A Reinforcement Learning approach in Industry 4.0 enabled production system

Maria Grazia Marchesano*, Emma Salatiello*, Guido Guizzi*, Liberatina Carmela Santillo*

**Università degli Studi di Napoli "Federico II", Dipartimento di Ingegneria Chimica, dei Materiali e della Produzione Industriale, P.le Tecchio, 80, 80125, Napoli, ITALY*

(e-mail: mariagrazia.marchesano@unina.it, emma.salatiello@unina.it, g.guizzi@unina.it, santillo@unina.it)

Abstract: Modern manufacturing systems require a high degree of production flexibility to adapt to more personalized market demands in a timely and cost-effective manner, which the Industry 4.0 paradigm's technologies enable. As a result, in today's manufacturing environment, it is critical to optimize the use of these technologies. Simultaneously, to remain competitive, firms must commit to addressing external and/or internal restrictions in the manufacturing system. As a result, considering the growing interest in artificial intelligence (AI) and the promising results of its industrial application, this paper proposes a novel approach to production control based on Reinforcement Learning (RL) for resolving production scheduling difficulties of varying complexity. In this way, human intervention in production scheduling can be reduced, while planning and decision-making capabilities are improved at the same time. To support this claim, a simulation study was conducted that aims to assess the behaviour of an automated factory regarding various external and internal operational constraints. Consider a Flow Shop production line working in an Industry 4.0 environment capable of adopting Cyber-Physical Systems (CPS) and the Internet of Things (IoT); this study provides a novel flexible dispatching rule based on production line performance monitoring. The performances of the proposed new approach are compared to that of previously suggested dispatching regulations in the scientific literature. The simulation results revealed several intriguing conclusions, emphasizing the rule's flexibility and practical use is given certain practical assumptions.

Keywords: Industry 4. 0, Flow Shop, Reinforcement learning, Neural networks in process control, DQN.

I. INTRODUCTION

Flexibility is a required feature in today's manufacturing environments, both in terms of extremely varied processing times and order arrival rates (Ivanov, Das, and Choi, 2018). While Eastern Europe and Germany have seen their industrial sectors grow steadily in recent decades, many Western European countries such as the United Kingdom and France have seen their market shares decrease (Hofmann and Rüsch, 2017). As a result, favourable conditions for economic growth have evolved, culminating in the advent of Industry 4.0 (Mittal *et al.*, 2018).

Industry 4.0 has resulted in a flood of technological developments, most notably Cyber-Physical Systems (CPS) and the Internet of Things (IoT) (Culot *et al.*, 2020). The concept of CPS in particular revolutionized our outlook on manufacturing: operations are now automated in terms of scheduling and processing management (Riedl *et al.*, 2014).

Simultaneously with technological innovation, a paradigm shift in production management logic is desirable, as these remain inextricably linked to highly centralised infrastructures, such as the Material Resource Planning system (i.e., MRP-II), and are therefore incapable of managing the new variability introduced in the production system. This is a critical area of research in contemporary manufacturing science, as decentralised Manufacturing Planning and Control (MPC) System architectures are being presented at an increasing rate. These provide more rapid production management in the event of unplanned production interruptions (e.g. unexpected variability of processing times, sudden maintenance requirements, etc.). One disadvantage of this strategy is that it tends to offload the entire complexity of the scheduling problem to entities that do not share the MRP system's global perspective, but rather a machine-specific one (Bendul and Blunck, 2019).

As a result, hybrid, semi-heterarchical approaches are preferable to completely decentralized systems. They proposed dividing the scheduling problem into different architectural levels. The most critical task is how to sequence orders (i.e., the dispatching rule) once they are released into the production system, drawing inspiration from recent innovations proposed by Grassi *et al.*, (2020).

Additionally, the scientific literature recommends using distinct dispatching criteria for orders that have been released to production (Vespoli *et al.*, 2019; Grassi *et al.*, 2021). Apart from the self-explanatory classic dispatch rules such as "First In - First Out" (FIFO), "Shortest Processing Time" (SPT), and "Longest Processing Time" (LPT), there are several novel rules to consider. For example, some writers have divided the challenge of selecting the best effective dispatching rule into two independent phases: a first phase using classification techniques and a second phase including rule generation (Bektur and Saraç, 2019; Uzun Araz, Eski, and Araz, 2019; Vlašić, Đurasević, and Jakobović, 2019). There are some intriguing methods to this problem that make use of evolutionary techniques, such as genetic algorithms, to perform both or a portion of the functions (Đurasević and Jakobović, 2021). For example, Heger et al., (2016) demonstrated how diverse dispatching policies might be assigned using evolutionary algorithms. Alternatively, Đurasević and Jakobović, (2019) offer an intriguing and novel strategy that uses evolutionary algorithms to generate new dispatching rules from a combination of these.

In addition to genetic algorithms, various proposals have been made to investigate the use of artificial intelligence and machine learning. Additionally, the approaches follow the same logic in this case: they begin with classification problems and subsequently provide dispatching rules (Heger *et al.*, 2016; Thürer and Stevenson, 2018; Kim *et al.*, 2020).

These approaches do not incorporate real-time data from the monitored manufacturing line. That is why we propose a Reinforcement Learning (RL) approach that enables the artificial intelligence system to make data elaboration through the interaction with the environment in which it operates.

The use of reinforcement learning enables the automation of adaptive judgments, which are often difficult for humans to make. We wish to demonstrate the method's potential by showing an example RL model that, given a certain configuration (Deep Q-Network, DQN), learns how to operate a manufacturing line to achieve the desired throughput level.

Deep reinforcement learning is a subset of reinforcement learning that uses deep neural networks to describe states and/or approximate functions. We decided to employ reinforcement learning because it exhibits the same learning properties as humans, i.e. it learns via trial and error which set of actions to apply (Sutton and Barto, 2018)

The requirement for a simple dispatching rule capable of continuously monitoring the productivity of the production line, selecting the most appropriate dispatching rule based on contingent circumstances, and deciding between the classic "FIFO," "SPT," and "LPT" dispatching rules is the aim for this work.

The present study distinguishes itself from previous work (Marchesano *et al.*, 2021) in that it recommends scheduling jobs based on a machine and production system attribute rather than on job qualities, as is the case with standard rules.

The remainder of the paper is set out as follows: Section 2 has a literature review; Section 3 has the proposed approach; Section 4 has the experimental plan executed to support the proposal; and lastly, Section 5 closes the work.

II. Reinforcement learning

Reinforcement Learning (RL) draws inspiration from a variety of other well-known fields that study decision-making under conditions of uncertainty.

In Reinforcement Learning (RL), the problem to address is described as a Markov Decision Process (MDP). MDP is a mathematical framework to describe an environment in reinforcement learning. The RL consists of two main elements: the agent and the environment. The former interacts with the least. The environment reacts to and rewards the seen or state-like impacts of these activities. These two components (Fig. 1) are in constant communication, with the agent attempting to affect the environment by its behaviour and the environment reacting to the agent's activities.

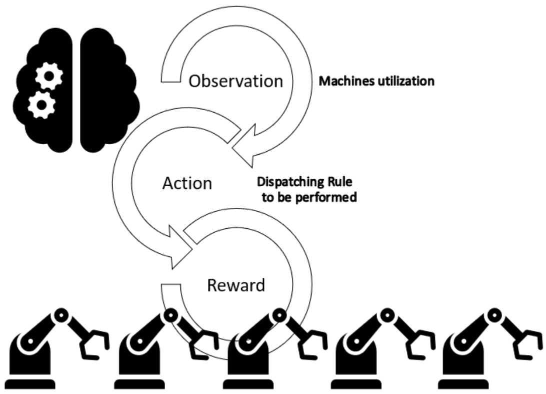

Fig. 1. RL approach proposed

The agent picks an action and observes what happens in the environment after it took the action. Then, it obtains a reward in correlation with the action and the state. The agent repeats the encounter numerous times and learns what action is ideal at each state (Hu *et al.*, 2020).

The environment's response to a specific activity is determined by a model that the agent may or may not be aware of. There are various ways for developing policies to solve tasks using deep reinforcement learning algorithms, each with its own set of benefits (Sutton and Barto, 2018). There is a distinction between model-based and model-free strengthening learning at the top level, which determines whether the algorithm attempts to learn a future model of the environment's dynamics.

The scheduling and production control problems have been handled in a variety of ways, using a variety of frameworks and algorithms characteristic of machine learning (Kusiak, 2020). RL is one of the strategies that may aid in the development of a more resilient manufacturing system capable of coping with manufacturing system complexity.

A novel method known as deep Q-network (DQN) was created in recent years, and it uses a traditional RL technique known as Q-Learning with deep neural networks (DNN). Mnih et al., 2015 created this algorithm. A RL approach for approximating functions is DQN. It is a development of the Q-learning approach in which a neural network takes the role of the state-action representation. The learning process in this technique involves backpropagating changes to the weights of the network's neurons. The change in weights based on the loss function provides the foundation for the DQN's learning of the value function:

$$L_t = \left(\mathrm{E}[\mathrm{r} + \gamma\, max_a \mathrm{Q}(s_{t+1}, a_t)] - \mathrm{Q}(s_t, a_t)\right)^2 ;$$

where $\mathrm{E}[\mathrm{r} + \gamma\, max_a \mathrm{Q}(s_{t+1}, a_t)]$represents the optimal expected reward related to the transition to the state s_{t+1}; r is the reward associated with the action a_t and to the state s_t; γ is the discount factor that is used to balance immediate and future reward; while $\mathrm{Q}(s_t, a_t)$ is the value estimated by the network. Backpropagation, which adheres to the principles of gradient descent, is the method used to spread mistakes calculated by the loss function throughout the network.

This algorithm is used in this study to train the RL agent to decide which dispatching rule is better to perform according to the production system's state.

III. Proposed approach

To address the problem, we consider a Flow Shop manufacturing line in an Industry 4.0 environment that enables the deployment of Cyber-Physical Systems (CPS) and Internet of Things (IoT). In this setting, important information can be acquired directly from the production line in this instance.

This study aims to develop a dispatching rule with a strong reactive behaviour that is based on production line performance monitoring. Without sacrificing generality, we will assume that the suggested flow-shop operates inside a Mass Customization paradigm, coping with a high level of variability entering the system. Then, because each job is unique, even if the technological cycle is followed, the consistency and specificity of each operation may vary, resulting in processing time variations. Taking as a starting point the work of Guizzi, Falcone, and De Felice (2019), which proposes an integrated parametric simulation model capable of combining production and maintenance processes, this study seeks to add greater generality by accounting for unforeseen events (e.g, breakdowns, potential rework, micro-stops, and unplanned machine setups). This is not the first time that this issue has been raised in the scientific literature. Wang et al. (2015), for example, address the proactive scheduling difficulty in the context of stochastic machine breakdown in worsening production environments. It provides a knowledge-based multi-objective evolutionary algorithm in which surrogate models based on support vector regression are employed to reduce computation costs.

To account for the variability of the productive process, in this work, an additional rate will be added to the job processing time. The gamma distribution is used to model this type of variability. This type of stochastic distribution is useful because it allows us to change the variability and simulate different scenarios by varying the distribution's parameters.

Given the complexity of the problem discussed we chose to employ RL approach.

RL is an agent-based learning approach that learns to accomplish a goal via interaction with the environment. The learning process is based on trial and error linked with a system of punishment or reward.

Using this tool, it may be feasible to choose which rule to employ in each machine queue, depending on data from the line in real-time or its digital twin (i.e., choosing the best appropriate rule based on the state of the system).

The challenge here is specifying both the observed state and the reward function that will be employed during the training phase.

A situation will be simulated in which the processing times will be sampled from a statistical distribution, as is typically done in the context of production scheduling problems.

In this case, however, an additional amount of time will be considered, representing a variability on the production line that cannot be predicted, such as micro-stops or reworking of the piece. For this micro-management of the scheduling of a production line, the RL approach is used for the scheduling of jobs in the line, considering the state of the machines (their utilization).

IV. Experimental approach

The scheduling issue will be assessed with a simulation model, which will be used to execute experiments using the AnyLogic multimethod simulation tool.

AnyLogic models are distinguished by a hierarchical structure, which allows one agent to surround other agents at variable levels. The top-level agent is the agent with the greatest degree of authority. It exemplifies the model's greatest level of abstraction. Each potential agent included in the top-level agent generates a lower level of abstraction. This property permits the development of a model at whatever level of detail desired, ultimately concealing an object's complexity. Furthermore, it offers a high level of modelling flexibility in terms of model structure and agent type.

To that aim, we provide a parametric model capable of representing a generic Flow-Shop system in a multi-method manner based on Discrete Event Simulation (DES) and Agent-based modelling. The decision to use such a methodology originated from our decision to parameterize the model in terms of resources (machines) and tasks to be performed (jobs), allowing us to use the same model to scale the size of the problem. Its parametric nature, in particular, enables you to define the system dimension in terms of the number of jobs and machines. The CONWIP (CONstant Work In Process) structure is used to control the WIP (Work In Process) of the Production System (PS) in the example.

In the simulation tool, two types of Agents are defined: Machines and Jobs. In the model, there are two low-level agents the jobs and the machines (Fig. 2) that "inhabit" the top-level agent the Main, which represents the manufacturing facility.

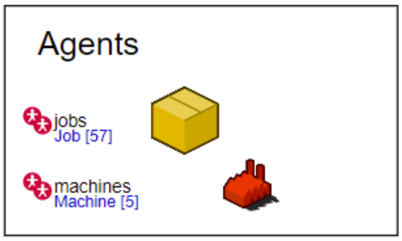

Fig. 2. Agents in the simulation model

In Fig. 3 there is the representation of the DES of the Machine agent in which the Job agents are being processed.

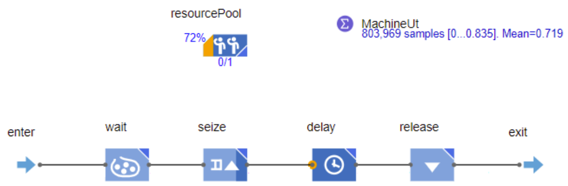

Fig. 3. Machine agent DES

In this simulation environment, the RL agent is thus trained through the RL experiment in which the characteristic elements of an MDP were identified, the observation of the state, the action to be performed, and the reward to be awarded to the agent. In particular, the experiment follows the configuration proposed in Mnih et al., 2015 in which a Deep Neural Network (DNN) is employed to approximate the representation of the state space and the action space and their relationship.

We studied a system with five machines and job processing times distributed as a gamma distribution with a value of α=1, so an exponential distribution, with an average of 10 minutes in the simulation scenario.

The input layer of the network is composed of 5 nodes representing the 5 utilization of the 5 machines, the output layer is composed of the number of combinations of the 3 dispatching rules on the 5 machines (3^5 243 nodes).

The network's main structure is a basic, fully linked, feed-forward network with one hidden layer made up of 150 nodes. As for the hyperparameters, the learning rate is 0.001, the discount factor γ equals t.99 and RMSProp (for Root Mean Square Propagation) is used as a gradient-ascent algorithm (Patterson, 2016). The training is made considering the structure described before and considering the following reward function:

$$reward = \begin{cases} -1 & TH < 1 \\ \beta \cdot TH - 1 & 1 \leq TH < r_b \\ 100 & TH > r_b \end{cases} \qquad (1)$$

In the (1) TH is the current throughput of the production line calculated using a mobile time window of 240 minutes. When it is minor of 1 the reward is -1 to punish the RL system. When it is in the interval 1 and r_b, the Bottleneck Rate of the line is the rate of the workstation that has the highest long-term utilization (Hopp and Spearman, 2011) the reward increases linearly (with β slope that in the experimental setting has been chosen equal to 3) as the TH increases. When the throughput is higher than r_b, the reward is 100 to maximize the TH of the line.

The experiments are made after the development of the policy resulting from the training of the RL agent.

The various scenarios differ in terms of the CONWIP value applied to the system and the dispatching rule applied to the machines.

The TH of the production line, as well as the coefficient of variation of the TH, are regarded as performance measures for analyzing the variability and hence governability of the phenomenon.

V. Results and Discussion

In the industrial area, whether in the form of micro-stops or extended downtime periods, there is the possibility of some inefficiency due to unforeseeable circumstances.

FIFO logic is the most often utilized rule in manufacturing to handle tasks. It is a basic logic that may result in some inefficiency when considering the whole processing time of the line, but its simplicity allows for efficient prediction of the amount of time required for the work to be finished when it is introduced into the production system.

On the other hand, the SPT dispatching rule is one of the most promising dispatching rules in the scientific literature in terms of performance. It excels in scheduling difficulties in both the information technology (IT) and industrial areas, especially when the aim is to optimize throughput. The downsides of the SPT are generally associated with the limited predictability of the Cycle Time as a consequence of the constant re-sequencing activity that may occur throughout processing.

The system that schedules the operation according to the Longest Processing Time (LPT) rule offers the greatest priority to tasks with the longest processing time. Schedulers will plan lengthier tasks at the beginning of the work schedule to decrease the amount of least time-consuming jobs after the work schedule. The benefit of this rule must be found in its capacity to maintain a steadier throughput over time, at the price of the poor predictability of the Cycle Time owing to any continuing re-sequencing activities.

Taking into consideration the strengths and shortcomings of the aforementioned rules, as well as the fact that no rule works effectively in a broad range of general operating settings, we offer a dispatching rule that identifies which DR to employ depending on the system's performance.

The goal here is to gather information about the system state before the event while accounting for the aleatory input into the system. The idea entails making the use of FIFO predictability as broadly as feasible, where machine utilization is almost the same for each machine.

Otherwise, if a certain machine needs to collect some utilization, it will agree to operate in an SPT logic to re-balance its utilization to the goal one. When the utilization is high, the machine is obliged to use LPT dispatching logic to proactively anticipate requests with longer processing durations, attempting to stabilize the Throughput.

The experimental results regarding the behaviour of the system reflect the considerations made above, in which the RL element goes on to choose a combination of rules on the machines such that it manages to balance the workload and achieve the productivity indicator, comparing the other two canonical rules, higher.

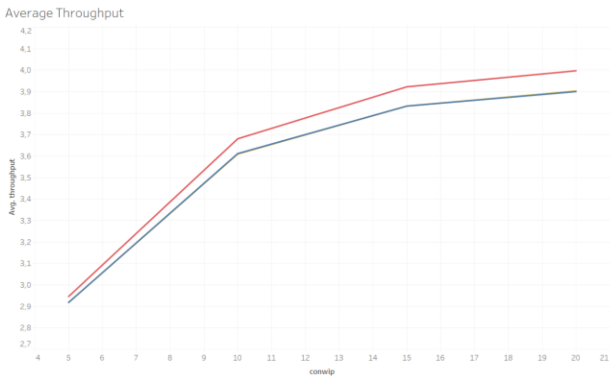

Fig. 4. The throughput (TH) comparison of the different rules

To begin with the simulation findings, we can state that the TH obtained using the RL technique is higher than in other dispatching rules in all of the CONWIP situations investigated. The same can be said about the coefficient of variance, which is lower when the RL approach is used rather than others. These findings are remarkable because the process itself can be much more predictable (Fig. 4 and 5). Fig. 4 shows the production line's throughput values as the CONWIP value varies. Fig. 5 depicts the CV values of the throughput as the CONWIP in the system varies.

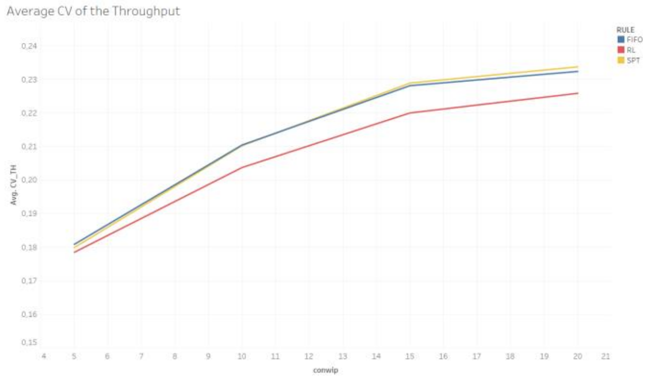

Fig. 5. The coefficient of variation for the TH

Considering the training process, you can see from the graphs that the loss function has a decreasing trend (Fig. 6). The function that represents the trend of the average reward is oscillating but increasing (Fig. 7). These two aspects show the goodness of the training but regarding the average reward, it should be considered an additional number of epochs to have a properly increasing trend.

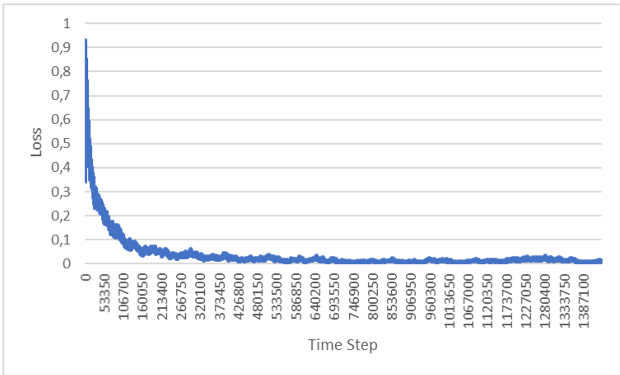

Fig. 6. The loss function

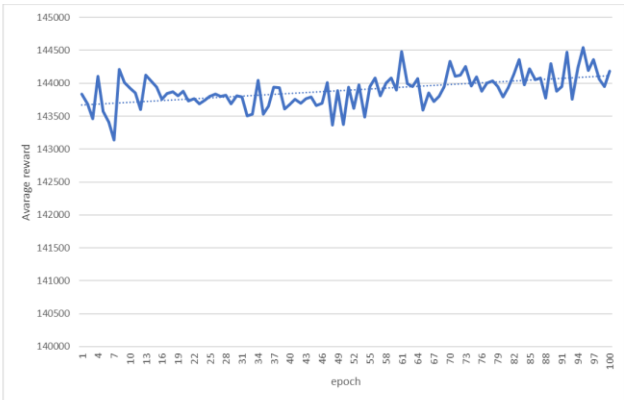

Fig. 7. The average reward

VI. Conclusions

The purpose of this work is to propose a novel dispatching rule capable of combining established dispatching rules with information about the monitored system's present condition.

The benefit is its adaptability and simplicity, which are regarded to be useful when applied to real-world hypotheses. To this end, and to test its performance, a parameterized simulation model has been constructed that incorporates a real-world industrial scenario with an unforeseen occurrence, and the suggested RL rule's performance has been examined and analyzed.

The simulation findings indicate that, in the studied situation, the suggested dispatching rule may strike a reasonable compromise between FIFO and SPT dispatching rules, balancing the trade-off between the SPT's throughput gains and the predictability of the orders allowed for production.

As a follow-up, it would be worthwhile to test additional scenarios in larger experimentation, including various configurations of the production line (e.g., hybrid-flow shop, open job-shop, or job-shop), to demonstrate that the proposed rule can be applied to a complex manufacturing configuration.

Additionally, the RL tool's potential may be evaluated and compared to the way human operators make choices on the production line, with the added advantage of eliminating normal human cognitive biases.

The proposal serves as a foundation for the development of a learning agent capable of comprehending how to accomplish a goal through interaction with the environment, acting in a predefined action space (which could be represented by the various Dispatching Rules into the various queues in front of the machine), and with a specified state read directly from the production line (e.g., the utilisation value of the machines).

This tool might be used by the production system to choose which rule to utilize downstream of the training based on data from the line in real-time or its digital twin.

To demonstrate significant benefits of monitoring the production system with more information than just the utilization value while making more dynamic judgments about how the DR should be employed in the future.

ACKNOWLEDGEMENTS

The research group would like to thank Tecnologica S.r.l. for granting the research.

REFERENCES

[1] Bektur, G. and Saraç, T. (2019) ‘A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server’, *Computers and Operations Research*, 103, pp. 46–63. doi: 10.1016/j.cor.2018.10.010.

[2] Bendul, J. C. and Blunck, H. (2019) ‘The design space of production planning and control for industry 4.0’, *Computers in Industry*, 105, pp. 260–272. doi: 10.1016/j.compind.2018.10.010.

[3] Culot, G. *et al.* (2020) ‘Behind the definition of Industry 4.0: Analysis and open questions’, *International Journal of Production Economics*, 226(August 2018), p. 107617. doi: 10.1016/j.ijpe.2020.107617.

[4] Đurasević, M. and Jakobović, D. (2021) ‘Automatic design of dispatching rules for static scheduling conditions’, *Neural Computing and Applications*, 33(10), pp. 5043–5068. doi: 10.1007/s00521-020-05292-w.

[5] Đurasević, M. and Jakobović, D. (2019) ‘Creating dispatching rules by simple ensemble combination’, *Journal of Heuristics*, 25(6), pp. 959–1013. doi: 10.1007/s10732-019-09416-x.

[6] Grassi, A. *et al.* (2020) ‘A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems’, *Manufacturing Letters*, 24, pp. 43–46. doi: 10.1016/j.mfglet.2020.03.007.

[7] Grassi, A. *et al.* (2021) ‘Assessing the performances of a novel decentralised scheduling approach in Industry 4.0 and cloud manufacturing contexts’, *International Journal of Production Research*, 59(20), pp. 6034–6053. doi: 10.1080/00207543.2020.1799105.

[8] Guizzi, G., Falcone, D. and De Felice, F. (2019) ‘An integrated and parametric simulation model to improve production and maintenance processes: Towards a digital factory performance’, *Computers and Industrial Engineering*, 137(September), p. 106052. doi: 10.1016/j.cie.2019.106052.

[9] Heger, J. *et al.* (2016) ‘Dynamic adjustment of dispatching rule parameters in flow shops with sequence-dependent set-up times’, *International Journal of Production Research*, 54(22), pp. 6812–6824. doi: 10.1080/00207543.2016.1178406.

[10] Hopp, W. J. and Spearman, M. L. (2011) *Factory Physics*. Third Edit. Edited by W. P. Inc.

[11] Hu, H. *et al.* (2020) ‘Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0’, *Computers and Industrial Engineering*, 149(August), p. 106749. doi: 10.1016/j.cie.2020.106749.

[12] Ivanov, D., Das, A. and Choi, T. M. (2018) ‘New flexibility drivers for manufacturing, supply chain and service operations’, *International Journal of Production Research*, 56(10), pp. 3359–3368. doi: 10.1080/00207543.2018.1457813.

[13] Kim, Y. G. *et al.* (2020) ‘Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system’, *Journal of Manufacturing Systems*, 57(August 2019), pp. 440–450. doi: 10.1016/j.jmsy.2020.11.004.

[14] Kusiak, A. (2020) ‘Convolutional and generative adversarial neural networks in manufacturing’, *International Journal of Production Research*, 58(5), pp. 1594–1604. doi: 10.1080/00207543.2019.1662133.

[15] Marchesano, M. G. *et al.* (2021) *Dynamic Scheduling in a Flow Shop Using Deep Reinforcement Learning, IFIP Advances in Information and Communication Technology*. Springer International Publishing. doi: 10.1007/978-3-030-85874-2_16.

[16] Mittal, S. *et al.* (2018) ‘A critical review of smart manufacturing & Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs)’, *Journal of Manufacturing Systems*, 49(October), pp. 194–214. doi: 10.1016/j.jmsy.2018.10.005.

[17] Mnih, V. *et al.* (2015) ‘Human-level control through deep reinforcement learning’, *Nature*, 518(7540), pp. 529–533. doi: 10.1038/nature14236.

[18] Patterson, J. (2016) *Deep Learning*, *Journal of Chemical Information and Modeling*. doi: 10.1017/CBO9781107415324.004.

[19] Sutton, R. S. and Barto, A. G. (1998) *SutSutton, R. S., & Barto, A. G. (1998). SuttonBartoSecondBook. In The Lancet (Vol. 258). https://doi.org/10.1016/S0140-6736(51)92942-XtonBartoSecondBook*, *The Lancet*.

[20] Thürer, M. and Stevenson, M. (2018) ‘On the beat of the drum: improving the flow shop performance of the Drum–Buffer–Rope scheduling mechanism’, *International Journal of Production Research*, 56(9), pp. 3294–3305. doi: 10.1080/00207543.2017.1401245.

[21] Uzun Araz, O., Eski, O. and Araz, C. (2019) ‘A reactive

scheduling approach based on fuzzy inference for hybrid flowshop systems', *International Journal of Simulation Modelling*, 18(1), pp. 5–18. doi: 10.2507/IJSIMM18(1)448.

[22] Vespoli, S. *et al.* (2019) 'Evaluating the advantages of a novel decentralised scheduling approach in the Industry 4.0 and Cloud cenManufacturing era', *IFAC-PapersOnLine*, 52(13), pp. 2170–2176. doi: 10.1016/j.ifacol.2019.11.527.

[23] Vlašić, I., Đurasević, M. and Jakobović, D. (2019) 'Improving genetic algorithm performance by population initialisation with dispatching rules', *Computers and Industrial Engineering*, 137(August), p. 106030. doi: 10.1016/j.cie.2019.106030.

[24] Wang, D. J. *et al.* (2015) 'A knowledge-based evolutionary proactive scheduling approach in the presence of machine breakdown and deterioration effect', *Knowledge-Based Systems*, 90, pp. 70–80. doi: 10.1016/j.knosys.2015.09.032.