On Reinforcement Learning in production control and its potentiality in manufacturing

Maria Grazia Marchesano, Guido Guizzi, Davide Castellano, Mario Di Nardo

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, guido.guizzi@unina.it, davide.castellano@unina.it, mario.dinardo@unina.it)

Abstract: In the new paradigm of industry 4.0, one of the open issues is to configure production and logistics systems to meet the increasingly customized market demand in a shorter time and at lower cost. It is suitable to implement measures and actions to make the production system much more resilient and self-organized to face all adversities. The technologies typical of Industry 4.0 help to meet these tasks. In the current era, it is therefore necessary to make even greater use of these tools. Considering the increasing interest in AI and the promising results of its application in industrial scenario, this paper proposes a new approach in production control using Reinforcement Learning (RL). A literature review is made to highlight the potential of RL application in production systems and how it could help in the decision making process. Among the applications found in the literature, an emphasis is placed on those specifically related to the world of manufacturing. The goal is to train a network to achieve a throughput target, keeping a certain amount of WIP constant on a Flow Shop line. A new approach where the state, action space and reward function are formulated. The system performance is compared with the known results of an analytical model (Practical Worst Case, PWC).

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

1. Introduction

The development of AI has now been going on for more than half a century and is making great steps towards a new phase of growth. Advances in new theories and technologies, such as Big Data, supercomputing, sensor networks, and data science, have made AI a legitimate component of any organization's automation strategy, thanks to strong demand and established trends in both economic and social terms. Artificial Intelligence, AI, is a great opportunity for the development of Industry 4.0. AI makes machines capable of performing activities and functions that until recently were exclusive to human intelligence. It can make a robot or software 'intelligent', i.e. able to learn and improve by learning. This is thanks to learning algorithms or machine learning. Machine learning is a particular branch of computer science that refers to all those mechanisms that allow an 'intelligent' machine to improve its capabilities and performance over time. The strength of machine learning lies in input, big data-based analysis, and data acquisition. Technically, the subject is closely related to computational statistics and probability. Therefore, it plays an important role in analysis for the optimization of business solutions. Regarding the machine learning, it is implemented mainly as:

- Supervised learning;
- Unsupervised learning;
- Reinforcement learning.

The first two methods use sets of data, that can be labeled or unlabeled, to classify, group, or cluster the samples, to make prediction, or to find new patterns. The last one, Reinforcement learning (RL), does not have data to work on and instead the learning process is made by the experience. Another step forward in ML is the development of Deep learning (DL), which is a set of techniques based on artificial neural networks organized in different layers, where each layer calculates the values for the next layer so that the information is processed more and more completely. Currently the themes of ML, DL and RL are widely investigated and implemented in many domains, from software computer and videogame (Van Hasselt, Guez and Silver, 2016), to tasks typical of the manufacturing domain as maintenance (Kuhnle, Jakubik and Lanza, 2019), production planning and control and fault detection (Xia *et al.*, 2020). The employment of these tools have proved the improvement of the performance of the overall system, and in addiction they have brought a certain self-regulation to the systems that in the manufacturing domain can be keystone to accomplish a resilient configuration.

In this paper, we would like to present RL in the domain of manufacturing and, in particular, in production control. The application of RL permits the automation of adaptive decisions, which are very often difficult for people to implement. We want to show the potential of this method by presenting an example RL model that, following a certain configuration (Deep Q-Network, DQN), learns how to control a production line to reach a targeted throughput level. The Deep reinforcement learning is a form of RL that uses deep neural networks for state representation and/or function approximation. We choose to use RL because it has the learning characteristics of humans, i.e. through trial and error it learns which is the best set of actions to implement (Sutton and Barto, 2018).

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.

2. Literature review

Reinforcement Learning (RL) is a field inspired by a number of other well-known disciplines that deal with decision-making under uncertainty.

Many studies have been conducted by researchers on the implementation of ML, DL and RL in the manufacturing domain from the perspective of Industry 4.0 (Wan *et al.*, 2020). The collection of studies reviewed can be divided according to the task in which the AI paradigms are involved in. Considering that our proposal is about the implementation of the RL-based model in the control of a production line, the literature review is conducted in the production control domain. Most control systems are based on static heuristics and models, which require a lot of expertise in the human field and thus do not fit the complex environment of manufacturing companies (Parente *et al.*, 2020). The scheduling and production control problem has been approached in different ways, using different types of frameworks and algorithms typical of ML. Reinforcement Learning (RL) is one of the techniques that could help to achieve a more resilient manufacturing system and cope with the complexities of manufacturing systems(Kuhnle *et al.*, 2019). RL is one of the Machine Learning (ML) systems that has been studied in the area of manufacturing control in recent years, along with Deep Learning (DL) (Usuga Cadavid *et al.*, 2020). Mezzogori, Romagnoli and Zammori, 2020 use the DL to predict precisely the delivery date in a make-to-order Job Shop managed by a Workload control. To deal with the complexity of the production system in the assembly job shop, Wang *et al.*, 2020 suggests an RL algorithm called dual Q-learning to enhance adaptability to environmental changes by self-learning. Wu *et al.*, 2020 suggest a combination of deep neural network (DNN) and Markov decision process (MDP) for dynamic scheduling of recurrent production systems. Thomas *et al.*, 2018 use Deep RL to account for uncertainties and achieve online dynamic scheduling in chemical production. In addition, a comparison is made with known heuristics or analytical models to prove the validity of the proposed approaches. This is also the case in Hofmann *et al.*, 2020 where the performance of the ML-based production schedule is compared with the static rule-based approach. The creation of a four-stage collaborative RL algorithm that provides a roadmap for two non-identical robots for non-identical machines is also part of the work by Arviv, Stern and Edan, 2016. In order to train a self-learning, intelligent, and autonomous agent for the decision problem of order dispatching in a complex job shop with strict time constraints, the authors Altenmüller *et al.*, 2020 use a Q-learning algorithm in conjunction with a process-based discrete-event simulation. The work by Leng *et al.*, 2021 proposed a loosely-coupled deep reinforcement learning (LCDRL) method for individualized Printed Circuit Board manufacturing order approval decision in Industry 4.0. In Chen, Fang and Tang, 2019, the Cloud Manufacturing perspective discusses the RL-based task assignment policy to support multi-project scheduling.

As shown before, it has not been discussed yet the problem of managing the throughput (TH) and work in progress (WIP) using RL to make the production systems more stable and self-controlled.

To address the problem of controlling a production line to overcome what are the issues of manufacturing inefficiency, our starting point is the work of Hopp and Spearman, 2011. They suggested to monitor Work-In-Process (WIP) in the system, and secure throughput from variance in the perspective of production control. Hopp and Spearman (2011) studied the actions of CONWIP (CONstant Work In Progress) lines in various scenarios in the literature. The scenarios are fascinating because they show how a CONWIP production line will work at its best or worst. They assess the performance of a production line in a real-world scenario in which job working times are exponentially distributed among the workstations, maintaining a balanced line (the average working time is the same for each work phase). The derived laws are summarized in Table 1.

Table 1 Relationship between Cycle time and Throughput(Hopp and Spearman, 2011).

Scenario	Cycle Time (CT)	Throughput (TH)
Best Case	$CT_{min} = \begin{cases} T_0 & if\ w \le W_0 \\ \frac{w}{r_b} & otherwise \end{cases}$	$TH_{max} = \begin{cases} \frac{w}{T_0} & if\ w \le W_0 \\ r_b & otherwise \end{cases}$
Worst Case	$CT_{max} = wT_0$	$TH_{min} = \frac{1}{T_0}$
Practical Worst-Case	$CT_{PWC} = T_0 + \frac{w-1}{r_b}$	$TH_{PWC} = \frac{w}{W_0 + w - 1} r_b$

- T_0 represents the Raw Processing Time of the line (the sum of long-term average process time of each workstation);
- r_b represents the Bottleneck Rate of the line (it is the rate of the workstation that have the highest long-term utilization);
- W_0 represents the Critical WIP of the line (it is the WIP level for which a line, with a defined Raw Processing Time and Bottleneck Rate, achieve the maximum throughput and the minimum cycle time without any variability).

So, starting from the hypothesis of PWC, we propose a method based on RL paradigm to accomplish the task of control a flow-shop line in terms of WIP and TH.

3. Proposed approach

Reinforcement Learning is a mathematical formalization of a problem involving decision-making.

RL is different from other Machine Learning methods because it focuses on goal-directed learning from interaction. The learning entity is not told what actions to take; instead, it must figure out for itself which actions result in the greatest reward, or objective, by putting them to the test through "trial and error". Furthermore, since current actions can decide future scenarios, these actions will impact not only the immediate reward but also future rewards, referred to as "delayed rewards" (how it happens in real life).

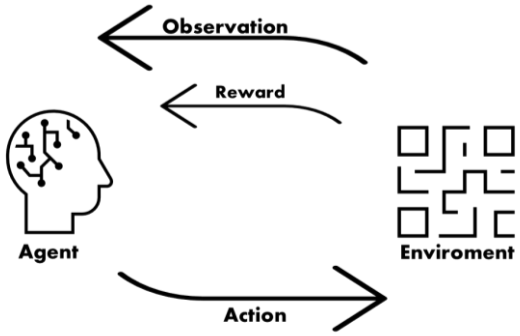

Figure 1 Reinforcement learning logic.

In RL there are two main components: an agent that has the function of making decisions (actions) to solve complex decision problems under uncertainty, an environment that is a "problem", i.e. everything that comes after the agent's decision. The environment responds to and rewards the effects of these behaviors, which are observations or states. These two core components (Figure 1) are actively communicating with each other, so that the agent is trying to control the environment through its behavior, and the environment is responding to the agent's actions. How the environment responds to a particular behavior is determined by a model that may or may not be known to the agent. There are several strategies for training policies to solve tasks with deep reinforcement learning algorithms, each with its own set of advantages (Sutton and Barto, 2018). At the top stage, a distinction exists between model-based and model-free strengthening learning, which indicates whether the algorithm tries to learn a future model of the dynamics of the environment.

In last years was developed a new algorithm called deep Q-network (DQN) it exploit a classic RL algorithm called Q-Learning with deep neural network (DNN). This algorithm was developed by Mnih *et al.*, 2015. DQN is an RL method for function approximation. It is a further development of the Q-learning method, in which the state-action representation is replaced by a neural network. In this algorithm, learning consists in adjusting the weights of the neurons composing the network by backpropagation. The learning of the value function in the DQN is based on the change of the weights depending on the loss 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. The errors computed by the loss function are propagated in the network by backpropagation, which follows the logic of the gradient descent. In fact, the gradient indicates the direction of the largest growth of a function, and by moving in the opposite direction, we reduce (to the maximum) the error. The behavior of the policy is given by an ε-greedy approach to strike a balance between exploring new states and exploiting already good policies.

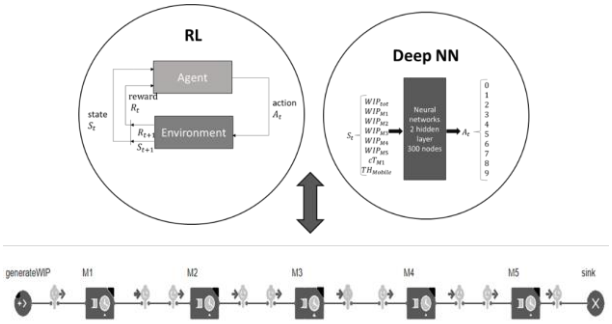

Figure 2 Proposed model.

We propose a model that employs the DQN in the context of production control. The control is made on a 5-machine Flow Shop production system with a FIFO dispatching rule for the jobs (Figure 2). The goal is to teach the system to decide by itself the quantity of WIP to keep constant in the system to achieve a targeted TH. The proposed system consists of a learning agent, which is a deep NN that approximates the learning function of RL. The learning agent refers to the entity that connects and learns from the environment through observations, actions and rewards. The environment is the virtual world in which the learning agent acts. As for the behavior, they reflect what the learning agent will do in the world. There are only some types of action that a learning agent can take depending on how much power the environment has (simulation model). In order to train the network to meet a certain goal TH, the collection of acts performed by the network injects as many jobs as possible to keep the WIP constant and reach the target TH at the same time.

The learning agent takes as input the observation of the state of the system, then a reward is calculated according to which the system will evolve to other states of the system in order to maximize the total reward. So the issue in this kind of problem is how to model the state, the reward and the set of possible action in order to describe the real system at its best. The issue of how to model the state in a RL approach has been investigated in Shi, Guo and Song, 2021. In this work the state is modelled in accordance to the feature of the system we want to control. So the following factors are considered: the completion time of the job on the first machine (cT_{M1}), the WIP on each machine, the initial total WIP in the system at each time step, and the current throughput in the line (TH_{Mobile}). The number of possible actions refers to the assessment of the number of orders to be placed in the system for achieving the throughput target. In order to achieve the set goal, a certain amount of job is injected into the system. This includes 10 potential actions: injecting 0 jobs when the

machines must discharge WIP; then injecting 1 job, 2 jobs etc. after the assessment, until 9 jobs. The reward is validation from the environment that helps to reinforce or punish the actions of the learning agent. This is expressed as a number, and it affects the way the learning agent chooses its actions. The reward function was chosen with the goal of receiving a reward of 1 when the TH system approaches the TH_{target}. The reward is given according to the distance of the current TH of the line and the target, the model gives a reward equals to 1 when the difference is smaller than 0.05 as we want to punish the situation in which the throughput is far away from the target because this would mean that or the system is not satisfying the production plan. The current throughput is measured using a mobile time window of 240 minutes.

$$S_t \begin{bmatrix} WIP_{tot} \\ WIP_{M1} \\ WIP_{M2} \\ WIP_{M3} \\ WIP_{M4} \\ WIP_{M5} \\ cT_{M1} \\ TH_{Mobile} \end{bmatrix} \quad A_t \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix}$$

$$reward: R_t = \begin{cases} 1 & if \ |TH_{target} - TH_{Mobile}| < 0.05 \\ 0 & else \end{cases};$$

A learned policy is produced after training. The policy developed during the training experiment would monitor the flow shop in terms of WIP and TH. The aim of the training experiment is to teach an artificial neural network how to operate a flow shop. It would do this by learning a strategy that, depending on the current state of the system, better governs the TH and WIP of the production line. The learner is not told which actions to perform, but must try them all to see which give the best results.

The time chosen for decision-making is 50 minutes, which is the raw time of line T_0, and the policy uses an observation of the model to estimate the appropriate response on the basis of the observations during training.

4. Experimental approach

To try the consideration made before a simulation model has been implemented in Anylogic, the multi-method simulation software, with the framework called rl4j (Reinforcement Learning for Java) that is integrated into the library DeepLearning4J. The environment of the RL is modelled as a discrete event simulation model (Figure 2), the observations of the state are the input layer of the DNN and it has 8 nodes. The action set is the output layer with a number of nodes equal to the number of possible actions in the model (10). The overall structure of the network is a simple, fully connected, feed-forward network with 2 hidden layers composed of 300 nodes.

The hyperparameter values used were chosen based on the scientific literature and the characteristics of our problem. The learning rate is 0.001, the discount factor γ equals to 0.99 (Patterson, 2016).The "L2" regularization algorithm is used, which applies a term to the objective function that reduces the squared weights. Regularization is a method to prevent overfitting. L2 increases generalization, smooths model performance on input transitions, and helps the network ignore weights it does not need (Patterson, 2016). RMSProp (for Root Mean Square Propagation) is used as a gradient-ascent algorithm, it is a method in which the learning rate is adapted for each of the parameters in the network.

The jobs that are processed in the system are modeled as agents with a basic state diagram (queue-work state-final state), and their processing time is expressed as a gamma distribution with a value of α=1, so an exponential distribution with a mean of 10 minutes.

In order to allow interaction between the model and the RL system, the required functions are used in the simulation model. There are thus two further functions in the Anylogic model: one for observation and one for action. The reward function is determined in the training experiment on the basis of the model's post-action measurements.

In this work the simulations are made to verify how good the employment of DQN is in the control of WIP and TH time. To measure the performance, we consider the mean error $TH_{target} - TH_{Mobile}$, the standard deviation of it and the mean WIP in the production line.

The simulations are made with two settings, one using the DQN and the other considering the hypothesis of the Practical Worst Case. To hit a steady-state of the production process, the simulation runs for two years. The throughput target is set to 4 unit/hour.

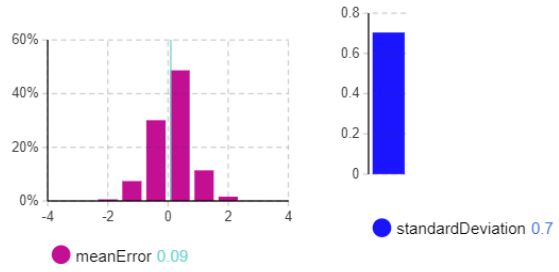

Figure 3 Histogram of the mean error and standard deviation with DQN.

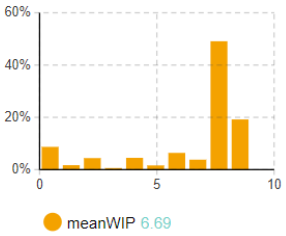

Figure 4 Histogram of the mean WIP with DQN.

The DQN model gives us a mean error of 0.09, a standard deviation of 0.7 and a mean WIP of 6.69 (Figure 5).

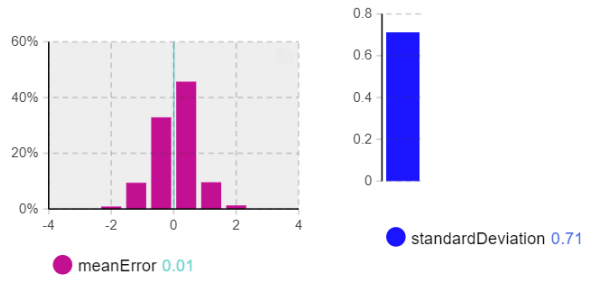

Figure 5 Histogram of the mean error and standard deviation PWC.

Comparing the results of the simulation model with the setting and the hypothesis of the PWC the results are: mean error of 0.01, a standard deviation of 0.71 and a mean WIP of 8 (Figure 6).

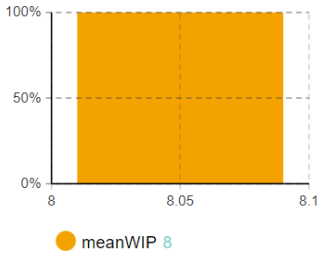

Figure 6 Histogram of mean WIP PWC.

Considering the system's average WIP, we can see that, while it is 6.68, the histogram(Figure 4) shows that, the frequency with which the system chooses to reach a WIP value of 8 (WIP of the PWC) is greater than the other. Since the overall results are so close to each other, in terms of mean error and standard deviation, we can say that the system integrated with the DQN learns the analytical law of the practical worst case without being taught it.

5. Conclusions

This work proposed to control the features of a 5-machine flow shop production line. The task is accomplished with an algorithm typical of the RL combined with the deep neural network, DQN. First of all, we have given the scientific context in which we are moving, examining the most significant works in the scientific literature for the treatment of our problem. then, in order to give a better understanding of the proposed model, we have described the most relevant characteristics. The data set is not previously established, but is collected using a simulation method. a description was given of how the state was modelled, the set of actions and how the reward is calculated. then the simulative tool was presented which allowed us to prove that the use of AI tools can teach the system to self-regulate and achieve the performance of analytical models found in literature (PWC). Given the high uncertainty of the experimented case, the findings of the proposed solution are promising. In the future, a deeper focus on how detailed the modeling of the state in respect to the problem studied will be analyzed. In addition, in future developments we want the experimental set to be expanded and some simulation parameters to vary in order to validate the proposed tool and perhaps compare it with more complicated algorithms.

Acknowledgement

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

References

Altenmüller, T. *et al.* (2020) 'Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints', *Production Engineering*, 14(3), pp. 319–328. doi: 10.1007/s11740-020-00967-8.

Arviv, K., Stern, H. and Edan, Y. (2016) 'Collaborative reinforcement learning for a two-robot job transfer flow-shop scheduling problem', *International Journal of Production Research*, 54(4), pp. 1196–1209. doi: 10.1080/00207543.2015.1057297.

Chen, S., Fang, S. and Tang, R. (2019) 'A reinforcement learning based approach for multi-projects scheduling in cloud manufacturing', *International Journal of Production Research*, 57(10), pp. 3080–3098. doi: 10.1080/00207543.2018.1535205.

Van Hasselt, H., Guez, A. and Silver, D. (2016) 'Deep reinforcement learning with double Q-Learning', *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 2094–2100.

Hofmann, C. *et al.* (2020) 'Autonomous production control for matrix production based on deep Q-learning', *Procedia CIRP*, 88, pp. 25–30. doi: 10.1016/j.procir.2020.05.005.

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

Kuhnle, A. *et al.* (2019) 'Design, implementation and evaluation of reinforcement learning for an adaptive order dispatching in job shop manufacturing systems', *Procedia CIRP*, 81, pp. 234–239. doi: 10.1016/j.procir.2019.03.041.

Kuhnle, A., Jakubik, J. and Lanza, G. (2019) 'Reinforcement learning for opportunistic maintenance optimization', *Production Engineering*, 13(1), pp. 33–41. doi: 10.1007/s11740-018-0855-7.

Leng, J. *et al.* (2021) 'A loosely-coupled deep reinforcement learning approach for order acceptance decision of mass-individualized printed circuit board manufacturing in industry 4.0', *Journal of Cleaner Production*, 280, p. 124405. doi: 10.1016/j.jclepro.2020.124405.

Mezzogori, D., Romagnoli, G. and Zammori, F. (2020) *Defining accurate delivery dates in make to order job-shops managed by workload control*, *Flexible Services and Manufacturing Journal*. Springer US. doi: 10.1007/s10696-020-09396-2.

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

Parente, M. *et al.* (2020) 'Production scheduling in the context of Industry 4.0: review and trends', *International Journal of Production Research*, 58(17), pp. 5401–5431. doi: 10.1080/00207543.2020.1718794.

Patterson, J. (2016) 'Deep Learning A Practitioner's Approach', *O'Reilly Media, Inc.*

Shi, L., Guo, G. and Song, X. (2021) 'Multi-agent based dynamic scheduling optimisation of the sustainable hybrid flow shop in a ubiquitous environment', *International Journal of Production Research*, 59(2), pp. 576–597. doi: 10.1080/00207543.2019.1699671.

Sutton, R. S. and Barto, A. G. (2018) 'Reinforcement Learning: An Introduction', *The MIT Press*.

Thomas, T. E. *et al.* (2018) 'Minerva: A reinforcement learning-based technique for optimal scheduling and bottleneck detection in distributed factory operations', in *2018 10th International Conference on Communication Systems and Networks, COMSNETS 2018*. Institute of Electrical and Electronics Engineers Inc., pp. 129–136. doi: 10.1109/COMSNETS.2018.8328189.

Usuga Cadavid, J. P. *et al.* (2020) 'Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0', *Journal of Intelligent Manufacturing*. Springer, pp. 1531–1558. doi: 10.1007/s10845-019-01531-7.

Wan, J. *et al.* (2020) 'Artificial-Intelligence-Driven Customized Manufacturing Factory: Key Technologies, Applications, and Challenges', *Proceedings of the IEEE*, pp. 1–22. doi: 10.1109/JPROC.2020.3034808.

Wang, H. *et al.* (2020) 'Adaptive scheduling for assembly job shop with uncertain assembly times based on dual Q-learning', *International Journal of Production Research*, 0(0), pp. 1–17. doi: 10.1080/00207543.2020.1794075.

Wu, C. H. *et al.* (2020) 'A deep learning approach for the dynamic dispatching of unreliable machines in re-entrant production systems', *International Journal of Production Research*, 58(9), pp. 2822–2840. doi: 10.1080/00207543.2020.1727041.

Xia, K. *et al.* (2020) 'A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence', *Journal of Manufacturing Systems*, (June), pp. 1–21. doi: 10.1016/j.jmsy.2020.06.012.