Wirtschafts- und Sozialwissenschaftliche Fakultät

Marcel Panzer | Benedict Bender | Norbert Gronau

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

Suggested citation referring to the original publication:
Journal of Manufacturing Systems 65 (2022) , pp. 743 - 766
DOI: https://doi.org/10.1016/j.jmsy.2022.10.019
ISSN: 0278-6125

Journal article | Version of record

Journal of Manufacturing Systems 65 (2022) 743–766

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Review

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

Marcel Panzer *, Benedict Bender, Norbert Gronau

Chair of Business Informatics, Processes and Systems, University of Potsdam, Germany

ARTICLE INFO

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

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.

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. [1], 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 [2,3]. In addition, machine learning (ML) is increasingly employed due to the growing capabilities of the given infrastructure in recent years [4]. ML can assist in performing multi-criteria optimization involving local and global objectives, multiple resources, machines, and factories [5] that demand a continuously optimized production control and schedule [4].

However, according to Cadavid et al. [6], 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. [7], 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 [8], but also by others such as Weichert et al. [9], Kang et al. [4], and Zhou et al. [10], highlighting the versatility of ML algorithms in various production scenarios. Nevertheless, Weichert et al. [9] 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 [6]. 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

* Correspondence to: Chair of Business Informatics, Processes and Systems, University of Potsdam, August-Bebel-Str. 89, 14482 Potsdam, Brandenburg, Germany.

E-mail address: marcel.panzer@wi.uni-potsdam.de (M. Panzer).

https://doi.org/10.1016/j.jmsy.2022.10.019
Received 8 February 2022; Received in revised form 14 October 2022; Accepted 25 October 2022
Available online 19 November 2022

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. [11] 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. [12] adopted what Baker [13] 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 [14] and Garetti and Taisch [15] 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 [5], or just in the context of other ML techniques [4], lacking a consolidation of NN-based PPC contributions. This also becomes apparent in the work of De Modesti et al. [16], who described the increased relevance of NNs but, like Çalis̜ and Bulkan [17] for job shop scheduling or Bertolini et al. [18] 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. [19]. The benefits of holonic systems were further outlined, among others, in Babiceanu and Chen [2] and recently in Derigent et al. [20], and was featured as one of 6 enablers for smart manufacturing control in Rojas and Rauch [21]. Beyond that, Lee and Kim [22] and Monostori et al. [23] outlined how MA systems enable robust and flexible production, similar to Gronauer and Diepold [3] or Herrera et al. [24], 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 8. Analysis tables with detailed review information can be found in Appendix.

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 [25]. 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 [26]. Related to Industry 4.0, the adoption of technologically advanced techniques in PPC can be deployed to improve performance [27]. In recent years, this has included NNs in particular, which have not only experienced great success with Google DeepMind [28] but are also increasingly implemented in production and can prevent extensive modeling or high dependence on human experience (as in [29]).

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 [30]. 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 [30,31].

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 [32,33]. According to Baker [13], an agent is a self-controlled software object that has its own values and communicates with other objects. Based on this, Patel et al. [34] 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 [35]. Early approaches in 1995 such as Zhang and Dietterich [36] and Zhang and Dietterich [37] 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 [38,39]. Riedmiller and Riedmiller [38] 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. [39] 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 [40] and Dorri et al. [41], respectively.

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 [42]. 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 [13]. The intermediate step between both

Table 1
Pursued taxonomy framework.

<table>
<tr><th>Characteristic</th><th colspan="12">Categories</th></tr>
<tr><td>(1) Focus</td><td colspan="3">Research outcomes</td><td colspan="3">Research methods</td><td colspan="3">Theories</td><td colspan="3">Applications</td></tr>
<tr><td>(2) Goal</td><td colspan="4">Integration</td><td colspan="4">Criticism</td><td colspan="4">Central issues</td></tr>
<tr><td>(3) Perspective</td><td colspan="6">Neutral representation</td><td colspan="6">Espousal of position</td></tr>
<tr><td>(4) Coverage</td><td colspan="3">Exhaustive</td><td colspan="3">Exhaustive & selective</td><td colspan="3">Representative</td><td colspan="3">Central/pivotal</td></tr>
<tr><td>(5) Organisation</td><td colspan="4">Historical</td><td colspan="4">Conceptual</td><td colspan="4">Methodological</td></tr>
<tr><td>(6) Audience</td><td colspan="3">Specialized scholars</td><td colspan="3">General scholars</td><td colspan="3">Practitioners/politicians</td><td colspan="3">General public</td></tr>
</table>

extremes is characterized as a holonic structure [43]. 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 [44]. 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.

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. [45] and Thomé et al. [46]. 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 [47]. Following Thomé et al. [46], we have organized the systematic literature review (SLR) into 8 (iterative) steps, from planning to updating the review, which are addressed next.

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. [48] and follows the associated taxonomy framework of Cooper [49], which is outlined in Table 1. Cells highlighted in gray represent the selection of underlying characteristics of the review and the associated objectives and focus areas.

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.

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.

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 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.

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 [50]). 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. [51] 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.

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 Fig. 1. Beginning with the database extraction and the 1794 papers initially obtained, duplicates were first removed and years filtered before applying thematic criteria.

Table 2
Keywords defined for the review.

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

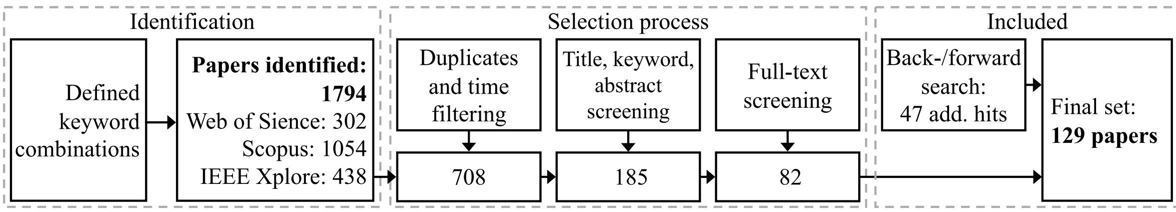


Fig. 1. Consolidated review process.

To ensure a high review quality, we screened the remaining 708 papers by title, keywords, and abstract according to Thomé et al. [46] 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 [52], 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.

3.2.4. Phase 4 - Data gathering

In accordance with Thomé et al. [46] and Webster and Watson [52], 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 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.

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 Fig. 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 Fig. 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.

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 4.4.

Besides the increasing tendency for total publication numbers from Fig. 2(a), a shift of research foci within the field becomes apparent in the system split shown on the left in Table 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.

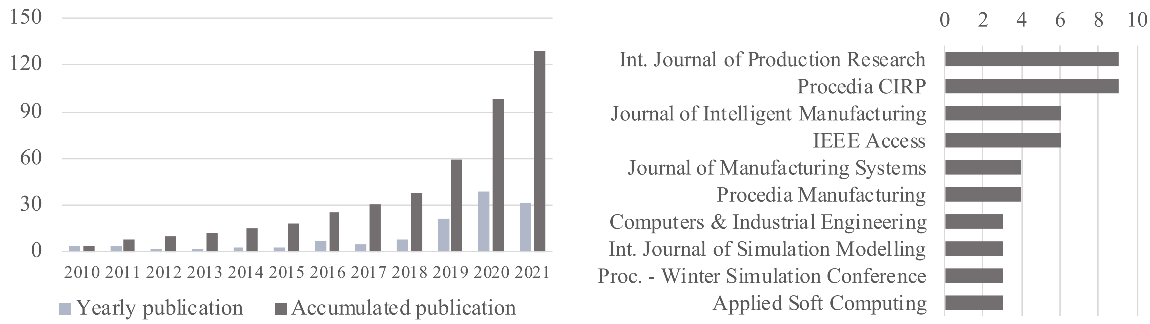
Although most of the publications within the categorical split on the right of Table 3 implemented plain approaches, especially in production control, one-fifth of the publications were based on MA approaches that benefit from a resulting complexity breakdown and increased scalability. The high embedded 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 [53]).

To ensure a consistent structure and address RQ1, the rest of the review is organized according to the section numbers given in Table 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.

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.

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 [54].



(a) Yearly and accumulated publications (b) Publications per outlet, 2010-2021

Fig. 2. Analysis of yearly and outlet publications.

Table 3
System and categorical split of the reviewed literature.

	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

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. [55]. Also in flow-shop scheduling, Marchesano et al. [56] 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 [57] and Lin et al. [58] outperformed heuristics such as FIFO or SPT in job-shop scheduling. For implementation, Lin et al. [58] 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 [59], a DQN RL approach to minimize makespan in dynamic scheduling [60], a double DQN in rescheduling color batches to minimize change-over cost [61], or a BP algorithm in lot sizing to minimize production, set-up, and inventory costs [62], among others listed in Table A.9.

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 [63] chose a hybrid fuzzy and NN-based approach to minimize the makespan and reduce time-consuming efforts, Ramanan et al. [64] 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 [65], 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 [66]. To improve slow convergences and avoid the local optima trap in job-shop scheduling, Zhang et al. [67] 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. [68], 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. [69], 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 [70,71] .

In a hybrid simulation, Sobottka et al. [72] 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 [73] 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 A.10.

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. [74] 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. [75] and Baer et al. [76] 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. [76] 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. [77] 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. [77], in which agents did not communicate and only perceived each other's actions, Park et al. [78] and Lee et al. [79] 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. [77], Pol et al. [80] 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. [81] 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 A.11.

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 [82]. To avoid terminological conflicts, we categorized each paper according to the key variable addressed.

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 Yanık [83] 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 [84] 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. [85], 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. [86] 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. [87] 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 [88] 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. [89] based on job and shop status information to estimate the due date. Based on the flow time, Karaoglan and Karademir [90] further estimated production costs to generate more precise price offers. Among other papers listed in Table A.12, Kramer et al. [91] predicted lead times assuming constantly changing environmental variables, which cannot be captured in regular models. Similarly, Göppert et al. [92] 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.

4.2.2. Embedded NN forecasting approaches

To forecast cycle times in wafer fabrication, Chen [53] 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. [93] 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. [94] in a three-stage make-to-order flow shop and Mezzogori et al. [95] 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 [96] or proposed a combined analytical and LSTM approach [97] 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 [82] 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. [98], other approaches are listed in Table A.13.

4.2.3. Multi-agent forecasting approaches

As the only MA forecasting approach, Morariu and Borangiu [99] 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.

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.

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. [100] 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. [101] 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 [102] and Bergmann et al. [103] applied a control strategy approximation approach to increase the accuracy of system reproduction and minimize manual interventions. Luo [104] 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 [105] and Zhao and Zhang [106] outperformed conventional approaches using NN-based rule selection depending on the flow- or job-shop system parameters. For this purpose, Zhao and Zhang [106] 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. [107] not only proved strong performance but also increased generalizability using the transfer strategy.

Other publications listed in Table A.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 [108], or implemented unit-cost minimization to mitigate the disadvantage of conventional methods' uncertain demands and long changeovers in a dishwasher wire-rack production system [109].

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. [110]. 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. [111]. 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 [112]. Kim et al. [113] 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 A.15.

4.3.3. Multi-agent control approaches

To cope with the inherent dynamics in job-shop scheduling, Hammami et al. [114] 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 [115] and Hofmann et al. [116]. In Dittrich and Fohlmeister [115], 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. [116] 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. [117] 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 [118]. [118] 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. [119] 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. [120] 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 A.15.

4.4. General analysis

The general analysis is briefly summarized in Table 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.

Referring to MA systems, most papers defined a global objective for agent-to-agent interaction (see Table 5). For this purpose, Pol et al. [80]

Table 4
Key statistics from the review process.

	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
MA system interaction and training approaches.

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%

derived a reward factor based on the total makespan and multiplied it by the local rewards for each agent. Waschneck et al. [117], 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 [115] or by partial activation of agents in Zhou et al. [81], 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. [75]. Such sharing of agent information was also referred to as indirect interaction [80] or sensing [77], indicating that agents must anticipate what the others might do next. A direct collaboration approach was also facilitated by bidding (as in [119]) or negotiation mechanisms (as in [121]).

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. [78] or Dittrich and Fohlmeister [115], 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 [79]). Others, such as Morariu and Borangiu [99] and Waschneck et al. [117], adopted a decentralized or decentralized iterative training approach, respectively. While Morariu and Borangiu [99] deployed LSTMs in parallel to learn machine cost patterns and generate bids, Waschneck et al. [117] 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. [80]. Where agents learned to meet local goals first in a decentralized manner, they were optimized to reach a global goal in a subsequent phase.

5. Taxonomy

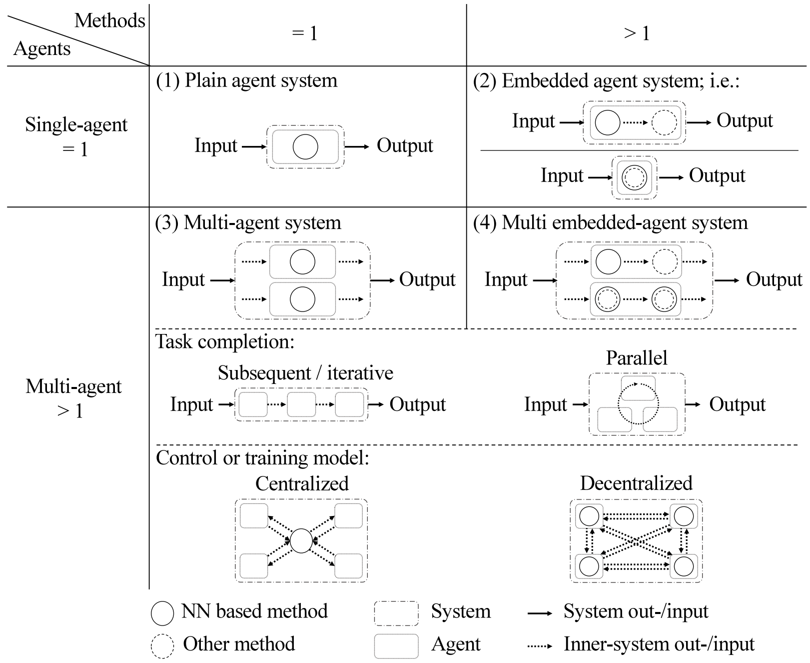
We propose a taxonomy to classify the implementation of NN in production and general systems, following the taxonomy development method of Nickerson et al. [122]. Proceeding from the empirical-to-conceptual path, we first identified object subsets based on the employed NNs and agent structures through our review and then condensed them into a coherent framework. The clustering of these is integrated into Table 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.

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 [123]). 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 [97]). 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 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. [34] 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.

Table 6
Proposed taxonomy for single- and multi-agent system interaction.



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 sills 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, the interaction between agents can adopt different forms. Based on this, Table 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 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.

Table 7
Types of interaction in multi-agent systems.

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

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).

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. [77] 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. [110] 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. [119]. To avoid instabilities, Gros et al. [118] and Waschneck et al. [117] 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. [77] 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 [118] or vibration during training [124]. 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. [125].
- **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 [126] and has a tremendous impact on the final performance [103,127]. In particular, Wu et al. [128] 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.

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. [117]. 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. [129] in a resource preemption environment, or by Kim et al. [130] 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. [131], 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. [94], depending on the specific complexity allocation.
Another non-NN-based example was presented in Minguillon and Lanza [132] by combining centralized and decentralized scheduling properties for the adjustment of degrees of freedom. As mentioned in Schwung et al. [133], 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 realized in learning factory environments, as discussed by Teichmann et al. [134].
- **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. [77], 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 [135] or nuclear mass training [136]. To achieve this within the relevant scope, Zang et al. [66] 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 [137], 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. [81] 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. [87] for decision support.

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.

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 adaption, can be facilitated.

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.

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.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Supplements and detailed review tables

See Tables A.8–A.15

Table A.8
List of abbreviations.

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 A.9
Plain NN based approaches in production planning.

	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. [60]
2	Dynamic scheduling	A2C	FFNN	Max. profitability	Superior	Continuous chemical process	Simulation	Hubbs et al. [59]
3	Dynamic scheduling	DQN	FFNN	Minimize completion time	–	General tasks, services	Simulation	Zhou et al. [138]
4	Dynamic scheduling	Policy gradient	FFNN	Maximize resource utilization	Similar	Cloud manufacturing	Simulation	Zhu et al. [139]
5	Flow-shop scheduling	DQN	FFNN	Maximize throughput	–	Flow-shop	Simulation	Marchesano et al. [56]
6	Flow-shop scheduling	Levenberg-Marquardt	FFNN	Minimize overall completion time	–	Flow-shop	Simulation	Rouhani et al. [140]
7	Flow-shop scheduling	REINFORCE	LSTM	Negative total tardiness	Superior	Medical mask production	Simulation	Wu et al. [55]
8	Job-shop scheduling	AC	RNN	Min. setup waste	Similar	Blown film extrusion	Simulation	Gannouni et al. [141]
9	Job-shop scheduling	HNN	HNN	Min. makespan	–	Job-shop	Simulation	Fnaiech et al. [142]
10	Job-shop Scheduling	DQN	FFNN	Minimize makespan	Superior	Job-shop	Simulation	Groth et al. [143]
11	Job-shop scheduling	DDDQN with PER	Conv.	Minimize makespan	Superior	Job-shop	Simulation	Han and Yang [57]
12	Job-shop scheduling	DQN	FFNN	Minimize lead-time	Superior	Job-shop	Simulation	Kardos et al. [144]
13	Job-shop scheduling	PPO	GNN	Minimize makespan	Superior	Job-shop	Simulation	Park et al. [145]
14	Job-shop scheduling	PPO	FFNN	Optimize exec. time, minimize makespan	Superior	Job-shop	Simulation	Wang et al. [146]
15	Job-shop scheduling	DQN	FFNN	Minimize makespan, costs, balance workloads	Superior	Job-shop	Simulation	Zhou et al. [147]
16	Job-shop scheduling	DQN	FFNN	Completion time, energy con., utilization	Superior	Reconfigurable production	Simulation	Chen et al. [148]
17	Job-shop scheduling	DQN	FFNN	Minimize makespan	Superior	Semiconductor	Simulation	Lin et al. [58]
18	Job-shop scheduling	DQN	FFNN	Minimize completion time, lateness	Superior	Single machine job-shop	Simulation	Xie et al. [149]
19	Job-shop scheduling	DRL	GCNN	Maximize fill rate	–	swv11 in OR library	Simulation	Seito et al. [150]
20	Lot scheduling	PPO	FFNN	Min. waiting times, amount, cost	Superior	Single machine	Simulation	Rummukainen et al. [126]
21	Lot-sizing	BP	FFNN	Minimize production, set-up, and inventory costs	Superior	Air supply and maintenance center	Simulation	Şenyiğit and Atici [62]
22	Re-entrant production	DQN	FFNN	Robustness	Similar	Single-product production	Simulation	Shi et al. [124]
23	Rescheduling	DQN	Conv.	Minimize tardiness	Superior	Semi-continuous extruders	Simulation	Palombarini et al. [151]
24	Rescheduling	DQN	Conv.	Minimize tardiness	Superior	Semi-continuous extruders	Simulation	Palombarini et al. [152]
25	Rescheduling	Double DQN	Conv.	Minimize changeover costs	Superior	Color batching	Simulation	Leng et al. [61]
26	Rush-order rescheduling	Supervised/ BP	FFNN	Precision and accuracy	Similar	Job-shop	Simulation	Madureira et al. [153]
27	Task scheduling	DQN	FFNN	Minimize makespan	Superior	Cloud manufacturing	Simulation	Dong et al. [154]

Table A.10
Embedded NN based approaches in production planning.

	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. [155]
29	Flow-shop scheduling	Supervised	FFNN	Minimize makespan	Superior	Hybrid fuzzy and NN based concept	Three echelon supply chain	Simulation	Kumar and Giri [63]
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. [64]
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. [65]
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. [66]
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. [156]
34	Job-shop scheduling	BP	FFNN	Minimize max. makespan	Superior	PSO-based NN optimization	Job-shop	Simulation	Zhang et al. [67]
35	Modeling	BP	FFNN	Accuracy	Superior	NN as meta-modeler for GA tuning	Industrial bakery	Hybrid	Sobottka et al. [72]
36	Order allocation	Lagrangian	FFNN	Utility of AM Cloud	Superior	Allocation and payment network	Additive mfg. order allocation	Simulation	Mashhadi et al. [157]
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. [68]
38	Production planning	BP	FFNN	Production cost	–	Hybrid monkey algorithm, stochastic simulation, NN	Fuel production	Simulation	Lan et al. [158]
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 [159]
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 [71]
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. [69]
42	Remanufacturing scheduling	BP	FFNN	Accuracy	–	FFNN into GA to calculate chromosome output	Cam-/crankshaft remanufacturing	Simulation	Wen et al. [70]
43	Rescheduling	–	FFNN	Minimize response time	Superior	Supervised dimensionality reduction, GRNN mapping, SVM rescheduling	Job-shop	Simulation	Wang et al. [160]

(continued on next page)

Table A.10 (*continued*).

	Embedded planning approaches								
	Subtopic	Algo.	NN	Objective	Super.	Embedding	Application	Simulation	Source
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 [161]
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. [73]
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. [162]
47	Task scheduling	BP	FFNN	Optimal evaluation	–	3-module system with stochastic classification, training/validation NN, and interactive validation	Knitting processes	Simulation	Baeza Serrato [163]

Table A.11
Multi-agent NN based approaches in production planning.

	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 crtl. by heuristics	Job-shop	Simulation	Baer et al. [76]
49	Job-shop scheduling	DQN	FFNN	Minimize makespan	–	None (sensing)	Joint-action learning	Job-shop	Simulation	Baer et al. [77]
50	Job-Shop scheduling	Asyn. DDPG	Conv.	Minimize makespan	Superior	Agent state information	Central and parallel training	Job-shop	Simulation	Liu et al. [75]
51	Job-shop scheduling	DQN	FFNN	Minimize makespan	Superior	Agent state information, global objective	Single NN instance	Job-shop	Simulation	Pol et al. [80]
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. [81]
53	Real-time scheduling	Simulated annealing	FFNN	Minimize tardiness	Superior	Agent information exchange, global objective	–	Job-shop	Simulation	Hammami et al. [74]
54	Robust scheduling	DQN	FFNN	Minimize makespan	Superior	None	Central training	Semicond. scheduling	Simulation	Park et al. [78]
55	Sustainable scheduling	DQN	FFNN	Minimize process time	Superior	None	Central training	Mold scheduling	Simulation	Lee et al. [79]

Table A.12
Plain NN based approaches in production forecasting.

	Plain forecasting approaches							
	Forecast	Subtopic	Algo.	NN	Super.	Application	Simulation	Source
56	Cycle-time	Dispatching	BP	FFNN	–	Semiconductor	Simulation	Chakravorty et al. [164]
57	Cycle-time	Flexible mfg.	BP	FFNN	Superior	Textile mfg.	Simulation	Onaran et al. [83]
58	Cycle-time	Production planning	BP	FFNN	–	Textile mfg.	Simulation	Cao and Ji [84]
59	Cycle-time	Virtual machine prototype	BP	FFNN	–	Job-shop	Simulation	Jain et al. [165]
60	Electricity price	Energy cost oriented planning	BP	FFNN	Superior	Electricity price forecast	Simulation	Windler et al. [86]
61	Energy cost / consumption	Production planning	Levenberg-Marquardt	FFNN	–	Rotary clinker furnace	Simulation	Pusnik et al. [166]
62	Energy consumption	Production planning	–	FFNN	Superior	Industrial facility	Simulation	Ramos et al. [167]
63	Failure occurrence time	Dynamic scheduling	–	FFNN	Superior	Pharmaceutical factory	Simulation	Azab et al. [168]
64	Flow-time	Cost estimation	BP	FFNN	Superior	Oil-/dry-type cast resin transformers	Simulation	Karaoglan et al. [90]
65	Flow-time	Job-shop scheduling	Levenberg-Marquardt	FFNN	Superior	Job-shop	Simulation	Silva et al. [89]
66	Lead-time	Job-shop scheduling	Supervised	FFNN	Superior	Job-shop	Simulation	Kramer et al. [91]
67	Lead-time	Job-shop scheduling	–	FFNN	–	Aluminum extrusion	Simulation	Sajko et al. [169]
68	Make-span	Online scheduling	AlphaZero	Conv. NN	–	Interconnected assembly	Simulation	Göppert et al. [92]
69	Number of mfg. products	Feasibility assessment	–	FFNN	–	Flywheel production	Simulation	Burduk et al. [170]
70	Liquid, oil, gas flow	Production back allocation	BP	FFNN	Superior	Samarang petrol mine	Reality	Pham and Phan [88]
71	Order compl. time	Job-shop control	BP	Deep belief network	Superior	RFID-driven job-shop	Simulation	Wang and Jiang [171]
72	Costs, output, quality	Black-box modeling	Levenberg-Marquardt	FFNN	–	Tennessee Eastman proc.	Simulation	Glavan et al. [85]
73	Processing times	Offline scheduling	BP	RNN	Inferior	Parallel machine sched.	Simulation	Yamashiro et al. [172]
74	Sequence deviation	Sequencing	BP	FFNN	Inferior	Automotive	Simulation	Stauder et al. [173]
75	Time constraint violations	Production planning	BP	RNN/LSTM	Similar	Job-shop	Simulation	May et al. [174]
76	Through-put	Process ctrl.	Supervised	FFNN	Superior	Geo-metallurgy	Simulation	Both et al. [175]
77	Through-put	Process ctrl./ order release	BP	FFNN	–	Color filter fabrication	Reality	Huang et al. [87]
78	WIP	Production planning	–	LSTM	Superior	Bottleneck machine	Simulation	Gallina et al. [176]

Table A.13
Embedded and multi-agent NN approaches in production forecasting.

	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 [53]
80	Cycle/ blockage/ starvation time	Bottleneck prediction	Levenberg–Marquardt	LSTM	Superior	2-staged cycle and starvation time prediction	Underbody assembly	Simulation	Lai et al. [93]
81	Energy con. patterns	Predictive planning	Unsupervised	LSTM	Superior	LSTM with prior classification and clustering	Job-shop	Simulation	Morariu et al. [177]
82	Gross demand	Master prod. scheduling	BP	FFNN	Superior	NN forecast for subsequent scheduling algorithms	Kalak Refinery System	Simulation	Sadiq et al. [96]
83	Job remaining time	Rescheduling	BP	FFNN	Superior	Deep autoencoder extracts features, NN predicts jobs remaining time forecast	Aeroengine production	Reality	Fang et al. [98]
84	Lead-time	Make-to-order manufacturing	BP	FFNN	Superior	Non-/Bottleneck forecasting separation	Three-stage flow-shop	Simulation	Schneckenreither et al. [94]
85	Lead-time	Workload control	–	FFNN	Superior	WLC based control with NN prediction to define delivery dates	Job-shop	Simulation	Mezzogori et al. [95]
86	Load-value	Production scheduling	BP	FFNN	–	Affinity propagation operations clustering with FFNN forecasting	Semiconductor	Simulation	Han et al. [178]
87	Order completion time	Production scheduling	BP	FFNN	Superior	NN for prediction, GA/SA for global/ local tuning	Job-shop	Simulation	Hu and Zhou [179]
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. [82]
89	Product completion time	Production scheduling	–	LSTM	Superior	NN prediction w.analytical model as baseline	Multi-product serial production	Simulation	Huang et al. [97]
90	Production process	Make-to-order manufacturing	BP	DBN	Superior	2-staged DBN based encoding and progress prediction	Job-shop	Simulation	Huang et al. [180]

	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 et al. [99]

Table A.14
Plain NN based approaches in production control.

	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. [181]
93	Dispatching	DQN	FFNN	WIP; util. ratio (1.); min. global time constraints (2.)	Superior	Semiconductor	Simulation	Altenmüller et al. [101]
94	Dispatching	TRPO	FFNN	Min. throughput time	Superior	Semiconductor	Simulation	Kuhnle et al. [182]
95	Dispatching	TRPO	FFNN	Max. utilization, min. lead time	Superior	Semiconductor	Simulation	Kuhnle et al. [183]
96	Dispatching	TRPO	FFNN	Max. utilization, min.throughput/ waiting time	Superior/similar	Semiconductor	Simulation	Kuhnle et al. [100]
97	Dispatching	DQN	FFNN	Max. utilization, min. lead times	Superior	Semiconductor	Simulation	Stricker et al. [184]
98	Dispatching	DDDQN	FFNN	Min. reconfiguration, min. makespan	Superior	Reconfigurable mfg. system	Simulation	Tang and Salonitis [185]
99	Dispatching	MDP	FFNN	Min. average cycle time	–	Re-entrant production	Simulation	Wu et al. [128]
100	Dispatching	DP	FFNN	Minimize total production cost	–	Re-entrant production	Simulation	Zhou et al. [127]
101	Flow control	–	FFNN	Min. makespan, cost, energy consumption	Superior	WIP bounding	Simulation	Danishvar et al. [186]
102	Flow control	DQN	FFNN	High throughput, min. WIP	Superior	WIP bounding	Simulation	Silva and Azevedo [187]
103	Flow-shop scheduling	–	FFNN	Minimize mean tardiness	Superior	Flow-shop	Simulation	Mouelhi-Chibani et al. [105]
104	Job-shop scheduling	BP algorithm	FFNN	Speed up modeling process, raise accuracy	–	Job-shop	Simulation	Bergmann and Stelzer [102]
105	Job-shop scheduling	BP algorithm	FFNN	Imitation of dispatching rule	–	Job-shop	Simulation	Bergmann et al. [103]
106	Job-shop scheduling	DoubleDQN	FFNN	Minimize total tardiness	Superior	Job-shop	Simulation	Luo [104]
107	Job-shop scheduling	DQN	FFNN	Minimize makespan	Superior	Job-shop	Simulation	Moon and Jeong [188]
108	Job-shop scheduling	PPO	FFNN	Max. productivity	–	Job-shop	Simulation	Overbeck et al. [189]
109	Job-shop scheduling	AC	Cong.	Min. makespan and total delay	Superior	Job-shop	Simulation	Zhao and Zhang [106]
110	Job-shop scheduling	REINFORCE	FFNN	Min. mean lateness, tardiness	Superior	Job-shop	Simulation	Zheng et al. [107]
111	Material flow control	Policy gradient	FFNN	Max. profit, min. cost, target deviation	Superior	Copper mining complex	Simulation	Kumar et al. [108]
112	Modular control	PPO	FFNN	Max. throughput	–	Modular production	Simulation	Mayer et al. [190]
113	Order release	A3C, Q-learning	FFNN	Min. tardiness, throughput time	Superior	Two-stage flow-shop	Simulation	Scheckenreither et al. [191]
114	Production and inventory control	ADP	FFNN	Min. total cost per unit	Superior	Dishwasher wire rack production system	Simulation	Wu et al. [109]

Table A.15
Embedded and multi-agent NN approaches in production control.

	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. [113]
116	Flow-shop scheduling	DQN	FFNN	Minimize mean tardiness	Superior	RL dynamically adjust scheduling k1/ k2 values	Flexible flow-shop	Simulation	Heger and Voss [112]
117	Flow-shop scheduling	NEAT	FFNN	Min. total tardiness and makespan	Superior	GA sets NN topology/ hyper-parameters	Flow-shop	Simulation	Lang et al. [192]
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. [110]
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. [111]
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. [193]
121	Job-shop scheduling	TRPO	FFNN	Explainability	Similar	RL scheduling and decision tree based control abstraction	Semicond.	Simulation	Kuhnle et al. [194]

	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. [121]
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. [115]
124	Job-shop scheduling	SA/ FFNN	Min. mean tardiness	–	Agent information exchange	Simultaneous learning with simulated annealing	Job-shop	Simulation	Hammami et al. [114]
125	Job-shop scheduling	DQN/ FFNN	Min. through-put time	Superior	Agent state info., global objective	Central DQN module	Matrix production	Simulation	Hofmann et al. [116]
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. [117]
127	Order dispatching	TD3/ FFNN	Minimum tardiness	Superior	Order bidding mechanism, global objective	Concurrent learning	Job-shop	Simulation	Malus et al. [119]
128	Re-ordering	DQN/ FFNN	Min. cost and decision time	Superior	None	Iterative curriculum learning	Car after paint buffer	Simulation	Gros et al. [118]
129	Routing, dispatchting, scheduling	PPO/ Conv. NN	Min. execution time, max. util. efficiency	Superior	Economic bidding/ global objective	–	Matrix production system	Simulation	May et al. [120]

References

[1] Bueno A, Godinho Filho M, Frank AG. Smart production planning and control in the industry 4.0 context: A systematic literature review. Comput Ind Eng 2020;149:106774. http://dx.doi.org/10.1016/j.cie.2020.106774.

[2] Babiceanu RF, Chen FF. Development and applications of holonic manufacturing systems: A survey. J Intell Manuf 2006;17(1):111–31. http://dx.doi.org/10.1007/s10845-005-5516-y.

[3] Gronauer S, Diepold K. Multi-agent deep reinforcement learning: a survey. Artif Intell Rev 2021. http://dx.doi.org/10.1007/s10462-021-09996-w.

[4] Kang Z, Catal C, Tekinerdogan B. Machine learning applications in production lines: A systematic literature review. Comput Ind Eng 2020;149:106773. http://dx.doi.org/10.1016/j.cie.2020.106773.

[5] Zhang J, Ding G, Zou Y, Qin S, Fu J. Review of job shop scheduling research and its new perspectives under Industry 4.0. J Intell Manuf 2019;30(4):1809–30. http://dx.doi.org/10.1007/s10845-017-1350-2.

[6] Cadavid JPU, Lamouri S, Grabot B, Fortin A. Machine learning in production planning and control: A review of empirical literature. IFAC-PapersOnLine 2019;52(13):385–90. http://dx.doi.org/10.1016/j.ifacol.2019.11.155.

[7] Liao Y, Deschamps F, Loures EdFR, Ramos LFP. Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal. Int J Prod Res 2017;55(12):3609–29. http://dx.doi.org/10.1080/00207543.2017.1308576.

[8] Panzer M, Bender B. Deep reinforcement learning in production systems: a systematic literature review. Int J Prod Res 2021;1–26. http://dx.doi.org/10.1080/00207543.2021.1973138.

[9] Weichert D, Link P, Stoll A, Rüping S, Ihlenfeldt S, Wrobel S. A review of machine learning for the optimization of production processes. Int J Adv Manuf Technol 2019;104(5–8):1889–902. http://dx.doi.org/10.1007/s00170-019-03988-5.

[10] Zhou L, Jiang Z, Geng N, Niu Y, Cui F, Liu K, et al. Production and operations management for intelligent manufacturing: a systematic literature review. Int J Prod Res 2022;60(2):808–46. http://dx.doi.org/10.1080/00207543.2021.2017055.

[11] Rabelo L, Alptekin S, Kiran A. Synergy of artificial neural networks and knowledge-based expert systems for intelligent FMS scheduling. In: 1990 IJCNN international joint conference on neural networks. Vol. 1, San Diego, CA, USA: IEEE; 1990, p. 359–66.

[12] Zhou D, Cherkassky V, Baldwin T, Hong D. Scaling neural network for job-shop scheduling. In: 1990 IJCNN international joint conference on neural networks. Vol. 3, San Diego, CA, USA: IEEE; 1990, p. 889–94. http://dx.doi.org/10.1109/IJCNN.1990.137947.

[13] Baker AD. A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull. J Manuf Syst 1998;17(4):297–320. http://dx.doi.org/10.1016/S0278-6125(98)80077-0.

[14] Zhang H-C, Huang SH. Applications of neural networks in manufacturing: a state-of-the-art survey. Int J Prod Res 1995;33(3):705–28. http://dx.doi.org/10.1080/00207549508930175.

[15] Garetti M, Taisch M. Neural networks in production planning and control. Prod Plan Control 1999;10(4):324–39. http://dx.doi.org/10.1080/095372899233082.

[16] De Modesti PH, Carvalhar Fernandes E, Borsato M. Production planning and scheduling using machine learning and data science processes. In: Säfsten K, Elgh F, editors. Advances in transdisciplinary engineering. IOS Press; 2020, http://dx.doi.org/10.3233/ATDE200153.

[17] Çaliş B, Bulkan S. A research survey: review of AI solution strategies of job shop scheduling problem. J Intell Manuf 2015;26(5):961–73. http://dx.doi.org/10.1007/s10845-013-0837-8.

[18] Bertolini M, Mezzogori D, Neroni M, Zammori F. Machine Learning for industrial applications: A comprehensive literature review. Expert Syst Appl 2021;175:114820. http://dx.doi.org/10.1016/j.eswa.2021.114820.

[19] Bitran GR, Haas EA, Hax AC. Hierarchical production planning: A two-stage system. Oper Res 1982;30(2):232–51. http://dx.doi.org/10.1287/opre.30.2.232.

[20] Derigent W, Cardin O, Trentesaux D. Industry 4.0: contributions of holonic manufacturing control architectures and future challenges. J Intell Manuf 2021;32(7):1797–818. http://dx.doi.org/10.1007/s10845-020-01532-x.

[21] Rojas RA, Rauch E. From a literature review to a conceptual framework of enablers for smart manufacturing control. Int J Adv Manuf Technol 2019;104(1–4):517–33. http://dx.doi.org/10.1007/s00170-019-03854-4.

[22] Lee J-H, Kim C-O. Multi-agent systems applications in manufacturing systems and supply chain management: a review paper. Int J Prod Res 2008;46(1):233–65. http://dx.doi.org/10.1080/00207540701441921.

[23] Monostori L, Valckenaers P, Dolgui A, Panetto H, Brdys M, Csáji BC. Cooperative control in production and logistics. IFAC Proc Vol 2014;47(3):4246–65. http://dx.doi.org/10.3182/20140824-6-ZA-1003.01026.

[24] Herrera M, Pérez-Hernández M, Kumar Parlikad A, Izquierdo J. Multi-agent systems and complex networks: Review and applications in systems engineering. Processes 2020;8(3):312. http://dx.doi.org/10.3390/pr8030312.

[25] Zipfel A, Braunreuther S, Reinhart G. Approach for a production planning and control system in value-adding networks. Proc CIRP 2019;81:1195–200. http://dx.doi.org/10.1016/j.procir.2019.03.291.

[26] Ramsauer C. Dezentrale PPS-systeme. Wiesbaden: Gabler Verlag; 1997, http://dx.doi.org/10.1007/978-3-322-89084-9.

[27] Kagermann H, Wahlster W, Helbig J. Recommendations for implementing the strategic initiative INDUSTRIE 4.0 – securing the future of german manufacturing industry. Acatech - National Academy of Science and Engineering; 2013.

[28] Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. 2017.

[29] Ding Y, Ma L, Ma J, Suo M, Tao L, Cheng Y, et al. Intelligent fault diagnosis for rotating machinery using deep Q-network based health state classification: A deep reinforcement learning approach. Adv Eng Inf 2019;42:100977. http://dx.doi.org/10.1016/j.aei.2019.100977.

[30] Mehlig B. Machine learning with neural networks: an introduction for scientists and engineers. first ed.. Cambridge University Press; 2021, http://dx.doi.org/10.1017/9781108860604.

[31] Arunraj NS, Ahrens D. A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. Int J Prod Econ 2015;170:321–35. http://dx.doi.org/10.1016/j.ijpe.2015.09.039.

[32] Csáji BC, Monostori L, Kádár B. Reinforcement learning in a distributed market-based production control system. Adv Eng Inf 2006;20(3):279–88. http://dx.doi.org/10.1016/j.aei.2006.01.001.

[33] Kádár B, Monostori L, Csáji B. Adaptive approaches to increase the performance of production control systems. In: Proceedings of 36th CIRP ISMS. 2003, p. 305–12.

[34] Patel M, Honavar V, Balakrishnan K, editors. Advances in the evolutionary synthesis of intelligent agents. Cambridge, Mass: MIT Press; 2001.

[35] Ueda K, Markus A, Monostori L, Kals H, Arai T. Emergent synthesis methodologies for manufacturing. CIRP Annals 2001;50(2):535–51. http://dx.doi.org/10.1016/S0007-8506(07)62994-1.

[36] Zhang W, Dietterich TG. High-performance job-shop scheduling with a time-delay TD network. In: Proceedings of the 14th international joint conference on artificial intelligence. 1995, p. 1024–30.

[37] Zhang W, Dietterich TG. A reinforcement learning approach to job-shop scheduling. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence. 1995, p. 1114–20.

[38] Riedmiller S, Riedmiller M. A neural reinforcement learning approach to learn local dispatching policies in production scheduling. In: Proceedings of the 16th international joint conference on artificial intelligence. Vol. 2, 1999, p. 764–9.

[39] Monostori L, Csáji B, Kádár B. Adaptation and learning in distributed production control. CIRP Annals 2004;53(1):349–52. http://dx.doi.org/10.1016/S0007-8506(07)60714-8.

[40] Aggarwal CC. Neural networks and deep learning: A textbook. Cham, Switzerland: Springer; 2018, p. 459–92.

[41] Dorri A, Kanhere SS, Jurdak R. Multi-agent systems: A survey. IEEE Access 2018;6:28573–93. http://dx.doi.org/10.1109/ACCESS.2018.2831228.

[42] Beigi A, Mozayani N. Dialogue strategy for horizontal communication in MAS organization. Comput Math Organ Theory 2016;22(2):161–83. http://dx.doi.org/10.1007/s10588-015-9201-1.

[43] Bongaerts L, Monostori L, McFarlane D, Kádár B. Hierarchy in distributed shop floor control. Comput Ind 2000;43(2):123–37. http://dx.doi.org/10.1016/S0166-3615(00)00062-2.

[44] Hoen PJ, Tuyls K, Panait L, Luke S, La Poutré JA. An Overview of Cooperative and Competitive Multiagent Learning. In: Tuyls K, Hoen PJ, Verbeeck K, Sen S, editors. Learning and adaption in multi-agent systems. Vol. 3898, Berlin, Heidelberg: Springer Berlin Heidelberg; 2006, p. 1–46. http://dx.doi.org/10.1007/11691839_1.

[45] Tranfield D, Denyer D, Smart P. Towards a methodology for developing evidence-informed management knowledge by means of systematic review. Br J Manag 2003;14(3):207–22. http://dx.doi.org/10.1111/1467-8551.00375.

[46] Thomé AMT, Scavarda LF, Scavarda AJ. Conducting systematic literature review in operations management. Prod Plan Control 2016;27(5):408–20. http://dx.doi.org/10.1080/09537287.2015.1129464.

[47] Petticrew M, Roberts H. Why do we need systematic reviews?. Oxford, UK: Blackwell Publishing Ltd; 2006, http://dx.doi.org/10.1002/9780470754887.ch1.

[48] Brocke J, Simons A, Niehaves B, Riemer K, Plattfaut R, Cleven A. Reconstructing the giant: On the importance of rigour in documenting the literature search process. In: Proceedings of the 17th European conference on information systems (ECIS). 2009.

[49] Cooper H. Organizing knowledge syntheses: A taxonomy of literature reviews. Knowl Soc 1988;1:104–26.
[50] Light RJ, Pillemer DB. Summing up: the science of reviewing research. Cambridge, Mass: Harvard University Press; 1984.
[51] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing atari with deep reinforcement learning. 2013, p. 1–9.
[52] Webster J, Watson RT. Analyzing the past to prepare for the future: writing a literature review. MIS Q 2002;26(2):13–23, Publisher: Management Information Systems Research Center, University of Minnesota.
[53] Chen T. Embedding a back propagation network into fuzzy c-means for estimating job cycle time: wafer fabrication as an example. J Ambient Intell Humaniz Comput 2016;7(6):789–800. http://dx.doi.org/10.1007/s12652-015-0336-1.
[54] Gelders LF, Van Wassenhove LN. Production planning: a review. European J Oper Res 1981;7(2):101–10. http://dx.doi.org/10.1016/0377-2217(81)90271-X.
[55] Wu C-X, Liao M-H, Karatas M, Chen S-Y, Zheng Y-J. Real-time neural network scheduling of emergency medical mask production during COVID-19. Appl Soft Comput 2020;97:106790. http://dx.doi.org/10.1016/j.asoc.2020.106790.
[56] Marchesano MG, Guizzi G, Santillo LC, Vespoli S. Dynamic scheduling in a flow shop using deep reinforcement learning. In: Dolgui A, Bernard A, Lemoine D, von Cieminski G, Romero D, editors. Advances in production management systems. artificial intelligence for sustainable and resilient production systems. Vol. 630, Cham: Springer International Publishing; 2021, p. 152–60. http://dx.doi.org/10.1007/978-3-030-85874-2_16.
[57] Han B-A, Yang J-J. Research on adaptive job shop scheduling problems based on dueling double DQN. IEEE Access 2020;8:186474–95. http://dx.doi.org/10.1109/ACCESS.2020.3029868.
[58] Lin C-C, Deng D-J, Chih Y-L, Chiu H-T. Smart manufacturing scheduling with edge computing using multiclass deep q network. IEEE Trans Ind Inf 2019;15(7):4276–84. http://dx.doi.org/10.1109/TII.2019.2908210.
[59] Hubbs CD, Li C, Sahinidis NV, Grossmann IE, Wassick JM. A deep reinforcement learning approach for chemical production scheduling. Comput Chem Eng 2020;141:106982. http://dx.doi.org/10.1016/j.compchemeng.2020.106982.
[60] Hu L, Liu Z, Hu W, Wang Y, Tan J, Wu F. Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. J Manuf Syst 2020;55:1–14. http://dx.doi.org/10.1016/j.jmsy.2020.02.004.
[61] Leng J, Jin C, Vogl A, Liu H. Deep reinforcement learning for a color-batching resequencing problem. J Manuf Syst 2020;56:175–87. http://dx.doi.org/10.1016/j.jmsy.2020.06.001.
[62] Şenyiğit E, Atici U. Artificial neural network models for lot-sizing problem: a case study. Neural Comput Appl 2013;22(6):1039–47. http://dx.doi.org/10.1007/s00521-012-0863-z.
[63] Kumar H, Giri S. A neural network-based algorithm for flow shop scheduling problems under fuzzy environment. Int J Process Manag Benchmarking 2020;10(2):282. http://dx.doi.org/10.1504/IJPMB.2020.106144.
[64] Ramanan TR, Sridharan R, Shashikant KS, Haq AN. An artificial neural network based heuristic for flow shop scheduling problems. J Intell Manuf 2011;22(2):279–88. http://dx.doi.org/10.1007/s10845-009-0287-5.
[65] Sim MH, Low MYH, Chong CS, Shakeri M. Job shop scheduling problem neural network solver with dispatching rules. In: 2020 IEEE international conference on industrial engineering and engineering management (IEEM). Singapore, Singapore: IEEE; 2020, p. 514–8.
[66] Zang Z, Wang W, Song Y, Lu L, Li W, Wang Y, et al. Hybrid deep neural network scheduler for job-shop problem based on convolution two-dimensional transformation. Comput Intell Neurosci 2019;2019:1–19. http://dx.doi.org/10.1155/2019/7172842.
[67] Zhang Z, Guan ZL, Zhang J, Xie X. A novel job-shop scheduling strategy based on particle swarm optimization and neural network. Int J Simul Model 2019;18(4):699–707. http://dx.doi.org/10.2507/IJSIMM18(4)CO18.
[68] Lan Y, Liu Y, Sun G. An approximation-based approach for fuzzy multi-period production planning problem with credibility objective. Appl Math Model 2010;34(11):3202–15. http://dx.doi.org/10.1016/j.apm.2010.02.013.
[69] Wen H, Liu M, Liu C, Liu C. Remanufacturing production planning with compensation function approximation method. Appl Math Comput 2015;256:742–53. http://dx.doi.org/10.1016/j.amc.2015.01.070.
[70] Wen H, Hou S, Liu Z, Liu Y. An optimization algorithm for integrated remanufacturing production planning and scheduling system. Chaos Solitons Fractals 2017;105:69–76. http://dx.doi.org/10.1016/j.chaos.2017.10.012.
[71] Zhang HP. Optimization of remanufacturing production scheduling considering uncertain factors. Int J Simul Model 2019;18(2):344–54. http://dx.doi.org/10.2507/IJSIMM18(2)CO8.
[72] Sobottka T, Kamhuber F, Faezirad M, Sihn W. Potential for machine learning in optimized production planning with hybrid simulation. Proc Manuf 2019;39:1844–53. http://dx.doi.org/10.1016/j.promfg.2020.01.254.
[73] Kumar A, Dimitrakopoulos R. Production scheduling in industrial mining complexes with incoming new information using tree search and deep reinforcement learning. Appl Soft Comput 2021;110:107644. http://dx.doi.org/10.1016/j.asoc.2021.107644.
[74] Hammami Z, Mouelhi W, Said LB. 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: IEEE; 2015, p. 494–501. http://dx.doi.org/10.1109/ISKE.2015.79.
[75] Liu C-L, Chang C-C, Tseng C-J. Actor-critic deep reinforcement learning for solving job shop scheduling problems. IEEE Access 2020;8:71752–62. http://dx.doi.org/10.1109/ACCESS.2020.2987820.
[76] Baer S, Bakakeu J, Meyes R, Meisen T. 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: IEEE; 2019, p. 22–5.
[77] Baer S, Turner D, Mohanty PK, Samsonov V, Bakakeu R, Meisen T. 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; 2020, p. 1–9.
[78] Park I-B, Huh J, Kim J, Park J. A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities. IEEE Trans Autom Sci Eng 2020;1–12. http://dx.doi.org/10.1109/TASE.2019.2956762.
[79] Lee S, Cho Y, Lee YH. Injection mold production sustainable scheduling using deep reinforcement learning. Sustainability 2020;12(20):8718. http://dx.doi.org/10.3390/su12208718.
[80] Pol S, Baer S, Turner D, Samsonov V, Meisen T. 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. 2021, p. 515–26. http://dx.doi.org/10.5220/0010455805150526, Online.
[81] Zhou T, Tang D, Zhu H, Zhang Z. Multi-agent reinforcement learning for online scheduling in smart factories. Robot Comput-Integr Manuf 2021;72:102202. http://dx.doi.org/10.1016/j.rcim.2021.102202.
[82] Worapradya K, Thanakijkasem P. Proactive scheduling for steelmaking-continuous casting plant with uncertain machine breakdown using distribution-based robustness and decomposed artificial neural network. Asia-Pac J Oper Res 2015;32(02):1550010. http://dx.doi.org/10.1142/S0217595915500104.
[83] Onaran E, Yanık S. Predicting cycle times in textile manufacturing using artificial neural network. In: Kahraman C, Cebi S, Cevik Onar S, Oztaysi B, Tolga AC, Sari IU, editors. Intelligent and fuzzy techniques in big data analytics and decision making. Vol. 1029, Cham: Springer International Publishing; 2020, p. 305–12. http://dx.doi.org/10.1007/978-3-030-23756-1_38.
[84] Cao H, Ji X. Prediction of garment production cycle time based on a neural network. Fibres Text East Eur 2021;29:8–12. http://dx.doi.org/10.5604/01.3001.0014.5036.
[85] Glavan M, Gradišar D, Strmčnik S, Mušič G. Production modelling for holistic production control. Simul Model Pract Theory 2013;30:1–20. http://dx.doi.org/10.1016/j.simpat.2012.07.010.
[86] Windler T, Busse J, Rieck J. One month-ahead electricity price forecasting in the context of production planning. J Clean Prod 2019;238:117910. http://dx.doi.org/10.1016/j.jclepro.2019.117910.
[87] Huang PB, Chen JC, Yu C-W, Chen C-C. Integration of simulation and neural network in forecasting the throughput for TFT-LCD colour filter fabs. Int J Comput Integr Manuf 2016;29(3):298–308. http://dx.doi.org/10.1080/0951192X.2015.1032356.
[88] Pham QT, Phan TKD. Apply neural network for improving production planning at Samarang petrol mine. Int J Intell Comput Cybern 2016;9(2):126–43. http://dx.doi.org/10.1108/IJICC-09-2015-0032.
[89] Silva C, Ribeiro V, Coelho P, Magalhães V, Neto P. Job shop flow time prediction using neural networks. Proc Manuf 2017;11:1767–73. http://dx.doi.org/10.1016/j.promfg.2017.07.309.
[90] Karaoglan AD, Karademir O. Flow time and product cost estimation by using an artificial neural network (ANN): A case study for transformer orders. Eng Econ 2017;62(3):272–92. http://dx.doi.org/10.1080/0013791X.2016.1185808.
[91] Kramer KJ, Wagner C, Schmidt M. Machine learning-supported planning of lead times in job shop manufacturing. In: Lalic B, Majstorovic V, Marjanovic U, von Cieminski G, Romero D, editors. Advances in production management systems. the path to digital transformation and innovation of production management systems. IFIP advances in information and communication technology, Vol. 591, Cham: Springer International Publishing; 2020, p. 363–70. http://dx.doi.org/10.1007/978-3-030-57993-7_41.
[92] Göppert A, Mohring L, Schmitt RH. Predicting performance indicators with ANNs for AI-based online scheduling in dynamically interconnected assembly systems. Prod Eng 2021;15(5):619–33. http://dx.doi.org/10.1007/s11740-021-01057-z.

[93] Lai X, Shui H, Ni J. 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; 2018, http://dx.doi.org/10.1115/MSEC2018-6678.

[94] Schneckenreither M, Haeussler S, Gerhold C. Order release planning with predictive lead times: a machine learning approach. Int J Prod Res 2021;59(11):3285–303. http://dx.doi.org/10.1080/00207543.2020.1859634.

[95] Mezzogori D, Romagnoli G, Zammori F. Deep learning and WLC: how to set realistic delivery dates in high variety manufacturing systems. IFAC-PapersOnLine 2019;52(13):2092–7. http://dx.doi.org/10.1016/j.ifacol.2019.11.514.

[96] Sadiq SS, Abdulazeez AM, Haron H. Solving multi-objective master production scheduling model of kalak refinery system using hybrid evolutionary imperialist competitive algorithm. J Comput Sci 2020;16(2):137–49. http://dx.doi.org/10.3844/jcssp.2020.137.149.

[97] Huang J, Chang Q, Arinez J. Product completion time prediction using a hybrid approach combining deep learning and system model. J Manuf Syst 2020;57:311–22. http://dx.doi.org/10.1016/j.jmsy.2020.10.006.

[98] Fang W, Guo Y, Liao W, Ramani K, Huang S. Big data driven jobs remaining time prediction in discrete manufacturing system: a deep learning-based approach. Int J Prod Res 2020;58(9):2751–66. http://dx.doi.org/10.1080/00207543.2019.1602744.

[99] Morariu C, Borangiu T. Time series forecasting for dynamic scheduling of manufacturing processes. In: 2018 IEEE international conference on automation, quality and testing, robotics (AQTR). Cluj-Napoca: IEEE; 2018, p. 1–6.

[100] Kuhnle A, Kaiser J-P, Theiß F, Stricker N, Lanza G. Designing an adaptive production control system using reinforcement learning. J Intell Manuf 2021;32(3):855–76. http://dx.doi.org/10.1007/s10845-020-01612-y.

[101] Altenmüller T, Stüker T, Waschneck B, Kuhnle A, Lanza G. Reinforcement learning for an intelligent and autonomous production control of complex jobshops under time constraints. Prod Eng 2020;14(3):319–28. http://dx.doi.org/10.1007/s11740-020-00967-8.

[102] Bergmann S, Stelzer S. Approximation of dispatching rules in manufacturing control using artificial neural networks. In: 2011 IEEE workshop on principles of advanced and distributed simulation. Nice, France: IEEE; 2011, p. 1–8.

[103] Bergmann S, Stelzer S, Strassburger S. On the use of artificial neural networks in simulation-based manufacturing control. J Simul 2014;8(1):76–90. http://dx.doi.org/10.1057/jos.2013.6.

[104] Luo S. Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. Appl Soft Comput 2020;91:106208. http://dx.doi.org/10.1016/j.asoc.2020.106208.

[105] Mouelhi-Chibani W, Pierreval H. Training a neural network to select dispatching rules in real time. Comput Ind Eng 2010;58(2):249–56. http://dx.doi.org/10.1016/j.cie.2009.03.008.

[106] Zhao Y, Zhang H. Application of machine learning and rule scheduling in a job-shop production control system. Int J Simul Model 2021;20(2):410–21. http://dx.doi.org/10.2507/IJSIMM20-2-CO10.

[107] Zheng S, Gupta C, Serita S. Manufacturing dispatching using reinforcement and transfer learning. In: Brefeld U, Fromont E, Hotho A, Knobbe A, Maathuis M, Robardet C, editors. Proceedings of joint European conference on machine learning and knowledge discovery in databases. 2020, p. 655–71. http://dx.doi.org/10.1007/978-3-030-46133-1_39.

[108] Kumar A, Dimitrakopoulos R, Maulen M. Adaptive self-learning mechanisms for updating short-term production decisions in an industrial mining complex. J Intell Manuf 2020;31(7):1795–811. http://dx.doi.org/10.1007/s10845-020-01562-5.

[109] Wu H, Evans G, Bae K-H. Production control in a complex production system using approximate dynamic programming. Int J Prod Res 2016;54(8):2419–32. http://dx.doi.org/10.1080/00207543.2015.1086035.

[110] Lang S, Behrendt F, Lanzerath N, Reggelin T, Muller M. 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; 2020, p. 3057–68. http://dx.doi.org/10.1109/WSC48552.2020.9383997.

[111] Luo S, Zhang L, Fan Y. Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. Comput Ind Eng 2021;159:107489. http://dx.doi.org/10.1016/j.cie.2021.107489.

[112] Heger J, Voss T. Dynamically adjusting the k -values of the ATCS rule in a flexible flow shop scenario with reinforcement learning. Int J Prod Res 2021;1–15. http://dx.doi.org/10.1080/00207543.2021.1943762.

[113] Kim H, Lim D-E, Lee S. Deep learning-based dynamic scheduling for semiconductor manufacturing with high uncertainty of automated material handling system capability. IEEE Trans Semicond Manuf 2020;33(1):13–22. http://dx.doi.org/10.1109/TSM.2020.2965293.

[114] Hammami Z, Mouelhi W, Ben Said L. On-line self-adaptive framework for tailoring a neural-agent learning model addressing dynamic real-time scheduling problems. J Manuf Syst 2017;45:97–108. http://dx.doi.org/10.1016/j.jmsy.2017.08.003.

[115] Dittrich M-A, Fohlmeister S. Cooperative multi-agent system for production control using reinforcement learning. CIRP Annals 2020;69(1):389–92. http://dx.doi.org/10.1016/j.cirp.2020.04.005.

[116] Hofmann C, Krahe C, Stricker N, Lanza G. Autonomous production control for matrix production based on deep Q-learning. Proc CIRP 2020;88:25–30. http://dx.doi.org/10.1016/j.procir.2020.05.005.

[117] Waschneck B, Reichstaller A, Belzner L, Altenmuller T, Bauernhansl T, Knapp A, et al. Deep reinforcement learning for semiconductor production scheduling. In: 2018 29th annual SEMI advanced semiconductor manufacturing conference (ASMC). NY, USA: IEEE; 2018, p. 301–6. http://dx.doi.org/10.1109/ASMC.2018.8373191.

[118] Gros TP, Gros J, Wolf V. Real-time decision making for a car manufacturing process using deep reinforcement learning. In: 2020 winter simulation conference (WSC). Orlando, FL, USA: IEEE; 2020, p. 3032–44. http://dx.doi.org/10.1109/WSC48552.2020.9383884.

[119] Malus A, Kozjek D, Vrabič R. Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. CIRP Ann 2020;69(1):397–400. http://dx.doi.org/10.1016/j.cirp.2020.04.001.

[120] May MC, Kiefer L, Kuhnle A, Stricker N, Lanza G. Decentralized multi-agent production control through economic model bidding for matrix production systems. Proc CIRP 2021;96:3–8. http://dx.doi.org/10.1016/j.procir.2021.01.043.

[121] Shin M, Ryu K, Jung M. Reinforcement learning approach to goal-regulation in a self-evolutionary manufacturing system. Expert Syst Appl 2012;39(10):8736–43. http://dx.doi.org/10.1016/j.eswa.2012.01.207.

[122] Nickerson R, Varshney U, Muntermann J. A method for taxonomy development and its application in information systems. Eur J Inf Syst 2013;22. http://dx.doi.org/10.1057/ejis.2012.26.

[123] Bellman R. Dynamic programming. Science 1966;153(3731):34–7.

[124] Shi D, Fan W, Xiao Y, Lin T, Xing C. Intelligent scheduling of discrete automated production line via deep reinforcement learning. Int J Prod Res 2020;58(11):3362–80. http://dx.doi.org/10.1080/00207543.2020.1717008.

[125] Hasselt Hv, Guez A, Silver D. Deep reinforcement learning with double Q-learning. In: Proceedings of the thirtieth AAAI conference on artificial intelligence. AAAI '16, AAAI Press; 2016, p. 2094–100.

[126] Rummukainen H, Nurminen JK. Practical reinforcement learning -experiences in lot scheduling application. IFAC-PapersOnLine 2019;52(13):1415–20. http://dx.doi.org/10.1016/j.ifacol.2019.11.397.

[127] Zhou F-Y, Wu C-H, Yu C-J. Dynamic dispatching for re-entrant production lines — A deep learning approach. In: 2017 13th IEEE conference on automation science and engineering (CASE). Xi'an: IEEE; 2017, p. 1026–31. http://dx.doi.org/10.1109/COASE.2017.8256238.

[128] Wu C-H, Zhou F-Y, Tsai C-K, Yu C-J, Dauzère-Pérès S. A deep learning approach for the dynamic dispatching of unreliable machines in re-entrant production systems. Int J Prod Res 2020;58(9):2822–40. http://dx.doi.org/10.1080/00207543.2020.1727041.

[129] Wang X, Zhang L, Lin T, Zhao C, Wang K, Chen Z. Solving job scheduling problems in a resource preemption environment with multi-agent reinforcement learning. Robot Comput-Integr Manuf 2022;77:102324. http://dx.doi.org/10.1016/j.rcim.2022.102324.

[130] Kim Y, Lee S, Son J, Bae H, Chung BD. Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system. J Manuf Syst 2020;57:440–50. http://dx.doi.org/10.1016/j.jmsy.2020.11.004.

[131] Pach C, Berger T, Bonte T, Trentesaux D. ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. Comput Ind 2014;65(4):706–20. http://dx.doi.org/10.1016/j.compind.2014.02.005.

[132] Minguillon FE, Lanza G. Coupling of centralized and decentralized scheduling for robust production in agile production systems. Proc CIRP 2019;79:385–90. http://dx.doi.org/10.1016/j.procir.2019.02.099, 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy.

[133] Schwung D, Yuwono S, Schwung A, Ding SX. Decentralized learning of energy optimal production policies using PLC-informed reinforcement learning. Comput Chem Eng 2021;152:107382. http://dx.doi.org/10.1016/j.compchemeng.2021.107382.

[134] Teichmann M, Ullrich A, Kotarski D, Gronau N. Facing the demographic change – recommendations for designing learning factories as age-appropriate teaching-learning environments for older blue-collar workers. SSRN Electron J 2021. http://dx.doi.org/10.2139/ssrn.3858716.

[135] Athanasopoulou L, Papacharalampopoulos A, Stavropoulos P. Context awareness system in the use phase of a smart mobility platform: A vision system for a light-weight approach. Proc CIRP 2020. http://dx.doi.org/10.1002/9780470754887.

[136] Zhao T, Zhang H. A new method to improve the generalization ability of neural networks: A case study of nuclear mass training. Nuclear Phys A 2022;1021:122420. http://dx.doi.org/10.1016/j.nuclphysa.2022.122420.
[137] Jones D. Artificial cognitive systems: the next generation of the digital twin. an opinion. Digit Twin 2021;1:3. http://dx.doi.org/10.12688/digitaltwin.17440.2.
[138] Zhou L, Zhang L, Horn BK. Deep reinforcement learning-based dynamic scheduling in smart manufacturing. Proc CIRP 2020;93:383–8. http://dx.doi.org/10.1016/j.procir.2020.05.163.
[139] Zhu H, Li M, Tang Y, Sun Y. A deep-reinforcement-learning-based optimization approach for real-time scheduling in cloud manufacturing. IEEE Access 2020;8:9987–97. http://dx.doi.org/10.1109/ACCESS.2020.2964955.
[140] Rouhani S, Fathian M, Jafari M, Akhavan P. Solving the problem of flow shop scheduling by neural network approach. In: Zavoral F, Yaghob J, Pichappan P, El-Qawasmeh E, editors. Networked digital technologies. Vol. 88, Berlin, Heidelberg: Springer Berlin Heidelberg; 2010, p. 172–83. http://dx.doi.org/10.1007/978-3-642-14306-9_18.
[141] Gannouni A, Samsonov V, Behery M, Meisen T, Lakemeyer G. 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: IEEE; 2020, p. 2640–7. http://dx.doi.org/10.1109/SMC42975.2020.9282869.
[142] Fnaiech N, Hammami H, Yahyaoui A, Varnier C, Fnaiech F, Zerhouni N. 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: IEEE; 2012, p. 5535–41.
[143] Groth M, Freier P, Schumann M. 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; 2021.
[144] Kardos C, Laflamme C, Gallina V, Sihn W. Dynamic scheduling in a job-shop production system with reinforcement learning. Proc CIRP 2021;97:104–9. http://dx.doi.org/10.1016/j.procir.2020.05.210.
[145] Park J, Chun J, Kim SH, Kim Y, Park J. Learning to schedule job-shop problems: representation and policy learning using graph neural network and reinforcement learning. Int J Prod Res 2021;59(11):3360–77. http://dx.doi.org/10.1080/00207543.2020.1870013.
[146] Wang L, Hu X, Wang Y, Xu S, Ma S, Yang K, et al. Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning. Comput Netw 2021;190:107969. http://dx.doi.org/10.1016/j.comnet.2021.107969.
[147] Zhou T, Tang D, Zhu H, Wang L. Reinforcement learning with composite rewards for production scheduling in a smart factory. IEEE Access 2021;9:752–66. http://dx.doi.org/10.1109/ACCESS.2020.3046784.
[148] Chen B, Wan J, Lan Y, Imran M, Li D, Guizani N. Improving cognitive ability of edge intelligent IIoT through machine learning. IEEE Netw 2019;33(5):61–7. http://dx.doi.org/10.1109/MNET.001.1800505.
[149] Xie S, Zhang T, Rose O. Online Single Machine Scheduling Based on Simulation and Reinforcement Learning. In: 18. ASIM fachtagung simulation in produktion und logistik. Chemnitz; 2019, p. 1–10.
[150] Seito T, Munakata S. 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; 2020, p. 766–72. http://dx.doi.org/10.5220/0009095207660772.
[151] Palombarini JA, Martínez EC. Closed-loop rescheduling using deep reinforcement learning. IFAC-PapersOnLine 2019;52(1):231–6. http://dx.doi.org/10.1016/j.ifacol.2019.06.067.
[152] Palombarini JA, Martinez EC. Automatic generation of rescheduling knowledge in socio-technical manufacturing systems using deep reinforcement learning. In: 2018 IEEE biennial congress of Argentina. San Miguel de Tucumán, Argentina: IEEE; 2018, p. 1–8. http://dx.doi.org/10.1109/ARGENCON.2018.8646172.
[153] Madureira A, Santos JM, Gomes S, Cunha B, Pereira J, Pereira I. Manufacturing rush orders rescheduling: a supervised learning approach. In: 2014 sixth world congress on nature and biologically inspired computing (NaBIC 2014). Porto, Portugal: IEEE; 2014, p. 299–304.
[154] Dong T, Xue F, Xiao C, Li J. Task scheduling based on deep reinforcement learning in a cloud manufacturing environment. Concurr Comput: Pract Exper 2020;32(11). http://dx.doi.org/10.1002/cpe.5654.
[155] Gahm C, Uzunoglu A, Wahl S, Ganschinietz C, Tuma A. Applying machine learning for the anticipation of complex nesting solutions in hierarchical production planning. European J Oper Res 2022;296(3):819–36. http://dx.doi.org/10.1016/j.ejor.2021.04.006.
[156] Zhao FQ, Zou JH, Yang YH. A hybrid approach based on artificial neural network (ANN) and differential evolution (DE) for job-shop scheduling problem. Appl Mech Mater 2010;26–28:754–7. http://dx.doi.org/10.4028/www.scientific.net/AMM.26-28.754.
[157] Mashhadi F, Salinas Monroy SA. Deep learning for optimal resource allocation in IoT-enabled additive manufacturing. In: 2020 IEEE 6th world forum on Internet of Things. LA, USA: IEEE; 2020, p. 1–6. http://dx.doi.org/10.1109/WF-IoT48130.2020.9221038.
[158] Lan Y, Zhao R, Tang W. Minimum risk criterion for uncertain production planning problems. Comput Ind Eng 2011;61(3):591–9. http://dx.doi.org/10.1016/j.cie.2011.04.014.
[159] Feng X, Yuan G. Optimizing two-stage fuzzy multi-product multi-period production planning problem. Int J Inf 2011;6(6):1879–93.
[160] Wang C, Jiang P. Manifold learning based rescheduling decision mechanism for recessive disturbances in RFID-driven job shops. J Intell Manuf 2018;29(7):1485–500. http://dx.doi.org/10.1007/s10845-016-1194-1.
[161] Yang S, Xu Z. Intelligent scheduling and reconfiguration via deep reinforcement learning in smart manufacturing. Int J Prod Res 2021;1–18. http://dx.doi.org/10.1080/00207543.2021.1943037.
[162] Liu Y, Zhang R, Wang M, Zhu X. A decomposition-based two-stage optimization algorithm for single machine scheduling problems with deteriorating jobs. Math Probl Eng 2015;2015:1–8. http://dx.doi.org/10.1155/2015/340769.
[163] Baeza Serrato R. Stochastic plans in SMEs: A novel multidimensional fuzzy logic system (mFLS) approach. Ingeniería E Investigación 2018;38(2):70–8. http://dx.doi.org/10.15446/ing.investig.v38n2.65357.
[164] Chakravorty S, Nagarur NN. 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; 2020, p. 1–5.
[165] Jain S, Lechevalier D, Narayanan A. Towards smart manufacturing with virtual factory and data analytics. In: 2017 winter simulation conference (WSC). Las Vegas, NV: IEEE; 2017, p. 3018–29.
[166] Pusnik M, Sucic B, Podgornik A, AL-Mansour F, Vuk T. Net fitting based production planning and decision support system for energy intensive industries. In: 2014 IEEE international energy conference. Cavtat, Croatia: IEEE; 2014, p. 1236–42.
[167] Ramos D, Faria P, Vale Z, Correia R. Short time electricity consumption forecast in an industry facility. IEEE Trans Ind Appl 2021;1. http://dx.doi.org/10.1109/TIA.2021.3123103.
[168] Azab E, Nafea M, Shihata LA, Mashaly M. A machine-learning-assisted simulation approach for incorporating predictive maintenance in dynamic flow-shop scheduling. Appl Sci 2021;11(24):11725. http://dx.doi.org/10.3390/app112411725.
[169] Sajko N, Kovacic S, Ficko M, Palcic I, Klancnik S. Management and production engineering review. 2020, http://dx.doi.org/10.24425/MPER.2020.134931, Publisher: Production Engineering Committee of the Polish Academy of Sciences.
[170] Burduk A, Chlebus T, Waszkowski R. 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. Vol. 637, Cham: Springer International Publishing; 2018, p. 179–88. http://dx.doi.org/10.1007/978-3-319-64465-3_18.
[171] Wang C, Jiang P. Deep neural networks based order completion time prediction by using real-time job shop RFID data. J Intell Manuf 2019;30(3):1303–18. http://dx.doi.org/10.1007/s10845-017-1325-3.
[172] Yamashiro H, Nonaka H. Estimation of processing time using machine learning and real factory data for optimization of parallel machine scheduling problem. Oper Res Perspect 2021;8:100196. http://dx.doi.org/10.1016/j.orp.2021.100196.
[173] Stauder M, Kühl N. 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. Flex Serv Manuf J 2021. http://dx.doi.org/10.1007/s10696-021-09430-x.
[174] May MC, Behnen L, Holzer A, Kuhnle A, Lanza G. Multi-variate time-series for time constraint adherence prediction in complex job shops. Proc CIRP 2021;103:55–60. http://dx.doi.org/10.1016/j.procir.2021.10.008.
[175] Both C, Dimitrakopoulos R. Applied machine learning for geometallurgical throughput prediction—A case study using production data at the tropicana gold mining complex. Minerals 2021;11(11):1257. http://dx.doi.org/10.3390/min11111257.
[176] Gallina V, Lingitz L, Breitschopf J, Zudor E, Sihn W. Work in progress level prediction with long short-term memory recurrent neural network. Proc Manuf 2021;54:136–41. http://dx.doi.org/10.1016/j.promfg.2021.07.047.
[177] Morariu C, Morariu O, Răileanu S, Borangiu T. Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. Comput Ind 2020;120:103244. http://dx.doi.org/10.1016/j.compind.2020.103244.
[178] Han Z, Zhang Q, Jiang Y, Duan B. Research on the production scheduling method of a semiconductor packaging test based with the clustering method. Int J Inf Syst Supply Chain Manag 2019;12(2):36–56. http://dx.doi.org/10.4018/IJISSCM.2019040103.
[179] Hu S, Zhou L. 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: IEEE; 2020, p. 1111–4.

[180] Huang S, Guo Y, Liu D, Zha S, Fang W. A two-stage transfer learning-based deep learning approach for production progress prediction in IoT-enabled manufacturing. IEEE Internet Things J 2019;6(6):10627–38. http://dx.doi.org/10.1109/JIOT.2019.2940131.

[181] Németh P, Ladinig T, Ferenczi B. Use of artificial neural networks in the production control of small batch production. In: Proceedings on the international conference on artificial intelligence (ICAI). 2016, p. 237–40.

[182] Kuhnle A, Röhrig N, Lanza G. Autonomous order dispatching in the semiconductor industry using reinforcement learning. Proc CIRP 2019;79:391–6. http://dx.doi.org/10.1016/j.procir.2019.02.101.

[183] Kuhnle A, Schäfer L, Stricker N, Lanza G. Design, implementation and evaluation of reinforcement learning for an adaptive order dispatching in job shop manufacturing systems. Proc CIRP 2019;81:234–9. http://dx.doi.org/10.1016/j.procir.2019.03.041.

[184] Stricker N, Kuhnle A, Sturm R, Friess S. Reinforcement learning for adaptive order dispatching in the semiconductor industry. CIRP Ann 2018;67(1):511–4. http://dx.doi.org/10.1016/j.cirp.2018.04.041.

[185] Tang J, Salonitis K. A deep reinforcement learning based scheduling policy for reconfigurable manufacturing systems. Proc CIRP 2021;103:1–7. http://dx.doi.org/10.1016/j.procir.2021.09.089.

[186] Danishvar M, Danishvar S, Katsou E, Mansouri SA, Mousavi A. Energy-aware flowshop scheduling: A case for AI-driven sustainable manufacturing. IEEE Access 2021;9:141678–92. http://dx.doi.org/10.1109/ACCESS.2021.3120126.

[187] Silva T, Azevedo A. Production flow control through the use of reinforcement learning. Proc Manuf 2019;38:194–202. http://dx.doi.org/10.1016/j.promfg.2020.01.026.

[188] Moon J, Jeong J. 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): IEEE; 2021, p. 1–8. http://dx.doi.org/10.1109/IMCOM51814.2021.9377434.

[189] Overbeck L, Hugues A, May MC, Kuhnle A, Lanza G. Reinforcement learning based production control of semi-automated manufacturing systems. Proc CIRP 2021;103:170–5. http://dx.doi.org/10.1016/j.procir.2021.10.027.

[190] Mayer S, Classen T, Endisch C. Modular production control using deep reinforcement learning: proximal policy optimization. J Intell Manuf 2021;32(8):2335–51. http://dx.doi.org/10.1007/s10845-021-01778-z.

[191] Schneckenreither M, Haeussler S. Reinforcement learning methods for operations research applications: The order release problem. In: Nicosia G, Pardalos P, Giuffrida G, Umeton R, Sciacca V, editors. Machine learning, optimization, and data science. Vol. 11331, Cham: Springer International Publishing; 2019, p. 545–59. http://dx.doi.org/10.1007/978-3-030-13709-0_46.

[192] Lang S, Reggelin T, Behrendt F, Nahhas A. 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; 2020, p. 1298–307.

[193] Zhao Y, Wang Y, Tan Y, Zhang J, Yu H. Dynamic jobshop scheduling algorithm based on deep q network. IEEE Access 2021;9:122995–3011. http://dx.doi.org/10.1109/ACCESS.2021.3110242.

[194] Kuhnle A, May MC, Schäfer L, Lanza G. Explainable reinforcement learning in production control of job shop manufacturing system. Int J Prod Res 2021;1–23. http://dx.doi.org/10.1080/00207543.2021.1972179.

Marcel Panzer, M.Sc. studied mechanical engineering at the Karlsruhe Institute of Technology and has been working as a research assistant at the University of Potsdam, Chair of Business Informatics, esp. Process and Systems since 2020. His research is focused on AI based production control.

Benedict Bender, Dr. studied business informatics at the University of Potsdam, the Humboldt University of Berlin as well as the University of St. Gallen. His research interests include Industry 4.0 and aspects of IT security. Furthermore, he deals with digital platforms and business ecosystems.

Norbert Gronau, Univ.-Prof. Dr.-Ing. received his doctorate in the Department of Computer Science at TU Berlin. Since 2004 he holds the chair of Business Informatics, Processes and Systems at University of Potsdam.