010). Implementations of this concept were made by Grassi et al. (2020) and Grassi et al. (2021) in different production levels. This facilitates a multi-agent system that enables the allocation of agents based on their functional scope. The semi-heterarchical concept was further implemented within a single control structure and by establishing domain-wise clustering by Borangiu et al. (2009) and Borangiu et al. (2010) in the field of product-driven scheduling and by Zambrano Rey et al. (2013) in the field of flexible manufacturing control. By using a 2-layer approach, Borangiu et al. (2010) fulfilled different objective horizons and obtained a comparably higher robustness and agility of the system. Through the semi-heterarchical approach, Zambrano Rey et al. (2013) was further able to achieve control over the otherwise myopic agent behavior.

8.2.2 Deep reinforcement learning based hyper-heuristic

To cope with the dynamic control requirements and allow for an adaptive control deep RL was implemented in production control approaches (Bahrpeyma and Reichelt, 2022; Estes et al., 2022). It made the leap to competitiveness especially with its successful implementation to the Atari environment, and has since become increasingly appealing for complex optimization problems (Mnih et al., 2013). Due to its particularly interactive learning strategy and the neural network's ability to process large state inputs, deep RL can be tailored to a variety of data-centric online applications (Baer et al., 2020). As indicated in recent reviews, particularly value-based RL

approaches were widely deployed in production control and demonstrated superior performances (Bahrpeyma and Reichelt, 2022; Panzer et al., 2022).

To facilitate the fundamental integration capability of deep RL to production control, the problem under consideration must satisfy the Markov property and correspond to a Markov Decision Process (MDP). Besides the rigid definition of the considered scope, the Markov assumption must be met, which implies that all future production states only depend on the current state. This constitutes the underlying assumption of our approach and of the later designed discrete-event based simulation (Sutton and Barto, 2017). Q-learning is a variant of RL, which is a model-free, off-policy RL algorithm that exploits a action- or Q-value function. The Q-value function, see Equation (1), is typically defined based on an agent's expected cumulative reward in Equation (2), which follows its current policy, derived from the Bellman equation (Bellman, 1957).

$$Q(s_t, a_t) = r + \gamma \max(Q(s_{t+1}, a_{t+1})) \quad (1)$$

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad (2)$$

In following functions $Q(s_t, a_t)$ resembles the Q-value for a state s_t and action a_t at a certain time t . r is the immediate reward received after taking action a in state s_t , γ is the discount factor, and $\max(Q(s_{t+1}, a_{t+1}))$ is the maximum Q-value over the next states s_{t+1} and actions a_{t+1} that can be executed from state s_{t+1} (Sutton and Barto, 2017).

Whereas in conventional Q-learning a table is used to map Q-values, deep RL exploits a deep neural network to function approximator to map the policy of an agent, which is frequently updated based on past made experiences. The neural network enables the agent to learn from high-dimensional and complex input data such as raw sensory information. It can handle non-linear and non-convex environments better than traditional RL, allowing it to be more accurate and efficient in learning. During learning process, often a batch replay is used, which iteratively trains and fits the network based on the stored batch data. The neural network, also known as the Q-network, takes the current state s_t of the agent as input and outputs a Q-value for each possible action a_t . The Q-network is trained to minimize the difference between the predicted Q-values and the target Q-values, which are calculated using the above mentioned Bellman equation. Thereby, Equation (3) is utilized for updating the weights of the Q-network in deep Q-learning (DQN), wherein w represents the weights of the Q-network, α denotes the learning rate, and E signifies the loss function, defined as the mean squared error between the predicted Q-values and the target Q-values, as shown in Equation (4).

$$w = w - \alpha \nabla_w(E) \quad (3)$$

$$E = (Q(s, a) - (r + \gamma \max(Q(s_{t+1}, a_{t+1}))))^2 \quad (4)$$

The DQN equation is derived by combining Equations (1)-(4), where the Q-network is trained to approximate the Q-values by minimizing the loss function using the Bellman equation, as summarized in Equation (5). To further stabilize learning and performance, a target network with weights θ^- is introduced and used to calculate $Q(s_{t+1}, a_{t+1})$ for the next states (Mnih et al., 2013; Mnih et al., 2015).

$$Q(s_t, a_t, \theta) \leftarrow Q(s_t, a_t, \theta) + \alpha [r + \gamma \max Q(s_{t+1}, a_{t+1}, \theta^-) - Q(s_t, a_t, \theta)] \quad (5)$$

Building up on deep RL, a hyper-heuristic is an optimization model that utilizes a machine or deep learning algorithm such as the DQN to learn a high-level policy for selecting and adapting low-level policies. Due to the deep learning algorithm, the hyper-heuristic possess the capability to adapt to specific optimization tasks, and thus effectively utilize the inherent capabilities and process logic's of low-level heuristics. This enables an automation of the design process, and allows for the utilization of knowledge from online machine learning algorithms as an optimizer, resulting in the derivation of near-optimal scheduling and dispatching policies, which are based on established and more comprehensible low-level heuristics (Burke et al., 2010, 2019; Drake et al., 2020). During operation, the smart agent is trained to select one suitable heuristic from a pre-defined set depending on the received production state. The objective of a deep RL based hyper-heuristic is to improve the performance of the underlying optimization problem by utilizing the strengths and process implications of multiple low-level heuristics as illustrated in Figure 8.1 (Van Ekeris et al., 2021; Zhang et al., 2022).

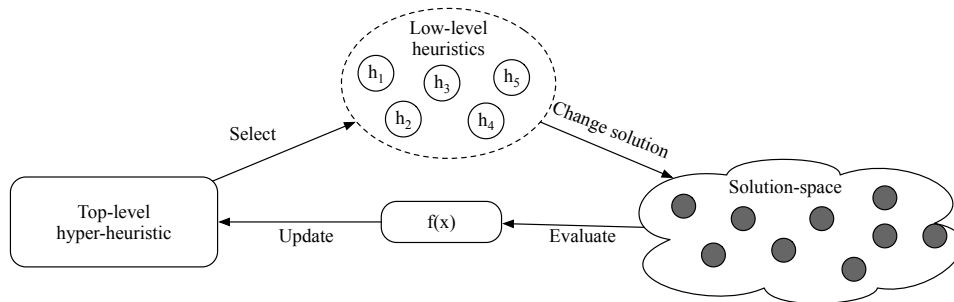


Figure 8.1 Hyper-heuristics based optimization approach (Cowling et al., 2001; Swiercz, 2017)

8.2.3 Deep RL and multi-agent based production control

Prevailing approaches in multi-agent based production control already try to leverage deep RL to benefit from a decentralized and online decision making and optimization process. To get a comprehensive overview of the research field and trends, we searched the databases *Scopus* and *WebofScience* to identify relevant scientific papers.

Several studies have explored different approaches to improve production performance indicators

such as utilization rates or order tardiness. Malus et al. (2020) proposed an order bidding mechanism for autonomous mobile robots that utilized a joint global reward to minimize delays, optimizing global utilization efficiency upon part completion and locally accepted bids. The agents could bid between 0 and 1 based on their respective states and proximal policy optimization (PPO) output, and the order with the highest bid was assigned for the dispatch task. The PPO ensured updates of the policy of not being too large, thus providing a balance between policy exploration and exploitation, making it more sample-efficient and stable compared to other RL algorithms (Schulman et al., 2017). To cope with the dynamics of inherent order scheduling, Hammami et al. (2017) proposed an multi-agent system based on simultaneous learning and information sharing between agents to reduce average delay. Decisional agents, responsible for overall decision-making process and order dispatching, were associated with choice agents that selected the best neural network for the decisional agent based on the desired performance optimization criteria. In the dispatching approaches conducted by Dittrich and Fohlmeister (2020) and Hofmann et al. (2020), a centralized DQN decision module was employed for training purposes. This central module functioned as a repository for storing and updating the dispatching policy among all agents, and provided the current control policy upon request. Dittrich and Fohlmeister (2020) created a job shop with three process steps (turning, milling, and assembly), where agents could access local and global information to allocate orders to machine within a machine group to optimize mean cycle time. These agents could request necessary local and global system information to facilitate informed decision-making which was facilitated by globally defined rewards, which were then propagated to individual agents. Other studies focused on different training strategies to improve production control. Waschneck et al. (2018) implemented a strategy where one network was trained at a time for stability and learning speed reasons, and subsequently each wafer manufacturing workstation was controlled by one neural network at a time to optimize for maximum uptime utilization as a global goal. The network input comprised all possible lot positions and an idle option, while the network output included machine states (capacity, availability, etc.) and job states (type, progress). To optimize the product sequence in car manufacturing, Gros et al. (2020) used an iterative learning strategy to prevent instabilities caused by parallel training of several agents, determining the output sequence of cars to a buffer after finishing paint jobs. This minimized costs caused by inefficient car sequences and non-balanced flow of goods in subsequent manufacturing processes. Overbeck et al. (2021) utilized PPO agents and hyper-parameter tuning, to determine the optimal action in an automated assembly system adhering to Chaku-Chaku principles, where workers were tasked with loading machines and transporting orders. An evaluation in a real assembly cell for automotive parts demonstrated an improvement in decision quality over time and a more produced parts.

The aforementioned approaches primarily dealt with control problems in conventional job shops.

Initial approaches in matrix systems were proposed by Gankin et al. (2021), May et al. (2021), and Hofmann et al. (2020). In Hofmann et al. (2020), agents received immediate rewards after each operational step for a chosen action and a delayed reward based on the total global cycle time after an order was completed, which accelerated the learning process and reduced order throughput times. The simulated system featured 10 workstations and several autonomous guided vehicles (AGVs) that could perform multiple process steps and were fully flexibly interconnected. Meanwhile, May et al. (2021) implemented an economic bidding approach to increase utilization efficiency in a matrix-structured production system. The approach to maximized the operational profit for each agent independently and optimized the execution time and resource utilization efficiency against conventional heuristics. Gankin et al. (2021) introduced a first large-scale matrix layout comprising 25 machines, which was based on the modular approach developed by Mayer et al. (2021). Two distinct product types were manufactured, each involving 13 process steps. To reduce decision complexity, an action masking mechanism was implemented for preventing the selection of incorrect actions as each process step could be performed only at specific machines. All 20 transport units were trained in parallel as DQN agents, and the same neural network and buffer were used as the central decision instance and to facilitate experience sharing between the agents.

8.2.4 Problem formulation and contribution

From the previous literature set, several performant applications can be observed, however, most approaches are rather specific, such as the wafer fabrication or the car paint buffer re-ordering. A more scalable and adaptive approach is given with the matrix approaches of Mayer et al. (2021) or Gankin et al. (2021). However, these assume a matrix structure and are less focused on the clustering of production units, and, in case of Gankin et al. (2021), exploit a central decision entity. Also, all mentioned approaches deploy a single-staged control organization at the operations level. In addition to the application and organization scope, which is summarized in Table 8.1, the algorithmic approaches are often self-contained AI algorithms.

Table 8.1 highlights three fields of potential research in production control, algorithmic (1), organizational (2), and optimization opportunities (3). The algorithmic field (1) currently lacks a deep RL based hyper-heuristics approach, which operates at a higher level and can quickly select lower-level heuristics, as opposed to meta-heuristics that serve as search process optimizers or general guidelines (Bányai, 2021). Previous research indicated that a deep RL-based hyper-heuristic can outperform population-based meta-heuristics, such as genetic algorithms, in terms of performance and interpretability (Zhang et al., 2022; Kallestad et al., 2023). Additionally, hyper-heuristics have benefit from fast computation of operations, as in Chang et al. (2022); Liu et al. (2020), making them particularly suitable for real-time environments.

Application	Algorithm	Training strategy	Control strategy	Agent interaction	Objective parameter	Orga. levels	Transfer scope	Year	Source
Car buffer	DQN	Iterative learning	Distributed	-	Cost/ decision time	1	Simulation	2020	Gros et al.
Chaku-chaku line	PPO	Shared PPO module	Central	-	Utilization/ throughput	1	Simulation	2021	Overbeck et al.
Job shop	SA	Concurrent learning	Distributed	Agent information exchange	Mean tardiness	1	Simulation	2017	Hammami et al.
	DQN	Iterative DQN/ heuristics learning	Distributed	Global rewards	WIP/ uptime utilization	1	Simulation	2018	Waschneck et al.
	DQN	Shared DQN module	Central	Agent information exchange	Mean cycle time	1	Simulation	2020	Dittrich et al.
	DRL (TD3)	Concurrent learning	Distributed	Order bidding mechanism	Tardiness	1	Simulation	2020	Malus et al.
Matrix production	DQN	Shared DQN module	-	Agent state information	Throughput time	1	Simulation	2020	Hofmann et al.
	DQN	Shared DQN module	Central	-	Throughput	1	Simulation	2021	Gankin et al.
	PPO	-	Distributed	Economic bidding	Execution time/ utilization eff.	1	Simulation	2021	May et al.
Matrix/modular production	DQN-based hyper-heuristic	Concurrent learning	Distributed	Agent and cell states	Throughput time/ priorities/ tardiness	> 1	Simulation and reality	Our approach	

Table 8.1 Deep RL based multi-agent approaches in production control

Regarding the organizational design (2), our approach deploys shopfloor and distribution layers to facilitate modular and semi-heterarchical production processes. The use of a deep RL-based multi-agent system is emphasized due to its collaborative possibilities and the ability to cope with larger systems requirements (Tampuu et al., 2017). Despite the current focus on single-agent environments (Esteso et al., 2022), our approach takes advantage of a distributed and semi-heterarchical agent organization and manages system complexity by decomposing the overall complexity into respective fragments of only processing relevant information. The approach will be applied in a modular production environment that allows pre-defined tool bundling and configuration of machine groups, similar to real production. The modularity aims to exploit product-specific machine synergies while reducing coordination complexity and increasing scalability. This aligns with the requirement for a common modeling approach, as articulated by Mourtzis (2020), which is supported by the proposed standardized production modules within the underlying simulation framework. Consequently, the simulation can serve as an evaluation tool for our deep learning control framework prior to its deployment in the intended (hybrid) real-world environment, functioning as an progressive Industry 4.0 test-bed, as emphasized by de Paula Ferreira et al. (2020) and de Paula Ferreira et al. (2022).

Regarding the optimization task (3), various approaches were proposed to optimize one or two objectives such as tardiness, throughput, or utilization. However, two variables that were not yet explored are order priority and urgency. Especially in modern production with customer-oriented services, order priority and urgency directly affect resource allocation and operational efficiency, to meet customers needs. Prioritized or premium customer groups as well as rush orders represent a significant source of revenue, and require a flexible production control that can adapt to fluctuating demands. To address this order features effectively, a combined measure or reward function is designed, that balances the prioritization of orders according to the pre-defined objectives of mean throughput-time, tardiness as well as order priority and urgency.

To the best of our knowledge, this is the first approach of a hyper-heuristics based production control in a modular production system. We seek to leverage production performance and control production complexity through a layered approach to enable a robust and adaptive control. Furthermore, this will be the first approach that transfers a control approach of deep RL based hyper-heuristics control to a real application.

8.3 Conceptual design

To ensure a systematic approach for reaching the research objectives, we followed the design science research methodology according to Peffers et al. (2007), see Figure 8.2. The first two steps of problem identification and objectives definition were addressed in the previous sections, which are now followed by constructing the research artefact as the third step. To satisfy the

emerging dynamic requirements and deliver an adaptable and scalable simulation approach, an appropriate simulation framework must first be chosen. The framework should seamlessly integrate the hyper-heuristic control approach, to enable decentralized decision-making and leverage production performance.

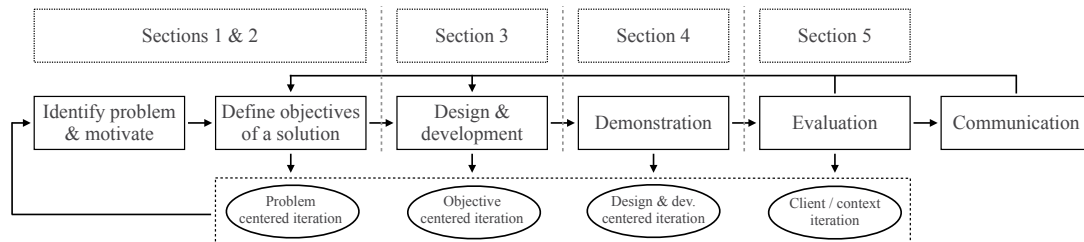


Figure 8.2 Pursued *DSRM* methodology (Peppers et al., 2007)

8.3.1 Simulation approach

To implement the simulation, the production simulation framework *CoBra*, developed at our research department, was utilized. *CoBra* is based on the *SimPy* simulation library, commonly used in the field of discrete-event production simulation, as done in previous works, e.g., Kuhnle et al. (2020, 2021); Liu et al. (2022). This tool enables the rapid creation of modular production environments, and supports the arbitrary design of production processes and control rules. A key requirement is the ability to seamlessly transfer the approach to a real environment. To achieve this, real-world requirements such as machine failures, maintenance efforts, randomly determined order sequences, and other process-dependent parameters were incorporated into the simulation. However, considering all the information for decision-making is impractical, necessitating a systematic construction of the state vector that is used for feeding the neural network, as outlined in Section 8.3.2.1.

In our approach, the modular production system consists of entities or groups of distributed autonomous agents, referred to as manufacturing (1) and distribution (2) modules (Giret and Botti, 2004). The agents operate in parallel, making decisions based on the information they individually receive, thereby reducing the need for a centralized control model. The base/bottom layer is composed of several manufacturing agents, that are responsible for the processing of goods (refer to the bottom of Figure 8.3). These agents serve in modules that are typically specialized in conducting specific manufacturing processes, such as welding or assembly. The upper layers consist of distribution modules, that are responsible for the production coordination and distribution of goods (see top three layers in Figure 8.3). All agents can work in collaboration with other agents to process intermediate products. They have the capability to make decisions based on a shared policy and the individually received information. This facilitates the system's adaptability to adapt to changing production conditions in a flexible and efficient manner. The

CoBra framework effectively enables the conceptual design of such semi-heterarchical systems up to a fully matrix-like production, that can be controlled by either deep learning based or conventional approaches. During the simulation, all agents are trained concurrently based on the obtained rewards, following the epsilon-greedy strategy. This facilitates an exploration of the action space and potential policies and provides a broad database for the batch replay.

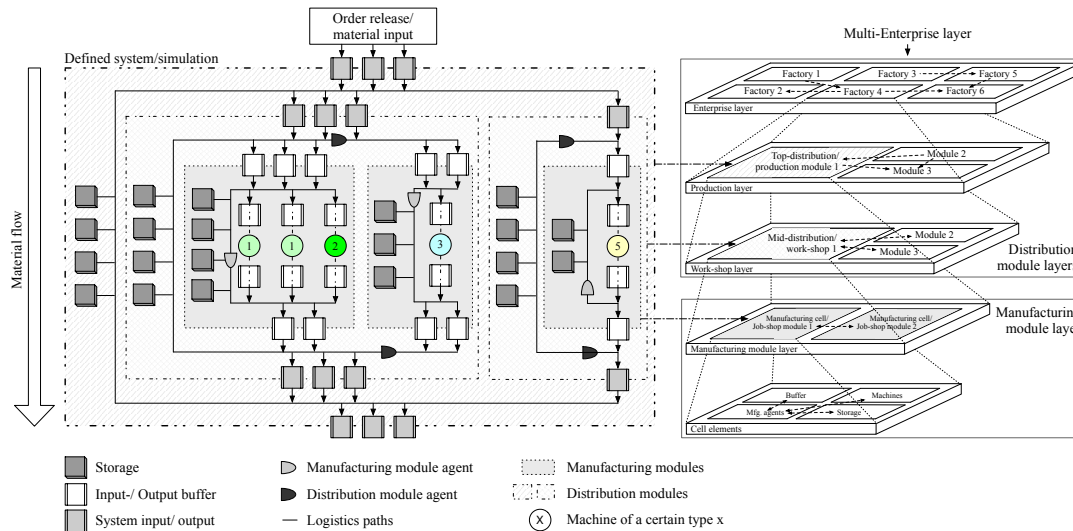


Figure 8.3 Projected multi-agent and semi-heterarchical system; right: adapted from Sallez et al. (2010)

Illustration of our modular production system organized into hierarchical layers, with manufacturing layer/shop-floor at the bottom and distribution layers for logistics activities at higher levels. Each layer comprises multiple interconnected modules, following a semi-heterarchical organization

8.3.2 Variable hyper-heuristic design

The development of a hyper-heuristic involves multiple steps that address the third step of the *DSRM* methodology (Peffer et al., 2007). Following problem identification and objective definition, the hyper-heuristic must be designed in compliance with system constraints, available information, and the performance indicators that require optimization. To enable an adaptive and online decision-making, and maintain the accessibility of the *CoBra* and *SimPy* simulation framework, *TensorFlow* was used for implementing deep learning functionalities. Based on that, the state vector design is first dealt with (see Section 8.3.2.1), that represents the current production state and also the system's interface that allows the hyper-heuristic to access the essential information for the decision making process. Second, in Section 8.3.2.2, lower-level heuristics are identified and the action space vector is constructed that addresses specific key performance criteria. Finally, in Section 8.3.2.3, the training and reward mechanisms are considered, which have an crucial impact on the learning process and system performance.

8.3.2.1 State space design

The state space design is an crucial step towards achieving an efficient and performant production control and should correlate to the targeted rewards and overall objectives (Kuhnle et al., 2020). The challenge is to select a state-set that contains all the essential information while avoiding the inclusion of unnecessary inputs. This may include general order information such as priorities, as well as process-related information pertaining to order throughput times or current tardiness. Machine information, including their operational status, maintenance needs, and current setup, could also be taken into account for specific scenarios if necessary. The state vector in our approach integrates buffers, storage, machine and agent order information, such as occupation type, along with associated order details, granting the specific agent's access to a comprehensive module state and global processing information, thereby facilitating situation-dependent decision-making.

Following other approaches (Kuhnle et al., 2021; Overbeck et al., 2021), we apply a *min-max-normalization* for the various state inputs to scale the gathered values within a predefined and constrained range. This should leverage the performance of the neural network and not only enables a smoother mapping between states and actions while mitigating outliers and disproportionate input variables.

For time-related state inputs, the normalization results in a state value range of $[-1, 1]$ for each order n out of all processable orders o within the respective module, as denoted in Equation (6). A processable order is defined by its feasibility of having one or more possible subsequent steps, i.e., when the input buffer of the machine with the next needed processing step is unblocked, or when the order can be transported from module input to an available storage slot. Concerning the part of Equation (6), $s_{n,tpt}$ is calculated individually for each order concerning their global systemic and local module start time. For the discrete state space of order priorities, the state inputs are discretized to $[0, 1]$, signifying an input of $s_{n,prio} = 0$ for normal orders, 0.5 for prioritized orders, and 1 for high-priority orders. This is intended to ensure more stable gradients, faster training, and correct weight initialization.

$$\begin{aligned}
[1] \quad s_{n,tpt} &= \left(1 - 2 \frac{t_{tpt,max} - t_{tpt,n}}{t_{tpt,max} - t_{tpt,min}} \right) \\
[2] \quad s_{n,due_to} &= \left(2 \frac{t_{dt,max} - t_{dt,n}}{t_{dt,max} - t_{dt,min}} - 1 \right) \\
[3] \quad s_{n,prio} &= \begin{cases} 0 & \text{if } prio_n = 0 \\ 0.5 & \text{if } prio_n = 1, \text{ (prioritized order)} \\ 1 & \text{if } prio_n = 2, \text{ (high priority order)} \end{cases}
\end{aligned} \tag{6}$$

The resulting state vector S_t , which is used as the input for the neural networks in subsequent

processes, was derived from iterative testing and consistent mapping to targeted production performance indicators, which encompass the order throughput time, tardiness, and order priorities. To achieve this, the due date, local and global start time, and priority of all processable orders o at each available slot are concatenated, as outlined in Equation (7). For orders that are blocked or reserved by other agents, undergoing processing by a machine, or situated in input buffers, the state input for each metric ($s_{tpt_due_to_prio}$) is assigned a value of 0 to maintain a constant state size. Additional module positions in Equation (7) correspond to those depicted in Figure 8.3.

$$S_t = \left(\underbrace{1, 0.8, -0.2}_{\text{Cell input buffer slots}}, \underbrace{\dots}_{\text{Further module positions}}, \underbrace{0, 0, -0.5}_{\text{Storage slots}} + s_{tpt\ global} + s_{due\ to} + s_{prio} \right) \quad (7)$$

8.3.2.2 Action space design

The action space design refers to the process of defining the set of possible actions that the deep RL agent can take at any given state and defines the manufacturing sequence. With the generic optimization approach according to Kanervisto et al. (2020), the objective is not a maximum number of actions, but a discretization of the action space and the selection of genuinely necessary actions. The former is given by the set of dispatching rules as control heuristics and the linking of each with a corresponding deep RL action. Dispatching rules can be deployed as low-level heuristics as they provide a quick and efficient selection of the next job to be processed, thereby reducing overall processing time and increasing the overall efficiency of the production process. Even though there are various dispatching rules available, some might be less effective, since they do not affect the desired performance parameter. Nevertheless, the idea of providing a wide range of production strategies can make it easier to tailor subsequent production scenarios and its specific optimization problem and constraints.

The selection of low-level dispatching rules is a crucial step before the training and optimization procedure and results in a representative rule-set that was derived from benchmarks and related approaches (Tay and Ho, 2007; Kaban et al., 2012; Bergmann et al., 2014; May et al., 2021). Also, due to the pre-defined set of dispatching rules, the action space does not increase with large layout sizes and there is no need to introduce masked actions for learning as the logic is mapped intrinsically. In the further course, we apply the local and global first-in-first-out (*FiFo*), shortest processing time (*SPT*), earliest due date (*EDD*) and highest priority (*HP*) dispatching rules as the low-level rule-set. The local and global *FiFo* rule determine the next order, out of the processable order set o , based on their local or global processing start time. The *SPT* rule selects

an order based on the time required to complete the remaining process step, which particularly beneficial in resource-constrained scenarios. The *EDD* rule selects an order based on the its due date, to meet tardiness objectives and the *HP* rule selects orders with a higher priority first.

8.3.2.3 Reward function design

The reward function is a fundamental component of the deep RL algorithm as it provides a scalar feedback signal to the agent, guiding its behavior towards maximizing the cumulative reward (Sutton and Barto, 2017). In the context of hyper-heuristics based production control and multi-objective optimization, the reward function is utilized to evaluate the performance of different low-level heuristics and guide the agent's selection towards a situation-specific and optimal control policy. To capture the desired performance criteria is crucial for the task completion and significantly affects system dynamics.

Based on the optimization criteria of throughput time, tardiness, and respective order priorities and urgencies, we derived a combined reward function that can be transferred to other scenarios. For this purpose, the total reward $R_{total} = \sum R_i$ for the chosen order n is composed of the mentioned optimization criteria i according to Equation (8). Normalizing the total return proved to be negative in the later tests. Although an optimal total return can be calculated for each state, it fluctuates and is complex to interpolate in between.

$$R_{total} = R_{tpt\ local} + R_{tpt\ global} + R_{due\ to} + R_{prio} \quad (8)$$

The rewards for the throughput time $R_{TPT\ local}$ and $R_{TPT\ global}$ are a measure of how long it takes for a order to be completed from start to finish and can be used to prioritize policies that support faster completion times within a single module or the whole system. The tardiness-related reward, $R_{due\ to}$, reflects the delay in completing an order, incentivizing the algorithm to prioritize solutions that minimize delays and achieve earlier completion times. Priority related rewards R_{prio} are used to assign different levels of relevance to the orders to prioritize orders with a higher priority orders first. The deep RL algorithm takes the rewards and penalties that are outlined in Equation (9) to adjust its decision-making process and improve the performance over time. To emphasise positive and negative actions, we normalized and raised the evaluation parameters in the respective range to calculate the reward with respect to the specific maximum reward $R_{dt}, R_{tpt}, R_{prio, 1/2}$.

$$\begin{aligned}
 [1] R_{n,tpt} &= \left(1 - 2 \frac{t_{tpt, max} - t_{tpt, n}}{t_{tpt, max} - t_{tpt, min}} \right)^5 * R_{tpt} \\
 [2] R_{n, due\ to} &= \left(2 \frac{t_{dt, max} - t_{dt, n}}{t_{dt, max} - t_{dt, min}} - 1 \right)^5 * R_{dt} \\
 [3] R_{n, prio} &= \begin{cases} 0 & \text{if } prio_n = 0 \\ R_{prio,1} & \text{if } prio_n = 1, \text{ (prioritized order)} \\ R_{prio,2} & \text{if } prio_n = 2, \text{ (high priority order)} \end{cases}
 \end{aligned} \tag{9}$$

The variable R_{tpt} represents throughput time and can be defined separately for global and local measures. Additionally, R_{prio} has a constraint that ensures its values are greater than zero, specifically $0 < R_{prio,1} < R_{prio,2}$. The testings under consideration involve rewards for orders and utilize the values $R_{dt} = 100$, and for both local and global $R_{tpt} = 100$. The rewards were iteratively determined through a sensitivity analysis and can be modified depending on the objectives. In the case that a higher-priority order is selected, it will receive a reward of either $R_{prio,1} = 100$ or a significantly increased $R_{prio,2} = 500$. Conversely, selecting a regular order when a high-priority order is available, this will result in a penalty.

8.4 Demonstration

In accordance with the *DSRM* demonstration step (Peffer et al., 2007), we will present the simulation pre-requisites in Section 8.4.1) and analyze its performance regarding the fulfillment of the pre-defined optimization performance criteria and conduct benchmark against conventional heuristics in Section 8.4.2.2. The results will facilitate making in-depth conclusions about the potential benefits and limitations of using deep RL as a top-level policy in multi-agent and multi-objective production control. The computations were carried out on a Intel Core i9-12900k CPU and 32GB of RAM. For each performance indicator, several simulations were run to obtain a representative set for training and benchmarking purposes.

8.4.1 Experimental settings

The simulation approach involves a *CoBra* model to emulate the behavior of the agents within the manufacturing system. The simulated system contains 3 layers for distributing orders and respective materials within the manufacturing system to fulfil machining steps at the distributed resources. The top distribution module $D1$ resembles the high-level control module, overseeing the operation of the mid-layer modules and system in- and output. Below, there are 2 mid-layer distribution modules $D1.1/D1.2$ that control the flow of goods between the underlying manufacturing modules. Each manufacturing module possesses specific process capabilities, and

there are 2 manufacturing mid-layer modules as illustrated in Figure 8.4 ($M1.1.1/2$; $M1.2.1/2$). The distances within and between the modules are determined based on 1 meter base heights and widths, with a distance between modules of 1 meter and a safe distance of 0.4 meters for the agents, respectively autonomous vehicles. The outer dimensions of the high-level distribution module are 13 meters in width and 6 meters in height. The speed of the agents is set to 0.5 meters per second, with a pick-up and unloading time of 0.1 minutes.

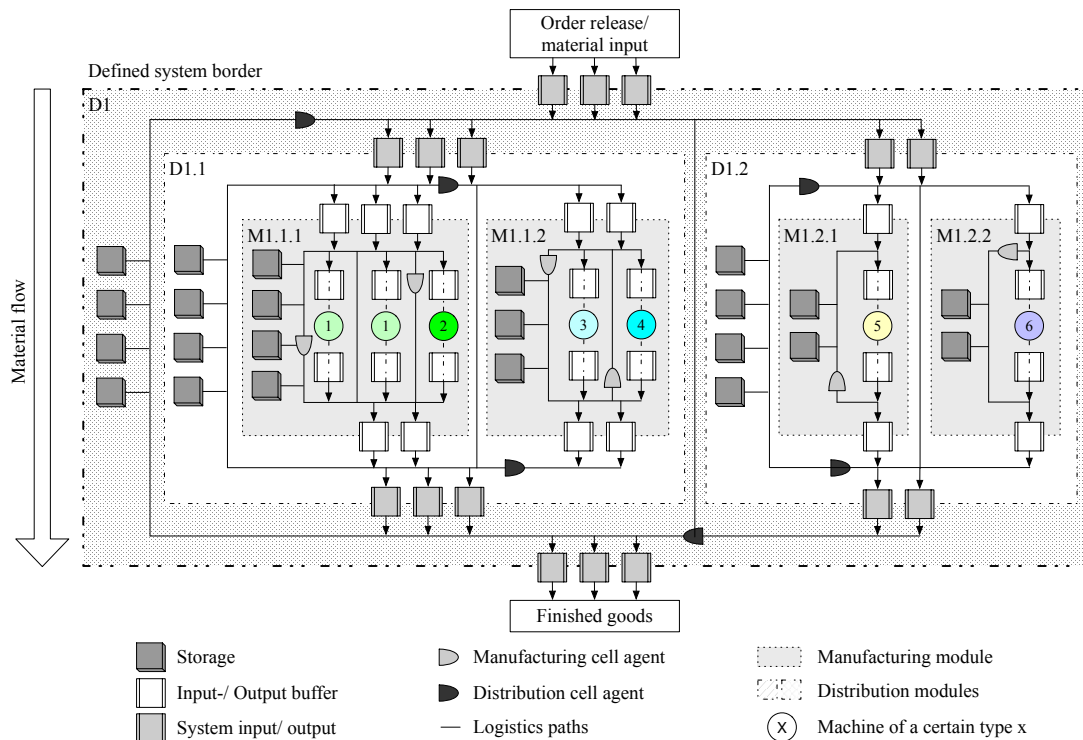


Figure 8.4 Simulated three-layer modular production system

The conducted simulations indicate that in the case of less complex modules, which i.e. contain a single machine and few add-on positions (as $M1.2.1$ or $M1.2.2$), a heuristic control mechanism is similar in effectiveness compared to utilizing a dedicated hyper-heuristic. Conversely, in cellular systems with larger state spaces and more complex dynamics and interactions, the deep RL approach proves to be superior. As a result, a hybrid approach, which employs both, heuristics and deep learning based hyper-heuristic was implemented. Due to the system modularity, this approach allows to leverage the advantages of both control approaches. For the smart modules, varying input layers were incorporated according to the module state space. The distribution agents in $D1$, $D1.1$ and $D1.2$ deploy 88, 84 and 52 neurons for the input layer, and agents in $M1.1.1$ and $M1.1.2$ deploy 80 and 60 neurons. In total, 20 neural networks are trained for the 10 deep learning based agents, with each agent deploying one online and one target network. The agents in the manufacturing modules $M1.2.1$ and $M1.2.2$ are controlled by a *FiFo* rule, following its benchmarking results in Balaji and Srinivasan (2010) or Kaban et al. (2012). The iteratively optimized algorithmic parameters were initially related to similar approaches (as in Gankin et

al. (2021)) and are listed in Table 8.2. A batch size of 128 was deployed as a balance between training performance and computational efficiency. Batch sizes smaller than 128 resulted in decreased performance, especially for rush and high-priority orders, due to limited exploitation of solution spaces and prioritization. Conversely, larger batch sizes resulted in significantly increased training times.

Parameter	Value
Batch size	128
Neurons in hidden layers	128/64
Learning rate α	0.005
Discount factor γ	0.98
Drop out ratio	0.01
Target update step	5
Minimum ϵ	0.01
ϵ -decay	0.997

Table 8.2 Parameter settings for the deep RL agents

The simulation seeks to represent a real production scenario with stochastically varying urgency and priority of orders, where the number of orders processed depends on the pre-defined system load. The order sequence is randomly determined based on the order frequency (see Table 8.3, left). The order urgency is reflected in the due to time, with 20% of orders designated as rush orders, receiving a due to time $T_{dt,n}$, which is the sum of the order release time $T_{release,n}$, the predicted processing time $T_{proc,n}$, assuming a low system load (see Table 8.3, right), and a load dependent factor T_{load} . The remaining orders (80%) are classified as standard orders, with an additional random distributed time buffer T_{buffer} between 30 and 60 minutes. Orders are further categorized as high-priority (10%, i.e. for a highly valuable customer), prioritized (15%), or standard orders (75%).

The simulation comprises two types of orders, steel shafts (1.) and aluminum shafts (2.). The processing steps, times, and order frequencies for each type and value-added service (e.g. labelling) are listed in Table 8.3, accounting for the unique properties of steel and aluminum, such as their strength, weight, and durability, that affect processing times. The value-added services, including labelling, coating, and packaging, can increase throughput time through additional steps. Process steps progress from left to right and are completed at the machine with the corresponding number in brackets, as pointed out in Figure 8.4. The time for a machine tool-set exchange between product groups and the time required to load and release an item is 0.5 minutes each. Machine failures occur at an average rate of 4 per 1000 minutes and have a stochastic duration with a repair time ranging from 10 to 20 minutes.

Order type	Order frequency		Processing time (at machine)			Avg. throughput time w. transportation (low-load)
			Milling	Grinding	Value-added-service	
Steel	1.1	13%	8 (1)	4 (2)	-	18.9
shaft	1.2	34%	8 (1)	4 (2)	Labelling/ packaging: 4 (6)	26.6
Aluminium	2.1	13%	3 (3)	3 (4)	-	13.8
shaft	2.2	40%	6 (3)	6 (4)	Coating/ packaging: 4 (5)	21.5

Table 8.3 Order type specifications with associated processing times [min.] and sequences

8.4.2 Experimental results

In Section 8.4.2.1, a high-load scenario was adapted during training process with a maximum system congestion, which increases task complexity of order selection and allows the mapping of a substantial number of state-action pairs for later operation. This implies an infinite number of planned orders and immediate order release to the input buffer throughout the whole training process. This scenario presents challenges to learn effective strategies to maintain control performance and efficiency in terms of order allocation and system management. In Section 8.4.2.2, a normal load scenario was adapted that reflects a real processing equilibrium for benchmarking purposes. In Section 8.4.2.3, the scalability and robustness of our approach are trial to ensure stability and efficiency during operation, thereby confirming its reliability for real-world applications. For conducting the performance analysis, the mean tardiness ($T_{td,mean} = \frac{1}{n} \sum_{i=1}^n \max(0, C_i - d_i)$), and the mean global throughput time ($T_{tpt,mean} = \frac{1}{n} \sum_{i=1}^n T_{tpt,i}$), for all n orders, were considered. A particular emphasis is put on the interconnected order indicators, regarding their priority and urgency.

8.4.2.1 Training process

Incorporating the comparative analysis of the hyper-heuristic approach against conventional heuristics within the training process, as depicted in Figure 8.5, we observe that the hyper-heuristic outperforms these traditional rules in terms of the received rewards. Despite of the initial random rule selection at the beginning of the training phase (with $\varepsilon = 1$, see left side in Figure 8.5), the hyper-heuristic demonstrates performance levels close to those of traditional dispatching rules at the start of the training. It maintains a functional policy that complies with process requirements which results in a working policy right from the start of training, without the need for action masking or other acceleration mechanisms. As the training process progresses, the optimization criteria are increasingly satisfied, leading to continuous learning and performance enhancements. This trend highlights the progressive performance convergence of the approach.

Upon completion of the training, the hyper-heuristic achieved a moving average score of 200, a significant improvement over conventional heuristics such as FiFo local (-50), EDD (-78),

and high priority rule (-207). This demonstrates the superiority of the deep learning based rule selection approach within the hyper-heuristic framework, which has significantly exploited its optimization potential compared to conventional heuristics.

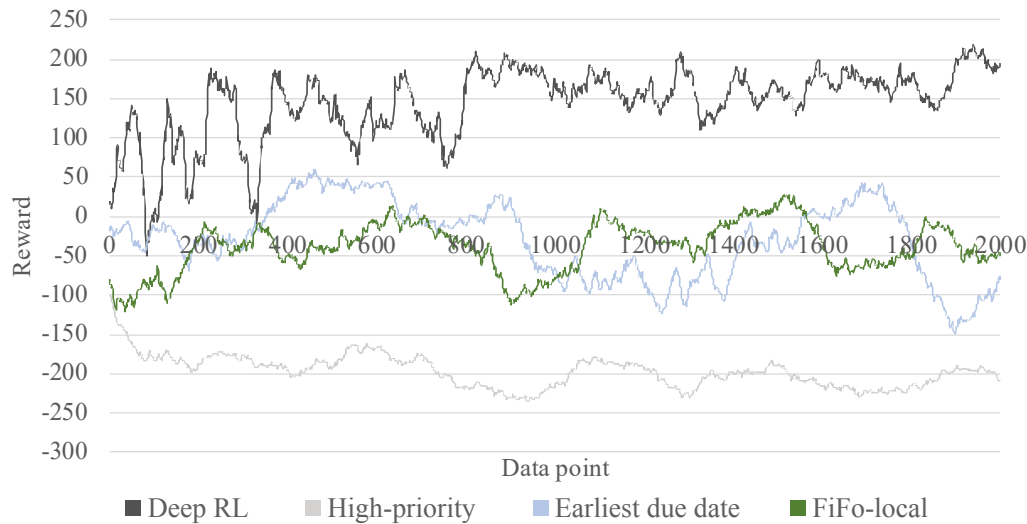


Figure 8.5 Moving average of obtained rewards for the top-layer *D1* agent

Figure 8.6 illustrates a throughput time related performance analysis of the hyper-heuristic, which is crucial for evaluating the true effectiveness of our approach. The results indicate a significant decrease in throughput time as the simulation progresses, which can be attributed to higher utilization rates despite the capacity limitations that restrict the simultaneous processing of orders at a machine. Regarding standard priority orders, the throughput time decreases by 43% and 51%, 59% for prioritized and high-priority orders. In contrast to the conventional rules, a noticeable decrease in fluctuations is observed with an increasing number of episodes, indicating that the deep RL algorithm is trending towards its optimal policy. Additionally, the analysis indicates that higher-priority orders have significantly faster throughput times compared to standard orders. Specifically, high-priority orders have a throughput time of 74.1 minutes, representing a 32.6% lower throughput time compared to standard orders, while prioritized orders exhibit a throughput time of 89.3 minutes, corresponding to an 18.7% decrease with a throughput time of 109.9 minutes. The green dotted line indicates the average Work-In-Progress (WIP) level. Despite a slight increase in throughput times as WIP levels rise, the robustness of control and optimization is evident with the occurring WIP peaks in Figure 8.6.

8.4.2.2 Benchmark results

In this Section, we conducted a benchmarking scenario that encompassed a simulation range of 7200 minutes, corresponding to three 8-hour shifts over a period of 5 days. The order amount of 2800 was iteratively determined to be near the system equilibrium. Unlike the training mode,

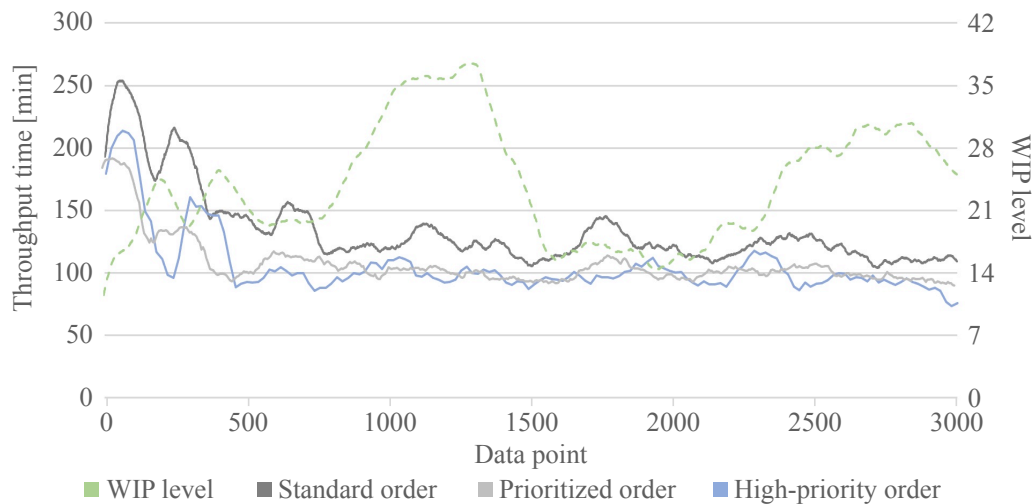


Figure 8.6 Moving average of throughput times related to order priorities

in which batch replay is required, the trained agents are now applied in an operational mode. To facilitate comprehensive analysis, we included a random rule in addition to the average benchmarks, which are summarized in Table 8.4. The left column of the table presents the hyper-heuristics results, while the rightmost column lists comparison indicators of the hyper-heuristic approach with the average of the individual dispatching rules (improved values are highlighted in green, worsened in red). The inclusion of the random rule allows for a more thorough evaluation of the other rules performances.

The results demonstrate that the summarized order values in the upper part were all improved using our approach. The mean tardiness was reduced by nearly 39.5% and the throughput was increased by 1.4%. In the individual benchmark comparison, the throughput time is comparable to the local *FiFo* rule, but priority-related indicators are improved. The order priority-related performance indicators are listed in the lower part for standard (0), prioritized (1) and high-priority (2) orders. With our approach, high priority orders were delivered with 60.8% reduced tardiness, and with a shorter throughput time. In contrast, both *FiFo* rules, as the most commonly used ones, performed worse regarding total order indicators compared to the proposed hyper-heuristic.

However, it was expected that rules that are specifically designed to optimize a particular indicator would perform better in that specific case. For instance, the *SPT* rule indicates shorter throughput times and the lowest work in progress levels, but it produced 78 fewer orders due to the blocking of buffers and storages by orders with additional process steps that are scheduled with higher throughput times. This led to a considerably high tardiness for these orders of 21.5 minutes. Consequently, all orders that comprised value-added services had to wait for orders with fewer remaining processing time. The mean tardiness was lowest with the *EDD* rule, which optimizes based on order due dates. However, none of the conventional dispatching rules were performing

	Hyper-heuristic	Fifo local	Fifo global	Earliest due date first	Shortest processing time first	Highest priority first	Random (excl. in avg. benchmark)	Hyper-heuristic against avg. benchmark
Total throughput [#]	2734	2722	2709	2718	2656	2678	2645	1.4%
Total order set	86.2	91.7	88.8	88.4	83.0	79.9	87.8	-0.2%
Mean tardiness [min.]	8.9	12.0	10.7	8.5	21.5	20.8	22.3	-39.5%
Work-in-progress [#]	28.3	29.5	30.2	29.7	27.0	28.9	29.9	-2.6%
Order priority	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2
Total throughput [#]	2066 402 266 2059 401 263 2049 399 261 2055 399 263 2012 389 254 2021 398 259 2005 387 253							1.3% 1.2% 2.3%
Priority related order set	88.3 83.6 73.3 91.1 93.9 93.1 88.4 89.6 90.5 88.1 90.3 88.0 81.4 85.1 93.2 91.5 44.5 44.1 87.6 85.9 87.8							0.2% 3.6% -10.4%
Mean tardiness [min.]	10.0 6.2 5.0 11.8 12.7 12.5 10.6 10.6 10.7 8.4 8.8 8.9 20.0 23.1 31.4 28.0 0.2 0.3 22.4 19.7 25.6							-36.5% -44.0% -60.8%
Work-in-progress [#]	22.2 3.8 2.3 22.7 4.1 2.8 23.2 4.1 2.9 22.8 4.1 2.7 20.6 3.6 2.8 25.2 2.2 1.5 23.0 3.9 3.0							-3.1% 5.0% -9.4%

Table 8.4 Multi-objective optimization benchmark incorporating order priorities

in combined measures and multi-objective view.

In the following, we compare the hyper-heuristic approach with individual dispatching rules in relation to the order priorities. In this regard, the hyper-heuristic outperformed the *FiFo*, *EDD*, and *SPT* rules, particularly for prioritized and high-priority orders. Specifically, for prioritized and high-priority orders, the hyper-heuristic reduced throughput by almost 7% to 83.6 minutes, and 20% 73.3 minutes, respectively, compared to the average time of the previously mentioned rules. Individually, the *HP* rule performed best for prioritised and high-priority orders, but had a 13.5% higher throughput time for standard orders than our approach, which account for 75% of the total order-set. Although the higher-priority orders are completed with a minimum mean tardiness with the *HP* rule, this leads to longer processing times for the much larger set of standard orders. Despite the hyper-heuristic's slightly poorer performance in the prioritized order class, it still outperformed the other rules from a multi-objective perspective. Additionally, the hyper-heuristic produced 2.7% more high-priority orders than the *HP* rule.

For a further analysis, we included Table 8.5 which demonstrates the relationship between order urgencies and priorities for the optimization of throughput times. The table includes a comparison of the corresponding throughput times with a high-priority and rush order as the base value in the lower section. As previous results indicate, the global *FiFo* and *random* rules were less performant compared to other the benchmarks, which is why they are excluded in further analysis.

Total orders	Hyper-heuristic		FiFo local		EDD		HP	
	Rush order	Standard order	Rush order	Standard order	Rush order	Standard order	Rush order	Standard order
High priority	72.1	73.6	90.0	90.7	59.1	96.3	45.1	43.9
Prioritized	72.5	86.2	86.2	90.4	56.1	98.6	44.0	44.6
Standard	87.1	88.6	87.5	88.7	57.5	96	93.8	90.9
In relation to high-priority and rush orders								
High priority	1	2.1%	1	0.8%	1	62.8%	1	-3%
Prioritized	0.6%	19.5%	-4.2%	0.4%	-5.2%	66.8%	-2%	-1%
Standard	20.8%	22.8%	-2.7%	-1.4%	-2.8%	62.6%	108%	101%

Table 8.5 Order urgency and priority dependent optimization of throughput times [min.]

One notable finding is the significant increase in throughput times of the hyper-heuristic (Table 8.5, left) for low prioritized and non-rush orders. The development is progressive, with a clear 22.8% increase from the highest to the lowest urgency and priority class. This suggests that non time-critical orders are processed at higher throughput times. However, due to the high priority relevance within the reward function, the increase is observable, but rather small with 2.1%. On the other hand, the *EDD* rule, as a single-criterion rule, only optimizes one criterion, which is the due date, equally for all priorities. Similarly, the *HP* rules also indicates a similar pattern for priorities, in which rush orders were processed 3% or 1% slower. The hyper-heuristic approach appears to be the more effective in optimizing combined urgency and priority measures. While

the *EDD* and *HP* rules may be suitable for constrained optimization, they may not perform well in stand-alone operation.

8.4.2.3 Analysis of optimization robustness and scalability

In addition to analyzing the performance of our approach, we also evaluated its robustness and scalability. It is essential to ensure that the approach can operate efficiently and reliably, even with increased order volumes, to guarantee long-term effectiveness. Previously, we demonstrated the control efficiency of our approach with increasing WIP levels in Figure 8.6. Now, we evaluate the robustness of our approach by measuring the rewards received during operation, which serve as an indicator of how well the desired production objectives were fulfilled. As illustrated in Figure 8.7, the received rewards of each dispatching rule were analyzed, with the random rule serving as the lower benchmark. The rewards of conventional rules exhibit significant fluctuations in both, short-term and long-term rewards. Despite occasional near-parity with the hyper-heuristic, on average, conventional rules were less robust than the hyper-heuristic in fulfilling the multiple defined optimization objectives.

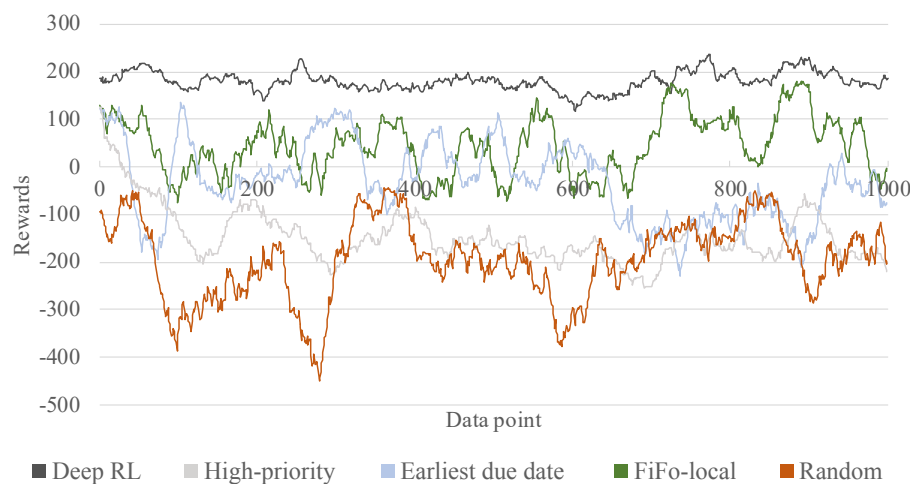


Figure 8.7 Moving average of agent rewards for the D1 distribution module

Table 8.6 summarizes the statistical values obtained from our robustness analysis. It can be observed that the hyper-heuristic approach achieved good performance in terms of robustness, which is in line with our earlier findings. However, the performance of the *HP* rule was surprisingly poor, which can be explained by the pre-defined rewards. Although its policy of favoring higher prioritized orders was followed, the larger amount of neglected standard orders led to a corresponding deterioration in performance. This trend is reflected in the decreasing reward trend, as the tardiness of the orders in the modules continued to increase over time (as indicated in Figures 8.5/8.7) which results in poor tardiness and throughput values and respective negative rewards. This highlights the importance of considering combined objectives,

as neglecting standard orders, although being less valuable per order, can have a detrimental effect on optimization objectives. Therefore, it is crucial to find a balance between prioritizing high-priority orders and ensuring that standard orders are also processed efficiently.

	Hyper-heuristic	FiFo local	Earliest due date	High-priority	Random
Arithmetic mean [-]	178.8	49.2	-24.1	-156.9	-193.2
Standard deviation [-]	21,2	60,9	85,5	59,8	84,6

Table 8.6 Summary of reward mean and standard deviation

The scalability of our approach is supported by the assumption of decentralized decision-making and the small state spaces of the neural networks that are used for action calculation. Due to the modular layout, also the overall task complexity was broken down among the agents which could thereby deploy compact neural networks. As a result, all agent computations were carried out in real-time, taking less than 0.01 seconds. We distinguish between the cases of pure training and the change of production organization during operation. Thereby, adding a new manufacturing module does not require the re-training of the entire system, as it would be the case with a central control instance. Instead, only the affected module and its overlying distribution module would need to be trained. Furthermore, if an identical manufacturing module is added, the logic can be learned via transfer learning, which reduces the training time to just training the overlying distribution module. Although training our scenario for 10,000 simulated minutes required approximately 36 hours (due to the iterative calculation), re-training a new network for a new module takes only a fraction of this time. Re-training the *D1.2* layer (see Figure 8.4) after an additional manufacturing module was added took only about 4 hours. From an organizational perspective, the scalability of our is not dependent on the complexity of the layout, as it can be broken down into sub-modules. Thus, our approach can be extended to more complex systems without requiring a complete overhaul of the existing training model. This scalability feature, combined with the real-time processing capabilities, leads to a suitable option for transferring it into a real production system.

8.5 Simulation to reality transfer

The transfer of the evolved and simulated approach was further deployed within the *Center for Industry 4.0* and validated using its modular manufacturing system. Figure 8.8 contains the real production environment (1), the updated simulation model (2), and the corresponding layout (3). The manufacturing cells are projected into the available machine cubes (numbers 1-5), and the input and output buffers are located within these. The real layout is similar to the simulated layout, with an additional robot manufacturing cell (see cell 4) and an additional storage slot in

the distribution cell to use the full storage capacity of the real environment (see number 8). For the transport of an order on the distribution level, the load carrier cubes are used as autonomous agents (9). If such a cube arrives at one of the machine cubes (1-5), an order is placed or picked up there and can then be transported to the next machine cube or to the system output. In the system input (6) and output (7), orders are generated (including the buffer capacity) and collected or deposited by the autonomous load carriers. The transfer of deep learning based instructions from the simulated agents to the real-world logistics elements was conducted by implementing the simulated system’s logic into the workload carriers. The instructions of the neural network were processed by the *FabOS* (factory operating system, Lass and Gronau (2020)). The action-set and state-vector remained consistent, with only the destinations being modified.

Since the agents are the bottleneck of this simulation due to their low speed, we based the performance evaluation on a fixed period of 5 hours. The path of the load carriers from one location to another is determined by the *FabOS* to avoid collisions and blockages. Due to time restrictions, the operation was carried out on the basis of previously trained agents. Control decisions were made after examining the status of new orders, after arriving at a destination, and after any changes in the system, when the agents had no assigned task. The results of the real test are listed in Table 8.7 and are compared against the *FiFo local* dispatching rule, due to its performance, but also its wide-spread use in real production systems. The best values are indicated in bold letters.

Although there was a slight decrease of one unit in throughput when compared to simulation performance, it is important to note that this may be attributed to the rather short duration of the conducted testing. Conversely, there was a 1.7% reduction in throughput time while maintaining a comparable level of tardiness for the total order-set. Additionally, higher-priority orders were processed more frequently and with an average tardiness that was 90% lower than that of lower-priority orders.

Scope/ order priority	Hyper-heuristic				FiFo local				Comparison (FiFo as base)			
	Total	0	1	2	Total	0	1	2	Total	0	1	2
Total throughput [#]	121	92	17	12	122	95	16	11	-0.8%	-3.2%	6.3%	9.1%
Throughput time [min.]	56,9	59,4	55,4	51,5	57,9	56,8	67,6	53,6	-1.7%	4.6%	-18.0%	-3.9%
Mean tardiness [min.]	9,8	12,8	1,6	1,1	9,8	9,4	13,9	14,1	0.0%	36.2%	-88.5%	-92.2%

Table 8.7 Performance benchmark within the hybrid production environment

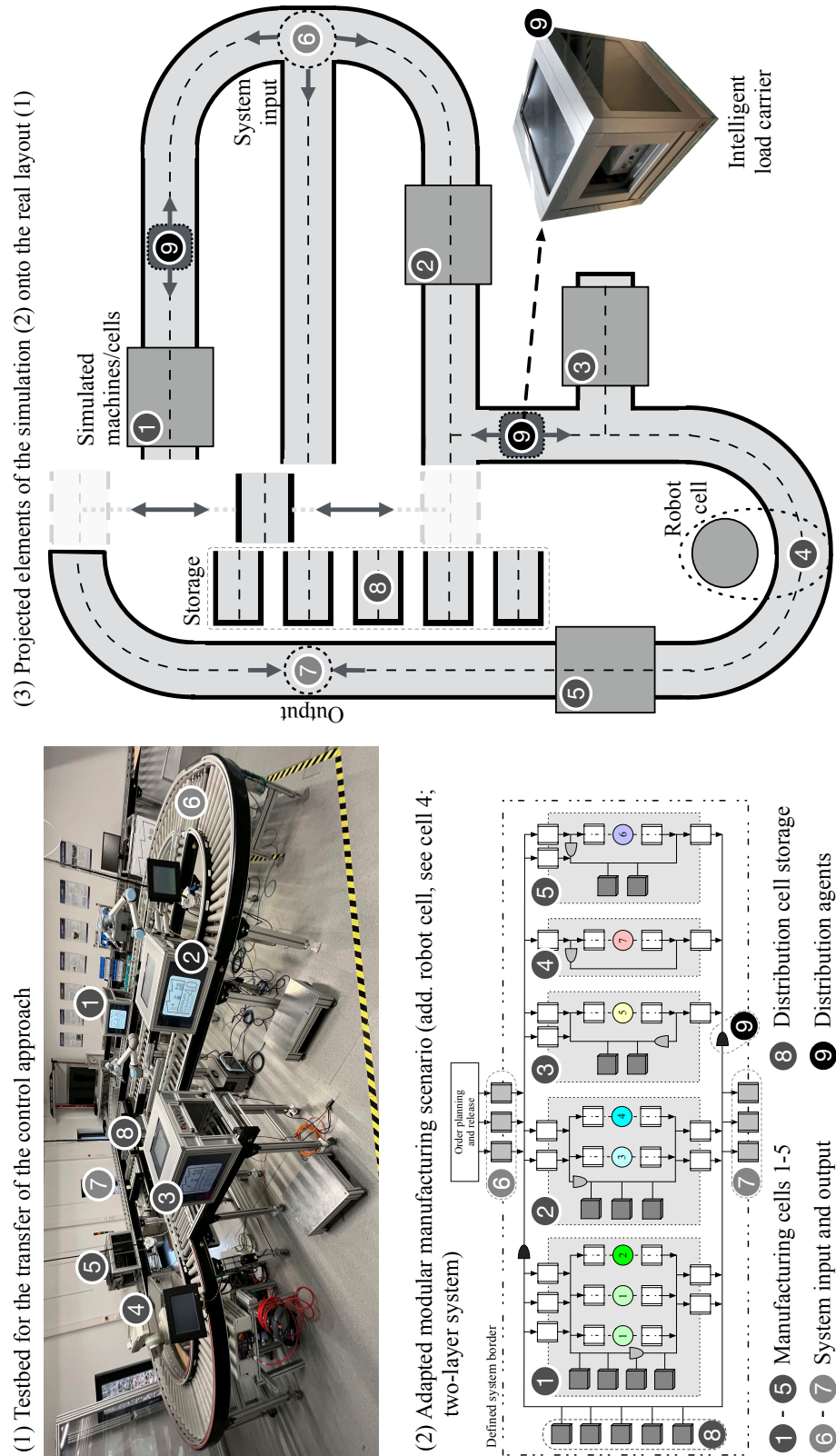


Figure 8.8 Testing setup of the hyper-heuristic within the hybrid production environment

8.6 Discussion

Today's production systems must cope with increasingly demanding customer requirements, shorter product and development cycles and short-term fluctuations in demand. One approach to address these challenges in production control is deep RL as a data-driven optimization tool, which differs from other machine learning methods mainly in its online adaptability and real-time processing of sensor data. In our approach, we have demonstrated the superior performance of deep RL compared to conventional rules that are still in widespread use. We also transferred our approach into a hybrid production environment, highlighting its real-world capabilities. It becomes clear that deep RL can master the link between input states, optimisation goal and the derivation of necessary actions and can help to achieve individual production goals. A multi-level system with distributed production resources was considered, which is composed of agents with different tasks. This is intended to cover different industry backgrounds and maximise transferability to specific applications.

Managerial insights The increasing connectivity of future factories and the growing complexity of products and processes require accelerated corporate adaptation cycles, especially in manufacturing. To meet these challenges, companies are well-advised to consider more sophisticated approaches to increase flexibility, mitigate process risks, and maximize production robustness. However, in addition to sustaining processes, it is also important to fully exploit competitiveness through the deliberate use of new algorithmic approaches and organizational capabilities. By using hyper heuristics, companies can limit their dependence on scarce human capital and proactively use data-driven operations to reduce costly manual processes. In contrast to conventional approaches, which can only react to changing conditions to a limited extent, the modular framework presented in our approach has a significantly higher transferability and can be adapted according to market requirements. In addition, the defined objectives were achieved in a combined manner more effectively, which increases cash flow and can reduce conversion costs. Furthermore, additional services such as rush orders and prioritized orders were integrated, which not only enable additional cash flow, but also integrate customer-oriented services on the shop floor.

8.7 Conclusion

This paper presents a novel hyper-heuristic based control approach for modular designed holonic production systems. Holons were modeled as autonomous manufacturing entities, providing high adaptability in a semi-heterarchical structure. Unlike previous multi-agent approaches that were limited to a single operational level or that implemented confined deep learning techniques, we deployed dispatching rules for facilitating a deep learning based performance optimization. Each agent within the production system had its own control policy and shared experiences with agents within the same cell. The differentiated contemplation of manufacturing agents at the shop-floor layer and the use of distribution agents within the upper transport layers was emphasized.

The control approach targeted several parameters for optimization, including global throughput time, adherence to due dates, and the processing of rush and prioritized orders. Simulations and real-world scenarios demonstrate the superiority of the hyper-heuristic approach in multi-objective optimization of average throughput time, total throughput, and tardiness. Prioritized and rush orders, which often emerge in practice, were processed with better performance than with dedicated dispatching rules. Likewise, not only were more orders processed within the real environment, but the existing rule-set was outperformed with regard to the defined performance indicators.

The hybrid and decentralized hyper-heuristic control approach integrates both, deep learning based and conventional elements, to facilitate a scenario-specific process optimization. Whereas the holonic approach allows for an easy adoption of new resources and processes, the hyper-heuristic prevents the selection of misleading actions by leveraging the rule-set integrated process logic. It resembles a self-configuring system that automatically adapts to changing production conditions without human intervention and leverages system scalability. The addition of a cell does not necessitate re-training of the entire system which leverages utilization efficiency. The distributed resources further avoid the need for large state and action spaces, as the modules and associated state or action parameters have a pre-defined scope.

Future research should focus on reducing the training effort required for comparable modules through the use of parameter learning strategies and the freeze and transfer of network layers to differing cells for faster scenario adoption. Moreover, to better adapt to varying environmental conditions, agents should be assigned specific action spaces based on cell objectives and levels, which can be kept variable, allowing different strategies to be executed. This will allow for an hybrid operational model, avoiding the prevalent pre-training in a digital twin, which further leverages production performances and adaptability. Future research should also target bridging the gap between the real-world and simulated environments, which will ultimately reduce operational barriers.

Copyright notice

This is an accepted version of this article published in:

Panzer, M., B. Bender and N. Gronau (2023). A deep reinforcement learning based hyper-heuristic for modular production control. *International Journal of Production Research*, p. 1-22. <https://doi.org/10.1080/00207543.2023.2233641>

Clarification of the copyright adjusted according to the guidelines of the publisher.

Contributor roles

This paper is the result of collaborative efforts where specific responsibilities were allocated to ensure the effective completion of the research and the preparation of the manuscript:

- **Marcel Panzer:** Assumed the lead role in most aspects of this publication. His responsibilities encompassed the conceptualization of the research, the design and execution of research methodology, simulations, evaluations, data collection and analysis, and primarily drafting the manuscript. Additionally, he contributed to compiling and editing the final manuscript during the publication reviews.
- **Norbert Gronau and Benedict Bender:** Both contributed to the development of this publication through their thorough reviews and guidance. Their contributions included critical reviews, providing insightful feedback and suggestions, which were instrumental in shaping the publication and further ensuring academic rigor.

The *Declaration of the Co-Authors* is inserted at the end of this thesis.

Publication 4 - References

- Baer, S., D. Turner, P. Mohanty, V. Samsonov, R. Bakakeu and T. Meisen (2020). Multi Agent Deep Q-Network Approach for Online Job Shop Scheduling in Flexible Manufacturing. In: *2020 International Conference on Manufacturing System and Multiple Machines*, Tokyo, Japan.
- Bahrpeyma, F. and D. Reichelt (2022). A review of the applications of multi-agent reinforcement learning in smart factories. *Frontiers in Robotics and AI* 9, p. 1027340. doi: 10.3389/frobt.2022.1027340.
- Baker, A. D. (1998). A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull. *Journal of Manufacturing Systems* 17(4), p. 297–320. doi: 10.1016/S0278-6125(98)80077-0.
- Balaji, P. G. and D. Srinivasan (2010). An Introduction to Multi-Agent Systems. In: J. Kacprzyk, D. Srinivasan, and L. C. Jain (Hrsg.), *Innovations in Multi-Agent Systems and Applications - I*, Volume 310, p. 1–27. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Bellman, R. (1957). A Markovian Decision Process. *Indiana University Mathematics Journal* 6(5), p. 679–684. doi: 10.1512/iumj.1957.6.56038.
- Borangiu, T., P. Gilbert, N.-A. Ivanescu and A. Rosu (2009). An implementing framework for holonic manufacturing control with multiple robot-vision stations. *Engineering Applications of Artificial Intelligence* 22(4-5), p. 505–521. doi: 10.1016/j.engappai.2009.03.001.
- Borangiu, T., S. Răileanu, D. Trentesaux and T. Berger (2010). Semi-heterarchical agile control architecture with intelligent product-driven scheduling. *IFAC Proceedings Volumes* 43(4), p. 108–113. doi: 10.3182/20100701-2-PT-4011.00020.
- Bueno, A., M. Godinho Filho and A. G. Frank (2020). Smart production planning and control in the Industry 4.0 context: A systematic literature review. *Computers & Industrial Engineering* 149, p. 106774. doi: 10.1016/j.cie.2020.106774.
- Burke, E. K., M. R. Hyde, G. Kendall, G. Ochoa, E. Özcan and J. R. Woodward (2010). A Classification of Hyper-Heuristic Approaches. In: M. Gendreau and J.-Y. Potvin (Hrsg.), *Handbook of Metaheuristics*, Volume 272, p. 453–477. Cham: Springer International Publishing. ISBN: 978-3-319-91086-4.
- Burke, E. K., M. R. Hyde, G. Kendall, G. Ochoa, E. Özcan and J. R. Woodward (2019). A Classification of Hyper-Heuristic Approaches: Revisited. In: M. Gendreau and J.-Y. Potvin (Hrsg.), *Handbook of Metaheuristics*, Volume 272, p. 453–477. Cham: Springer International Publishing. ISBN: 978-3-319-91085-7.
- Bányai, T. (2021). Optimization of Material Supply in Smart Manufacturing Environment: A

- Metaheuristic Approach for Matrix Production. *Machines* 9(10), p. 220. doi: 10.3390/machines9100220.
- Chang, J., D. Yu, Z. Zhou, W. He and L. Zhang (2022). Hierarchical Reinforcement Learning for Multi-Objective Real-Time Flexible Scheduling in a Smart Shop Floor. *Machines* 10(12), p. 1195. doi: 10.3390/machines10121195.
- Cowling, P., G. Kendall and E. Soubeiga (2001). A Hyperheuristic Approach to Scheduling a Sales Summit. In: G. Goos, J. Hartmanis, J. van Leeuwen, E. Burke, and W. Erben (Hrsg.), *Practice and Theory of Automated Timetabling III*, Volume 2079, p. 176–190. Berlin, Heidelberg: Springer Berlin Heidelberg.
- de Paula Ferreira, W., F. Armellini and L. A. De Santa-Eulalia (2020). Simulation in industry 4.0: A state-of-the-art review. *Computers & Industrial Engineering* 149, p. 106868. doi: 10.1016/j.cie.2020.106868.
- de Paula Ferreira, W., F. Armellini, L. A. de Santa-Eulalia and V. Thomasset-Laperrière (2022). A framework for identifying and analysing industry 4.0 scenarios. *Journal of Manufacturing Systems* 65, p. 192–207. doi: 10.1016/j.jmsy.2022.09.002.
- Dittrich, M.-A. and S. Fohlmeister (2020). Cooperative multi-agent system for production control using reinforcement learning. *CIRP Annals* 69(1), p. 389 – 392. doi: 10.1016/j.cirp.2020.04.005.
- Drake, J. H., A. Kheiri, E. Özcan and E. K. Burke (2020). Recent advances in selection hyper-heuristics. *European Journal of Operational Research* 285(2), p. 405–428. doi: 10.1016/j.ejor.2019.07.073.
- Esteso, A., D. Peidro, J. Mula and M. Díaz-Madroñero (2022). Reinforcement learning applied to production planning and control. *International Journal of Production Research*, p. 1–18. doi: 10.1080/00207543.2022.2104180.
- Gankin, D., S. Mayer, J. Zinn, B. Vogel-Heuser and C. Endisch (2021). Modular Production Control with Multi-Agent Deep Q-Learning. In: *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vasteras, Sweden, p. 1–8. IEEE. doi: 10.1109/ETFA45728.2021.9613177.
- Giret, A. and V. Botti (2004). Holons and agents. *Journal of Intelligent Manufacturing* 15(5), p. 645–659. doi: 10.1023/B:JIMS.0000037714.56201.a3.
- Grassi, A., G. Guizzi, L. C. Santillo and S. Vespoli (2020). A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters* 24, p. 43–46. doi: 10.1016/j.mfglet.2020.03.007.
- Grassi, A., G. Guizzi, L. C. Santillo and S. Vespoli (2021). Assessing the performances

- of a novel decentralised scheduling approach in Industry 4.0 and cloud manufacturing contexts. *International Journal of Production Research* 59(20), p. 6034–6053. doi: 10.1080/00207543.2020.1799105.
- Greschke, P., M. Schönemann, S. Thiede and C. Herrmann (2014). Matrix Structures for High Volumes and Flexibility in Production Systems. *Procedia CIRP* 17, p. 160–165. doi: 10.1016/j.procir.2014.02.040.
- Groover, M. P. (2019). *Automation, production systems, and computer-integrated manufacturing* (Fifth edition ed.). Hudson Street, New York: Pearson Education.
- Gros, T. P., J. Gros and V. Wolf (2020). Real-Time Decision Making for a Car Manufacturing Process Using Deep Reinforcement Learning. In: *2020 Winter Simulation Conference (WSC)*, Orlando, FL, USA, p. 3032–3044. IEEE. doi: 10.1109/WSC48552.2020.9383884.
- Hammami, Z., W. Mouelhi and L. Ben Said (2017). On-line self-adaptive framework for tailoring a neural-agent learning model addressing dynamic real-time scheduling problems. *Journal of Manufacturing Systems* 45, p. 97–108. doi: 10.1016/j.jmsy.2017.08.003.
- Herrera, M., M. Pérez-Hernández, A. Kumar Parlikad and J. Izquierdo (2020). Multi-Agent Systems and Complex Networks: Review and Applications in Systems Engineering. *Processes* 8(3), p. 312. doi: 10.3390/pr8030312.
- Hofmann, C., C. Krahe, N. Stricker and G. Lanza (2020). Autonomous production control for matrix production based on deep Q-learning. *Procedia CIRP* 88, p. 25–30. doi: 10.1016/j.procir.2020.05.005.
- Kaban, A. K., Z. Othman and D. S. Rohmah (2012). Comparison of dispatching rules in job-shop scheduling problem using simulation: a case study. *International Journal of Simulation Modelling* 11(3), p. 129–140. doi: 10.2507/IJSIMM11(3)2.201.
- Kallestad, J., R. Hasibi, A. Hemmati and K. Sørensen (2023). A General Deep Reinforcement Learning Hyperheuristic Framework for Solving Combinatorial Optimization Problems. *European Journal of Operational Research*, p. S037722172300036X. doi: 10.1016/j.ejor.2023.01.017.
- Kanervisto, A., C. Scheller and V. Hautamaki (2020). Action Space Shaping in Deep Reinforcement Learning. In: *2020 IEEE Conference on Games (CoG)*, Osaka, Japan, p. 479–486. IEEE. doi: 10.1109/CoG47356.2020.9231687.
- Kapoor, K., A. Z. Bigdeli, Y. K. Dwivedi and R. Raman (2021). How is COVID-19 altering the manufacturing landscape? A literature review of imminent challenges and management interventions. *Annals of Operations Research*. doi: 10.1007/s10479-021-04397-2.
- Kuhnle, A., J.-P. Kaiser, F. Theiß, N. Stricker and G. Lanza (2020). Designing an adaptive pro-

- duction control system using reinforcement learning. *Journal of Intelligent Manufacturing* 32, p. 855–876. doi: 10.1007/s10845-020-01612-y.
- Kuhnle, A., M. C. May, L. Schäfer and G. Lanza (2021). Explainable reinforcement learning in production control of job shop manufacturing system. *International Journal of Production Research*, p. 1–23. doi: 10.1080/00207543.2021.1972179.
- Lass, S. and N. Gronau (2020). A factory operating system for extending existing factories to Industry 4.0. *Computers in Industry* 115, p. 103128. doi: 10.1016/j.compind.2019.103128.
- Lee, J., B. Bagheri and C. Jin (2016). Introduction to cyber manufacturing. *Manufacturing Letters* 8, p. 11–15. doi: 10.1016/j.mfglet.2016.05.002.
- Lee, J., B. Bagheri and H.-A. Kao (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters* 3, p. 18–23. doi: 10.1016/j.mfglet.2014.12.001.
- Lee, J., H. Davari, J. Singh and V. Pandhare (2018). Industrial Artificial Intelligence for industry 4.0-based manufacturing systems. *Manufacturing Letters* 18, p. 20–23. doi: 10.1016/j.mfglet.2018.09.002.
- Liu, C., C. Chang and C. Tseng (2020). Actor-Critic Deep Reinforcement Learning for Solving Job Shop Scheduling Problems. *IEEE Access* 8, p. 71752–71762. doi: 10.1109/ACCESS.2020.2987820.
- Liu, R., R. Piplani and C. Toro (2022). Deep reinforcement learning for dynamic scheduling of a flexible job shop. *International Journal of Production Research* 60(13), p. 4049–4069. doi: 10.1080/00207543.2022.2058432.
- Malus, A., D. Kozjek and R. Vrabič (2020). Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. *CIRP Annals* 69(1), p. 397 – 400. doi: 10.1016/j.cirp.2020.04.001.
- May, M. C., L. Kiefer, A. Kuhnle, N. Stricker and G. Lanza (2021). Decentralized Multi-Agent Production Control through Economic Model Bidding for Matrix Production Systems. *Procedia CIRP* 96, p. 3–8. doi: 10.1016/j.procir.2021.01.043.
- Mayer, S., T. Classen and C. Endisch (2021). Modular production control using deep reinforcement learning: proximal policy optimization. *Journal of Intelligent Manufacturing* 32(8), p. 2335–2351. doi: 10.1007/s10845-021-01778-z.
- Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller (2013). Playing Atari with Deep Reinforcement Learning. p. arXiv:1312.5602.
- Mourtzis, D. (2020). Simulation in the design and operation of manufacturing systems: state of the art and new trends. *International Journal of Production Research* 58(7), p. 1927–1949.

- doi: 10.1080/00207543.2019.1636321.
- Mönch, L., J. Fowler and S. J. Mason (2013). *Production planning and control for semiconductor wafer fabrication facilities: modeling, analysis, and systems*. Number vol. 52 in Operations research/computer science interfaces series. New York: Springer. doi: OCLC: ocn794710214.
- Overbeck, L., A. Hugues, M. C. May, A. Kuhnle and G. Lanza (2021). Reinforcement Learning Based Production Control of Semi-automated Manufacturing Systems. *Procedia CIRP 103*, p. 170–175. doi: 10.1016/j.procir.2021.10.027.
- Panzer, M. and B. Bender (2022). Deep reinforcement learning in production systems: a systematic literature review. *International Journal of Production Research 60*(13), p. 4316–4341. doi: 10.1080/00207543.2021.1973138.
- Panzer, M., B. Bender and N. Gronau (2022). Neural agent-based production planning and control: An architectural review. *Journal of Manufacturing Systems 65*, p. 743–766. doi: 10.1016/j.jmsy.2022.10.019.
- Parente, M., G. Figueira, P. Amorim and A. Marques (2020). Production scheduling in the context of Industry 4.0: review and trends. *International Journal of Production Research 58*(17), p. 5401–5431. doi: 10.1080/00207543.2020.1718794.
- Parunak, H. V. D., J. F. White, P. W. Lozo, R. Judd, B. W. Irish and J. Kindrick (1986). An Architecture for Heuristic Factory Control. In: *1986 American Control Conference*, Seattle, WA, USA, p. 548–558. IEEE. doi: 10.23919/ACC.1986.4789001.
- Peppers, K., T. Tuunanen, M. A. Rothenberger and S. Chatterjee (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems 24*(3), p. 45–77. doi: 10.2753/MIS0742-1222240302.
- Ritterbusch, G. D. and M. R. Teichmann (2023). Defining the Metaverse: A Systematic Literature Review. *IEEE Access 11*, p. 12368–12377. doi: 10.1109/ACCESS.2023.3241809.
- Sallez, Y., T. Berger, S. Raileanu, S. Chaabane and D. Trentesaux (2010). Semi-heterarchical control of FMS: From theory to application. *Engineering Applications of Artificial Intelligence 23*(8), p. 1314–1326. doi: 10.1016/j.engappai.2010.06.013.
- Samsonov, V., M. Kemmerling, M. Paegert, D. Lütticke, F. Sauermann, A. Gützlaff, G. Schuh and T. Meisen (2021). Manufacturing Control in Job Shop Environments with Reinforcement Learning. In: *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*, Online Streaming, — Select a Country —, p. 589–597. SCITEPRESS - Science and Technology Publications. doi: 10.5220/0010202405890597.
- Schenk, M., S. Wirth and E. Muller (2010). *Factory planning manual: situation-driven production facility planning*. Berlin ; London ; New York: Springer. doi: OCLC: ocn428029418.

- Schmidtke, N., A. Rettmann and F. Behrendt (2021). Matrix Production Systems - Requirements and Influences on Logistics Planning for Decentralized Production Structures. doi: 10.24251/HICSS.2021.201.
- Schulman, J., F. Wolski, P. Dhariwal, A. Radford and O. Klimov (2017). Proximal Policy Optimization Algorithms. *arXiv*. doi: 10.48550/arXiv.1707.06347.
- Sutton, R. S. and A. G. Barto (2017). *Reinforcement learning: an introduction* (2nd ed.). Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press. ISBN: 978-0-262-03924-6.
- Swiercz, A. (2017). Hyper-Heuristics and Metaheuristics for Selected Bio-Inspired Combinatorial Optimization Problems. In: J. D. S. Lorente (Hrsg.), *Heuristics and Hyper-Heuristics - Principles and Applications*. InTech. doi: 10.5772/intechopen.69225.
- Tampuu, A., T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru and R. Vicente (2017). Multiagent cooperation and competition with deep reinforcement learning. *PLOS ONE* 12(4), p. e0172395. doi: 10.1371/journal.pone.0172395.
- Tao, F., Q. Qi, A. Liu and A. Kusiak (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems* 48, p. 157–169. doi: 10.1016/j.jmsy.2018.01.006.
- Tay, J. C. and N. B. Ho (2007). Designing Dispatching Rules to Minimize Total Tardiness. In: J. Kacprzyk, K. P. Dahal, K. C. Tan, and P. I. Cowling (Hrsg.), *Evolutionary Scheduling*, Volume 49, p. 101–124. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Valckenaers, P., F. Bonneville, H. Van Brussel, L. Bongaerts and J. Wyns (1994). Results of the holonic control system benchmark at KU Leuven. In: *Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, Troy, NY, USA, p. 128–133. IEEE. doi: 10.1109/CIMAT.1994.389083.
- Van Ekeris, T., R. Meyes and T. Meisen (2021). Discovering Heuristics And Metaheuristics For Job Shop Scheduling From Scratch Via Deep Reinforcement Learning. doi: 10.15488/11231.
- Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmuller, T. Bauernhansl, A. Knapp and A. Kyek (2018). Deep reinforcement learning for semiconductor production scheduling. In: *2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, Saratoga Springs, NY, USA. doi: 10.1109/ASMC.2018.8373191.
- Weiss, G. (Hrsg.) (2001). *Multiagent systems: a modern approach to distributed artificial intelligence* (3. print ed.). Cambridge, Mass.: MIT Press.
- Zambrano Rey, G., C. Pach, N. Aissani, A. Bekrar, T. Berger and D. Trentesaux (2013). The control of myopic behavior in semi-heterarchical production systems: A holonic framework. *Engineering Applications of Artificial Intelligence* 26(2), p. 800–817. doi:

10.1016/j.engappai.2012.08.011.

Zhang, Y., R. Bai, R. Qu, C. Tu and J. Jin (2022). A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties. *European Journal of Operational Research* 300(2), p. 418–427. doi: 10.1016/j.ejor.2021.10.032.

Enhancing economic efficiency in modular production systems through deep reinforcement learning

Marcel Panzer^{1a} and Norbert Gronau^a

^a *Chair of Business Informatics, Processes and Systems, University of Potsdam,
Karl-Marx-Street 67, 14482 Potsdam, Germany*

ABSTRACT

In times of increasingly complex production processes and volatile customer demands, the production adaptability is crucial for a company's profitability and competitiveness. The ability to cope with rapidly changing customer requirements and unexpected internal and external events guarantees robust and efficient production processes, requiring a dedicated control concept at the shop floor level. Yet in today's practice, conventional control approaches remain in use, which may not keep up with the dynamic behaviour due to their scenario-specific and rigid properties. To address this challenge, deep learning methods were increasingly deployed due to their optimization and scalability properties. However, these approaches were often tested in specific operational applications and focused on technical performance indicators such as order tardiness or total throughput. In this paper, we propose a deep reinforcement learning based production control to optimize combined techno-financial performance measures. Based on pre-defined manufacturing modules that are supplied and operated by multiple agents, positive effects were observed in terms of increased revenue and reduced penalties due to lower throughput times and fewer delayed products. The combined modular and multi-staged approach as well as the distributed decision-making further leverage scalability and transferability to other scenarios.

Keywords

Modular Production; Production Control; Multi-Agent System; Deep Reinforcement Learning; Discrete Event Simulation

¹Corresponding author

Submitted to the Procedia CIRP on 25 May 2023, accepted on 3 September 2023.

9.1 Introduction

In today's emerging economy, production systems are increasingly exposed to external and internal pressures and must adapt to shifting demands, volatile supply markets, and increasing customer requirements such as shortening delivery times. These influences, combined with growing product complexity, necessitate the implementation of a robust production control that effectively monitors and optimizes central operational process parameters, such as order tardiness or throughput times. Furthermore, this system should facilitate the integration of customer-centric services to cope with the prevalence of an increasingly large number of customized individual product offerings that drive mass customization. These come along with the emergence of customer related add-on services and process related supplementary services, including maintenance, repairs, and ongoing modifications of the production system, which substantially heightened the demand for an effective and robust shop-floor control.

To cope with the growing amounts of data and the increasing optimization complexity, in recent years, an increasing number of Industry 4.0 and machine learning-driven approaches were deployed to improve process planning and control (Lee et al., 2016; Lass and Gronau, 2020). The pertinent use of these technologies can lead to a significant exploitation of given process potentials that cannot be leveraged with conventional tool-sets (Grassi et al., 2020; Parente et al., 2020). However, machine learning approaches were only used to a limited extent in practical use-cases, as conventional production control strategies, such as First-in-First-out (FiFo) or Earliest-Due-Date (EDD) rules, continue to be predominantly applied in production control (Kuhnle et al., 2021). Nevertheless, current approaches highlight the potential of using deep learning, in particular deep reinforcement learning (RL), due to their high adaptability and optimization performance (Samsonov et al., 2021). Deep RL is characterized by its interactive and trial-and-error-based learning with its environment (Sutton and Barto, 2017), thereby meeting global and local objectives in multi-agent systems (Tampuu et al., 2017). Its performance, which often outperformed other conventional methods, is primarily attributed to its high adaptability and responsiveness, which enables a rapid decision making in diverse and dynamic environments, compared to meta-heuristics and other machine learning methods (Panzer et al., 2022). These characteristics make deep RL particularly attractive for production control, where real-time decisions need to be made about the disposition of products and intralogistics processes (Chang et al., 2022).

In practice, most deep RL based control approaches were implemented in job shop systems, however, only a few approaches were conducted on flexible and modular production systems. Concerning the control optimization, operational parameters such as lead time or machine utilization were mainly optimized, but there was no approach that addressed a combined techno-financial optimization (Panzer and Bender, 2022). Thus, there is a need for a reactive control

approach that not only controls multiple agents in a re-configurable modular production system, but simultaneously leverages customer-centric requirements and objective parameters for its intelligent decision-making process. In this paper, we present a novel deep learning based control approach that combines the adaptability of deep RL in conjunction with a modular production system. It focuses on the optimization of customer-centric services as well as technical performance indicators to leverage adaptability and profitability for the control of multiple agents.

The remainder of this paper is structured as follows. The next Section outlines the basics of modular production and deep RL. In section 9.3, the discrete event-based (DES) simulation framework is presented in detail, explaining how agents are trained and rewarded, while operations are conducted. Section 9.4 presents optimization results from a test study. Finally, section 9.5 provides a conclusion about achieved milestones and summarizes the main findings.

9.2 Related work

The adaptability of a production system is predicated not solely on its control mechanisms, but also on the re-configurability of the system components and their ability to generate new system competencies from existing process elements. In this context, modular production systems were frequently utilized in control research, an aspect we focus in the following section (May et al., 2021).

9.2.1 Modular production systems

Modular production systems are characterized by their ability to define almost arbitrary process flows, similar to those known from matrix systems that feature a product flow independent system control. Frequently, autonomous guided vehicles (AGVs) are used to facilitate such flexible production (as in Mayer et al. (2021)), to leverage shared control experience in intelligent control approaches, and to increase utilization and reduce buffer inventories (Greschke et al., 2014). However, the increased optimization space associated with free process flows requires dedicated control strategies that take advantage of the increasing amounts of data collected by distributed machine sensors, smart products, or AGVs. In this context, previous approaches often deployed decentralized control frameworks that allow the optimization problem to be divided among multiple agents within modules, thus leveraging the capabilities of individual resources in the overall system (Balaji and Srinivasan, 2010). Our DES framework further adopts a semi-heterarchical control approach that combines the flexibility and high local reactivity of a heterarchical production concept with the long-term optimization capabilities of a more centralized hierarchical production control (Bongaerts et al., 2000; Groover, 2019). In this way, global objectives, which specifically include customer-related metrics such as order tardiness,

are taken into account that enhance reliability, and increase overall efficiency. Simultaneously, local technical parameters can be considered such as buffer stocks or machine uptime rates to reduce bottlenecks.

9.2.2 Deep reinforcement learning

To reach the adaptability of the modular production system within the control framework, it is imperative to encapsulate it within the algorithmic model. Therefore, deep RL algorithms, which are capable of executing real-time decisions predicated on current system information, were often used in dynamic control tasks. In deep RL, the control policy of a deep RL agent is represented by the neural network and is perpetually updated on previously made experiences. The DQN algorithm, which is deployed in this study, utilizes a Q-value, that represents an indicative measure of success for the execution of an action a_t in a state s_t , and is computed based on the Bellman equation, considering the cumulative rewards that can be expected from that state onward (Bellman, 1961; Mnih et al., 2013).

Throughout the learning process, the weights of the neural network are updated based on the actual observed state transitions towards s_{t+1} , beginning from s_t on the basis of the selected action a_t , as indicated in Equation 9.1. To circumvent unstable learning behavior, the learning rate, denoted as α , modulates the speed of the learning process, thereby facilitating an iterative learning mechanism predicated on the received rewards and resulting Q-values of the subsequent state s_{t+1} . Moreover, we employ an online network and a target network, denoted by weights θ and θ^- respectively, with prescribed update cycles to stabilize the DQN learning process as further outlined in Sutton and Barto (2017).

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha(r + \gamma \underset{max}{Q}(s_{t+1}, a_{t+1})) \quad (9.1)$$

9.2.3 Deep reinforcement learning-based production control

Although there were some integration approaches of deep RL in production control, only a few have focused on multi-agent based production control in modular production (Panzer and Bender, 2022). To find such, we searched the databases *Scopus* and *Web of Science*.

In non-modular production, some strategies were explored to improve production performance indicators, such as utilization rates or order delays. Malus et al. (2020) introduced an order bidding mechanism for autonomous mobile robots, based on a proximal policy optimization (PPO). Agents could place bids between 0 and 1, with the order acquiring the highest bid being chosen for the dispatching operation. Hammami et al. (2017) proposed a multi-agent system based on simultaneous learning and information dissemination between agents to reduce average order delays. Decision agents were connected to selection agents, which selected the best neural

network for the decision agent. Dittrich and Fohlmeister (2020); Hofmann et al. (2020) utilized a central DQN decision module for training purposes, which stored and updated the disposition policies of all agents and made them available upon request. Waschneck et al. (2018) trained separate neural networks for wafer fabrication stations to optimize maximum uptime utilization. Gros et al. (2020) leveraged an iterative learning strategy to optimize product sequencing in automotive production, thereby reducing costs associated with inefficient car sequencing. Overbeck et al. (2021) employed PPO agents and hyper-parameter tuning to determine the optimal action in an automated assembly system, resulting in an increased number of parts produced.

The aforementioned approaches focused mainly on production control in conventional job-shop systems. In contrast, Gankin et al. (2021) suggested a large-scale matrix layout based on a modular approach. The production system comprised 25 stations and 20 units and produced two different product types, each with 13 process steps. An action masking mechanism was used to reduce decision complexity. A shared DQN and buffer were used as the decision-making authority, facilitating experience sharing among agents. May et al. (2021) implemented an economic bidding approach to increase utilization efficiency in a matrix-structured production system. The approach maximized operational profit for each agent independently and optimized execution time and resource utilization efficiency compared to conventional heuristics. Hofmann et al. (2020) simulated matrix system with 10 workstations and multiple AGVs. Agents received immediate rewards after each process step and delayed rewards based on the global cycle time after completing an operational step which accelerated the learning process and reduced job cycle times.

These approaches revealed initial advancements in the application of deep RL in matrix and modular systems, by using economic bidding mechanisms, and the use of global rewards to improve utilization efficiency and reduce lead times. However, the consideration of combined techno-financial performance indicators in a flexible modular environment is still pending and will be addressed in this paper.

9.3 Proposed algorithm - a deep RL control framework

The structure of the adopted modular production approach necessitates a production control system that is equally modular, exhibiting both, high flexibility and robustness. Therefore, the simulation framework distinguishes between two types of agents, manufacturing and distribution agents. The former are responsible for supplying orders to machines and managing the local operations, while the latter are tasked with intra-logistics operations and distributing orders between modules. Strategic decisions and coordination, particularly those concerning multiple modules, are made by a higher-level distribution agents, that reflect a more hierarchical control

to manage inter-module dependencies and to ensure that the overall objectives are met. The semi-heterarchical concept of our framework allows to adapt to dynamic changes in real-time, while still maintaining an overall coordination to optimize the whole system’s productivity and efficiency. This balance between local autonomy and global control is key to achieving a robust, flexible, and efficient production system.

Fig. 9.1 illustrates the modular production system, encompassing two manufacturing modules M1.1/M1.2, a quality assurance station Q1, and the high-level distribution module D1. In the subsequent simulated scenario, two order groups 1 and 2 are manufactured on the machines within the respective modules M1.1 and M1.2. The DES is predicated on pre-defined events, satisfying the Markov property and enabling a precise calculation of pre-established production states and potential state transitions. Every agent is equipped with one online and one target network and possesses the capability to dispatch a variety of products from a pickup point to a predetermined destination. These destinations could include a machine, a module storage area, or a transfer buffer between two modules. Thereby, each order has specific global and local process information that, cumulatively, constitute the state space of an agent. Each agent receives an individual state for decision-making, enabling an optimal and situation-dependent dispatching process. The resulting data set, which stores past-made state transitions, respectively chosen actions, and rewards, s_t, s_{t+1}, a_t, r_t , is stored in a batch. This batch then serves as the training basis during the batch replay and facilitates a continuous policy update.

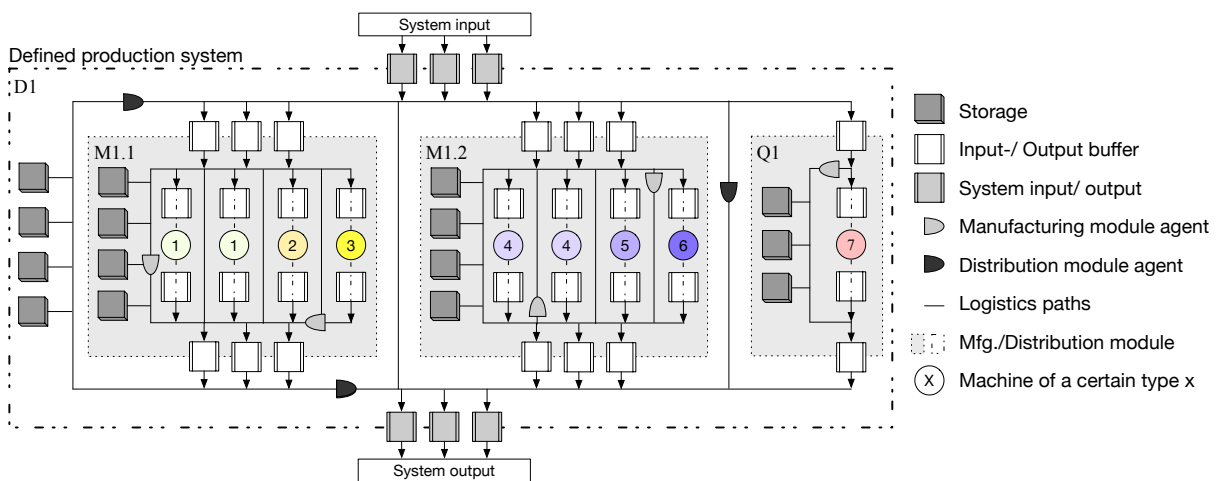


Figure 9.1 Simulated modular production system

9.3.1 Agent state space

Prior to each dispatching decision, the agents receive a tailored state according to the type and size of their module along with its constituent module resources. For the state space calculation, a min-max normalization is employed to harmonize the input parameters and stabilize the learning

process. This method constraints continuous process information inputs within the range of $[-1, 1]$. This makes it possible for the agent to still derive insights regarding the urgency or waiting time of an order at any given production state. For discrete values like order priority or urgency, discrete variables are utilized as indicated in Table 9.1. The variables are computed for each available space within an agent’s respective module, containing information about the other agents as well, and are concatenated into an overall vector, which is then inputted into the neural network.

State	Notation	Description
Throughput time - local	s_{tptg}	Order entry timestamp - system
Throughput time - local	s_{tptl}	Order entry timestamp - module
Due date	s_{dd}	Point of time, at which an order is due
Priority	s_{prio}	Order priority is determined by a binary variable, where $s_{prio} = 0/1$ indicates a non-/ priority order
Urgency	s_{urg}	Order urgency is determined by a binary variable, where $s_{urg} = 0/1$ indicates a non-/ urgent order

Table 9.1 Production state parameters

9.3.2 Agent action space

The DQN, as a value-based RL algorithm, features a discrete action space wherein the Q-values represent the success estimate for each action. The discrete action $a = \operatorname{argmax}_a Q(s_t, a)$, that offers the highest Q-value, is executed with a probability of $1 - \varepsilon$. The epsilon factor ε decreases with each training step and enables a sufficient exploration during the beginning of the training process. The defined actions should enable an efficient order processing and to fulfil local and global objectives such as reducing local buffer stocks and global order through-put time or tardiness. To achieve this, the deep RL agents select the most appropriate action from a set of dispatching rules that consists of a FiFo rule on a global or local scale, a EDD, and a High-Priority (HP) rule. This approach offers the advantage of mapping the process logic into the low-level rule-set, thereby circumventing the need of an agent to learn process interdependencies and system constraints. This can reduce the optimization complexity and accelerate the learning process towards the optimal control policy. Each agent employs the same set of rules and carries out its actions individually, based on the intrinsically developed policy.

9.3.3 Designing the reward function

The design of the reward function is crucial for the learning performance indicators and should align with the set objectives of optimizing technical and financial parameters. It's noteworthy that these objectives can conflict with each other, necessitating a reliable reward design that can be accurately modeled in the DES. For this purpose, we developed a combined reward function that was derived from the optimization criteria of optimizing order tardiness and throughput, and the additional services offered for urgent and priority orders. This function is generic and can be adapted to other scenarios. The total reward is calculated by the sum of the partial rewards $r_{ges} = \sum r_i$ for a given order. The individual optimization criteria r_i are equally weighted in subsequent testings but can be adjusted to specific scenarios. A cumulative reward of $r_i = 200$ is given for selecting the order with the longest waiting time in the system/module, with the most urgent due date, or a high priority or urgency, respectively. In contrast, $r_i = -200$ is assigned, if the order with the poorest metric is chosen. In cases where the order lies between the worst and best options for continuous metrics (such as throughput time), a linear relationship is established, i.e. resulting in a reward of 0 when the selected order is precisely between the worst and best possibilities.

9.4 Simulation results and analysis

The following section provides a detailed description of the simulated scenario, that was used to conduct the techno-financial analysis. The calculations were performed on an Intel Core i9-12900k CPU with 32GB of RAM. Unless specified otherwise, three simulation runs were run for each metric.

9.4.1 Experimental settings

The simulated system was previously visualized in Fig. 9.1. A total of 8 dispatching agents were applied in the production system, each of which was controlled by two neural networks. Relevant network parameters were transferred from Mnih et al. (2013). The techno-financial process parameters pertaining to the production of the two order groups are listed in Table 9.2. The order groups consist of two distinct types of steel shafts, with varying characteristics between group 1 and 2. Each base order undergoes milling/grinding and hardening processes throughout the manufacturing process. Additionally, an supplementary labelling service is available for each group and there is a 10% probability for a quality check. Further, there is a 25% probability of an order being classified as a rush or priority order. A rush order may be directly requested by customers or is critical for the operation of neighboured systems. A priority order is not necessarily related to time urgency but for other reasons such as the order value or

business considerations like the strategic customer importance. The remaining order parameters, including the due date or system events such as machine failures, are stochastic parameters and are determined randomly.

	Order			
	Type 1.1	Type 1.2	Type 2.1	Type 2.2
Order frequency	30%	20%	30%	20%
Machining steps	1→2	1→2→3	4→5	4→5→6
Add. services	-	Labelling	-	Labelling
Processing times [sec.]	19.2	24.4	26.6	34.3
Revenue [\$]	100	110	150	160
Proc. cost [\$]	60	65	85	90

Table 9.2 Experimental order settings for simulation

The sales data and unit costs provided in Table 9.2 were used to compute the key financial indicators. Additional sales and corresponding penalty costs resulting from priority and urgent orders were calculated using constant fees or penalties. Specifically, a fee of \$30 was applied for priority processing, \$30 for urgent service, and \$50 for combined processing. Regarding order tardiness, the penalty for standard orders amounted to \$0.2 per minute of tardiness, while rush orders incurred a penalty of \$0.5 per minute of tardiness. For prioritized orders, the penalty escalated to \$2 per minute. These various amounts were aggregated to yield a final profit.

9.4.2 Experimental results

The assessment of training efficacy and the achievement of the desired performance objectives can be initially analyzed on the basis of the earned rewards. Fig. 9.2 displays the learning curve, which indicates a significant upward trend in the obtained reward signals. Notably, the performance, which was initially modest, not only exhibits consistent improvement over time, but there are also significantly reduced fluctuations in performance.

As a benchmark, we employed the EDD rule as well as the widely used global FIFO rule, similar to comparable approaches as discussed in Popper and Ruskowski (2022). As summarized in the Tables 9.3 and 9.4, our approach was able to improve the financial and technical indicators of customer-centric order processing. Specifically, the imposition of service fees in the form of penalties has been significantly reduced, leading to a notable 6% increase in total profit. This not only opens up opportunities for more intricate multi-objective optimization strategies, but also integrates pertinent customer-centric services that have the potential to increase revenues from additional services when being appropriately integrated into shop-floor operations.

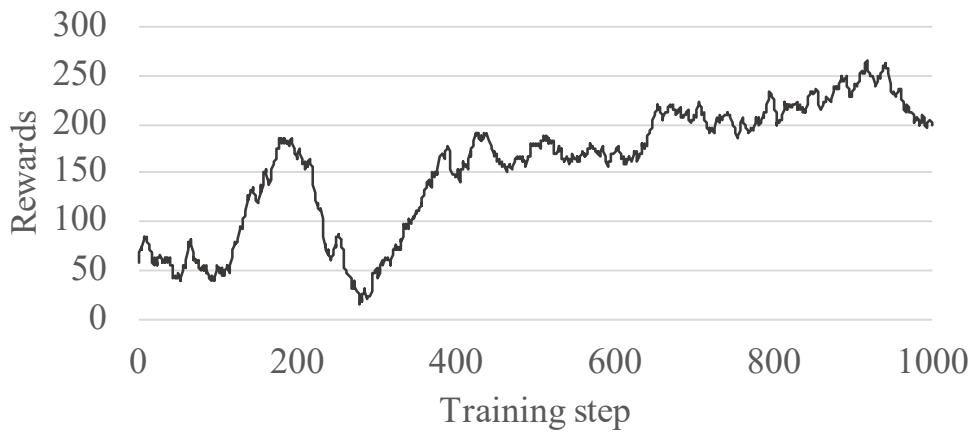


Figure 9.2 Reward progression of a deep RL agent during the training phase

	Deep RL	EDD		FiFo global	
Total revenue	\$55856,7	\$55766,7	-0,2%	\$55526,7	-0,6%
Thereof service	\$5110,0	\$5066,7	-0,8%	\$4980,0	-2,5%
Processing cost	\$29293,3	\$29266,7	-0,1%	\$29180,0	-0,4%
Service penalties	\$1915,2	\$3233,7	68,8%	\$3339,9	74,4%
Profit	\$24648,1	\$23266,3	-6,3%	\$23006,8	-7,3%

Table 9.3 Summary of key financial indicators

Table 9.4 provides a comprehensive breakdown of throughput times T_{tpt} and tardiness rates T_{tardi} for the different order classed. The obtained metrics indicate a significant improvement in the processing of priority, rush, and combined orders. However, it also reveals a slight increase in order throughput times and tardiness for standard orders, as expected. This analysis clearly reveals that in particular the priority and combined urgent-priority orders, represented in the lower rows of the table, were processed significantly faster with a throughput time of 57.9 and 70.8 seconds and with a highly reduced tardiness of 7.8 and 0.6 seconds. Compared to the EDD and FiFo rules, this also yields in the aforementioned optimized financial metrics. However, it is also clear that the standard orders in the first row exhibited slower processing, resulting in a slightly increased tardiness of 18.9 seconds compared to 15.5 seconds with the global FiFo rule. However, given the lower individual order significance, this increased tardiness is less critical.

Notable improvements in the indicators of combined order metrics were achieved, which may be less attainable with conventional rules. In addition, individual performance indicators can be further adjusted to accommodate specific production conditions through a scenario-specific reward weighting, which facilitates the development of use-case optimal policies. The modular production further facilitates the generation of transferable knowledge across different scenarios, resulting in significant savings in computation resources. Moreover, this approach capitalizes on shared knowledge and eliminates the need for system re-training in each instance, thus enabling

Priority	Urgency	Deep RL		EDD		FiFo global	
		T_{tpt}	T_{tardi}	T_{tpt}	T_{tardi}	T_{tpt}	T_{tardi}
Non-prio	Standard	83,0	18,9	81,9	18,5	77,0	15,5
	Urgent	75,7	7,7	71,5	4,8	86,9	17,8
Prio	Standard	70,8	7,8	82,0	18,7	74,3	13,0
	Urgent	57,9	0,6	61,0	3,1	69,9	9,8

Table 9.4 Summary of key technical indicators [sec.]

the utilization and exploitation of existing knowledge.

9.5 Conclusion

In times of increased market fluctuations and production complexities, production control approaches must effectively address the increased dynamic requirements. To accommodate a high degree of adaptability to varying fabrication processes, modular production approaches were increasingly implemented, although they entail a high degree of control complexity. In this study, we addressed the increased control complexity by deploying a deep RL based control framework with multiple dispatching agents. We conducted a techno-financial optimization of the production processes to mitigate the challenges posed by the complex operational production.

The multiple DQN based agents were trained based on concurrent learning and were able to process priority and rush orders significantly faster and with less tardiness. Based on a pre-defined multi-objective reward function, not only technical key indicators were improved, but the deep learning control also led to substantial reductions in tardiness penalties, resulting in increased profits. Conventional rules were significantly outperformed, thus fostering an incentive for the integration and optimization of the presented deep learning based control approach to facilitate the integration customer-centric services.

The incorporation of the desired objectives into a representative reward function offers a promising approach to simultaneously address multiple technical and financial objectives in production systems. The decentralized modular control approach further enables an easy adoption of new resources and processes. Future research should focus on improving operational performance by reducing training efforts and bridging the gap between real-world and simulated environments.

Copyright notice

This is an accepted version of this article, which is in publishing process:

Panzer, M. and N. Gronau (forthcoming, expected 2024). Enhancing economic efficiency in modular production systems through deep reinforcement learning. *Procedia CIRP*. Accepted for publication.

Clarification of the copyright adjusted according to the guidelines of the publisher.

Contributor roles

This paper is the result of collaborative efforts where specific responsibilities were allocated to ensure the effective completion of the research and the preparation of the manuscript:

- **Marcel Panzer:** Took the lead role in the majority of the work associated with this publication. This included the conceptualization of the research, the design and execution of the methodology, simulations, and evaluations, the collection and analysis of data, and the primary responsibility for writing the original draft of the manuscript. The contribution also extended to the compilation and editing of the final manuscript during the publication reviews.
- **Norbert Gronau:** Played a role in the development of this publication through his extensive review and guidance. His contributions included critical review, providing insightful feedback and suggestions. His guidance was instrumental in shaping the publication, ensuring academic rigor and alignment with the intended research objectives.

The *Declaration of the Co-Authors* is inserted at the end of this thesis.

Publication 5 - References

- Balaji, P. G. and D. Srinivasan (2010). An Introduction to Multi-Agent Systems. In: J. Kacprzyk, D. Srinivasan, and L. C. Jain (Hrsg.), *Innovations in Multi-Agent Systems and Applications - I*, Volume 310, p. 1–27. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-14435-6.
- Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton, N.J.: Princeton University Press.
- Bongaerts, L., L. Monostori, D. McFarlane and B. Kádár (2000). Hierarchy in distributed shop floor control. *Computers in Industry* 43(2), p. 123–137. doi: 10.1016/S0166-3615(00)00062-2.
- Chang, J., D. Yu, Z. Zhou, W. He and L. Zhang (2022). Hierarchical Reinforcement Learning for Multi-Objective Real-Time Flexible Scheduling in a Smart Shop Floor. *Machines* 10(12), p. 1195. doi: 10.3390/machines10121195.
- Dittrich, M.-A. and S. Fohlmeister (2020). Cooperative multi-agent system for production control using reinforcement learning. *CIRP Annals* 69(1), p. 389 – 392. doi: 10.1016/j.cirp.2020.04.005.
- Gankin, D., S. Mayer, J. Zinn, B. Vogel-Heuser and C. Endisch (2021). Modular Production Control with Multi-Agent Deep Q-Learning. In: *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vasteras, Sweden, p. 1–8. IEEE. doi: 10.1109/ETFA45728.2021.9613177.
- Grassi, A., G. Guizzi, L. C. Santillo and S. Vespoli (2020). A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters* 24, p. 43–46. doi: 10.1016/j.mfglet.2020.03.007.
- Greschke, P., M. Schönemann, S. Thiede and C. Herrmann (2014). Matrix Structures for High Volumes and Flexibility in Production Systems. *Procedia CIRP* 17, p. 160–165. doi: 10.1016/j.procir.2014.02.040.
- Groover, M. P. (2019). *Automation, production systems, and computer-integrated manufacturing* (Fifth edition ed.). New York: Pearson Education. ISBN: 978-0-13-460546-3.
- Gros, T. P., J. Gros and V. Wolf (2020). Real-Time Decision Making for a Car Manufacturing Process Using Deep Reinforcement Learning. In: *2020 Winter Simulation Conference (WSC)*, Orlando, FL, USA, p. 3032–3044. IEEE. doi: 10.1109/WSC48552.2020.9383884.
- Hammami, Z., W. Mouelhi and L. Ben Said (2017). On-line self-adaptive framework for tailoring a neural-agent learning model addressing dynamic real-time scheduling problems. *Journal of Manufacturing Systems* 45, p. 97–108. doi: 10.1016/j.jmsy.2017.08.003.
- Hofmann, C., C. Krahe, N. Stricker and G. Lanza (2020). Autonomous production con-

- trol for matrix production based on deep Q-learning. *Procedia CIRP* 88, p. 25–30. doi: 10.1016/j.procir.2020.05.005.
- Kuhnle, A., M. C. May, L. Schäfer and G. Lanza (2021). Explainable reinforcement learning in production control of job shop manufacturing system. *International Journal of Production Research*, p. 1–23. doi: 10.1080/00207543.2021.1972179.
- Lass, S. and N. Gronau (2020). A factory operating system for extending existing factories to Industry 4.0. *Computers in Industry* 115. doi: 10.1016/j.compind.2019.103128.
- Lee, J., B. Bagheri and C. Jin (2016). Introduction to cyber manufacturing. *Manufacturing Letters* 8, p. 11–15. doi: 10.1016/j.mfglet.2016.05.002.
- Malus, A., D. Kozjek and R. Vrabič (2020). Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. *CIRP Annals* 69(1), p. 397 – 400. doi: 10.1016/j.cirp.2020.04.001.
- May, M. C., L. Kiefer, A. Kuhnle, N. Stricker and G. Lanza (2021). Decentralized Multi-Agent Production Control through Economic Model Bidding for Matrix Production Systems. *Procedia CIRP* 96, p. 3–8. doi: 10.1016/j.procir.2021.01.043.
- Mayer, S., T. Classen and C. Endisch (2021). Modular production control using deep reinforcement learning: proximal policy optimization. *Journal of Intelligent Manufacturing* 32(8), p. 2335–2351. doi: 10.1007/s10845-021-01778-z.
- Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller (2013). Playing Atari with Deep Reinforcement Learning. p. arXiv:1312.5602.
- Overbeck, L., A. Hugues, M. C. May, A. Kuhnle and G. Lanza (2021). Reinforcement Learning Based Production Control of Semi-automated Manufacturing Systems. *Procedia CIRP* 103, p. 170–175. doi: 10.1016/j.procir.2021.10.027.
- Panzer, M. and B. Bender (2022). Deep reinforcement learning in production systems: a systematic literature review. *International Journal of Production Research* 60(13), p. 4316–4341. doi: 10.1080/00207543.2021.1973138.
- Panzer, M., B. Bender and N. Gronau (2022). Neural agent-based production planning and control: An architectural review. *Journal of Manufacturing Systems* 65, p. 743–766. doi: 10.1016/j.jmsy.2022.10.019.
- Parente, M., G. Figueira, P. Amorim and A. Marques (2020). Production scheduling in the context of Industry 4.0: review and trends. *International Journal of Production Research* 58(17), p. 5401–5431. doi: 10.1080/00207543.2020.1718794.
- Popper, J. and M. Ruskowski (2022). Using Multi-Agent Deep Reinforcement Learning For Flexible Job Shop Scheduling Problems. *Procedia CIRP* 112, p. 63–67.

10.1016/j.procir.2022.09.039.

- Samsonov, V., M. Kemmerling, M. Paegert, D. Lütticke, F. Sauermann, A. Gützlaff, G. Schuh and T. Meisen (2021). Manufacturing Control in Job Shop Environments with Reinforcement Learning. In: *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*, p. 589–597. doi: 10.5220/0010202405890597.
- Sutton, R. S. and A. G. Barto (2017). *Reinforcement learning: an introduction* (2nd ed.). Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press. ISBN: 978-0-262-03924-6.
- Tampuu, A., T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru and R. Vicente (2017). Multiagent cooperation and competition with deep reinforcement learning. *PLOS 12*(4). doi: 10.1371/journal.pone.0172395.
- Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmuller, T. Bauernhansl, A. Knapp and A. Kyek (2018). Deep reinforcement learning for semiconductor production scheduling. In: *29th Annual SEMI Advanced Semiconductor Manufacturing Conference*, Saratoga Springs, NY, USA. doi: 10.1109/ASMC.2018.8373191.

10 Discussion

This chapter presents a comprehensive analysis of the research outcomes. The findings reveal a broad application range for deep learning in production research. However, in the realm of production control, there is a predominant reliance on job-shop scenarios, as highlighted in publications 1 and 2 (Chapters 4, 5). The identified gaps and derived requirements in current methodologies informed the development of a deep learning based production control artifact, as detailed in publications 3 to 5 (Chapters 7, 8, 9).

The subsequent sections commence with an integrative analysis of the research outcomes in Section 10.1. Following this, Section 10.2 addresses the transferability of these findings and their broader implications for academic scholars as well as for practitioners and managerial stakeholders.

10.1 Integration of the results

This thesis focuses on creating an adaptive production control framework. It prioritizes rapid decision-making across various objectives, robust processes, and adaptability to system changes and disruptions. Conventional optimization methods, like meta-heuristics, are often challenged by the condensed nature of real-time control problems. Deep reinforcement learning demonstrated promising results in complex job-shop systems, but its use has been mainly limited to single-agent and centralized systems, or specific configurations. To address this gap, the following sections answer the formulated research questions. It starts with an examination of the established requirements in Section 10.1.1, proceeds to evaluate production complexity in Section 10.1.2, and examines the framework's generalizability in Section 10.1.3. Ultimately, it integrates these insights to tackle the primary research question in Section 10.1.4.

10.1.1 Requirements for deep learning based control optimization methodologies

In the initial set of publications, the significant opportunity presented by integrating deep learning into production systems was emphasized, particularly in the context of autonomous and decentralized production processes. Understanding the unique requirements and challenges specific to these systems is crucial to fully leverage the potential of deep learning based production approaches. This leads to the formulation of the first sub-research question, S-RQ1.

S-RQ1: What requirements do production systems impose on deep learning based control optimization methodologies?

In the first publication (see Chapter 4), the focus was on algorithmic analysis, revealing deep learning's effectiveness in production applications, especially deep reinforcement learning for

complex optimization challenges. However, it highlighted a research gap in applying these methods beyond traditional job-shop or matrix production models, pointing to a need for better resource integration and optimization. The second publication (see Chapter 5) provides a comprehensive view of organizational aspects, noting a dominance of single-layer and single-agent systems. Based on the results of this first bundle of publications, this thesis categorizes the requirements for novel control optimization methodologies into structural, organizational, and algorithmic aspects, extensively discussed in Section 6.1, which significantly influenced the thesis's direction.

First, from a structural perspective, these systems require a high degree of adaptability and robustness, similar to the flexibility observed in dynamic and complex job-shop and matrix production structures. This includes the ability to withstand fluctuations in internal and external parameters, such as order sizes or system interruptions. However, there's an need for resource consolidation. Unlike the rather low productivity of job-shop and matrix production structures, novel methodologies in deep learning based research should allow coping with modular designed production systems that bundle and group resources. This approach entails the use of standardized and object-oriented modules to leverage throughput performance and system efficiency.

Second, organizational requirements for complex scenarios increasingly focus on shifting from single-agent, centralized systems to decentralized, multi-agent structures. This shift, underscored in recent research like Zhou et al. (2021), is critical for managing task complexity, enhancing scalability, and responsiveness in large-scale production. However, integrating deep learning based production agents in multi-layered structures is still under explored, limiting the optimization potential of these agents. Such integration requirements are essential for aligning optimization strategies with various production layers, and require not only defining the quantity and interactions of autonomous agents but also leveraging the expertise of domain-specific specialists. These specialists, more adaptable and efficient than generalists, play a key role in enhancing system adaptability across various operational areas like intra-logistics or material dispatching.

Third, from an algorithmic perspective, deep learning based control approaches predominantly utilized deep reinforcement learning. These facilitated real-time control decisions and handled multi-criteria decision-making. Despite this capability, optimization was generally confined to one or two parameters, with only rare cases extending to three. This indicates a need for broader evaluation scopes, incorporating customer-centric metrics into the reward function and evaluation, beyond technical performance indicators. The structural and organizational requirements of dynamic systems like job-shops, marked by high flexibility and adaptability, also demand the deep learning control approaches for rapid adaption to new environments. This includes dealing with a wide array of influencing factors such as product diversity, machine

type variations, and other complex elements, necessitating standardized input parameters. Also, coping with the increased complexity in optimization not only requires decentralized multi-agent structures but also intra-agent embedded optimization strategies that break down the overall task into manageable sub-tasks. Ensuring robust models is crucial, as agents must adeptly adjust to fluctuating production parameters, facing the challenge of choosing appropriate actions from a vast array of possibilities. This highlights the intricate control and optimization challenges inherent in deploying and fine-tuning deep learning methodologies in complex production systems.

10.1.2 Managing decision-making and optimization complexity

Production system complexity has been extensively studied in various research works. Lödging (2016) highlights material flow complexity as emerging from interactions between process steps and their return flows. ElMaraghy et al. (2012b) takes a broader view, considering manufacturing systems' complexity alongside factors from product design and market perspectives. This includes differentiating between static and dynamic complexity. The overarching insight from these studies is that minimizing complexity is one key to enhance system performance. Kuhnle (2020) presents a nuanced perspective on complexity in production control, focusing on the decision-making process regarding production orders and their allocation to specific devices. Notably, prior studies have incorporated deep-learning algorithms in addressing optimization complexity. This predominantly algorithm-centric implementation approach leads to the formulation of the second sub-research question.

S-RQ2: How can the decision-making and optimization complexity of large systems be distributed among autonomous system components?

In addressing the distribution of decision-making and optimization complexity for large manufacturing systems, this thesis adopts structural, organizational, and algorithmic complexity perspectives, as illustrated in Figure 10.1. From a structural perspective, the adoption of adaptable techniques is imperative to meet evolving consumer demands and technological advancements (Boyer, 2000; Bordoloi et al., 2009). This thesis emphasizes the need to balance core processing synergies with the ability to adapt to systemic changes. A key aspect of this strategy is using standardized production modules to capitalize on deep learning driven control and operational efficiencies. The proposed modular system facilitates an approach between the widely adopted flow-shop and job-shop processes. By segmenting the system into discrete, manageable modules, localized decision-making is enabled, effectively containing complexity within defined module boundaries. This modular approach ensures adaptability and scalability without proportionally increasing the optimization complexity for the individual agents. Nonetheless, the emphasis on structural hardware components presents challenges in adaptability and incurs significant

costs, underscoring the necessity for a strong focus on organizational and algorithmic strategies, especially in optimizing existing systems in brown-field approaches.

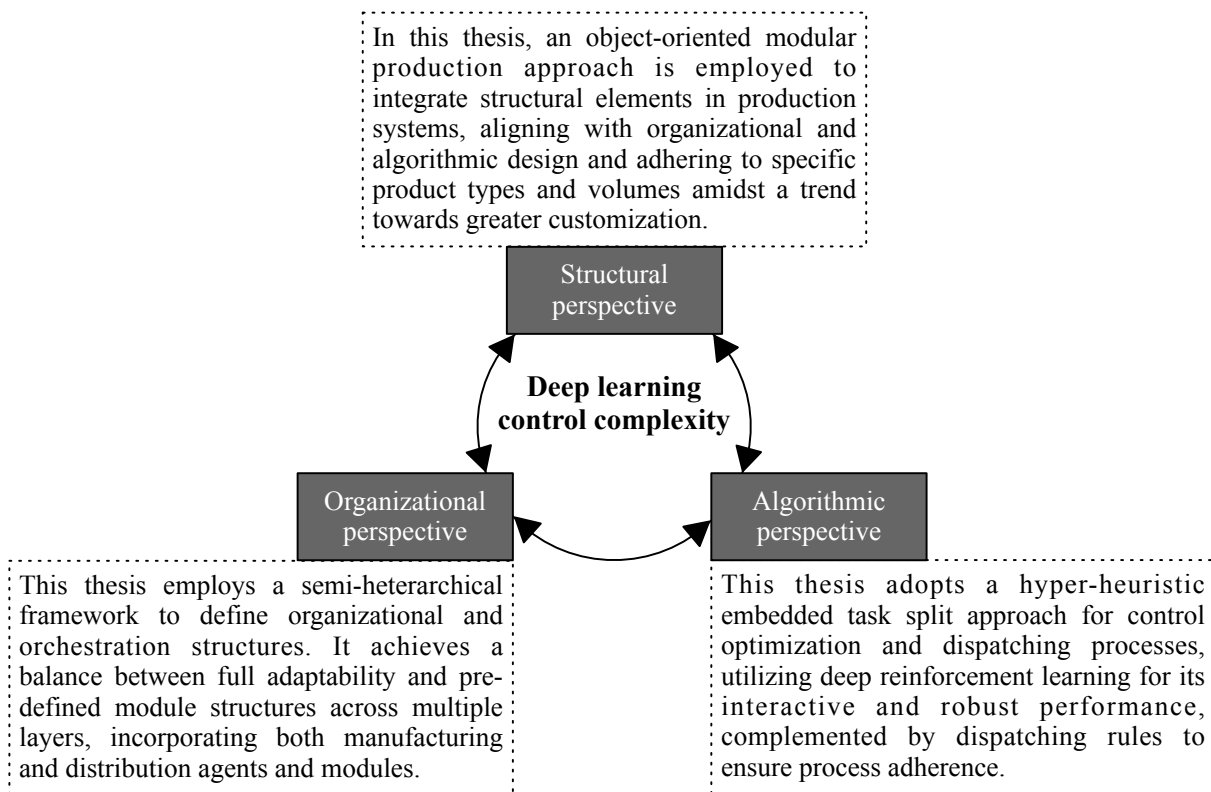


Figure 10.1 Threefold reduction in optimization and control complexity with proposed elements

From the organizational perspective, this thesis emphasizes the critical role of coordinated optimization in large systems, a decision between structured, hierarchical systems and heterarchical, flexible systems, or a hybrid of both for enhanced proactiveness planning and responsiveness. It highlights the necessity for operational agents to assimilate both global and localized data streams and reward mechanisms into their decision-making policies. Evaluations of these agents are proposed to be on an individual basis while incorporating multi-agent and multi-criteria performance metrics, with scalability being a paramount concern. The thesis proposes the use of multi-agent systems to efficiently distribute control complexity across individual agents or groups of agents within large production systems. A semi-heterarchical organization is proposed for the specification of inter- and intra-modular or resource-based organization, which confines complexities to independent modules, allowing heterarchical and autonomous dependencies, which prevents escalating complexities due to specific task allocation. This approach significantly reduces the state and action spaces for deep learning agents and simplifies data processing and neural network learning in the proposed manufacturing and distribution modules and layers.

From an algorithmic perspective, this thesis identifies a prevalent reliance on basic deep learning

techniques, while the benefits of embedded methods in reducing optimization complexity have not been exploited in production control. In large systems, where the control state space experiences exponential growth, segmenting complex tasks becomes increasingly essential. This issue is particularly emphasized in the second publication, highlighting embedded techniques in simplifying complex control tasks (see Chapter 5). Thereby, a single complex control task can be divided into distinct sub-tasks, executable either serially or in parallel. Although control decomposition is utilized in other fields, its extensive application in deep learning based production is limited. This thesis specifically addresses this by splitting task complexity into optimization and operational tasks using a hyper-heuristic approach. While the deep reinforcement learning based top-level heuristics enhances system performance, the allocation of control complexity to a decoupled low-level heuristic achieves robust operation, effectively reducing deadlocks and non-learning behavior. This approach, although it confines the solution space to predefined heuristics, also enhances the explainability of action choices and can be finely tuned for specific applications.

In addressing the complexities of large production systems, this thesis emphasizes the need to not only consider stand-alone perspectives but also their interconnections. For instance, it proposes a modular control structure for segmenting production complexity. However, it must be acknowledged that this approach may not suit all company use-cases. For instance, in systems with a linear flow-shop for manufacturing standardized products, a semi-heterarchical control might prove less efficient than a hierarchical planning system, thus requiring alternative strategies for reducing complexity. This highlights the necessity of an integrated approach that combines structural, organizational, and algorithmic perspectives to manage system complexity effectively.

10.1.3 Generalizability of the developed control framework

The management of complexity in production systems is closely related to the final sub-research question (S-RQ3). Within this framework, the system generalizability, similar to complexity, is not determined by a singular variable. Instead, it was addressed from the interplay of various production resources and integrated into the proposed *CoBra* control framework.

S-RQ3: How can a high level of control generalizability be ensured across varying production scenarios?

The *CoBra* framework's systemic generalizability is primarily attributed to its modular and semi-heterarchical design. The manufacturing and distribution layers enable the framework to be scaled from a single manufacturing module to an entire factory. This is crucial for a holistic process management, including machine occupancy, downtime, and work-in-process balancing. The semi-heterarchical organization enhances generalizability by leveraging a multi-layered production with decentralized decision-making. This allows for the addition and modification of

modules at respective levels without disrupting the overall system, ensuring responses to varying manufacturing environments. Central to each module is the use of deep reinforcement learning, which continuously refines control strategies of the autonomous agents.

This integration of deep learning with conventional rule-based methodologies forms a hyper-heuristic approach, as defined by Cowling et al. (2001). Its hybrid strategy aims to prevent initial and ongoing process disturbances and instabilities for new production scenarios or products, addressing training and operational limitations, such as dead-locks. Additionally, the hyper-heuristic is generalizable across various production contexts. It incorporates adaptable reward functions and flexible action spaces, which are based on widely-used dispatching rules. By segmenting tasks, agents have standardized state and action spaces, which can be further tailored to specific requirements, enhancing process efficiency.

In summary, the holistic generalizability of the *CoBra* control framework is characterized by its structural, organizational and algorithmic features. It supports diverse modular layouts and allows for rapid expansion and adjustments in agent, layer, and module count. The modular design not only facilitates the reuse and modification of trained networks but also enables automated network generation for quick scenario adaptation.

10.1.4 Answering the central research question

The answers to the sub-research questions facilitate to address the central research question of this thesis.

How can a data-driven and autonomous control optimization be designed for adaptive production systems?

Previous research in deep learning based production control has predominantly focused on single-agent systems and standard deep reinforcement learning algorithms. These methods often require retraining for new systems or scenarios and struggle with capturing the logic of large, complex systems, sometimes necessitating action masking to determine permissible actions. Moreover, there has been a bias towards studying job-shop and matrix production systems, known for their adaptability but limited productivity.

This thesis introduces a control framework tailored for dynamic production environments. Central to this framework is its integration into a modular production concept, supported by a semi-heterarchical production organization. This approach effectively tackles the scalability challenges often encountered in centralized and single-agent systems. It does so by using manufacturing and distribution modules that enable task-specific process optimization. Additionally, the approach facilitates plug-and-play simulation configuration, allowing for more flexible and scalable solutions in various operational contexts. Drawing from advancements

in deep learning, the framework uses deep reinforcement learning for real-time, autonomous decision-making and optimization. The continuous training further enables ongoing improvement through feedback and data-driven policy refinement. When embedded within a hyper-heuristic framework, this learning component effectively adapts to various production challenges, like machine breakdowns, fluctuations in demand, and changing order parameters. Performance testings indicate improvements in multi-objective optimization, encompassing both technical and customer-specific parameters. The hyper-heuristic also improves explainability through standardized action spaces based on common dispatching rules, avoiding the common *black-box* problem.

The results of this thesis meet the initial objectives outlined in the introductory section, to create an adaptive and performing control framework. The framework is integrated into various scenarios, thereby outperforming common dispatching rules. A key component in achieving this is the reward function of the deep reinforcement learning algorithm, which effectively integrates and prioritizes various order and process metrics. This component is adaptable to different agent types, thereby fulfilling the second objective of facilitating flexible multi-objective optimization. The last objective, achieving scalability and generalizability, is realized through a structured, multi-agent, and multi-layered approach. In this approach, optimization is conducted using standardized modules, which helps maintain consistent states across diverse scenarios and restricts the exponential expansion of state spaces. These states are managed by the hyper-heuristic system, employing clear and consistent dispatching rules for effective operation.

10.2 Transferability of the results

The thesis presents findings that can hold relevance in both scientific and practical domains, offering a detailed analysis and interpretation. It contributes to the field by addressing a research gap with theoretical foundations and a practical artifact. The insights provided are useful for understanding economic metrics and business applications. By taking a holistic approach, the research aims to facilitate the application of its results in diverse contexts, enhancing the overall comprehension and utility of the knowledge acquired. In the following, the overall research contribution is summarized in Section 10.2.1. Subsequently, a potential research transfer procedure is proposed and the control concept is classified within a broader planning and control context in Section 10.2.2. Finally, managerial insights are presented in Section 10.2.3.

10.2.1 Research contribution

Recent studies indicated a growing interest among researchers in structuring complex system landscapes, focusing on their interdependencies and architectures. This interest is particularly evident in production environments, where dynamic processes are increasingly benefiting from

deep learning based optimization methodologies. However, translating these theoretical methods into practical solutions remains a challenge for both generalists and specialists. While current research reveals dedicated approaches, their integration into a broader context was limited. Therefore, it's vital that these approaches are not only effective but also beneficial to applied research. To aid in this, a taxonomy has been proposed to help researchers identify suitable algorithmic strategies and choose between single-agent or multi-agent models, and between serial or parallel embedded algorithms for process optimization and complexity reduction.

This thesis further proposes a production control framework that incorporates deep learning methodologies in an accessible manner. The framework introduces a multi-layered, embedded approach to deep learning based production control. It connects organizational, structural, and algorithmic perspectives and is divided into a top-level optimization component and a low-level process component for robust training and ongoing optimization. Thereby, a special focus of this thesis was on addressing the control optimization problem and reaching a threefold reduction of its complexity. The control framework extends beyond technical parameters to include order-related aspects, marking a step towards the development of intelligent products with autonomous process navigation. The integrated hyper-heuristic algorithm optimizes conflicting objectives and enhances decision-making transparency, addressing the common *black-box* issue in deep learning. The approach further indicates robustness against order-induced variations. Validating the hybrid scenario implications was crucial, with performance metrics observed in simulations confirmed in an actual application.

A qualitative categorization between the research and practice can be proposed based on the first bundle of publications and the relationship between product diversity and volume proposed by Koren (2010). Therefore, Figure 10.2 expands on the initial Figure 1.2 of Koren (2010) and proposes an integration of the reviews findings. As indicated by Koren (2010), there's an increasing trend towards individualization, enabling parts of individualization, also known as mass customization, through line flows and robotics, thus achieving moderate variety. In contrast, deep learning based production control research, as emphasized in the first two publications, is often applied to job-shop scenarios. While being highly adaptive, these job-shops exhibit limited productivity and may not adequately map required production volumes, as indicated in Figure 10.2. Despite these differences, current research pursues highly adaptive control approaches from a structural perspective, in contrast to practice, which focuses on efficiency and optimized productivity. Modular systems, as suggested by Reichwald and Dietel (1991); Zäpfel (2000); Kellner et al. (2020), can be used to achieve synergies and potential throughput increases, categorized by lower variety but higher volume. This research trajectory could lead to closer alignment with real-world production scenarios, suggesting a shift towards larger volumes with less variety. However, this is an assumption, and a validation of this trajectory in comparison to other strategies must be subject to future research, as outlined in Section 11.2.

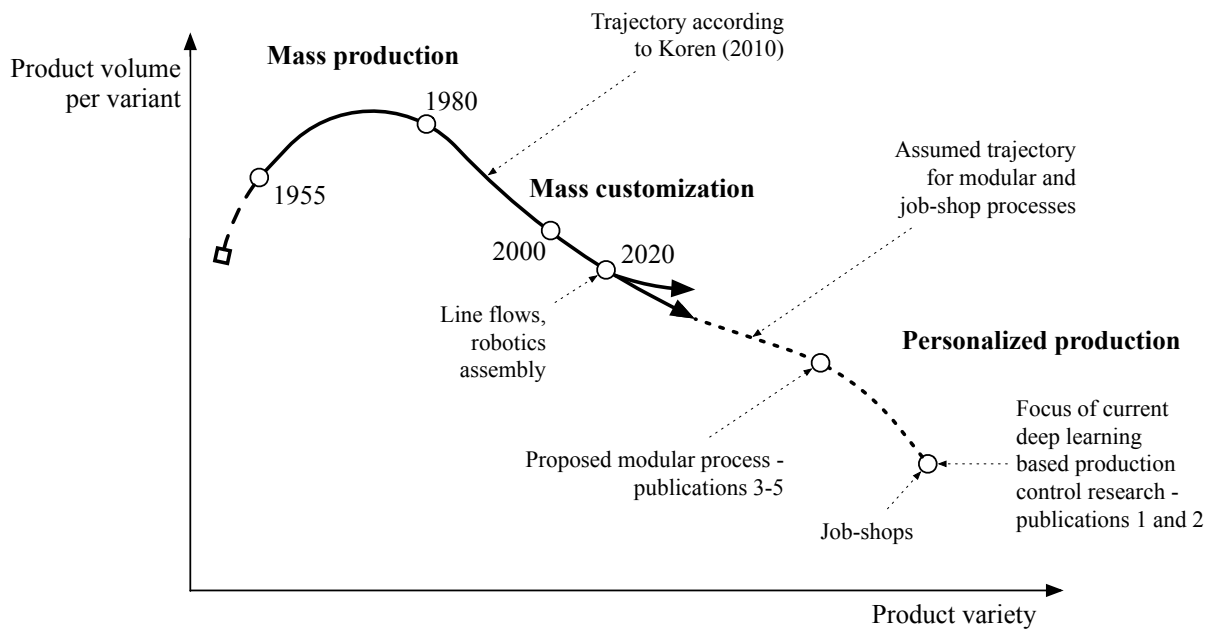


Figure 10.2 Production variety and volume - deep learning research and industry with own classification, the left trajectory is adapted from Koren (2010), while the dotted line resembles the proposed trajectory

10.2.2 Research transfer to operational practice

Transitioning a theoretical framework into practical application necessitates the creation of an integrative process model. The control framework proposed in this thesis was transferred to the *Center for Industry 4.0*. The adaptability of this framework to various real-world settings, leveraging the merits of hyper-heuristics, becomes a point of inquiry. An iterative four-step procedure model is suggested, detailed further by sub-steps in Figure 10.3. Each sub-phase is elaborated upon, supplemented by representative questions and focal points. Also, examples linking directly to the discussed framework are discussed.

Converging and integrating simulation results into operations necessitates thorough preliminary analysis. The initial focus is on evaluating product parameters, as indicated in point 1.1 of Figure 10.3, going beyond just production quantity and variety specifications. A detailed requirements definition and review of sub-variants and corresponding processes is pivotal since foundational decisions for ensuing simulations depend on these insights. The correlation of product and process directly influences the required optimization complexity. Hence, factoring in process steps, their duration, and potential challenges during modeling is vital for practical application. The product thus acts as the starting point for an in-depth process analysis, which is essential for defining all structural, organizational, and algorithmic framework conditions. Concurrently, an analysis of extant process data (1.2) determines the available, pertinent data and any further data collection needs, like product-specific machine failure rates, which can serve as auxiliary

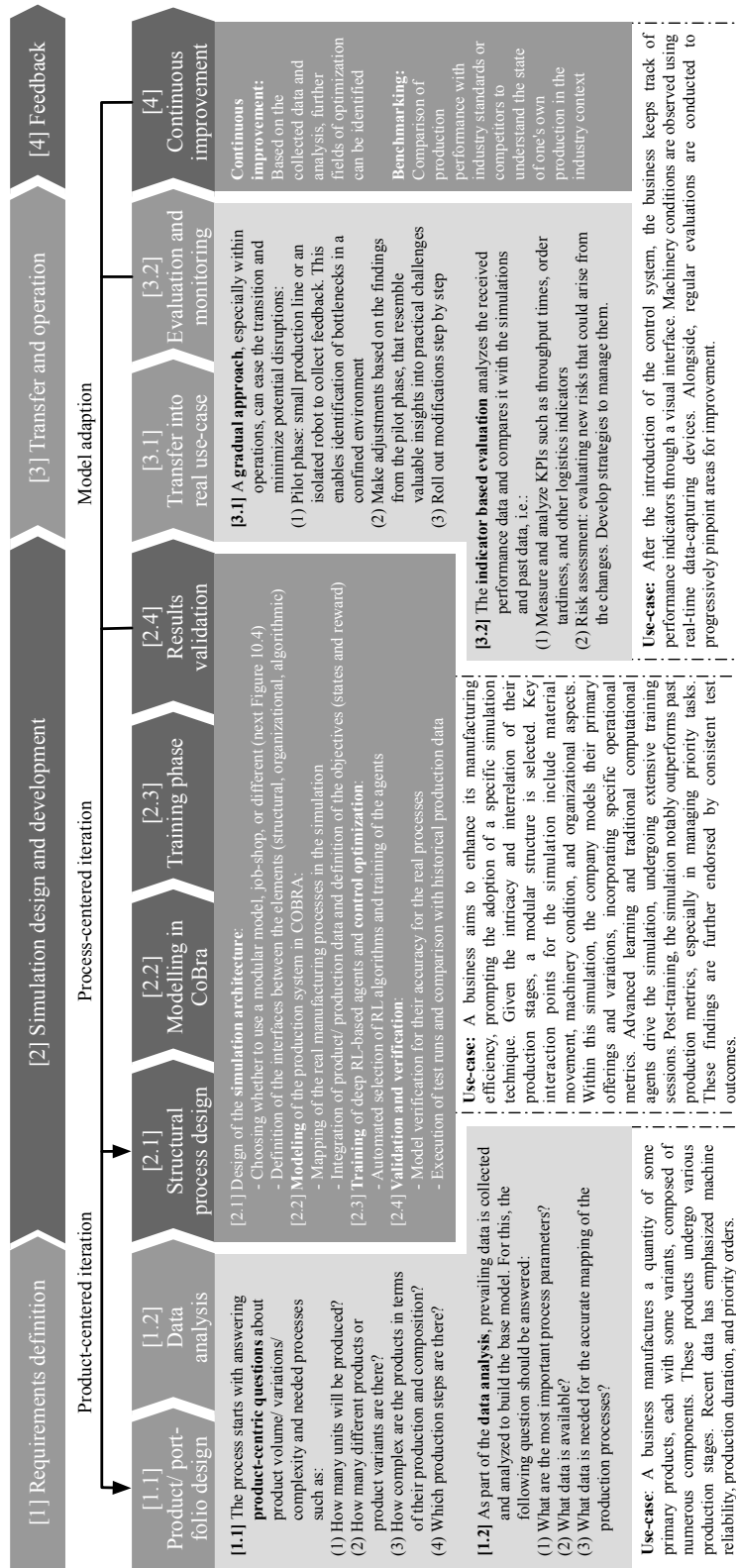


Figure 10.3 Integrative and interactive process model for the framework implementation

quantitative process data.

Subsequent to this foundational analysis and consideration of broader structural parameters, the attention shifts towards examining pre-existing stipulations for the simulation design and development (2), especially in the brownfield sector. Often, these stipulations can limit full process overhauls both technically and financially. This necessitates a comprehensive structural analysis of the system levels under consideration, systematically illustrated in Figure 10.4.

The system levels in Figure 10.4 are adapted from the management and communication levels as per Langmann (2010). It's crucial to note that not every system can consistently apply the same production control as proposed in this thesis, which resembles a limitation in terms of transferability. Within this scope, the systemic intersection where planning activities end and operational control begins, is typically termed the order release point, or operational equilibrium, as indicated on the left in Figure 10.4. In horizontal perspectives, on the integration of information within supply chain networks, such an equilibrium is less achieved due to the systems being less connected. This issue is further compounded by additional vertical planning complexities in the operational equilibrium. Owing to the escalating complexity of planning and the expense associated with planning data generation and preparation, providing error-free planning data becomes highly expensive. This often leads to the generation of non-executable production schedules (Lödding, 2016; Mayer et al., 2016; Lödding, 2019).

The challenges in planning and execution are continued in the critical distinction between either centralized or decentralized planning and control systems (Leitão and Restivo, 2006; Meissner et al., 2017). This distinction is then further manifested in either hierarchical or heterarchical organizations (Sallez et al., 2010). Also, in current research on deep learning based production, the equilibrium or the lack thereof between planning and control, as well as between hierarchical and heterarchical, and centralized and decentralized organization, has been assessed predominantly from one perspective (see Chapters 4, 5, Publications 1 and 2). Consequently, it becomes imperative to leverage deep learning based methodologies from their constrained contexts and situate them within a broader planning and control framework. Rigorous analysis of this operational equilibrium could facilitate optimal control adaptability grounded in structure while preserving planning proficiency in deep learning based systems, thereby demarcating decision boundaries in deep learning driven production and ensuring manageable complexity. This methodology not only mitigates the escalating *Curse of Dimensionality* in complex production systems, but also addresses the structural control complexity as delineated in Duffie (1982) or Duffie and Piper (1986). The latter suggests an exponential increase in complexity as a function of system size, emphasizing the need for advanced and potentially modular production control strategies. This is also a part of the future research directions discussed in Section 11.2.

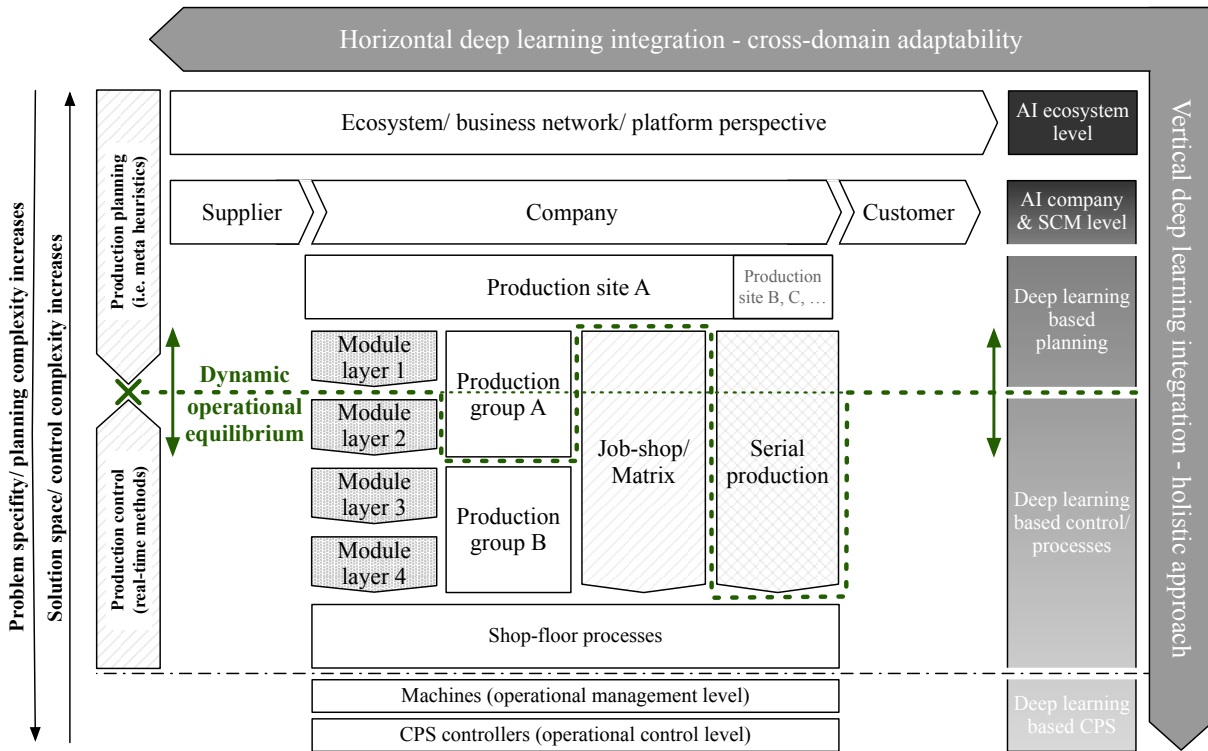


Figure 10.4 Taxonomy of deep learning model integration capabilities

In a modular, multi-layered production system, the boundaries between modules and layers can be flexibly defined. This allows for a operational equilibrium of control and planning between different layers of modules. For example, in Figure 10.4 on the left, this equilibrium is depicted between the first and second module layers. In the first layer, there can be a focus on planning activities, while the layers below can utilize real-time optimization control methods, as proposed in this thesis. This decentralized system enables multiple agents to reduce decision complexity and incorporate specialized optimization and operational knowledge. At higher levels, the centralized planning is responsible for long- to medium-term optimization, considering network and intra-logistics effects like batch tasks. Lower module layers then handle reactive control tasks.

However, this clear differentiation between module layers poses challenges in other production concepts, such as serial production or job-shops. For instance, serial productions, with production schedules planned days in advance, analogous to the *Perlenkette* approach in automotive manufacturing (Weyer, 2002; Herlyn, 2012), might leave only a limited space for control optimization, as indicated by the green dotted line at the bottom. Given the complexity of automotive products, supply orders would need pre-scheduling several days ahead, followed by just-in-time execution. Consequently, businesses must establish the operational equilibrium considering their products and holistic ecosystem information. In contrast, job-shops, often one-layered as analyzed in the first bundle of publications, can allow a broader control scope. Simultaneously, job-shop systems

could also rely on fully planned operations, which would change the dotted line to the bottom in Figure 10.4.

Each scenario impacts the system and the equilibrium between production control and planning differently, as shown by the arrows on the left side in Figure 10.4. The transition from centralized to decentralized principles also plays a significant role in decision-making for both single-agent and multi-agent systems. In this context, the *ADACOR* reference architecture for holonic systems suggests designing a system that is as decentralized as possible, yet as centralized as necessary (Leitão and Restivo, 2006). Therefore, the equilibrium must be set individually, giving modular or grouped production systems more flexibility in determining and establishing such a systemic equilibrium.

Beyond the planning and control levels, one can expand perspectives to both the CPS at the bottom and the organizational level at the top. This encompasses both microscopic aspects, like robotics control, and macroscopic ones, including analysis of the entire corporate structure with its suppliers and customers. A bottom-up approach, moving from higher levels to the shop floor, brings increased specificity. In contrast, a top-down perspective broadens the solution range, with rising control complexity. The effectiveness of this complexity varies. As specificity intensifies, such as in individual assembly processes, localized solutions become more appealing.

Referring back to the design and development process depicted in Figure 10.3, following the initial structure and inherent organization definition, there are critical steps within the conceptual procedure model that demand systematic execution for building the simulation (2). A primary task entails converting relevant process and order data into the states and the reward function. Moreover, integrating product processes is crucial, ensuring the incorporation of previously identified processing data. Post-integration, the automated training phase commences, detailing the neural network specifics and seeking appropriately trained matches. Within this framework, analytical tools are incorporated for process performance evaluation and benchmarking. These tools facilitate comparisons with various standards and historical data, like previously recorded throughput times.

The next step (3), transferring the devised model into a real-world context, is pivotal. While various approaches might be suitable for this transition, a gradual implementation with consistent evaluation and monitoring is advised. Ideally, the agent should obtain data from the real system that mirrors what it garnered during simulation, indicating a direct equivalence in the logistic process definition. An illustrative application is seen with autonomous mobile robots. They can receive, aggregate, and compile system data, with the control framework filtering and relaying only pertinent information to the neural network. Based on the particularly efficient Pandas data-frames used in the simulation, real-time decision-making is enabled.

In the last step (4), to guarantee the approach's efficacy, ongoing refinement of the model is

essential for a continuous improvement process. This phase encompasses routine benchmarking to maintain the model's relevance and efficiency. It involves recurrent navigation through the iterative product- and process-centric loops, aiming to maintain a robust model. The adaptation of models in Figure 10.3 becomes essential as soon as the system conditions change in such a way that either a major or minor adaptation of the control policy is required, which necessitates re-training efforts. In the case of a major product change, it is required to run through the entire process again through the *product-centered iteration*. This is because the majority of processes are affected by such changes through the introduction of new manufacturing processes or the adaptation of process chains. In this context, a re-evaluation of the structural conditions is required, and the modules must be adapted at the manufacturing and distribution levels. This contrasts with *process-centered iteration*. If the structural conditions remain constant and only minor modifications need to be made, Process-Centered Iteration allows intra-modular adjustments that only cause reduced re-training efforts. Such changes may include, for example, the introduction of an additional process step, the implementation of a sub-product, or the addition of another quality assurance level.

10.2.3 Managerial insights

Scientific discussions have frequently highlighted the potential for the proposed approach to be effortlessly integrated into real-world applications (Mohammed et al., 2020). Given the current economic landscape, there is an increasing need for managers to enhance cost and usage efficiency in manufacturing companies through novel control approaches which requires evaluating a diverse set of strategies. As data accumulation continues to expand across various sources, deriving meaningful insights and best practices from this data can position companies advantageously for future challenges (Horng et al., 2020).

As production systems become more interconnected, it leads to the evolution of novel business models. Concurrently, the fastening product cycles necessitate systems to exhibit heightened adaptability. Factors like 24-hour deliveries, bespoke customer demands, and loyalty programs can enhance sales and improve customer retention, but they require proficient control and oversight (Alshurideh et al., 2020; Kim et al., 2021). An expanded parameter set can amplify the complexities involved in process management and optimization as indicated throughout the techno-economical evaluation. Conventional algorithms, requiring manual calibration, may soon be inadequate for expansive systems, potentially binding companies to specific software vendors. To circumvent this, companies should consider strategies that boost process adaptability, diminish process variability, and improve production robustness. It's pivotal for these entities to regard emerging technologies as an integral facet of their future system strategy. Particularly regarding deep learning, often misconstrued as a *black box*, it's crucial to assess and substantiate

its efficacy through tangible outcomes.

In the production context, deep reinforcement learning demonstrates its efficacy as a tool for process optimization across diverse application areas. Its inherent adaptability and robustness contribute to decreased operational costs, increased uptime, and substantially reduced manual process interference. In this thesis, by employing hyper-heuristics, there was an observed profit increase of approximately 7% in the investigated scenario. Additionally, service penalties decreased by as much as 40%. Notably, when benchmarked with conventional rules, the tardiness of critical orders witnessed a 100% reduction in specific instances. Therefore, through deep reinforcement learning, there's potential for in-depth analysis and optimization of both customer-centric parameters and on-site operational procedures.

The framework's reward and optimization function provide considerable flexibility for the control objectives and can be tailored using various parameters or weightings, contingent on the business and process model. The linear reward function facilitates the inclusion of conflicting objectives that necessitate explicit trade-offs and objective balancing. Different process management strategies can further be implemented at distinct levels.

For the realization of the suggested approach, the *operational equilibrium* was outlined, facilitating a nuanced distinction between planning and control tasks. To authenticate the viability and effectiveness of this equilibrium, simulation methods, such as the aforementioned *CoBra* framework, can represent cost-efficient evaluation tools. By utilizing these simulations, businesses can gain preliminary insights and pinpoint viable application areas. It's worth highlighting that both the simulation and the foundational neural models of the control framework are open source, further lowering barriers for first-time users. For the tangible translation and assessment of this theory into industrial practices, initiating pilot studies is advised. These studies aim to evaluate the value-added and relevance of deep reinforcement learning within specific scenarios. Concurrently, active participation from management is paramount, their involvement is pivotal in initiating organizational shifts and facilitating the adoption of pioneering technologies.

11 Summary

In this thesis, the emphasis was on the development of a deep learning based production control framework. This framework should not only be highly adaptable but should also facilitate the improvement of key performance indicators such as throughput times or tardiness. According to the *VDI5201* definition, an adaptive system should efficiently execute systemic changes with minimal effort, enabling it to continuously adapt to evolving external and internal conditions, thereby enhancing company competitiveness (VDI, 2017). Such adaptations encompass the system responsiveness to machine breakdowns, the flexibility in the face of fluctuating order volumes, or the incorporation of system functionalities and capacities.

In the first bundle of publications, the field of deep learning based production was analyzed and structured. It was observed that while deep learning made substantial advancements in recent years, it was primarily applied to constrained problem domains. To date, there's a noticeable absence of comprehensive frameworks for multi-agent systems. Additionally, a taxonomy was proposed to classify deep learning production methodologies, offering a transparent representation of both the arrangement of the agents involved and the foundational algorithmic structure.

Based on these findings, specific requirements are identified, which are subsequently converted into design specifications for a technical artifact, adhering to the *DSRM* approach, in the second bundle of publications, encompassing publications three to five. An iterative development process was employed, leveraging system structure, organization, and algorithm as a threefold foundation for leveraging system adaptability.

The detailed findings are incorporated into a deep learning based production control framework. The simulated control framework varies across multiple levels and agents, each controlled by a hyper-heuristic that embeds a deep reinforcement learning algorithm with dispatching rules. The modular components offer flexibility in configuration and adaptation, facilitating the creation of systems with diverse sizes and capacities. Due to the standardization of the modules and agents, neural networks can be re-used or trained for new modules. The neural networks are stored in a stack accessible to all agents. It can be specifically defined for each production agent whether specific networks should be reused. If not, the control framework automatically creates new neural networks to be integrated into the corresponding agents. This guarantees not only a high degree of scenario adaptability but also rapid transitions without compromising the foundational knowledge base.

Individual agents, like distribution and manufacturing agents, undergo parallel training and optimization for specific performance indicators. Results show that the deep learning based control framework outperforms traditional dispatching rules used in the industry, enabling multi-objective optimization. Improvements are observed in technical indicators such as work-

in-progress rates, order indicators like throughput times, and economic indicators, including profit. The optimization strategy also shows robustness in handling varying order volumes. The integration of deep reinforcement learning enables real-time control during operational mode without straining computational resources. Additionally, the developed technology undergoes practical evaluation in a real-world environment, specifically at the *Center for Industry 4.0*.

While this thesis addresses the underlying research questions, it's important to acknowledge its limitations for a balanced evaluation. These limitations, detailed in Section 11.1, lead to fields for future research, that are further elaborated in Section 11.2.

11.1 Critical appraisal of the thesis

In this thesis, a deep learning based control framework was designed to optimize production processes. Tailored to address pivotal research queries, it's imperative to consistently conduct a critical evaluation of both the developed artifact and its derived outcomes.

First, during the design and evolution phase of the control framework, efforts focused on expanding and deepening its scope. Although the framework does not fully represent real conditions or a complete digital twin, this limitation is intentional. It incorporates only aspects relevant to the specific problem and scope, thus reducing optimization complexity and implementation efforts. This constrained modular approach, however, may limit the framework's applicability to other productions areas, such as serial production lines. Therefore, it is important to contextualize the simulations and evaluations within the given scientific and practical contexts. While the framework could be applied in different industries, implementing a modular and semi-hierarchical structure isn't always feasible. Also, as highlighted by Heger (2007), Wiendahl et al. (2015), and ElMaraghy and Wiendahl (2019), further adaptability factors such as scalability, modularity, compatibility, mobility, and universality are critical, especially in brownfield scenarios. Therefore, future developments should also respect and include elements like conveyor systems, that are widely used in practice, in addition to autonomous mobile robots. Also, many companies use specialized software for production simulations, and while the adaptability of the control framework presented in this thesis is notable, its transferability to such systems needs further exploration.

Second, when examining performance from the perspective of key indicators, it's beneficial to consider additional evaluative parameters for a more in-depth understanding. One of such parameters can be the optimal modularity factor. While these parameters offer valuable insights into networked systems, their focus is less algorithmic, as highlighted by Newman and Girvan (2004). In terms of benchmarking and optimization evaluation, more exhaustive indicators and interconnected perspectives, such as the production operating curve, that combines work-in-progress levels, throughput times, and capacity utilization (Nyhuis and Wiendahl, 2012), require

further analysis, both in simulations and hybrid testing at the *Center for Industry 4.0*. Nyhuis and Wiendahl (2012) also discusses the flow rate as a central process metric. Referring to this flow rate and the data from the fifth publication, specifically Tables 9.4 and 9.5, for all orders, an average minimum throughput time of 25.5 seconds was observed, transportation times included. For prioritized rush orders, the average unweighted flow rate is approximately 2.3, derived from an average unweighted throughput time of 57.9 seconds. For regular orders with a throughput time of 83.0 seconds, the flow rate is around 3.2, which appears reasonable when compared to the metric for prioritized rush orders. However, it is crucial to benchmark these metrics against industry standards to ensure they are contextualized within the specific application under consideration. This is important to accurately assess the system's efficiency and competitiveness in a real-world industrial context.

Third, in the context of the deep learning control framework, it's important to acknowledge the data-intensive training and operation process. This brings up the crucial question, if there is sufficient data to integrate the control framework into real applications. In real environments, according to Lödding (2019), there is often a gap between the data used for planning and the actual manufacturing system, which would also represent a central implementation hurdle for the developed framework. Also, further evaluations focusing on control reliability and safety would be crucial to uncover any unexpected behaviours, potential errors, or anomalies not identified in this thesis. Regarding the choice of the hyper-heuristics top-level component, a deep Q-Network was selected due to its impressive results. Although overfitting was not an issue in this study, further learning and operational investigations are necessary, particularly to further enhance stability and to prevent worst-case behaviors such as catastrophic forgetting (during training phases). From a broader control perspective, this thesis focuses only on control activities that enable adaptive behavior and real-time responses. However, it falls short in incorporating planning capabilities for medium- to long-term optimization. Therefore, while deep reinforcement learning can integrate mid- to long-term goals into the reward function, planning approaches might be more suitable for strategic operations. This planning limitation is addressed in detail in the future research section.

11.2 Fields for future research

Based on the conducted research and the formulated limitations, there are potential fields of research that warrant deeper exploration in the future. These are presented in the bullets below.

- **Structural and process perspective - enhancing the framework:**

First, the potential for detailed simulation of support processes is a key area for future exploration. This also includes containers with homogeneous or fluid substances like plastics, as well as modeling complex assembly processes with diverse, product-dependent

sequences, and handling batch processes and re-entrant flows. Additionally, the framework can be improved with more structural details, surpassing the capabilities of methods in the second bundle of publications. Compared to the publications in the second bundle, further refinements and extensions were already made accessible and tested, offering a broader scope for two-dimensional design. This is illustrated in Figure 11.1, which is seamlessly integrated into the automated neural network generation and control optimization process within the *CoBra* framework.

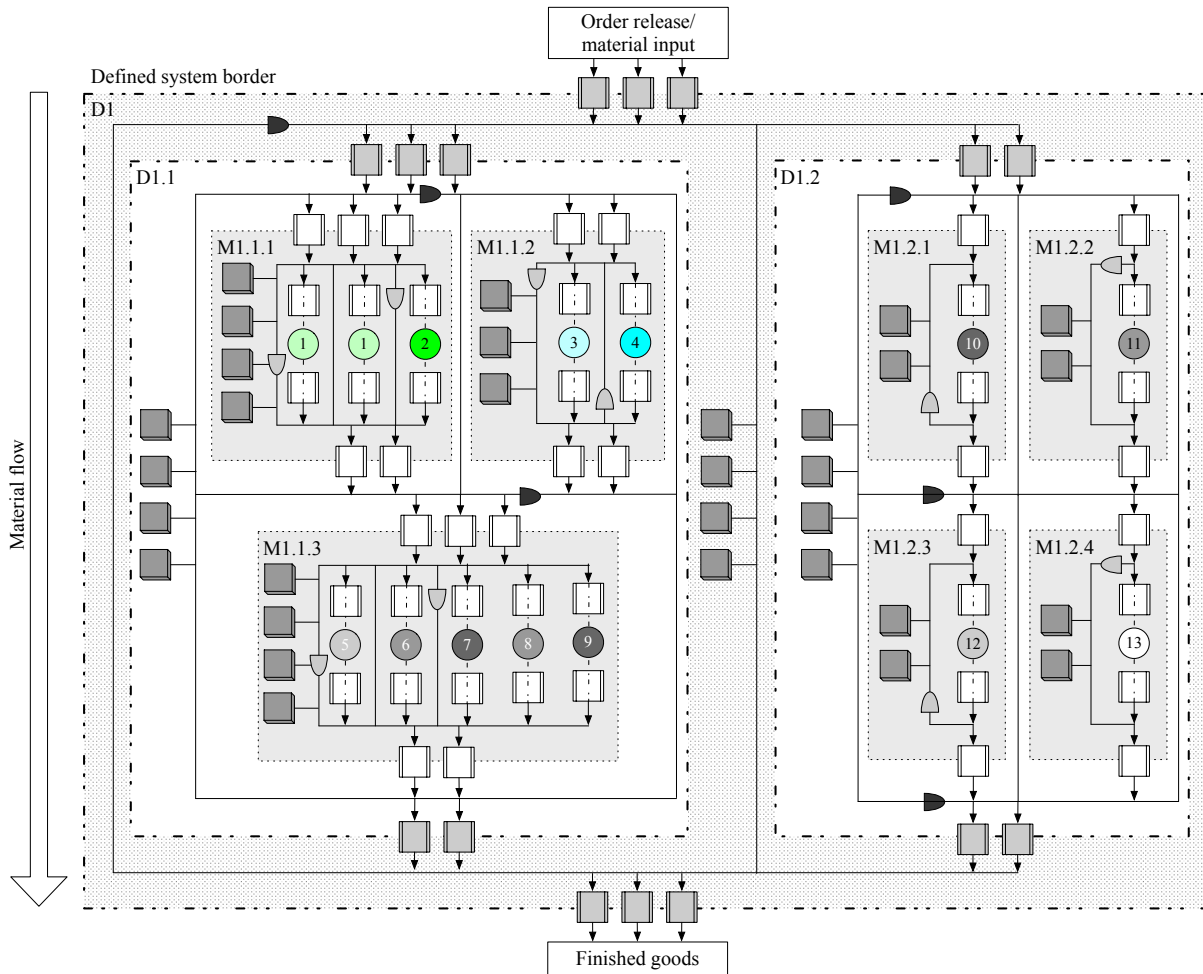


Figure 11.1 Extended simulation framework

Second, a vital field for future research is the layout and process optimization. Although less explored in this thesis, using the existing simulation tool and overlaying control framework for targeted layout optimization can enhance process efficiency and routing. Integrating layout optimization strategies, particularly by analyzing production bottlenecks and logistics routes, offers substantial potential to leverage both the efficiency and effectiveness of simulated production processes. This approach emphasizes the importance of a comprehensive optimization strategy, aiming at maximizing system performance through

focused but integrative optimization efforts. The layout optimization also encompasses comparisons and benchmarks between job-shop processes and modular system designs, aimed at validating their respective strengths and potential synergies.

Third, it's important to recognize that the control framework's application extends beyond just modular production. Its flexible design allows for use in various contexts, not only in production settings but also in solving computational challenges like real-time scheduling and vehicle routing. Its adaptability to other intra-logistics problems, such as warehousing, is a key feature. The framework's ability to control systems defined by discrete events broadens its potential uses beyond its initial production focus. With suitable modifications, the framework can be adapted for various scientific and technical fields, underscoring its interdisciplinary value.

- **Organizational perspective - interdisciplinary approach:**

Integrating additional planning activities into the order release and ongoing planning process presents a notable opportunity to connect production planning and control more effectively. This area, particularly the balance between these elements, has been largely overlooked in existing deep learning based production research. Figure 10.4 suggests an equilibrium that merits further investigation, potentially leading to a dynamic model that encompasses these considerations. Additionally, the shift from centralized planning to decentralized control, both within and between companies, has not been extensively studied, as indicated by Mayer et al. (2016). This transition is crucial for achieving a balance between long-term global optimization and short-term local responsiveness, a concept yet to be fully explored in current research.

The concept of a fluid organization and system boundaries might be beneficial in the context of the operational equilibrium. This idea is represented in the matrix production system shown in Figure 11.2. The system is modular, aligning with the *Divide and Conquer* principle by ElMaraghy et al. (2009), which advocates for dividing a whole system into multiple manageable units. This modularization also draws inspiration from modern warehouse intralogistics, such as those in Amazon Robotics fulfillment centers. In this setup, agents (or robots) can adjust their actions based on their location and move between different virtual areas and distribution modules within the organization, as illustrated by the colored modules in Figure 11.2.

Future research should investigate the feasibility and efficiency of robots moving in and out of these modules in a semi-heterarchical control layout. In this design, robots would have the autonomy to switch policies, potentially eliminating stops at input-output buffers and enabling continuous transport. However, this approach might lead to occasionally empty modules and reduced process efficiency. To address this, the reward function for

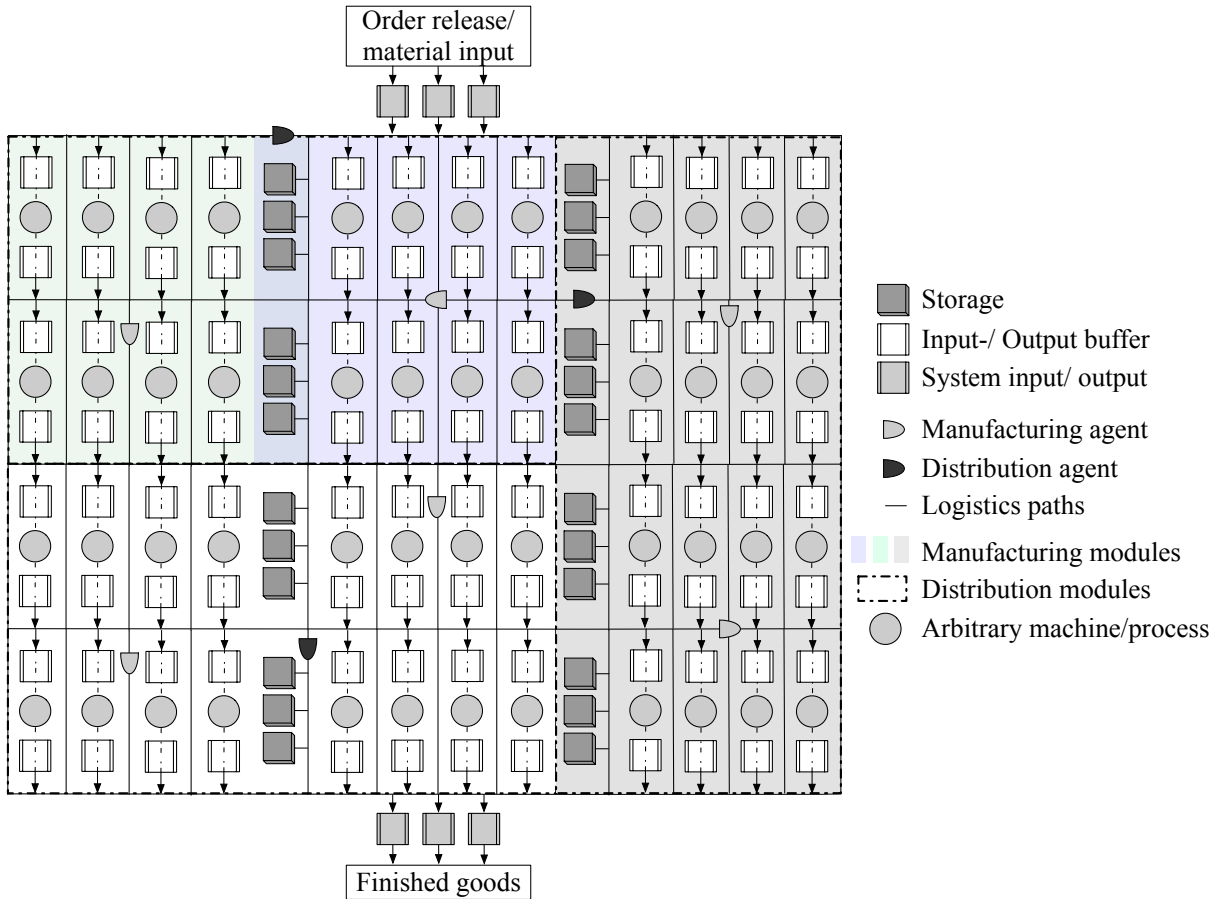


Figure 11.2 Projected semi-heterarchical organization with virtual structure

the system should include considerations for module occupancy, penalizing situations where a module remains empty. This approach underscores the complex interplay between planning and control. Medium-term planning, informed by upcoming order volumes, could strategically influence module occupation and utilization. This expands the scope of planning from just managing transitions and order releases to also include medium-term capacity control and reward parameter adjustments.

• **Algorithmic perspective:**

Expanding on the prior mentioned operational equilibrium, incorporating planning capabilities into the top distribution modules is a significant advancement not previously addressed in this thesis. This leads to the development of a deep learning algorithm that acts as a master algorithm or as a prior predictive planning tool. This algorithm would adjust its overarching strategy according to the overall state of production, thereby modifying real-time operations in anticipation of future needs. This also supports the matrix fusion concept, invented by Siegert et al. (2018). This approach could improve the balance or fusion of planning and control, especially within the lower modules. For instance, during periods with high levels of work-in-progress, the planning strategy could shift

to prioritize overall throughput and throughput time in the deep reinforcement learning reward function, rather than focusing on specific order prioritization. This strategic shift aims to achieve mid- to long-term global objectives, addressing the *dilemma of process planning* as described by Gutenberg (1971) and Nyhuis and Wiendahl (2012). It goes beyond the tactical scope of hyper-heuristics, introducing a wider strategic perspective.

Additionally, the development of intermediate-level algorithms, like those based on a factor model, offers further potential for strategic policy generation. A relevant example is the genetic algorithm for creating suitable neural networks, as explored by Lang et al. (2020). It's important to note that the optimal use of such algorithms would be during the configuration and training phase, rather than in real-time operations, due to the high operational costs of real-time creation and optimization. Keeping computational demands and calculation times manageable is crucial for enabling feasible decentralized and real-time decision-making.

From a broader perspective, this thesis highlights the further need for effective transfer of scientific research into practical applications, particularly in the realm of hybrid model factories. Future initiatives should therefore concentrate on enhancing this transfer and implementation process. A gradual rollout strategy is recommended, prioritizing continuous access to essential information for detailed evaluations.

12 Conclusion

In the current production landscape, companies are confronted with continuously increasing requirements such as shorter product life cycles, increasingly complex technical processes, and fluctuating procurement and sales markets. For this reason, innovative new control approaches are needed to remain competitive. The objective is to reduce manual intervention and minimize the need for expert knowledge through data-driven manufacturing processes that allow continuous process monitoring, learning, and optimization. Within this context, the integration of *Industry 4.0* concepts and other advanced industrial technologies gained prominence in recent years.

In this thesis, a production control framework based on deep learning is developed, following the *DSRM* approach by Peffers et al. (2007) and the iterative design process that incorporates the relevance and rigor cycles as proposed by Hevner (2007). The objective is not only to meet specific use-case requirements, but to structure and systematically reduce the inherent control complexity. A two-staged approach is adopted to ensure a comprehensive analysis. First, this thesis contributes to a deeper penetration and structuring of deep learning based production research, focusing on production planning and control. Second, it presents an adaptive and self-configuring control framework based on deep reinforcement learning. This framework serves as a tool to implement a widely applicable system for various modular production scenarios.

As part of the first fundamental research phase, specific studies are conducted to analyze the prevailing research field of deep learning based production systems. Thereby, deep reinforcement learning is identified as particularly suitable for the dynamic and continuously evolving domain of production control. This technique is characterized by its interactive learning capabilities and, in contrast to other deep learning methods, requires comparatively few computational resources in operational mode while being significantly faster in calculation. In addition to the superior performances achieved so far, this emphasizes its suitability for its deployment in real-time systems. Subsequently, employing a novel organizational algorithmic taxonomy, existing approaches are investigated regarding their combined use in terms of agent composition and orchestration. In these studies, it is further emphasized that modular production systems based on deep learning and advanced organizational methodologies, such as multi-layered or multi-agent based production systems, garnered minimal attention in prior research, both in individual and combined analyses.

In this thesis, standardized control and processing modules are exploited to ensure scalability and to limit the optimization space. These modules, configurable and modifiable with integrated resources, facilitate direct systemic changes and procedural adjustments. Their design reduces the need for large state spaces, thus decreasing structural and dynamic process complexities. The modules are organized in semi-heterarchical manufacturing and distribution layers, that can be designed for different performance indicators and trained with distinct reward functions,

supported by a standardized neural network stack for knowledge transfer and automated network generation. The holistic approach integrates structural, organizational, and algorithmic elements, postulating that a comprehensive reduction in production control complexity necessitates the simultaneous consideration of all three aspects. This thesis leverages a deep reinforcement learning algorithm by using a hyper-heuristic that differentiates between optimization and operation activities. This hybrid and decentralized approach merges a deep learning based top-level heuristic with low-level dispatching rules. The deep reinforcement learning component leverages control optimization, learning, adaptability, and scenario-specific process optimization. In parallel, conventional rules provide consistent process adherence and reliability.

The conducted simulated and real evaluations confirm the performance of the control concept for multi-objective optimization. The framework exhibit a structural and procedural adaptability that enables it to react flexibly to order fluctuations and to handle varying order quantities. It also reveals a high degree of robustness in optimization, especially in the face of varying work-in-progress levels and order quantities. The framework also enables balanced optimization by dynamically evaluating optimization parameters. Prioritized and urgent jobs thus exhibit significantly shorter throughput times and lower tardiness than standard orders, which contributes to improved response times. This leads to a scenario-specific focus on pre-defined objectives, which can also be economic in nature. The application of conventional rules also helps to elevate the explainability of the chosen actions, mitigating the typical characterization of deep learning methods as black-box approaches.

Finally, this thesis highlights the capabilities of deep learning in optimizing performance indicators in modular production systems while ensuring system adaptability and robustness. However, it is observed that such applications are notably sparse in the broader field of deep learning based production. The provided framework aims to be a foundational tool for researchers and industry professionals, assisting them in exploring the potential of deep learning in production control and designing their own application scenarios. Given the industrial advancements and the *Industry 4.0* capabilities, combined with the evident gaps in real-world implementations, there's a growing imperative for both research and industry to explore the full scope and applicability of deep learning in complex and large-scale production environments and to transfer novel approaches into operational practice.

Bibliography

- Adadi, A. (2021). A survey on data-efficient algorithms in big data era. *Journal of Big Data*, 8(1):24. doi: 10.1186/s40537-021-00419-9.
- Alexander, S. (1987). An expert system for the selection of scheduling rules in a job shop. *Computers & Industrial Engineering*, 12(3):167–171. doi: 10.1016/0360-8352(87)90010-6.
- Alshurideh, M., Gasaymeh, A., Ahmed, G., Alzoubi, H., and Kurd, B. A. (2020). Loyalty program effectiveness: Theoretical reviews and practical proofs. *Uncertain Supply Chain Management*, pages 599–612. doi: 10.5267/j.uscm.2020.2.003.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., and Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1):53. doi: 10.1186/s40537-021-00444-8.
- Amer, M. and Maul, T. (2019). A review of modularization techniques in artificial neural networks. *Artificial Intelligence Review*, 52(1):527–561. doi: 10.1007/s10462-019-09706-7.
- Antons, O. and Arlinghaus, J. C. (2022). Distributing decision-making authority in manufacturing – review and roadmap for the factory of the future. *International Journal of Production Research*, 60(13):4342–4360. doi: 10.1080/00207543.2022.2057255.
- Arinez, J. F., Chang, Q., Gao, R. X., Xu, C., and Zhang, J. (2020). Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook. *Journal of Manufacturing Science and Engineering*, 142(11):110804. doi: 10.1115/1.4047855.
- Armstrong, R., Hall, B. J., Doyle, J., and Waters, E. (2011). 'Scoping the scope' of a cochrane review. *Journal of Public Health*, 33(1):147–150. doi: 10.1093/pubmed/fdr015.
- Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). A Brief Survey of Deep Reinforcement Learning. *IEEE Signal Processing Magazine*, 34(6):26–38. doi: 10.1109/MSP.2017.2743240.
- Arwa, E. O. and Folly, K. A. (2020). Reinforcement Learning Techniques for Optimal Power Control in Grid-Connected Microgrids: A Comprehensive Review. *IEEE Access*, 8:208992–209007. doi: 10.1109/ACCESS.2020.3038735.
- Baer, S., Bakakeu, J., Meyes, R., and Meisen, T. (2019). Multi-Agent Reinforcement Learning for Job Shop Scheduling in Flexible Manufacturing Systems. In *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*, Laguna Hills, CA, USA. ISBN: 978-1-72814-087-2.
- Baker, A. D. (1998). A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull. *Journal of Manufacturing Systems*,

- 17(4):297–320. doi: 10.1016/S0278-6125(98)80077-0.
- Balaji, P. G. and Srinivasan, D. (2010). An Introduction to Multi-Agent Systems. In Kacprzyk, J., Srinivasan, D., and Jain, L. C., editors, *Innovations in Multi-Agent Systems and Applications - I*, volume 310, pages 1–27. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-642-14434-9, 978-3-642-14435-6.
- Bansal, R. C. (2005). Optimization Methods for Electric Power Systems: An Overview. *International Journal of Emerging Electric Power Systems*, 2(1). doi: 10.2202/1553-779X.1021.
- Barbati, M., Bruno, G., and Genovese, A. (2012). Applications of agent-based models for optimization problems: A literature review. *Expert Systems with Applications*, 39(5):6020–6028. doi: 10.1016/j.eswa.2011.12.015.
- Barbosa, J., Leitão, P., Adam, E., and Trentesaux, D. (2015). Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Computers in Industry*, 66:99–111. doi: 10.1016/j.compind.2014.10.011.
- Bellman, R. (1957). *Dynamic programming*, volume 1. Princeton University Press, Princeton, NJ, USA. ISBN: 978-0-691-07951-6.
- Bendul, J. C. and Blunck, H. (2019). The design space of production planning and control for industry 4.0. *Computers in Industry*, 105:260–272. doi: 10.1016/j.compind.2018.10.010.
- Bergmann, S., Stelzer, S., and Strassburger, S. (2014). On the use of artificial neural networks in simulation-based manufacturing control. *Journal of Simulation*, 8(1):76–90. doi: 10.1057/jos.2013.6.
- Bertolini, M., Mezzogori, D., Neroni, M., and Zammori, F. (2021). Machine Learning for industrial applications: A comprehensive literature review. *Expert Systems with Applications*, 175:114820. doi: 10.1016/j.eswa.2021.114820.
- Bertrand, J. W. M., Wortmann, J. C., and Wijngaard, J. (1990). *Production control: a structural and design oriented approach*. Number 11 in Manufacturing research and technology. Elsevier, Amsterdam ; New York. ISBN: 978-0-444-88122-9.
- Bitsch, G. and Senjic, P. (2022). Short-Time Adaption and Reconfiguration of Cyber-Physical Production Systems. *Procedia CIRP*, 112:209–213. doi: 10.1016/j.procir.2022.09.074.
- Blackstone, J. H., Phillips, D. T., and Hogg, G. L. (1982). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research*, 20(1):27–45. doi: 10.1080/00207548208947745.
- Bondi, A. B. (2000). Characteristics of scalability and their impact on performance. In *Proceedings of the second international workshop on Software and performance - WOSP '00*, pages 195–203, Ottawa, Ontario, Canada. ACM Press. ISBN: 978-1-58113-195-6.

- Bongaerts, L., Monostori, L., McFarlane, D., and Kádár, B. (2000). Hierarchy in distributed shop floor control. *Computers in Industry*, 43(2):123–137. doi: 10.1016/S0166-3615(00)00062-2.
- Borangiu, T., Gilbert, P., Ivanescu, N.-A., and Rosu, A. (2009). An implementing framework for holonic manufacturing control with multiple robot-vision stations. *Engineering Applications of Artificial Intelligence*, 22(4-5):505–521. doi: 10.1016/j.engappai.2009.03.001.
- Borangiu, T., Răileanu, S., Trentesaux, D., and Berger, T. (2010). Semi-heterarchical agile control architecture with intelligent product-driven scheduling. *IFAC Proceedings Volumes*, 43(4):108–113. doi: 10.3182/20100701-2-PT-4011.00020.
- Borangiu, T., Răileanu, S., Berger, T., and Trentesaux, D. (2015). Switching mode control strategy in manufacturing execution systems. *International Journal of Production Research*, 53(7):1950–1963. doi: 10.1080/00207543.2014.935825.
- Bordoloi, S. K., Cooper, W. W., and Matsuo, H. (2009). FLEXIBILITY, ADAPTABILITY, AND EFFICIENCY IN MANUFACTURING SYSTEMS. *Production and Operations Management*, 8(2):133–150. doi: 10.1111/j.1937-5956.1999.tb00366.x.
- Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization Methods for Large-Scale Machine Learning. *SIAM Review*, 60(2):223–311. doi: 10.1137/16M1080173.
- Boyer, K. K. (2000). Flexibility in Manufacturing. In Swamidass, P. M., editor, *Encyclopedia of Production and Manufacturing Management*, pages 209–213. Springer US. ISBN: 978-0-7923-8630-8.
- Browne, J., O’Kelly, M., and Davies, B. (1982). Scheduling in a batch or job shop production environment. *Engineering Management International*, 1(3):173–184. doi: 10.1016/0167-5419(82)90016-3.
- Brucker, P. (2007). *Scheduling algorithms*. Springer, Berlin ; New York, 5th ed edition. ISBN: 978-3-540-69515-8.
- Brunoe, T. D., Soerensen, D. G., and Nielsen, K. (2021). Modular Design Method for Reconfigurable Manufacturing Systems. *Procedia CIRP*, 104:1275–1279. doi: 10.1016/j.procir.2021.11.214.
- Buckhorst, A. F., Grahn, L., and Schmitt, R. H. (2022). Decentralized Holonic Control System Model for Line-less Mobile Assembly Systems. *Robotics and Computer-Integrated Manufacturing*, 75:102301. doi: 10.1016/j.rcim.2021.102301.
- Bueno, A., Godinho Filho, M., and Frank, A. G. (2020). Smart production planning and control in the Industry 4.0 context: A systematic literature review. *Computers & Industrial Engineering*, 149:106774. doi: 10.1016/j.cie.2020.106774.
- Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., and Qu, R. (2013).

- Hyper-heuristics: a survey of the state of the art. *Journal of the Operational Research Society*, 64(12):1695–1724. doi: 10.1057/jors.2013.71.
- Burke, E. K., Hyde, M. R., Kendall, G., Ochoa, G., Özcan, E., and Woodward, J. R. (2010). A Classification of Hyper-Heuristic Approaches. In Gendreau, M. and Potvin, J.-Y., editors, *Handbook of Metaheuristics*, 146, pages 453–477. Springer International Publishing, Cham. ISBN: 978-3-319-91085-7, 978-3-319-91086-4.
- Burke, E. K., Hyde, M. R., Kendall, G., Ochoa, G., Özcan, E., and Woodward, J. R. (2019). A Classification of Hyper-Heuristic Approaches: Revisited. In Gendreau, M. and Potvin, J.-Y., editors, *Handbook of Metaheuristics*, 272, pages 453–477. Springer International Publishing, Cham. ISBN: 978-3-319-91085-7, 978-3-319-91086-4.
- Cadavid, J. P. U., Lamouri, S., Grabot, B., and Fortin, A. (2019). Machine Learning in Production Planning and Control: A Review of Empirical Literature. *IFAC-PapersOnLine*, 52(13):385–390. doi: 10.1016/j.ifacol.2019.11.155.
- Caesar, B., Grigoleit, F., and Unverdorben, S. (2019). (Self-)adaptiveness for manufacturing systems: challenges and approaches. *SICS Software-Intensive Cyber-Physical Systems*, 34(4):191–200. doi: 10.1007/s00450-019-00423-8.
- Chang, J., Yu, D., Zhou, Z., He, W., and Zhang, L. (2022). Hierarchical Reinforcement Learning for Multi-Objective Real-Time Flexible Scheduling in a Smart Shop Floor. *Machines*, 10(12):1195. doi: 10.3390/machines10121195.
- Chapman, S. N. (2006). *The fundamentals of production planning and control*. Pearson/Prentice Hall, Upper Saddle River, NJ. ISBN: 978-0-13-017615-8.
- Chen, B. and Matis, T. I. (2013). A flexible dispatching rule for minimizing tardiness in job shop scheduling. *International Journal of Production Economics*, 141(1):360–365. doi: 10.1016/j.ijpe.2012.08.019.
- Choi, B. K. and Kang, D. (2013). *Modeling and simulation of discrete-event systems*. John Wiley & Sons Inc, Hoboken, New Jersey. ISBN: 978-1-118-73276-2, 978-1-118-73285-4, 978-1-118-73287-8.
- Chollet, F. (2015). *Keras*. Keras Documentation, last accessed on 08.01.2024.
- Christensen, J. (1994). Holonic Manufacturing Systems: Initial Architecture and Standards Directions. In *Proceedings of the 1. Euro Workshop on Holonic Manufacturing Systems*.
- Coello, C. A. C., Lamont, G., and van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation Series. Springer US, Boston, MA. ISBN: 978-0-387-33254-3.
- Cooper, H. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowl-*

- edge in Society*, 1:104–126.
- Cowling, P., Kendall, G., and Soubeiga, E. (2001). A Hyperheuristic Approach to Scheduling a Sales Summit. In Goos, G., Hartmanis, J., van Leeuwen, J., Burke, E., and Erben, W., editors, *Practice and Theory of Automated Timetabling III*, volume 2079, pages 176–190. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-540-42421-5, 978-3-540-44629-3.
- Csáji, B. C., Monostori, L., and Kádár, B. (2006). Reinforcement learning in a distributed market-based production control system. *Advanced Engineering Informatics*, 20(3):279 – 288. doi: 10.1016/j.aei.2006.01.001.
- Dantas, A., Rego, A. F. D., and Pozo, A. (2021). Using deep Q-network for selection hyper-heuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1488–1492, Lille France. ACM. ISBN: 978-1-4503-8351-6.
- Deshmukh, A. V., Talavage, J. J., and Barash, M. M. (1998). Complexity in manufacturing systems, Part 1: Analysis of static complexity. *IIE Transactions*, 30(7):645–655. doi: 10.1080/07408179808966508.
- Dey, A. (2016). Machine Learning Algorithms: A Review. *International Journal of Computer Science and Information Technologies*, 7:1174–1179. doi: 10.21275/ART20203995.
- Dittrich, M.-A. and Fohlmeister, S. (2020). Cooperative multi-agent system for production control using reinforcement learning. *CIRP Annals*, 69(1):389 – 392. doi: 10.1016/j.cirp.2020.04.005.
- Dokeroglu, T., Kucukyilmaz, T., and Talbi, E.-G. (2024). Hyper-heuristics: A survey and taxonomy. *Computers & Industrial Engineering*, 187:109815. doi: 10.1016/j.cie.2023.109815.
- Doya, K. (2000). Reinforcement Learning in Continuous Time and Space. *Neural Computation*, 12(1):219–245. doi: 10.1162/089976600300015961.
- Drake, J. H., Kheiri, A., Özcan, E., and Burke, E. K. (2020). Recent advances in selection hyper-heuristics. *European Journal of Operational Research*, 285(2):405–428. doi: 10.1016/j.ejor.2019.07.073.
- Duffie, N. A. (1982). An Approach to the Design of Distributed Machinery Control Systems. *IEEE Transactions on Industry Applications*, IA-18(4):435–442. doi: 10.1109/TIA.1982.4504105.
- Duffie, N. A. and Piper, R. S. (1986). Nonhierarchical control of manufacturing systems. *Journal of Manufacturing Systems*, 5(2):141. doi: 10.1016/0278-6125(86)90036-1.
- Durasević, M. and Jakobović, D. (2019). Creating dispatching rules by simple ensemble combination. *Journal of Heuristics*, 25(6):959–1013. doi: 10.1007/s10732-019-09416-x.
- Durão, L. F., Guimarães, M. O., Salerno, M. S., and Zancul, E. (2019). Uncertainty Management in Advanced Manufacturing Implementation: The Case for Learning Factories. *Procedia*

- Manufacturing*, 31:213–218. doi: 10.1016/j.promfg.2019.03.034.
- ElMaraghy, H., AlGeddawy, T., Azab, A., and ElMaraghy, W. (2012a). Change in Manufacturing – Research and Industrial Challenges. In ElMaraghy, H. A., editor, *Enabling Manufacturing Competitiveness and Economic Sustainability*, pages 2–9. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-642-23859-8, 978-3-642-23860-4.
- ElMaraghy, H., Azab, A., Schuh, G., and Pulz, C. (2009). Managing variations in products, processes and manufacturing systems. *CIRP Annals*, 58(1):441–446. doi: 10.1016/j.cirp.2009.04.001.
- ElMaraghy, H., Monostori, L., Schuh, G., and ElMaraghy, W. (2021). Evolution and future of manufacturing systems. *CIRP Annals*, 70(2):635–658. doi: 10.1016/j.cirp.2021.05.008.
- ElMaraghy, H. A. and Wiendahl, H.-P. (2019). Changeable Manufacturing. In Chatti, S., Laperrière, L., Reinhart, G., and Tolio, T., editors, *CIRP Encyclopedia of Production Engineering*, pages 219–226. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-662-53119-8, 978-3-662-53120-4.
- ElMaraghy, W., ElMaraghy, H., Tomiyama, T., and Monostori, L. (2012b). Complexity in engineering design and manufacturing. *CIRP Annals*, 61(2):793–814. doi: 10.1016/j.cirp.2012.05.001.
- Epitropakis, M. and Burke, E. (2018). Hyper-heuristics. In *Handbook of Heuristics*, pages 489–545. doi: 10.1007/978-3-319-07124-4_32.
- Erixon, G., Von Yxkull, A., and Arnström, A. (1996). Modularity – the Basis for Product and Factory Reengineering. *CIRP Annals*, 45(1):1–6. doi: 10.1016/S0007-8506(07)63005-4.
- Esteso, A., Peidro, D., Mula, J., and Díaz-Madroñero, M. (2022). Reinforcement learning applied to production planning and control. *International Journal of Production Research*, 61(16):1–18. doi: 10.1080/00207543.2022.2104180.
- Fan, J., Han, F., and Liu, H. (2014). Challenges of Big Data analysis. *National Science Review*, 1(2):293–314. doi: 10.1093/nsr/nwt032.
- Fischer, K., Schillo, M., and Siekmann, J. (2003). Holonic Multiagent Systems: A Foundation for the Organisation of Multiagent Systems. In Goos, G., Hartmanis, J., Van Leeuwen, J., Mařík, V., McFarlane, D., and Valckenaers, P., editors, *Holonic and Multi-Agent Systems for Manufacturing*, volume 2744, pages 71–80. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-540-40751-5, 978-3-540-45185-3.
- Fowler, J. W., Mönch, L., and Ponsignon, T. (2015). Discrete-event simulation for semiconductor wafer fabrication facilities: a tutorial. *International Journal of Industrial Engineering*, 22(5). doi: 10.23055/IJNETAP.2015.22.5.2276.

- Fuchigami, H., Sarker, R., and Rangel, S. (2018). Near-Optimal Heuristics for Just-In-Time Jobs Maximization in Flow Shop Scheduling. *Algorithms*, 11(4):43. doi: 10.3390/a11040043.
- Gankin, D., Mayer, S., Zinn, J., Vogel-Heuser, B., and Endisch, C. (2021). Modular Production Control with Multi-Agent Deep Q-Learning. volume 21364735, pages 1–8, Vasteras, Sweden. IEEE. ISBN: 978-1-72812-989-1.
- Geiger, C. D., Uzsoy, R., and Aytuğ, H. (2006). Rapid Modeling and Discovery of Priority Dispatching Rules: An Autonomous Learning Approach. *Journal of Scheduling*, 9(1):7–34. doi: 10.1007/s10951-006-5591-8.
- Geurtsen, M., Didden, J. B., Adan, J., Atan, Z., and Adan, I. (2023). Production, maintenance and resource scheduling: A review. *European Journal of Operational Research*, 305(2):501–529. doi: 10.1016/j.ejor.2022.03.045.
- Ghaleb, M., Zolfagharinia, H., and Taghipour, S. (2020). Real-time production scheduling in the Industry-4.0 context: Addressing uncertainties in job arrivals and machine breakdowns. *Computers & Operations Research*, 123:105031. doi: 10.1016/j.cor.2020.105031.
- Ghobakhloo, M. (2020). Industry 4.0, digitization, and opportunities for sustainability. *Journal of Cleaner Production*, 252:119869. doi: 10.1016/j.jclepro.2019.119869.
- Glover, F. (1977). HEURISTICS FOR INTEGER PROGRAMMING USING SURROGATE CONSTRAINTS. *Decision Sciences*, 8(1):156–166. doi: 10.1111/j.1540-5915.1977.tb01074.x.
- Gonzalez, P. L., Framinan, J. M., and Ruiz-Usano, R. (2010). A multi-objective comparison of dispatching rules in a drum–buffer–rope production control system. *International Journal of Computer Integrated Manufacturing*, 23(2):155–167. doi: 10.1080/09511920903440362.
- Grabot, B. and Geneste, L. (1994). Dispatching rules in scheduling Dispatching rules in scheduling: a fuzzy approach. *International Journal of Production Research*, 32(4):903–915. doi: 10.1080/00207549408956978.
- Grassi, A., Guizzi, G., Santillo, L. C., and Vespoli, S. (2020). A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 24:43–46. doi: 10.1016/j.mfglet.2020.03.007.
- Greschke, P. (2016). *Matrix-Produktion als Konzept einer taktunabhängigen Fließfertigung*. Schriftenreihe des Instituts für Werkzeugmaschinen und Fertigungstechnik der Technischen Universität Braunschweig. Vulkan Verlag, Essen, 1 edition. ISBN: 978-3-8027-8344-9.
- Greschke, P., Schönemann, M., Thiede, S., and Herrmann, C. (2014). Matrix Structures for High Volumes and Flexibility in Production Systems. *Procedia CIRP*, 17:160–165. doi: 10.1016/j.procir.2014.02.040.

- Gronauer, S. and Diepold, K. (2021). Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55:895 – 943. doi: 10.1007/s10462-021-09996-w.
- Groover, M. P. and Jayaprakash, G. (2016). *Automation, production systems, and computer-integrated manufacturing*. Always learning. Pearson Prentice Hall, Upper Saddle River, NJ, 4. ed., global ed edition. ISBN: 978-1-292-07611-9.
- Gros, T. P., Gros, J., and Wolf, V. (2020). Real-Time Decision Making for a Car Manufacturing Process Using Deep Reinforcement Learning. In *2020 Winter Simulation Conference (WSC)*, volume 20512838, pages 3032–3044, Orlando, FL, USA. IEEE. ISBN: 978-1-72819-499-8.
- Gutenberg, E. (1971). *Grundlagen der Betriebswirtschaftslehre*, volume 1. Springer Berlin Heidelberg, Berlin, Heidelberg, 24 edition. ISBN: 978-3-662-21966-9, 978-3-662-21965-2.
- Han, B.-A. and Yang, J.-J. (2020). Research on Adaptive Job Shop Scheduling Problems Based on Dueling Double DQN. *IEEE Access*, 8:186474–186495. doi: 10.1109/ACCESS.2020.3029868.
- Hayes, R. H. and Wheelwright, S. C. (1984). *Restoring our competitive edge: competing through manufacturing*. Wiley, New York. ISBN: 978-0-471-05159-6.
- Heger, C. L. (2007). *Bewertung der Wandlungsfähigkeit von Fabrikobjekten*. Number 2007,1 in Berichte aus dem IFA. PZH, Produktionstechn. Zentrum, Garbsen. ISBN: 978-3-939026-43-3.
- Heger, J. and Voss, T. (2021). Dynamically adjusting the k -values of the ATCS rule in a flexible flow shop scenario with reinforcement learning. *International Journal of Production Research*, pages 1–15. doi: 10.1080/00207543.2021.1943762.
- Helkiö, P. and Tenhiälä, A. (2013). A contingency theoretical perspective to the product-process matrix. *International Journal of Operations & Production Management*, 33(2):216–244. doi: 10.1108/01443571311295644.
- Henn, G. and Kühnle, H. (1999). Strukturplanung. In Eversheim, W. and Schuh, G., editors, *Gestaltung von Produktionssystemen.*, pages 1–117. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-540-65453-7, 978-3-642-58399-5.
- Herlyn, W. (2012). *PPS im Automobilbau: Produktionsprogrammplanung und -steuerung von Fahrzeugen und Aggregaten*. Fahrzeugtechnik. Hanser, München. ISBN: 978-3-446-41370-2.
- Herrera, M., Pérez-Hernández, M., Kumar Parlikad, A., and Izquierdo, J. (2020). Multi-Agent Systems and Complex Networks: Review and Applications in Systems Engineering. *Processes*, 8(3):312. doi: 10.3390/pr8030312.
- Herrera Vidal, G. and Coronado Hernández, J. R. (2021). Complexity in manufacturing systems: a literature review. *Production Engineering*, 15(3-4):321–333. doi: 10.1007/s11740-020-01013-3.

- Herrmann, J.-P., Tackenberg, S., Padoano, E., and Gamber, T. (2021). A literature review and cluster analysis of the Aachen production planning and control model under Industry 4.0. *Procedia Computer Science*, 180:208–218. doi: 10.1016/j.procs.2021.01.158.
- Hershauer, J. C. and Ebert, R. J. (1975). Search and Simulation Selection of a Job-Shop Sequencing Rule. *Management Science*, 21(7):833–843. doi: 10.1287/mnsc.21.7.833.
- Hevner, March, Park, and Ram (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75. doi: 10.2307/25148625.
- Hevner, A. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19.
- Hofmann, C., Brakemeier, N., Krahe, C., Stricker, N., and Lanza, G. (2019). The Impact of Routing and Operation Flexibility on the Performance of Matrix Production Compared to a Production Line. In Schmitt, R. and Schuh, G., editors, *Advances in Production Research*, pages 155–165. Springer International Publishing, Cham. ISBN: 978-3-030-03450-4, 978-3-030-03451-1.
- Hofmann, C., Krahe, C., Stricker, N., and Lanza, G. (2020). Autonomous production control for matrix production based on deep Q-learning. *Procedia CIRP*, 88:25–30. doi: 10.1016/j.procir.2020.05.005.
- Hofmann, E. and Knébel, S. (2013). Alignment of manufacturing strategies to customer requirements using analytical hierarchy process. *Production & Manufacturing Research*, 1(1):19–43. doi: 10.1080/21693277.2013.846835.
- Holthaus, O. and Rajendran, C. (1997). Efficient dispatching rules for scheduling in a job shop. *International Journal of Production Economics*, 48(1):87–105. doi: 10.1016/S0925-5273(96)00068-0.
- Hornig, M.-F., Kung, H.-Y., Chen, C.-H., and Hwang, F.-J. (2020). Deep Learning Applications with Practical Measured Results in Electronics Industries. *Electronics*, 9(3):501. doi: 10.3390/electronics9030501.
- Hu, Y., Zhang, Z., Wang, J., Wang, Z., and Liu, H. (2021). Task Decomposition Based on Cloud Manufacturing Platform. *Symmetry*, 13(8):1311. doi: 10.3390/sym13081311.
- Huang, J., Chang, Q., and Arinez, J. (2020). Deep reinforcement learning based preventive maintenance policy for serial production lines. *Expert Systems with Applications*, 160:113701. doi: 10.1016/j.eswa.2020.113701.
- Huang, Z., Zhou, Y., Chen, Z., He, X., Lai, X., and Xia, X. (2021). Running Time Analysis of MOEA/D on Pseudo-Boolean Functions. *IEEE Transactions on Cybernetics*, 51(10):5130–5141. doi: 10.1109/TCYB.2019.2930979.

- Hunt, H., Pollock, A., Campbell, P., Estcourt, L., and Brunton, G. (2018). An introduction to overviews of reviews: planning a relevant research question and objective for an overview. *Systematic Reviews*, 7(1):39. doi: 10.1186/s13643-018-0695-8.
- Jacobs, F. R., Berry, W. L., Whybark, D. C., and Vollmann, T. E. (2018). *Manufacturing planning and control for supply chain management: the CPIM reference/ planning and control for supply chain management: \$b the CPIM reference*. McGraw-Hill Education, New York, second edition edition. ISBN: 978-1-260-10838-5.
- Jamwal, A., Agrawal, R., Sharma, M., and Giallanza, A. (2021). Industry 4.0 Technologies for Manufacturing Sustainability: A Systematic Review and Future Research Directions. *Applied Sciences*, 11(12):5725. doi: 10.3390/app11125725.
- Jimenez, J.-F., Bekrar, A., Zambrano-Rey, G., Trentesaux, D., and Leitão, P. (2017). Pollux: a dynamic hybrid control architecture for flexible job shop systems. *International Journal of Production Research*, 55(15):4229–4247. doi: 10.1080/00207543.2016.1218087.
- Jones, D., Mirrazavi, S., and Tamiz, M. (2002). Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1):1–9. doi: 10.1016/S0377-2217(01)00123-0.
- Kaban, A. K., Othman, Z., and Rohmah, D. S. (2012). Comparison of dispatching rules in job-shop scheduling problem using simulation: a case study. *International Journal of Simulation Modelling*, 11(3):129–140. doi: 10.2507/IJSIMM11(3)2.201.
- Kagermann, H., Wahlster, W., and Helbig, J. (2013). *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0 – Securing the Future of German Manufacturing Industry*. Acatech - National Academy of Science and Engineering.
- Kallestad, J., Hasibi, R., Hemmati, A., and Sörensen, K. (2023). A General Deep Reinforcement Learning Hyperheuristic Framework for Solving Combinatorial Optimization Problems. *European Journal of Operational Research*, 209(1):446–468. doi: 10.1016/j.ejor.2023.01.017.
- Kang, Z., Catal, C., and Tekinerdogan, B. (2020). Machine learning applications in production lines: A systematic literature review. *Computers & Industrial Engineering*, 149:106773. doi: 10.1016/j.cie.2020.106773.
- Karageorgos, A., Mehandjiev, N., Weichhart, G., and Hämmerle, A. (2003). Agent-based optimisation of logistics and production planning. *Engineering Applications of Artificial Intelligence*, 16(4):335–348. doi: 10.1016/S0952-1976(03)00076-9.
- Kellner, F., Lienland, B., and Lukesch, M. (2020). Produktionsfaktoren. In *Produktionswirtschaft*, pages 31–159. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-662-61445-7, 978-3-662-61446-4.

- Kim, J. J., Steinhoff, L., and Palmatier, R. W. (2021). An emerging theory of loyalty program dynamics. *Journal of the Academy of Marketing Science*, 49(1):71–95. doi: 10.1007/s11747-020-00719-1.
- Klemmt, A., Horn, S., Weigert, G., and Wolter, K.-J. (2009). Simulation-based optimization vs. mathematical programming: A hybrid approach for optimizing scheduling problems. *Robotics and Computer-Integrated Manufacturing*, 25(6):917–925. doi: 10.1016/j.rcim.2009.04.012.
- Koestler, A. (1970). Beyond Atomism and Holism—the Concept of the Holon. *Perspectives in Biology and Medicine*, 13(2):131–154. doi: 10.1353/pbm.1970.0023.
- Kolisch, R. and Hartmann, S. (1999). Discrete-Time Stable Generalized Self-Learning Optimal Control With Approximation Errors. *IEEE Transactions on Neural Networks and Learning Systems*, 29(4):1226–1238. doi: 10.1007/978-1-4615-5533-9 7.
- Koren, Y. (2010). *The global manufacturing revolution: product-process-business integration and reconfigurable systems*. Wiley series in systems engineering and management. Wiley, Hoboken, N.J. ISBN: 978-0-470-58377-7.
- Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel, H. (1999). Reconfigurable Manufacturing Systems. *CIRP Annals*, 48(2):527–540. doi: 10.1016/S0007-8506(07)63232-6.
- Kotzur, L., Nolting, L., Hoffmann, M., Groß, T., Smolenko, A., Priesmann, J., Büsing, H., Beer, R., Kullmann, F., Singh, B., Praktiknjo, A., Stolten, D., and Robinius, M. (2021). A modeler’s guide to handle complexity in energy systems optimization. *Advances in Applied Energy*, 4:100063. doi: 10.1016/j.adapen.2021.100063.
- Koç, E., Delibaş, M. B., and Anadol, Y. (2022). Environmental Uncertainties and Competitive Advantage: A Sequential Mediation Model of Supply Chain Integration and Supply Chain Agility. *Sustainability*, 14(14):8928. doi: 10.3390/su14148928.
- Kuhnle, A. (2020). *Adaptive order dispatching based on reinforcement learning: application in a complex job shop in the semiconductor industry*. Number Band 241 in Forschungsberichte aus dem wbk, Institut für Produktionstechnik, Karlsruher Institut für Technologie (KIT). Shaker Verlag GmbH, Düren. ISBN: 978-3-8440-7762-9.
- Kuhnle, A., Kaiser, J.-P., Theiß, F., Stricker, N., and Lanza, G. (2020). Designing an adaptive production control system using reinforcement learning. *Journal of Intelligent Manufacturing*, 32:855–876. doi: 10.1007/s10845-020-01612-y.
- Kuhnle, A., Röhrig, N., and Lanza, G. (2019). Autonomous order dispatching in the semiconductor industry using reinforcement learning. *Procedia CIRP*, 79:391–396. doi: 10.1016/j.procir.2019.02.101.

- Kumar, A. and Dimitrakopoulos, R. (2021). Production scheduling in industrial mining complexes with incoming new information using tree search and deep reinforcement learning. *Applied Soft Computing*, 110:107644. doi: 10.1016/j.asoc.2021.107644.
- Kurinov, I., Orzechowski, G., Hämäläinen, P., and Mikkola, A. (2020). Automated Excavator Based on Reinforcement Learning and Multibody System Dynamics. *IEEE Access*, 8:213998–214006. doi: 10.1109/ACCESS.2020.3040246.
- Kádár, B., Monostori, L., and Csáji, B. (2003). Adaptive approaches to increase the performance of production control systems. *Proceedings of 36th CIRP ISMS*, pages 305–312.
- Lang, S., Reggelin, T., Behrendt, F., and Nahhas, A. (2020). Evolving Neural Networks to Solve a Two-Stage Hybrid Flow Shop Scheduling Problem with Family Setup Times. doi: 10.24251/HICSS.2020.160.
- Langmann, R., editor (2010). *Taschenbuch der Automatisierung*. Fachbuchverl. Leipzig im Carl-Hanser-Verl, München, 2., neu bearb. Aufl. edition. ISBN: 978-3-446-42112-7.
- Le-Anh, T. and De Koster, M. B. M. (2005). On-line dispatching rules for vehicle-based internal transport systems. *International Journal of Production Research*, 43(8):1711–1728. doi: 10.1080/00207540412331320481.
- Lee, A. S. and Baskerville, R. L. (2003). Generalizing Generalizability in Information Systems Research. *Information Systems Research*, 14(3):221–243. doi: 10.1287/isre.14.3.221.16560.
- Lee, J.-H. and Kim, C.-O. (2008). Multi-agent systems applications in manufacturing systems and supply chain management: a review paper. *International Journal of Production Research*, 46(1):233–265. doi: 10.1080/00207540701441921.
- Lee, S., Cho, Y., and Lee, Y. H. (2020). Injection Mold Production Sustainable Scheduling Using Deep Reinforcement Learning. *Sustainability*, 12(20):8718. doi: 10.3390/su12208718.
- Leitão, P. and Restivo, F. (2006). ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*, 57(2):121–130. doi: 10.1016/j.compind.2005.05.005.
- Liaqait, R. A., Hamid, S., Warsi, S. S., and Khalid, A. (2021). A Critical Analysis of Job Shop Scheduling in Context of Industry 4.0. *Sustainability*, 13(14):7684. doi: 10.3390/su13147684.
- Lin, J. (2019). Backtracking search based hyper-heuristic for the flexible job-shop scheduling problem with fuzzy processing time. *Engineering Applications of Artificial Intelligence*, 77:186–196. doi: 10.1016/j.engappai.2018.10.008.
- Lin, J., Li, Y.-Y., and Song, H.-B. (2022). Semiconductor final testing scheduling using Q-learning based hyper-heuristic. *Expert Systems with Applications*, 187:115978. doi: 10.1016/j.eswa.2021.115978.

- Liu, H. and Dong, J. J. (1996). Dispatching rule selection using artificial neural networks for dynamic planning and scheduling. *Journal of Intelligent Manufacturing*, 7(3):243–250. doi: 10.1007/BF00118083.
- Luczak, H., Heiderich, T., and Kaiser, H. (1998). The Aachener PPC-model. An integrated model for production planning and control. Maui, Hawaii.
- Luo, M., Lin, J., and Xu, L. (2020). Solving Flexible Job-shop Problem with Sequence-dependent Setup Times by Using Selection Hyper-heuristics. In *Proceedings of the 2nd International Conference on Artificial Intelligence and Advanced Manufacture*, pages 428–433, Manchester United Kingdom. ACM. ISBN: 978-1-4503-7553-5.
- Luo, S. (2020). Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Applied Soft Computing*, 91:106208. doi: 10.1016/j.asoc.2020.106208.
- Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y., and Kim, D. I. (2019). Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. *IEEE Communications Surveys Tutorials*, 21(4):3133–3174. doi: 10.1109/COMST.2019.2916583.
- Lödding, H. (2016). *Verfahren der Fertigungssteuerung*. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-662-48458-6, 978-3-662-48459-3.
- Lödding, H., editor (2019). *PPS-Report 2019: Studienergebnisse*. TEWISS - Technik und Wissen GmbH, Garbsen. ISBN: 978-3-95900-402-2.
- Mahadevan, S. and Theocharous, G. (1998). Optimizing Production Manufacturing Using Reinforcement Learning. In *Proceedings of the Eleventh International FLAIRS Conference*, pages 372–377.
- Malhan, R. and Gupta, S. K. (2023). The Role of Deep Learning in Manufacturing Applications: Challenges and Opportunities. *Journal of Computing and Information Science in Engineering*, 23(6):060816. doi: 10.1115/1.4062939.
- Malus, A., Kozjek, D., and Vrabič, R. (2020). Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. *CIRP Annals*, 69(1):397 – 400. doi: 10.1016/j.cirp.2020.04.001.
- Marcucci, G., Antomarioni, S., Ciarapica, F. E., and Bevilacqua, M. (2022). The impact of Operations and IT-related Industry 4.0 key technologies on organizational resilience. *Production Planning & Control*, 33(15):1417–1431. doi: 10.1080/09537287.2021.1874702.
- Mason, S. J., Fowler, J. W., and Matthew Carlyle, W. (2002). A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *Journal of Scheduling*, 5(3):247–262. doi: 10.1002/jos.102.

- Maulana, A., Jiang, Z., Liu, J., Back, T., and Emmerich, M. T. M. (2015). Reducing complexity in many objective optimization using community detection. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 3140–3147, Sendai, Japan. IEEE. ISBN: 978-1-4799-7492-4.
- May, M. C., Kiefer, L., Kuhnle, A., Stricker, N., and Lanza, G. (2021). Decentralized Multi-Agent Production Control through Economic Model Bidding for Matrix Production Systems. *Procedia CIRP*, 96:3–8. doi: 10.1016/j.procir.2021.01.043.
- Mayer, J., Pielmeier, J., Berger, C., Engehausen, F., Hempel, T., and Hünnekes, P. (2016). *Aktuellen Herausforderungen der Produktionsplanung und -steuerung mittels Industrie 4.0 begegnen: Studienergebnisse*. TEWISS-Technik und Wissen GmbH, Garbsen. ISBN: 978-3-95900-104-5.
- Mayer, S., Classen, T., and Endisch, C. (2021). Modular production control using deep reinforcement learning: proximal policy optimization. *Journal of Intelligent Manufacturing*, 32(8):2335–2351. doi: 10.1007/s10845-021-01778-z.
- Mayer, S., Hohme, N., Gankin, D., and Endisch, C. (2019). Adaptive Production Control in a Modular Assembly System – Towards an Agent-based Approach. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, pages 45–52, Helsinki, Finland. IEEE. ISBN: 978-1-72812-927-3.
- McKay, K. N. and Wiers, V. C. S. (2003). Planning, scheduling and dispatching tasks in production control. *Cognition, Technology & Work*, 5(2):82–93. doi: 10.1007/s10111-002-0117-4.
- Mehta, S. V. (1999). Predictable scheduling of a single machine subject to breakdowns. *International Journal of Computer Integrated Manufacturing*, 12(1):15–38. doi: 10.1080/095119299130443.
- Meissner, H., Ilse, R., and Aurich, J. C. (2017). Analysis of Control Architectures in the Context of Industry 4.0. *Procedia CIRP*, 62:165–169. doi: 10.1016/j.procir.2016.06.113.
- Meudt, T., Malte, P., and Metternich, J. (2017). Die Automatisierungspyramide - Ein Literaturüberblick. <https://tuprints.ulb.tu-darmstadt.de/id/eprint/6298>, last accessed on 08.01.2024.
- Mishra, M., Nayak, J., Naik, B., and Abraham, A. (2020). Deep learning in electrical utility industry: A comprehensive review of a decade of research. *Engineering Applications of Artificial Intelligence*, 96:104000. doi: 10.1016/j.engappai.2020.104000.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. *arXiv*, 1312.5602. doi: 10.48550/arXiv.1312.5602.

- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533. doi: 10.1038/nature14236.
- Modrak, V. and Soltysova, Z. (2023). Influence of Manufacturing Process Modularity on Lead Time Performances and Complexity. *Applied Sciences*, 13(12):7196. doi: 10.3390/app13127196.
- Mohammed, M. Q., Chung, K. L., and Chyi, C. S. (2020). Review of Deep Reinforcement Learning-Based Object Grasping: Techniques, Open Challenges, and Recommendations. *IEEE Access*, 8:178450–178481. doi: 10.1109/ACCESS.2020.3027923.
- Monostori, L., Csáji, B., and Kádár, B. (2004). Adaptation and Learning in Distributed Production Control. *CIRP Annals*, 53(1):349–352. doi: 10.1016/S0007-8506(07)60714-8.
- Monostori, L., Váncza, J., and Kumara, S. (2006). Agent-Based Systems for Manufacturing. *CIRP Annals*, 55(2):697–720. doi: 10.1016/j.cirp.2006.10.004.
- Moon, J. and Jeong, J. (2021). Smart Manufacturing Scheduling System: DQN based on Cooperative Edge Computing. In *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pages 1–8, Seoul, Korea (South). IEEE. ISBN: 978-1-66542-318-2.
- Mosavi, A., Faghan, Y., Ghamisi, P., Duan, P., Ardabili, S. F., Salwana, E., and Band, S. S. (2020). Comprehensive Review of Deep Reinforcement Learning Methods and Applications in Economics. *Mathematics*, 8(10):1640. doi: 10.3390/math8101640.
- Mourtzis, D. (2020). Simulation in the design and operation of manufacturing systems: state of the art and new trends. *International Journal of Production Research*, 58(7):1927–1949. doi: 10.1080/00207543.2019.1636321.
- Mourtzis, D., Fotia, S., Boli, N., and Vlachou, E. (2019). Modelling and quantification of industry 4.0 manufacturing complexity based on information theory: a robotics case study. *International Journal of Production Research*, 57(22):6908–6921. doi: 10.1080/00207543.2019.1571686.
- Márkus, A., Kis Váncza, T., and Monostori, L. (1996). A Market Approach to Holonic Manufacturing. *CIRP Annals*, 45(1):433–436. doi: 10.1016/S0007-8506(07)63096-0.
- Mönch, L., Fowler, J., and Mason, S. J. (2013). *Production planning and control for semiconductor wafer fabrication facilities: modeling, analysis, and systems*. Number vol. 52 in Operations research/computer science interfaces series. Springer, New York. ISBN: 978-1-4614-4471-8, 978-1-4614-4472-5.

- Nachum, O., Gu, S., Lee, H., and Levine, S. (2018). Data-Efficient Hierarchical Reinforcement Learning. doi: 10.48550/ARXIV.1805.08296.
- Nasiri, M. M., Yazdanparast, R., and Jolai, F. (2017). A simulation optimisation approach for real-time scheduling in an open shop environment using a composite dispatching rule. *International Journal of Computer Integrated Manufacturing*, 30(12):1239–1252. doi: 10.1080/0951192X.2017.1307452.
- Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113. doi: 10.1103/PhysRevE.69.026113.
- Nyhuis, P. (2008). *Produktionskennlinien — Grundlagen und Anwendungsmöglichkeiten*. Beiträge zu einer Theorie der Logistik. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-540-75641-5, 978-3-540-75642-2.
- Nyhuis, P., Rochow, N. E., Krause, M., Pischke, D., Seitz, M., and Kuprat, V. K. (2021). Organisationsformen der Produktion. *Journal of Production Systems and Logistics*, (1). doi: 10.15488/11332.
- Nyhuis, P. and Wiendahl, H.-P. (2012). *Logistische Kennlinien*. VDI-Buch. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-540-92838-6, 978-3-540-92839-3.
- Oh, Y., Zhou, C., and Behdad, S. (2018). Part decomposition and assembly-based (Re) design for additive manufacturing: A review. *Additive Manufacturing*, 22:230–242. doi: 10.1016/j.addma.2018.04.018.
- Oluyisola, O. E., Sgarbossa, F., and Strandhagen, J. O. (2020). Smart Production Planning and Control: Concept, Use-Cases and Sustainability Implications. *Sustainability*, 12(9):3791. doi: 10.3390/su12093791.
- Omar, Y. M., Minoufekar, M., and Plapper, P. (2019). Business analytics in manufacturing: Current trends, challenges and pathway to market leadership. *Operations Research Perspectives*, 6:100127. doi: 10.1016/j.orp.2019.100127.
- Orfi, N., Terpenney, J., and Sahin-Sariisik, A. (2011). Harnessing Product Complexity: Step 1—Establishing Product Complexity Dimensions and Indicators. *The Engineering Economist*, 56(1):59–79. doi: 10.1080/0013791X.2010.549935.
- Ouelhadj, D. and Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12(4):417–431. doi: 10.1007/s10951-008-0090-8.
- Pantke, F., Edelkamp, S., and Herzog, O. (2016). Symbolic discrete-time planning with continuous numeric action parameters for agent-controlled processes. *Mechatronics*, 34:38–62. doi: 10.1016/j.mechatronics.2015.06.015.
- Panzer, M. and Bender, B. (2022). Deep reinforcement learning in production systems: a

- systematic literature review. *International Journal of Production Research*, 60(13):4316–4341. doi: 10.1080/00207543.2021.1973138.
- Panzer, M., Bender, B., and Gronau, N. (2022). Neural agent-based production planning and control: An architectural review. *Journal of Manufacturing Systems*, 65:743–766. doi: 10.1016/j.jmsy.2022.10.019.
- Parunak, H. V. D., White, J. F., Lozo, P. W., Judd, R., Irish, B. W., and Kindrick, J. (1986). An Architecture for Heuristic Factory Control. In *1986 American Control Conference*, pages 548–558, Seattle, WA, USA. IEEE. doi: 10.23919/ACC.1986.4789001.
- Paul, M., Ramanan, T. R., and Sridharan, R. (2018). Simulation modelling and analysis of dispatching rules in an assembly job shop production system with machine breakdowns. *International Journal of Advanced Operations Management*, 10(3):234. doi: 10.1504/I-JAOM.2018.093739.
- Paul, S., Kurin, V., and Whiteson, S. (2019). Fast Efficient Hyperparameter Tuning for Policy Gradients. In *Advances in Neural Information Processing Systems*.
- Peffer, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3):45–77. doi: 10.2753/MIS0742-1222240302.
- Peres, R. S., Jia, X., Lee, J., Sun, K., Colombo, A. W., and Barata, J. (2020). Industrial Artificial Intelligence in Industry 4.0 - Systematic Review, Challenges and Outlook. *IEEE Access*, 8:220121–220139. doi: 10.1109/ACCESS.2020.3042874.
- Perwitz, J., Sobottka, T., Beicher, J.-N., and Gaal, A. (2022). Simulation-based evaluation of performance benefits from flexibility in assembly systems and matrix production. *Procedia CIRP*, 107:693–698. doi: 10.1016/j.procir.2022.05.047.
- Philipp, T., de Beer, C., Windt, K., and Scholz-Reiter, B. (2007). Evaluation of Autonomous Logistic Processes — Analysis of the Influence of Structural Complexity. In Hülsmann, M. and Windt, K., editors, *Understanding Autonomous Cooperation and Control in Logistics*, pages 303–324. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-540-47449-4, 978-3-540-47450-0.
- Pinedo, M. (2012). *Scheduling: theory, algorithms, and systems*. Springer, New York, 4th ed edition. ISBN: 978-1-4614-1986-0.
- Pinho, T., Coelho, J., Oliveira, P., Oliveira, B., Marques, A., Rasinmäki, J., Moreira, A., Veiga, G., and Boaventura-Cunha, J. (2021). Routing and schedule simulation of a biomass energy supply chain through SimPy simulation package. *Applied Computing and Informatics*, 17(1):36–52. doi: 10.1016/j.aci.2018.06.004.

- Pritschow, G. and Wiendahl, H.-P. (1995). Application of Control Theory for Production Logistics – Results of a Joint Project. *CIRP Annals*, 44(1):421–424. doi: 10.1016/S0007-8506(07)62355-5.
- Pérez-Lara, M., Saucedo-Martínez, J. A., Marmolejo-Saucedo, J. A., Salais-Fierro, T. E., and Vasant, P. (2020). Vertical and horizontal integration systems in Industry 4.0. *Wireless Networks*, 26(7):4767–4775. doi: 10.1007/s11276-018-1873-2.
- Qiao, D. and Wang, Y. (2021). A review of the application of discrete event simulation in manufacturing. *Journal of Physics: Conference Series*, 1802(2):022066. doi: 10.1088/1742-6596/1802/2/022066.
- Qin, W., Zhuang, Z., Huang, Z., and Huang, H. (2021). A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem. *Computers & Industrial Engineering*, 156:107252. doi: 10.1016/j.cie.2021.107252.
- Rauf, M., Guan, Z., Sarfraz, S., Mumtaz, J., Shehab, E., Jahanzaib, M., and Hanif, M. (2020). A smart algorithm for multi-criteria optimization of model sequencing problem in assembly lines. *Robotics and Computer-Integrated Manufacturing*, 61:101844. doi: 10.1016/j.rcim.2019.101844.
- Reddy, M. and Nagesh Kumar, D. (2020). Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review. *H2Open Journal*, 3(1):135–188. doi: 10.2166/h2oj.2020.128.
- Reichwald, R. and Dietel, B. (1991). Produktionswirtschaft. In Picot, A., editor, *Industrie-triebslehre*, pages 395–622. Gabler Verlag, Wiesbaden. ISBN: 978-3-322-87162-6, 978-3-322-87161-9.
- Rey, G. Z., Pach, C., Aissani, N., Bekrar, A., Berger, T., and Trentesaux, D. (2013). The control of myopic behavior in semi-heterarchical production systems: A holonic framework. *Engineering Applications of Artificial Intelligence*, 26(2):800–817. doi: 10.1016/j.engappai.2012.08.011.
- Rogers, G. G. and Bottaci, L. (1997). Modular production systems: A new manufacturing paradigm. *Journal of Intelligent Manufacturing*, 8(2):147–156. doi: 10.1023/A:1018560922013.
- Rummukainen, H. and Nurminen, J. K. (2019). Practical Reinforcement Learning - Experiences in Lot Scheduling Application. *IFAC-PapersOnLine*, 52(13):1415–1420. doi: 10.1016/j.ifacol.2019.11.397.
- Sabadka, D., Molnár, V., and Fedorko, G. (2019). Shortening of Life Cycle and Complexity Impact on the Automotive Industry. *TEM Journal*, 8(4):1295–1301. doi: 10.18421/TEM84-27.
- Sallez, Y., Berger, T., Raileanu, S., Chaabane, S., and Trentesaux, D. (2010). Semi-heterarchical

- control of FMS: From theory to application. *Engineering Applications of Artificial Intelligence*, 23(8):1314–1326. doi: 10.1016/j.engappai.2010.06.013.
- Samsonov, V., Ben Hicham, K., and Meisen, T. (2022). Reinforcement Learning in Manufacturing Control: Baselines, challenges and ways forward. *Engineering Applications of Artificial Intelligence*, 112:104868. doi: 10.1016/j.engappai.2022.104868.
- Sarang, N. and Poullis, C. (2023). Tractable large-scale deep reinforcement learning. *Computer Vision and Image Understanding*, 232:103689. doi: 10.1016/j.cviu.2023.103689.
- Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3):160. doi: 10.1007/s42979-021-00592-x.
- Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016). Prioritized Experience Replay. In *International Conference on Learning Representations*, San Juan, Puerto Rico.
- Scheer, A.-W. (1997). *Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse*. Springer, Berlin Heidelberg, 7., durchges. Aufl. edition. ISBN: 978-3-540-62967-2.
- Schenk, M., Wirth, S., and Müller, E. (2010). *Factory planning manual: situation-driven production facility planning*. Springer, Berlin ; London ; New York. ISBN: 978-3-642-03634-7, 978-3-642-03635-4.
- Scherer, E., editor (1998). *Shop floor control - a systems perspective: from deterministic models towards agile operations management ; with 6 tables*. Springer, Berlin Heidelberg. ISBN: 978-3-642-60313-6, 978-3-642-64349-1, 978-3-540-64002-8.
- Schmidt, M. and Nyhuis, P. (2021). *Produktionsplanung und -steuerung im Hannoveraner Lieferkettenmodell: innerbetrieblicher Abgleich logistischer Zielgrößen*. Springer Vieweg, Berlin [Heidelberg]. ISBN: 978-3-662-63896-5.
- Schmidtke, N., Rettmann, A., and Behrendt, F. (2021). Matrix Production Systems - Requirements and Influences on Logistics Planning for Decentralized Production Structures. doi: 10.24251/HICSS.2021.201.
- Schneeweiss, C. (2003). Distributed decision making—a unified approach. *European Journal of Operational Research*, 150(2):237–252. doi: 10.1016/S0377-2217(02)00501-5.
- Schneider and Kirkpatrick, S. (2006). *Stochastic Optimization*. Scientific Computation. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-540-34559-6.
- Scholz-Reiter, B., Windt, K., and Liu, H. (2011). Modelling dynamic bottlenecks in production networks. *International Journal of Computer Integrated Manufacturing*, 24(5):391–404. doi: 10.1080/0951192X.2010.511655.
- Schuh, G., Brandenburg, U., and Cuber, S. (2012). *Aufgaben. Produktionsplanung und -steuerung*. 1. Springer Vieweg, Berlin Heidelberg, 4., überarb. Aufl. edition. ISBN: 978-3-642-25422-2.

- Schuh, G. and Kampker, A. (2012). *Aachener PPS-Modell*. Produktionsplanung und -steuerung. 1. Springer Vieweg, Berlin Heidelberg, 4., überarb. aufl edition. ISBN: 978-3-642-25422-2.
- Schuh, G., Reuter, C., Prote, J.-P., Brambring, F., and Ays, J. (2017). Increasing data integrity for improving decision making in production planning and control. *CIRP Annals*, 66(1):425–428. doi: 10.1016/j.cirp.2017.04.003.
- Schuh, G. and Schmidt, C., editors (2014). *Produktionsmanagement: Handbuch Produktion und Management 5*. VDI-Buch. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-642-54287-9, 978-3-642-54288-6.
- Schuh, G., Schmitz, S., Maetschke, J., Janke, T., and Eisbein, H. (2023). Application of a Reinforcement Learning-based Automated Order Release in Production. doi: 10.15488/13500.
- Schwab, K. (2016). *The fourth industrial revolution*. Crown Business, New York, first u.s. edition edition. ISBN: 978-1-5247-5886-8.
- Schwartz, H. M. (2014). *Multi-agent machine learning: a reinforcement approach*. John Wiley & Sons, Hoboken, NJ. ISBN: 978-1-118-88448-5, 978-1-118-88447-8.
- Schönemann, M., Herrmann, C., Greschke, P., and Thiede, S. (2015). Simulation of matrix-structured manufacturing systems. *Journal of Manufacturing Systems*, 37:104–112. doi: 10.1016/j.jmsy.2015.09.002.
- Sculli, D. and Tsang, K. (1990). Priority dispatching rules in a fabrication/assembly shop. *Mathematical and Computer Modelling*, 13(3):73–79. doi: 10.1016/0895-7177(90)90372-T.
- Seliger, G. and Kruetzfeldt, D. (1999). Agent-Based Approach for Assembly Control. *CIRP Annals*, 48(1):21–24. doi: 10.1016/S0007-8506(07)63123-0.
- Selke, C. (2005). *Entwicklung von Methoden zur automatischen Simulationsmodellgenerierung*. Number Bd. 193 in Forschungsberichte IWB. Utz, München. ISBN: 978-3-8316-0495-1.
- Sethi, A. and Sethi, S. (1990). Flexibility in manufacturing: A survey. *International Journal of Flexible Manufacturing Systems*, 2(4). doi: 10.1007/BF00186471.
- Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D Nonlinear Phenomena*, 404. doi: 10.1016/j.physd.2019.132306.
- Siegert, J., Schlegel, T., and Bauernhansl, T. (2018). Matrix Fusion Factory. *Procedia Manufacturing*, 23:177–182. doi: 10.1016/j.promfg.2018.04.013.
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv*. arXiv: 1712.01815.

- Simpson, T. W., Siddique, Z., and Jiao, J. R., editors (2006). *Product Platform and Product Family Design*. Springer US, New York, NY. ISBN: 978-0-387-25721-1, 978-0-387-29197-0.
- Song, H.-B. and Lin, J. (2021). A genetic programming hyper-heuristic for the distributed assembly permutation flow-shop scheduling problem with sequence dependent setup times. *Swarm and Evolutionary Computation*, 60:100807. doi: 10.1016/j.swevo.2020.100807.
- Spencer, M. S. and Cox, J. F. (1995). An analysis of the product-process matrix and repetitive manufacturing. *International Journal of Production Research*, 33(5):1275–1294. doi: 10.1080/00207549508930209.
- Sutton, R. S. and Barto, A. G. (2017). *Reinforcement learning: an introduction*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts, 2nd edition. ISBN: 978-0-262-03924-6.
- Swiercz, A. (2017). Hyper-Heuristics and Metaheuristics for Selected Bio-Inspired Combinatorial Optimization Problems. *Heuristics and Hyper-Heuristics - Principles and Applications*, 1:3–20. doi: 10.5772/intechopen.69225.
- Tao, F., Qi, Q., Liu, A., and Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*, 48:157–169. doi: 10.1016/j.jmsy.2018.01.006.
- Thomé, A. M. T., Scavarda, L. F., and Scavarda, A. J. (2016). Conducting systematic literature review in operations management. *Production Planning & Control*, 27(5):408–420. doi: 10.1080/09537287.2015.1129464.
- Tinini, R. I., Santos, M. R. P. D., Figueiredo, G. B., and Batista, D. M. (2020). 5GPy: A SimPy-based simulator for performance evaluations in 5G hybrid Cloud-Fog RAN architectures. *Simulation Modelling Practice and Theory*, 101:102030. doi: 10.1016/j.simpat.2019.102030.
- Tranfield, D., Denyer, D., and Smart, P. (2003). Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. *British Journal of Management*, 14(3):207–222. doi: 10.1111/1467-8551.00375.
- Trentesaux, D. (2009). Distributed control of production systems. *Engineering Applications of Artificial Intelligence*, 22(7):971–978. doi: 10.1016/j.engappai.2009.05.001.
- Trierweiler, M., Foith-Förster, P., and Bauernhansl, T. (2020). Changeability of Matrix Assembly Systems. *Procedia CIRP*, 93:1127–1132. doi: 10.1016/j.procir.2020.04.029.
- Uhlemann, T. H.-J., Lehmann, C., and Steinhilper, R. (2017). The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0. *Procedia CIRP*, 61:335–340. doi: 10.1016/j.procir.2016.11.152.
- Uzsoy, R., Church, L. K., Ovacik, I. M., and Hinchman, J. (1993). Performance evaluation of dispatching rules for semiconductor testing operations. *Journal of Electronics Manufacturing*,

- 03(02):95–105. doi: 10.1142/S0960313193000115.
- Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., and Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry*, 37(3):255–274. doi: 10.1016/S0166-3615(98)00102-X.
- van Hasselt, H., Guez, A., and Silver, D. (2016). Deep Reinforcement Learning with Double Q-learning. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2094–2100. arXiv: 1509.06461.
- van Veen, F. (2017). *Neural Network Zoo Prequel: Cells and Layers*. Publication Title: THE ASIMOV INSTITUTE.
- VDI (2017). *Adaptability: description and measurement of the adaptability of manufacturing companies (medical device industry*, volume VDI 5201 Part 1. Beuth Verlag GmbH.
- Vieira, G. E., Herrmann, J. W., and Lin, E. (2003). Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. *Journal of Scheduling*, 6(1):39–62. doi: 10.1023/A:1022235519958.
- Voß, S. (2008). Metaheuristics. In Floudas, C. A. and Pardalos, P. M., editors, *Encyclopedia of Optimization*, pages 2061–2075. Springer US, Boston, MA. ISBN: 978-0-387-74758-3, 978-0-387-74759-0.
- Wang, G., Li, X., Gao, L., and Li, P. (2021a). Energy-efficient distributed heterogeneous welding flow shop scheduling problem using a modified MOEA/D. *Swarm and Evolutionary Computation*, 62:100858. doi: 10.1016/j.swevo.2021.100858.
- Wang, J., Sun, Y., Zhang, W., Thomas, I., Duan, S., and Shi, Y. (2016). Large-Scale Online Multitask Learning and Decision Making for Flexible Manufacturing. *IEEE Transactions on Industrial Informatics*, 12(6):2139–2147. doi: 10.1109/TII.2016.2549919.
- Wang, L., Hu, X., Wang, Y., Xu, S., Ma, S., Yang, K., Liu, Z., and Wang, W. (2021b). Dynamic Job-shop Scheduling in Smart Manufacturing using Deep Reinforcement Learning. *Computer Networks*, page 107969. doi: 10.1016/j.comnet.2021.107969.
- Waschneck, B., Bauernhansl, T., Altenmüller, T., and Kyek, A. (2017). Production Scheduling in Complex Job Shops from an Industrie 4.0 Perspective: A Review and Challenges in the Semiconductor Industry. doi: 10.5281/ZENODO.495155.
- Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., and Kyek, A. (2018). Optimization of global production scheduling with deep reinforcement learning. *Procedia CIRP*, 72:1264 – 1269. doi: 10.1016/j.procir.2018.03.212.
- Wassim, B. (2023). Hyper-heuristics applications to manufacturing scheduling: overview and opportunities. *IFAC-PapersOnLine*, 56(2):935–940. doi: 10.1016/j.ifacol.2023.10.1685.

- Weichert, D., Link, P., Stoll, A., Rüping, S., Ihlenfeldt, S., and Wrobel, S. (2019). A review of machine learning for the optimization of production processes. *The International Journal of Advanced Manufacturing Technology*, 104(5-8):1889–1902. doi: 10.1007/s00170-019-03988-5.
- Weiss, G., editor (2001). *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT Press, Cambridge, Mass., 3. print edition. ISBN: 978-0-262-73131-7, 978-0-262-23203-6.
- Weyer, M. (2002). *Das Produktionssteuerungskonzept Perlenkette und dessen Kennzahlensystem*. Helmesverlag, Karlsruhe. ISBN: 978-3-9808133-5-8.
- White, K. P. and Ingalls, R. G. (2018). The basics of simulation. In *2018 Winter Simulation Conference (WSC)*, pages 147–161, Gothenburg, Sweden. IEEE. ISBN: 978-1-5386-6572-5.
- Wiendahl, H.-P. (1997). *Fertigungsregelung: logistische Beherrschung von Fertigungsabläufen auf Basis des Trichtermodells*. Hanser, München Wien. ISBN: 978-3-446-19084-9.
- Wiendahl, H.-P., ElMaraghy, H., Nyhuis, P., Zäh, M., Wiendahl, H.-H., Duffie, N., and Brieke, M. (2007). Changeable Manufacturing - Classification, Design and Operation. *CIRP Annals*, 56(2):783–809. doi: 10.1016/j.cirp.2007.10.003.
- Wiendahl, H.-P., Reichardt, J., and Nyhuis, P. (2015). *Handbook Factory Planning and Design*. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-662-46390-1, 978-3-662-46391-8.
- Wiendahl, H.-P. and Scholtissek, P. (1994). Management and Control of Complexity in Manufacturing. *CIRP Annals*, 43(2):533–540. doi: 10.1016/S0007-8506(07)60499-5.
- Wolpert, D. and Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82. doi: 10.1109/4235.585893.
- Wu, Y., Frizelle, G., and Efstathiou, J. (2007). A study on the cost of operational complexity in customer–supplier systems. *International Journal of Production Economics*, 106(1):217–229. doi: 10.1016/j.ijpe.2006.06.004.
- Xie, J., Gao, L., Peng, K., Li, X., and Li, H. (2019). Review on flexible job shop scheduling. *IET Collaborative Intelligent Manufacturing*, 1(3):67–77. doi: 10.1049/iet-cim.2018.0009.
- Xu, L. D., Xu, E. L., and Li, L. (2018). Industry 4.0: state of the art and future trends. *International Journal of Production Research*, 56(8):2941–2962. doi: 10.1080/00207543.2018.1444806.
- Yonaga, K., Miyama, M., Ohzeki, M., Hirano, K., Kobayashi, H., and Kurokawa, T. (2022). Quantum Optimization with Lagrangian Decomposition for Multiple-process Scheduling in Steel Manufacturing. *ISIJ International*, 62(9):1874–1880. doi: 10.2355/isijinternational.ISIJINT-

2022-019.

- Zhang, H. and Roy, U. (2019). A semantics-based dispatching rule selection approach for job shop scheduling. *Journal of Intelligent Manufacturing*, 30(7):2759–2779. doi: 10.1007/s10845-018-1421-z.
- Zhang, J., Ding, G., Zou, Y., Qin, S., and Fu, J. (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing*, 30(4):1809–1830. doi: 10.1007/s10845-017-1350-2.
- Zhang, L., Yang, C., Yan, Y., and Hu, Y. (2022a). Distributed Real-Time Scheduling in Cloud Manufacturing by Deep Reinforcement Learning. *IEEE Transactions on Industrial Informatics*, 18(12):8999–9007. doi: 10.1109/TII.2022.3178410.
- Zhang, Q., Vonderembse, M. A., and Lim, J.-S. (2003). Manufacturing flexibility: defining and analyzing relationships among competence, capability, and customer satisfaction. *Journal of Operations Management*, 21(2):173–191. doi: 10.1016/S0272-6963(02)00067-0.
- Zhang, Y., Bai, R., Qu, R., Tu, C., and Jin, J. (2022b). A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties. *European Journal of Operational Research*, 300(2):418–427. doi: 10.1016/j.ejor.2021.10.032.
- Zhang, Y., Zhou, Z., Shi, Z., Meng, L., and Zhang, Z. (2021). Online Scheduling Optimization for DAG-Based Requests Through Reinforcement Learning in Collaboration Edge Networks. *IEEE Access*, 8:72985–72996. doi: 10.1109/ACCESS.2020.2987574.
- Zheng, T., Ardolino, M., Bacchetti, A., and Perona, M. (2020). The applications of Industry 4.0 technologies in manufacturing context: a systematic literature review. *International Journal of Production Research*, pages 1–33. doi: 10.1080/00207543.2020.1824085.
- Zheng, W., Tan, Y., Meng, L., and Zhang, H. (2018). An improved MOEA/D design for many-objective optimization problems. *Applied Intelligence*, 48(10):3839–3861. doi: 10.1007/s10489-018-1183-5.
- Zhong, R. Y., Xu, X., Klotz, E., and Newman, S. T. (2017). Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering*, 3(5):616–630. doi: 10.1016/J.ENG.2017.05.015.
- Zhou, B. and Zhao, L. (2022). A multi-objective decomposition evolutionary algorithm based on the double-faced mirror boundary for a milk-run material feeding scheduling optimization problem. *Computers & Industrial Engineering*, 171:108385. doi: 10.1016/j.cie.2022.108385.
- Zhou, J., Li, P., Zhou, Y., Wang, B., Zang, J., and Meng, L. (2018). Toward New-Generation Intelligent Manufacturing. *Engineering*, 4(1):11–20. doi: 10.1016/j.eng.2018.01.002.
- Zhou, T., Tang, D., Zhu, H., and Zhang, Z. (2021). Multi-agent reinforcement learning for online scheduling in smart factories. *Robotics and Computer-Integrated Manufacturing*, 72:102202.

doi: 10.1016/j.rcim.2021.102202.

- Zhou, Y., Yang, J.-j., and Huang, Z. (2020). Automatic design of scheduling policies for dynamic flexible job shop scheduling via surrogate-assisted cooperative co-evolution genetic programming. *International Journal of Production Research*, 58(9):2561–2580. doi: 10.1080/00207543.2019.1620362.
- Zhou, Y., Yang, J.-J., and Zheng, L.-Y. (2019). Multi-Agent Based Hyper-Heuristics for Multi-Objective Flexible Job Shop Scheduling: A Case Study in an Aero-Engine Blade Manufacturing Plant. *IEEE Access*, 7:21147–21176. doi: 10.1109/ACCESS.2019.2897603.
- Zimmermann, H.-J. (2008). *Operations Research*. Vieweg+Teubner Verlag, Wiesbaden. ISBN: 978-3-8348-0455-6, 978-3-8348-9461-8.
- Zäpfel, G. (2000). *Taktisches Produktions-Management*. Oldenbourg Wissenschaftsverlag. ISBN: 978-3-486-25464-8.
- Zäpfel, G. (2001). *Grundzüge des Produktions- und Logistikmanagement*. Internationale Standardlehrbücher der Wirtschafts- und Sozialwissenschaften. Oldenbourg, München Wien, 2 edition. ISBN: 978-3-486-25618-5.

List of Figures

1.1	Production planning and control hierarchy	2
1.2	Trend towards higher product variety	3
1.3	Adaptability and flexibility measures	8
1.4	Framework for production planning and control design	10
1.5	Pursued design science research methodology	11
1.6	Pursued methodology for the scientific approach	13
1.7	Thesis structure	15
2.1	Structure of the fundamentals chapter	17
2.2	Exemplary job-shop process	19
2.3	Comparison of a conventional and matrix assembly	20
2.4	Classification of agent based systems	22
2.5	Hayes-Wheelwright	25
2.6	Aachen production planning and control model	26
2.7	Closed loop production control model	27
2.8	Hyper-heuristics principle	32
2.9	Agent - environment interaction loop, adapted from Sutton and Barto (2017)	34
2.10	Value- and policy-based neural networks	35
2.11	Neural network-based policy approximation in deep RL	39
2.12	Types of neural network	39
2.13	DQN operating with target network	41
3.1	Elaboration of the fundamental research base	44
3.2	SLR review steps	47
3.3	Pursued taxonomy framework	47
4.1	Agent-Environment Interaction	64
4.2	Eight step approach to conduct a SLR	66
4.3	Conducted review process	68
4.4	Analysis of yearly deep RL publications, 2021 includes Jan./Feb.	69
4.5	Number of publications per outlet; 2010-2021	70
4.6	Number of publications allocated to the production disciplines	71
4.7	Quantitative analysis of applied algorithms and testing environments	84

List of Figures

5.1	Consolidated review process	114
5.2	Analysis of yearly and outlet publications	115
6.1	Applied algorithms in deep learning based production research	167
6.2	System re-configuration capabilities according to <i>VDI5201</i>	169
6.3	Semi-heterarchical control framework	172
6.4	Decreasing heuristics share during initial learning process	176
6.5	Summarized research gap, requirements, and specifications	177
6.6	Design and control framework with proposed artifact elements	178
6.7	Breakdown of throughput time components as per Nyhuis and Wiendahl (2012): (a) order- and (b) process step-related processing times	180
7.1	Pursued <i>DSRM</i> methodology, Peffers et al. (2007)	193
7.2	Hyper-heuristics and DQN based optimization, inspired by Goos et al. (2001), Lorente et al. (2001)	194
7.3	Distributed decision-making and parallel processing of the dispatching agents; left: Sutton and Barto (2017)	195
7.4	Descriptive manufacturing and distribution modules within the simulation	196
7.5	Simulation framework with flexible module recognition	197
7.6	Enhancing simulation scalability through a standardized neural network stack	198
7.7	Simulated 3-staged modular production system for PCB and electric drive fabri- cation	201
7.8	Development order throughput times and tardiness during the training process	204
7.9	Moving average of chosen actions for a D1 module agent throughout the training process	205
7.10	Moving average of a another agent in D1, emphasizing a clear trend towards explainable action selection	205
8.1	Hyper-heuristics based optimization approach	226
8.2	Pursued <i>DSRM</i> methodology (Peffers et al., 2007)	231
8.3	Projected multi-agent and semi-heterarchical system	232
8.4	Simulated three-layer modular production system	237
8.5	Moving average of obtained rewards for the top-layer <i>D1</i> agent	240
8.6	Moving average of throughput times related to order priorities	241
8.7	Moving average of agent rewards for the D1 distribution module	244
8.8	Testing setup of the hyper-heuristic within the hybrid production environment	247
9.1	Simulated modular production system	264
9.2	Reward progression of a deep RL agent during the training phase	268
10.1	Threefold reduction in optimization and control complexity with proposed elements	278
10.2	Production variety and volume - research versus reality	283

10.3 Integrative and interactive process model for the framework implementation . .	284
10.4 Taxonomy of deep learning model integration capabilities	286
11.1 Extended simulation framework	294
11.2 Projected semi-heterarchical organization with virtual structure	296

List of Tables

1.1	Research guidelines for design science by Hevner et al. (2004) and coverage in this thesis	12
2.1	Production process categories	18
2.2	Organization dependent optimization, according to Trentesaux (2009); Sallez et al. (2010); Borangiu et al. (2015)	23
2.3	Popular dispatching rules	30
2.4	Overview of machine learning methods	33
2.5	Overview of reinforcement learning methods	34
4.1	Taxonomy framework of the SLR	66
4.2	Defined keywords for the SLR	67
4.3	Summary of deep RL applications in process control	72
4.4	Summary of deep RL applications in production scheduling, dispatching, and (intra-) logistics	76
4.5	Summary of deep RL applications in assembly and robotics	79
4.6	Summary of deep RL applications in maintenance, energy management, and (process) design	82
4.7	Summary of deep RL applications in quality control and further applications	82
4.8	Summary of the key findings from the review analysis	83
5.1	Pursued taxonomy framework	112
5.2	Keywords defined for the review	113
5.3	System and categorical split of the reviewed literature	116
5.4	Key statistics from the review process	124
5.5	MA system interaction and training approaches	125
5.6	Proposed taxonomy for single- and multi-agent system interaction	126
5.7	Types of interaction in multi-agent systems	129
5.8	List of abbreviations	155
5.9	Plain NN based approaches in production planning	156
5.10	Embedded NN based approaches in production planning	157
5.11	Multi-agent NN based approaches in production planning	158
5.12	Plain NN based approaches in production forecasting	159

List of Tables

5.13	Embedded and multi-agent NN approaches in production forecasting	160
5.14	Plain NN based approaches in production control	161
5.15	Embedded and multi-agent NN approaches in production control	162
6.1	Deep learning and multi-agent based approaches	164
6.2	Comparison of job-shop, matrix and modular production specifications	171
7.1	Overview of prevailing simulation frameworks	188
7.2	Summary of deep RL based control approaches in multi-agent production systems	192
7.3	Available state parameters within the simulation framework	194
7.4	Summarized reward elements for individual and common rewards	199
7.5	Summarized processing times of both product groups; in [min.]	202
7.6	Iteratively defined deep RL and training parameter settings	202
7.7	Control optimization and benchmark against conventional heuristics	207
7.8	Assessment of system resilience against fluctuating order loads	208
8.1	Deep RL based multi-agent approaches in production control	229
8.2	Parameter settings for the deep RL agents	238
8.3	Order type specifications with associated processing times [min.] and sequences	239
8.4	Multi-objective optimization benchmark incorporating order priorities	242
8.5	Order urgency and priority dependent optimization of throughput times [min.] .	243
8.6	Summary of reward mean and standard deviation	245
8.7	Performance benchmark within the hybrid production environment	246
9.1	Production state parameters	265
9.2	Experimental order settings for simulation	267
9.3	Summary of key financial indicators	268
9.4	Summary of key technical indicators [sec.]	269

Declaration on Honour

Versicherung

Nachname	Panzer
Vorname	Marcel
geb. am	12.06.1994
in:	Göttingen
Titel der Dissertation	Design of a hyper-heuristics based control framework for modular production systems

Ich habe mich noch in keinem Promotionsverfahren befunden.

Ich habe mich bereits früher von _____ bis _____ (Zeitraum angeben) in einem Promotionsverfahren befunden und habe dieses erfolgreich/erfolglos (unzutreffendes bitte streichen) beendet.

Die hier eingereichte Arbeit oder wesentliche Teile sind in keinem anderen Verfahren zur Erlangung eines akademischen Grades vorgelegt worden.

Ich versichere, dass:

- die Dissertation selbständig und ohne fremde Hilfe verfasst wurde,
- nur die angegebenen Quellen und Hilfsmittel benutzt wurden,
- die wörtlich oder sinngemäß entnommenen Stellen aus den benutzten Werken als solche kenntlichgemacht wurden,
- sowie die Grundsätze zur Sicherung guter wissenschaftlicher Praxis der DFG eingehalten wurden.

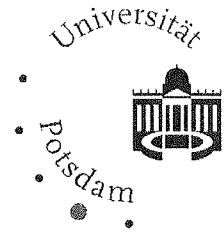
Einer Überprüfung mittels einer Plagiatssoftware stimme ich zu.

Ort/Datum

Unterschrift

Statements by the co-authors

KO-AUTOR*INNENERKLÄRUNG
DECLARATION OF CO-AUTHORSHIP



Name des Kandidaten/ Name of the candidate:	
Marcel Panzer	
Titel des Artikels/ Title of the article:	
Deep reinforcement learning in production systems: a systematic literature review	
<input type="checkbox"/>	nicht eingereicht/ <i>not submitted</i>
<input type="checkbox"/>	eingereicht bei/ <i>submitted to:</i>
<input checked="" type="checkbox"/>	zur Veröffentlichung angenommen oder veröffentlicht in/ <i>accepted for publication or published in:</i>
International Journal of Production Research	
Arbeitsanteil des Kandidaten an vorgenanntem Artikel/ Quantification of candidate's contribution to the article overall:	
<input type="checkbox"/>	hat zur Arbeit beigetragen/ <i>has contributed to the work (<1/3)</i>
<input type="checkbox"/>	hat wesentlich zur Arbeit beigetragen/ <i>has made a substantial contribution (1/3 to 2/3)</i>
<input checked="" type="checkbox"/>	hat einen Großteil der Arbeit allein erbracht/ <i>did the majority of the work independently (>2/3)</i>
Ko-Autoren/ Co-authors (Name und Kontaktdaten/ full name and contact):	
Dr. Benedict Bender Karl-Marx-Straße 67, 14482 Potsdam	
Hiermit bestätige ich die Richtigkeit des oben beschriebenen Arbeitsanteils des Kandidaten./ <i>I hereby confirm the candidate's contribution as quantified above.</i>	
Potsdam, 05.01.2024	
Ort, Datum/ <i>Place, Date</i>	Unterschrift Ko-Autor/ <i>Signature Co-Author</i>

Alle personenbezogenen Angaben gelten stets für Frauen und Männer gleichermaßen.

KO-AUTOR*INNENERKLÄRUNG
 DECLARATION OF CO-AUTHORSHIP



Name des Kandidaten/ Name of the candidate:	
Marcel Panzer	
Titel des Artikels/ Title of the article:	
Neural agent-based production planning and control: an architectural review	
<input type="checkbox"/>	nicht eingereicht/ <i>not submitted</i>
<input type="checkbox"/>	eingereicht bei/ <i>submitted to:</i>
<input checked="" type="checkbox"/>	zur Veröffentlichung angenommen oder veröffentlicht in/ <i>accepted for publication or published in:</i>
Journal of Manufacturing Systems	
Arbeitsanteil des Kandidaten an vorgenanntem Artikel/ Quantification of candidate's contribution to the article overall:	
<input type="checkbox"/>	hat zur Arbeit beigetragen/ <i>has contributed to the work (<1/3)</i>
<input type="checkbox"/>	hat wesentlich zur Arbeit beigetragen/ <i>has made a substantial contribution (1/3 to 2/3)</i>
<input checked="" type="checkbox"/>	hat einen Großteil der Arbeit allein erbracht/ <i>did the majority of the work independently (>2/3)</i>
Ko-Autoren/ Co-authors (Name und Kontaktdaten/ full name and contact):	
Univ.-Prof. Dr.-Ing. habil. Norbert Gronau, Karl-Marx-Straße 67, 14482 Potsdam	
Hiermit bestätige ich die Richtigkeit des oben beschriebenen Arbeitsanteils des Kandidaten./ <i>I hereby confirm the candidate's contribution as quantified above.</i>	
Potsdam, 22.12.2023	_____
Ort, Datum/ <i>Place, Date</i>	Unterschrift Ko-Autor/ <i>Signature Co-Author</i>

Alle personenbezogenen Angaben gelten stets für Frauen und Männer gleichermaßen.

KO-AUTOR*INNENERKLÄRUNG
DECLARATION OF CO-AUTHORSHIP



Name des Kandidaten/ Name of the candidate:	
Marcel Panzer	
Titel des Artikels/ Title of the article:	
Neural agent-based production planning and control: an architectural review	
<input type="checkbox"/>	nicht eingereicht/ <i>not submitted</i>
<input type="checkbox"/>	eingereicht bei/ <i>submitted to:</i>
<input checked="" type="checkbox"/>	zur Veröffentlichung angenommen oder veröffentlicht in/ <i>accepted for publication or published in:</i>
Journal of Manufacturing Systems	
Arbeitsanteil des Kandidaten an vorgenanntem Artikel/ Quantification of candidate's contribution to the article overall:	
<input type="checkbox"/>	hat zur Arbeit beigetragen/ <i>has contributed to the work (<1/3)</i>
<input type="checkbox"/>	hat wesentlich zur Arbeit beigetragen/ <i>has made a substantial contribution (1/3 to 2/3)</i>
<input checked="" type="checkbox"/>	hat einen Großteil der Arbeit allein erbracht/ <i>did the majority of the work independently (>2/3)</i>
Ko-Autoren/ Co-authors (Name und Kontaktdaten/ full name and contact):	
Dr. Benedict Bender Karl-Marx-Straße 67, 14482 Potsdam	
Hiermit bestätige ich die Richtigkeit des oben beschriebenen Arbeitsanteils des Kandidaten./ <i>I hereby confirm the candidate's contribution as quantified above.</i>	
Potsdam, 05.01.2024	_____
Ort, Datum/ <i>Place, Date</i>	Unterschrift Ko-Autor/ <i>Signature Co-Author</i>

Alle personenbezogenen Angaben gelten stets für Frauen und Männer gleichermaßen.

KO-AUTOR*INNENERKLÄRUNG
DECLARATION OF CO-AUTHORSHIP



Name des Kandidaten/ Name of the candidate:	
Marcel Panzer	
Titel des Artikels/ Title of the article:	
Designing an adaptive and deep learning based control framework for modular production systems	
<input type="checkbox"/>	nicht eingereicht/ <i>not submitted</i>
<input type="checkbox"/>	eingereicht bei/ <i>submitted to:</i>
<input checked="" type="checkbox"/>	zur Veröffentlichung angenommen oder veröffentlicht in/ <i>accepted for publication or published in:</i>
Journal of Intelligent Manufacturing	
Arbeitsanteil des Kandidaten an vorgenanntem Artikel/ Quantification of candidate's contribution to the article overall:	
<input type="checkbox"/>	hat zur Arbeit beigetragen/ <i>has contributed to the work (<1/3)</i>
<input type="checkbox"/>	hat wesentlich zur Arbeit beigetragen/ <i>has made a substantial contribution (1/3 to 2/3)</i>
<input checked="" type="checkbox"/>	hat einen Großteil der Arbeit allein erbracht/ <i>did the majority of the work independently (>2/3)</i>
Ko-Autoren/ Co-authors (Name und Kontaktdaten/ full name and contact):	
Univ.-Prof. Dr.-Ing. habil. Norbert Gronau, Karl-Marx-Straße 67, 14482 Potsdam	
Hiermit bestätige ich die Richtigkeit des oben beschriebenen Arbeitsanteils des Kandidaten./ <i>I hereby confirm the candidate's contribution as quantified above.</i>	
Potsdam, 22.12.2023	_____
Ort, Datum/ <i>Place, Date</i>	Unterschrift Ko-Autor/ <i>Signature Co-Author</i>

Alle personenbezogenen Angaben gelten stets für Frauen und Männer gleichermaßen.

KO-AUTOR*INNENERKLÄRUNG
DECLARATION OF CO-AUTHORSHIP



Name des Kandidaten/ Name of the candidate:	
Marcel Panzer	
Titel des Artikels/ Title of the article:	
A deep reinforcement learning based hyper-heuristic for modular production control	
<input type="checkbox"/>	nicht eingereicht/ <i>not submitted</i>
<input type="checkbox"/>	eingereicht bei/ <i>submitted to:</i>
<input checked="" type="checkbox"/>	zur Veröffentlichung angenommen oder veröffentlicht in/ <i>accepted for publication or published in:</i>
International Journal of Production Research	
Arbeitsanteil des Kandidaten an vorgenanntem Artikel/ Quantification of candidate's contribution to the article overall:	
<input type="checkbox"/>	hat zur Arbeit beigetragen/ <i>has contributed to the work (<1/3)</i>
<input type="checkbox"/>	hat wesentlich zur Arbeit beigetragen/ <i>has made a substantial contribution (1/3 to 2/3)</i>
<input checked="" type="checkbox"/>	hat einen Großteil der Arbeit allein erbracht/ <i>did the majority of the work independently (>2/3)</i>
Ko-Autoren/ Co-authors (Name und Kontaktdaten/ full name and contact):	
Univ.-Prof. Dr.-Ing. habil. Norbert Gronau, Karl-Marx-Straße 67, 14482 Potsdam	
Hiermit bestätige ich die Richtigkeit des oben beschriebenen Arbeitsanteils des Kandidaten./ <i>I hereby confirm the candidate's contribution as quantified above.</i>	
Potsdam, 22.12.2023	
Ort, Datum/ <i>Place, Date</i>	Unterschrift Ko-Autor/ <i>Signature Co-Author</i>

Alle personenbezogenen Angaben gelten stets für Frauen und Männer gleichermaßen.

KO-AUTOR*INNENERKLÄRUNG
 DECLARATION OF CO-AUTHORSHIP



Name des Kandidaten/ Name of the candidate:	
Marcel Panzer	
Titel des Artikels/ Title of the article:	
A deep reinforcement learning based hyper-heuristic for modular production control	
<input type="checkbox"/>	nicht eingereicht/ <i>not submitted</i>
<input type="checkbox"/>	eingereicht bei/ <i>submitted to:</i>
<input checked="" type="checkbox"/>	zur Veröffentlichung angenommen oder veröffentlicht in/ <i>accepted for publication or published in:</i>
International Journal of Production Research	
Arbeitsanteil des Kandidaten an vorgenanntem Artikel/ Quantification of candidate's contribution to the article overall:	
<input type="checkbox"/>	hat zur Arbeit beigetragen/ <i>has contributed to the work (<1/3)</i>
<input type="checkbox"/>	hat wesentlich zur Arbeit beigetragen/ <i>has made a substantial contribution (1/3 to 2/3)</i>
<input checked="" type="checkbox"/>	hat einen Großteil der Arbeit allein erbracht/ <i>did the majority of the work independently (>2/3)</i>
Ko-Autoren/ Co-authors (Name und Kontaktdaten/ full name and contact):	
Dr. Benedict Bender Karl-Marx-Straße 67, 14482 Potsdam	
Hiermit bestätige ich die Richtigkeit des oben beschriebenen Arbeitsanteils des Kandidaten./ <i>I hereby confirm the candidate's contribution as quantified above.</i>	
Potsdam, 05.01.2024	
Ort, Datum/ <i>Place, Date</i>	Unterschrift Ko-Autor/ <i>Signature Co-Author</i>

Alle personenbezogenen Angaben gelten stets für Frauen und Männer gleichermaßen.

KO-AUTOR*INNENERKLÄRUNG
DECLARATION OF CO-AUTHORSHIP



Name des Kandidaten/ Name of the candidate:	
Marcel Panzer	
Titel des Artikels/ Title of the article:	
Enhancing economic efficiency in modular production systems through deep reinforcement learning	
<input type="checkbox"/>	nicht eingereicht/ <i>not submitted</i>
<input type="checkbox"/>	eingereicht bei/ <i>submitted to:</i>
<input checked="" type="checkbox"/>	zur Veröffentlichung angenommen oder veröffentlicht in/ <i>accepted for publication or published in:</i>
11th CIRP Global Web Conference (CIRPe 2023)	
Arbeitsanteil des Kandidaten an vorgenanntem Artikel/ Quantification of candidate's contribution to the article overall:	
<input type="checkbox"/>	hat zur Arbeit beigetragen/ <i>has contributed to the work (<1/3)</i>
<input type="checkbox"/>	hat wesentlich zur Arbeit beigetragen/ <i>has made a substantial contribution (1/3 to 2/3)</i>
<input checked="" type="checkbox"/>	hat einen Großteil der Arbeit allein erbracht/ <i>did the majority of the work independently (>2/3)</i>
Ko-Autoren/ Co-authors (Name und Kontaktdaten/ full name and contact):	
Univ.-Prof. Dr.-Ing. habil. Norbert Gronau, Karl-Marx-Straße 67, 14482 Potsdam	
Hiermit bestätige ich die Richtigkeit des oben beschriebenen Arbeitsanteils des Kandidaten./ <i>I hereby confirm the candidate's contribution as quantified above.</i>	
Potsdam, 22.12.2023	_____
Ort, Datum/ <i>Place, Date</i>	Unterschrift Ko-Autor/ <i>Signature Co-Author</i>

Alle personenbezogenen Angaben gelten stets für Frauen und Männer gleichermaßen.