Chapter 14
Artificial Intelligence for Solving Flowshop and Jobshop Scheduling Problems: A Literature Review

P. Gomez-Gasquet, A. Boza, A. Navarro, and D. Pérez-Perales

Abstract With recent advances in artificial intelligence (AI), it is time to take a review of learning process as an approach for production scheduling. Neural networks, reinforcement learning, multi-agent systems, etc., have been successfully applied to solve a variety of complex problems. However, although combinatorial problems are also complex, it is not evident that the application of AI techniques can help to solve them in a satisfactory way and specifically in the field of production scheduling. At this time, it is interesting to know if researchers propose AI applications to solve scheduling problems in a global way and these are more efficient than those used up to now, or on the contrary, the dominant research lines focus on some partial aspect of the resolution. This paper makes a review of the different contributions that the AI field has made in recent years on the problem of the flowshop and jobshop scheduling. The work aims to see which are the AI methods that have been used, which have greater presence and what possibilities they offer in future.

Keywords AI · Machine learning · Jobshop · Flowshop · Scheduling

P. Gomez-Gasquet (✉) · A. Boza
Centro de Investigación de Gestión e Ingeniería de Producción (CIGIP), Universitat Politècnica de València. Cno/ de Vera S/N, 46022 Valencia, Spain
e-mail: pgomez@cigip.upv.es

A. Boza
e-mail: aboza@omp.upv.es

A. Navarro · D. Pérez-Perales
Departamento de Organización de Empresas, Universitat Politècnica de València. Cno/ de Vera S/N, 46022 Valencia, Spain
e-mail: alnaal@etsii.upv.es

D. Pérez-Perales
e-mail: dapepe@omp.upv.es

C. Avilés-Palacios and M. Gutierrez (eds.), *Ensuring Sustainability*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-95967-8_14

14.1 Introduction

The main objective of this article is to review the methods and algorithms applied to the resolution of a set of production scheduling problems whose approach has modified its traditional approach to incorporate contributions from the discipline that studies AI.

AI is a discipline that could be divided into a wide number of fields, which in many cases are interconnected and share a wide spectrum of knowledge. One of these fields is machine learning, which provides methods to improve efficiency by automating learning processes. Other fields, linked to the previous one that have come to have their own space, are those related to multi-agent systems, rule-based systems or case-based systems. And also, we can mention lines of research within AI based on the improvement and expansion of the applications of certain transversal methods such as neural networks, fuzzy logic or Bayesian networks, to give a few examples. Within these lines, it is necessary to highlight the one that has to do with the development of algorithms based on artificial life or evolutionary computation (genetic, ant, bee algorithms, etc.) since it provides an important set of techniques that converge with others from classic operation research. Throughout this paper, it will be revealed how the field of machine learning is the one that arouses the most interest among researchers focused on production scheduling. A classic classification of machine learning establishes three main areas of action: supervised or unsupervised learning and reinforcement learning [17].

Scheduling problems have been widely studied in the past decade, and solutions based on bioinspired algorithms which maintains a strong relationship with artificial life of the AI have been very successful. However, in this paper this area will not be addressed, as it is well known, and the contribution of machine learning techniques and methods will be sought. In this sense, the use of transversal AI techniques that are not used from the ML approach is out of focus. For example, Sabuncuoglu [8] provides a set of applications to the flowshop and jobshop workshops with neuronal network but without the indicated approach.

The set of scheduling problems that has been considered is wide since it includes any variation that addresses a configuration in jobshop or flowshop. What we try is to analyse how AI has modified the structure of the resolution method, either by providing a way to perform pre- or post-processing of the solution of any of the algorithms commonly used in this type of problems, or either by changing the approach to any of its key functions, or by simply using a typical AI technique instead of those algorithms. In the following section, the methodology that has been applied is commented; finally, the analysis of the results is addressed, in Sect. 14.3 and conclusions in Sect. 14.4.

14.2 Methodology

Once the scope of study is established, the methodology proposed is as follows: (1) select the sources of information; (2) search contributions; (3) debug the results; (4) analyse the results.

According to the methodology, the first step has been to choose Scopus as a source of information, since within the scientific field it has a better balance between the quality of the contributions and the breadth of search. There are other alternatives, such as Google Scholar, with a greater number of references but including the ones of Scopus and the differential is usually achieved with marginal contributions.

The second step is to properly choose the keywords for the search. The words “artificial intelligence”, “machine learning” and “reinforcement learning” were used, on the one hand, in combination with “jobshop” and “flowshop”, and on the other hand, in combination with “production scheduling”. As a consequence of the debugging process, in step three of the methodology, a set of filtering layer has been used. First, we filtered depending on the year because we discovered that most papers were published in the last ten years as it is shown in Fig. 14.1. Second, we got rid of papers that were not related to research.

Third, we took into account the title and the abstract of the paper to decide whether such paper talks about the topic we were concerned with. Fourth, we found out that, because of the huge paper database that we were managing, there were some papers which were duplicated; consequently, we removed them. Finally, a set of 21 papers which were directly related to the scope were found.

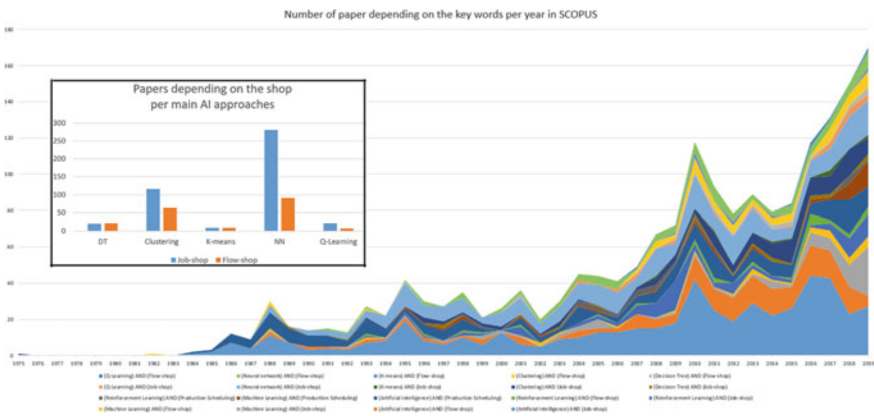

Fig. 14.1 Quantitative overview of papers found according to keywords

14.3 Results and Analysis

As a result of searching keywords in Scopus, we came up with an important number of papers which leads us to Fig. 14.1.

In Fig. 14.1, we can see two charts. The big one shows the evolution of the publication of papers in Scopus in a timeline taking into account the keywords that we looked for first. We clearly see that the publication is increasing and that in the last ten years it has dramatically gone up. Besides, the small chart represents the number of papers published in Scopus per shop within the main artificial intelligence approaches. We can see that the jobshop papers are greater in number to the flowshop ones and that neural network approaches are the most common ones in order to deal with both shops. The picture shows the tendency of the publications of papers towards this topic which is interesting to perceive that it presents a growing tendency.

Now, we were able to filter and keep those papers which are related to our field of study; but, also thanks to the findings we were also able to figure out what is currently the tendency within this topic (Table 14.1).

The set of selected paper have been classified as contributions in the context of supervised, unsupervised [14] or reinforcement learning [17]. The first two have been widely validated and used in other fields, and the reinforcement learning is a fashionable issue to address complex problems in recent years.

14.3.1 AI Contribution to the Flowshop Scheduling Problem

A flowshop scheduling problem consists in a set of jobs $J = \{1, 2, \ldots \mathrm{i}, \ldots \mathrm{n}\}$, and a set of operations $P = \{1, 2,..\mathrm{o},..\mathrm{P}\}$ have to be carried out with the same route for each job. In this way, a set of stages $S = \{1, 2,..s,..S\}$ is considered such that Po is carried out in So and $P = S$. A sequence for each stage must be found in order to minimize any objective function. Any type of additional restriction (set-ups, buffer restrictions, multiple machines in each stage, etc.) will be welcome in this work since no additional hypotheses are established.

The few proposals that have been found in the field of supervised learning are aimed at predicting the behaviour of some element of the problem. The proposal

Table 14.1 Results of selected bibliography after debugging

	Supervised	Unsupervised	Reinforcement learning		
	No specific	Clustering	Q-Learning		
Flowshop	[13], [5], [7] (Bartosz Sadel 2016)	[22]	[2, 3, 6]		
		NN	Q-Learning	NN	DQN
Jobshop		[21]	[4, 12, 18, 19, 23],	[1, 9, 11, 16, 20]	[24][10]

of Pavelski et al. [13] tries to identify the metaheuristic (Hill Climbing, Simulated Annealing, Tabu Search, ILS) that lower makespan or flowtime get given an instance in a flowshop problem, which may be permutation, no-wait, no- idle. Although the proposed approach has presented low accuracy in some cases, the resulting models showed interesting relations between the problem features and metaheuristics characteristics. In the case addressed by Hao et al. [7], they are oriented to use the learning capacity to identify the probabilities that help to locate a job in a position of the sequence of one of the phases of the estimation distribution algorithm (EDA) in a dynamic permutation flowshop problem with the objective of minimizing the total tardiness. For Sadel and Sniezynski [15], the focus is on identifying the most suitable machine to execute an operation in a hybrid flowshop that considers a job arrival following a Poisson distribution with the aim of minimizing the idle time of the machines. In the contribution of [13], a learning methodology is proposed whose central element is the identification of statistically significant differences using the Kruskal–Wallis test. However, Sadel and Sniezynski [15] perform a classification process through a software agent architecture, where in addition to supervised learning, reinforced learning is used. The previous proposals focus their efforts on improving sequencing; however, [5] presents a contribution whose objective is not to improve the sequence but to predict the lead time of the jobs. In a mass customization environment, they use the data from a MES to predict lead time using several predictive learning algorithms (linear regression, tree models, support vector regression).

In the field of unsupervised learning, Wang and Tang [22] propose to address the problem of the classic permutation flowshop with the variant of being multi-objective using AI in the exploitation phase of the local search process. The contribution of learning is focused on a clustering method that efficiently groups the non-dominated solutions, improving the local search phase: "There are two main features in the proposed ML-MOMA. First, each solution is assigned with an individual archive to store the non-dominated solutions found by it and based on these individual archives a new population update method is presented. Second, an adaptive multi-objective local search is developed, in which the analysis of historical data accumulated during the search process is used to adaptively determine which non-dominated solutions should be selected for local search and how the local search should be applied" [22].

In the reinforcement learning field, all the papers found apply the Q-learning algorithm with the aim of minimizing makespan. In a first approximation, Fonseca-Reyna et al. [3] addressed the problem of permutation flowshop to later [2] apply what has been learned to the hybrid flowshop with unrelated machines, setup times depending on the sequence and machines not eligible for some jobs. In both cases, the algorithm is executed at each stage of the workshop, which is represented by an agent. The state is the representation of the sequence that has been constructed in each agent until a given instant. The action consists in deciding which job in the queue of the corresponding stage is chosen to be operated. The reward is the inverse value of the makespan of the sequence generated by the agent. In a non-deterministic hybrid workshop [6], redefine the problem in terms of a Markov process where the states are defined by the tuple (stage, job), the decisions are the machines where

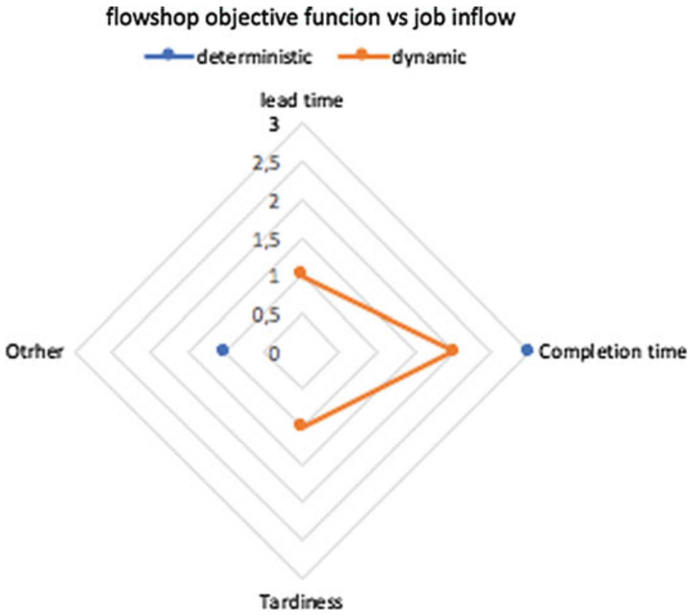

Fig. 14.2 Review of the objective function vs job inflow found in jobshop papers

it operates in the next stage, and the reward is a function inversely proportional to the waiting time of job in the next stage. The algorithm applies a greedy variable criterion that facilitates exploration at the beginning and reduces it as more has been learned. Two methods are used to adjust the greedy coefficient, one of them based on the concept of Boltzmann temperature (simulated annealing).

In addition, we have also analysed the objective function used and how the job arrival is considered in each paper that is shown in Fig. 14.2. As can be seen, the number of papers that consider deterministic and dynamic flowshop is balanced; however, the predominant objective function is the makespan.

14.3.2 AI Contribution to the Jobshop Scheduling Problem

A jobshop scheduling problem consists in a set of jobs $J = \{1, 2, \ldots i, \ldots n\}$, and a set of operations $P = \{1, 2,..o,..P\}$ have not to be carried out with the same route for each job. In this way, a set of stages $S = \{1, 2,..s,..S\}$ is considered such that Po is carried out in So and $P = S$. A sequence for each stage must be found in order to minimize any objective function.

In the jobshop context, no references of interest that use supervised learning have been found. The only proposal found in the field of unsupervised learning [21] is aimed at predicting makespan without the need to calculate it. To be able to specify the date, the system has the data generated by the real-time RFID system that feeds a deep belief network (DBN).

The rest of the contributions are concentrated in the field of reinforcement learning, and all of them apply a Q-learning method, a neural network or a combination of both. The authors who apply Q-learning mainly focus on static workshops, except [18, 19]. Thus, [12, 23] propose a multi-agent system that implements a Q-learning. In a more complete proposal, [23] defines that the states are determined by 4 characteristics of the scheduling and the workshop and are previously reduced to a K cluster in a deterministic shop for minimizing earliness and tardiness penalty. An adaptive scheduling is proposed, which consists in the strategy being modified according to the conditions of each moment, for which they propose a multi-agent system (agent work, agent machine, agent buffer, agent state). On this system, an adaptation of the Q-learning algorithm is proposed in which the status is determined by 4 characteristics of the programme and the workshop (average of penalties for early/late, average of the delivery factor, load and utilization ratio of the machines) and is previously reduced to a cluster K. The actions are the heuristic rules that can be used (EDD, SPT, FIFO, etc.). Reward is the negative value of the sum of the early and tardy or 1 otherwise. The contribution of [4] is very similar to that commented in the case of the flowshop. In a dynamic context, [18] decides so on the configuration of the batches and their sequence. In the first place, the problem focuses on when a batch of jobs is closed to be operated on a machine, considering that it is uncertain when each job arrives, and therefore, it is not known when the desirable size will be reached. Second, in what order the jobs are run. To do this, they propose to use a neural network based on the Q-learning algorithm in which the state is defined by 4 variables in each machine (the number of jobs being travelling to, being waiting in front of, being processed on machine and the estimated remaining time that machine m becomes idle), and the actions are tied to the chosen batch on a machine, including the option to not process any. And the reward is made up of two parts, one related to the chosen action associated with the waiting time and batch size, and the other to the cost of maintaining a job in the system. Meanwhile, [19] address a system with 2 agents dealing with a dynamic hybrid jobshop problem with the goal of minimizing makespan. A system with 2 agents is proposed. The first is responsible for creating sequences and acts when an event occurs that involves the location of a job. The solution proposal is based on a Q-learning method. In the proposal, the state is comprised of a vector with 3 parts: existence of an idle resource, existence of a job in queue and ratio between current and desired performance. Actions consist of assigning a job to a resource pool. The reward is a high and positive number if the end is reached with a throughput greater than the threshold and negative if this value is not reached; for intermediate states, it will be a much lower positive value and proportional to the throughput. The second agent uses a learning method in the form of a classifier that allows identifying the machines that are causing bottlenecks.

In the context of the authors who apply the neuronal network, a balance between dynamic and static problems is perceived. The contributions of [1, 11] address dynamic problems with minimization of makespan, being a flexible jobshop in the last case. In the proposal of Mao et al. [11] is addressed a computing system that resembles a dynamic jobshop whose objective is to minimize the average makespan. In this area, it is important to have systems that adapt to the variability of the load. For this, they propose a scalable reinforced learning system and various training methods. The use of a Neuronal Network whose the most novel aspect in the consideration of a set of job characteristics by means of an embedded graph that allows more or less attributes to be included without modifying the design of the network. On the other hand, [1] consider dependence on variable job and time. It also has the characteristics that jobs tend to be repetitive and have a similar duration. In the static context, [20] address a bi-objective jobshop and propose a reinforced neural network for each objective with a single layer of NxM neurons. The results are adjusted using a couple of complementary algorithms to avoid not complying with the precedence restrictions. The results are analysed by an experimental procedure. Setiawan [16] orient the problem as a critical path problem with the objective of minimizing makespan. The proposed solution consists of a reinforced learning based on temporary difference learning (TDL) based on a gradient descent method that is generated through a neural network. And [9] propose a reinforced learning method, which is implied in an agent. Due to the large size of the matrix (state, action), it is replaced by an estimator formed by fully connected neural network that uses policy gradient to train its learnable parameters.

Finally, there is a set of contributions that apply Q-learning using a neural network to calculate the matrix Q, known as DQN. In [24], a dynamic hybrid jobshop is addressed in order to reduce delays using deep reinforcement learning (DRL) in two phases. In phase A, it is applied individually in each work centre, and in phase B, it is done with the objective of balancing the workshop as a whole. In its training phase, the system compares its decisions with those of an expert system, supervised learning, and receives rewards based on the degree of affinity of the response. In proposal of Lin et al. [10] is desired to minimize the makespan. The authors propose the resolution of a jobshop with the aim of minimizing the makespan. The authors follow a strategy that assigns a dispatch rule (SPT, LPT, FIFO, etc.) to each station, and they may be different. There is centralized information in the cloud that allows information to be received and distributed from the entire plant. The authors use a multiclass DQL (MDQL) to assign a rule to each station. The MDQL is formed by a forward propagation neural network with 3 layers. In the input layer, characteristics of the orders and the system are considered. When the NN provides the rules, these are used to generate a programme that is then adjusted, the result is used to calculate the values of the matrix Q, and later the characteristics give input to the NN. Finally, an experimental analysis is carried out with standard data.

In addition, we have also analysed the objective function used and how the job arrival is considered in each paper that is shown in Fig. 14.3.

In the case of jobshop, there is a greater tendency to study deterministic cases than dynamic ones, and the predominance of makespan as an objective function is greater than in the case of flowshop.

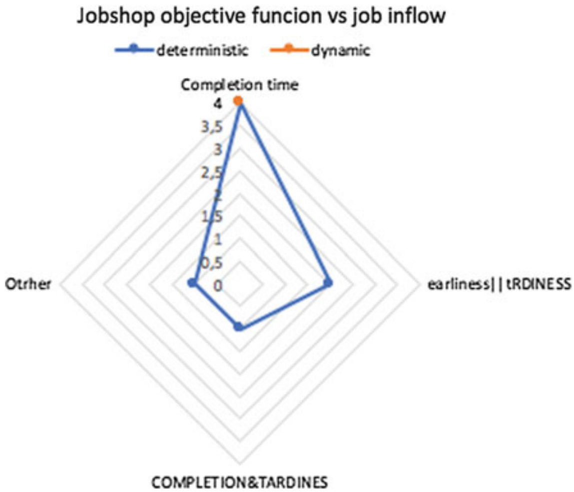

Fig. 14.3 Review of the objective function versus job inflow found in jobshop papers

14.4 Conclusions

This paper has made a review through the best contributions that have been found on a set of more than 1000 references. Most of them were published in the last 2–3 years. In general, it has been observed how the techniques that have been classified as reinforcement learning used for the generation of scheduling or selection of heuristic rules are the ones that present the most applications for the reviewed workshops, and as the rest of the techniques are in many cases used as an aid or complements to scheduling. In addition, it should be noted that the applications tend to focus on the most complex problems that are usually encountered in this field.

Because of all this analysis process, we can conclude that most of the research towards this topic is currently growing in the field of RL. It looks like these algorithms are able to fit the scheduling problem better than others, and consequently, research on this field is supposed to go up in the following years.

Finally, this paper is part of a research project within the application of AI techniques to production scheduling and it will lead us to develop our own RL algorithm in order to solve a flowshop scheduling problem. We have currently developed a Q-learning and SARSA algorithm without having very good results; but in future, we will work in a deep Q network algorithm to improve the accuracy of the scheduler.

Acknowledgements This research is part of the I+D+i project 'Integración de la Toma de Decisiones de los Niveles Táctico-Operativo para la Mejora de la Eficiencia del Sistema de Productivo en Entornos Industria 4.0 (NIOTOME)' (Ref. RTI2018-102020-B-I00) funded by MCIN/ AEI/10.13039/501100011033/ ERDF A way of making Europe.

References

1. Cheong M, Lee H, Yeom L, Woo H (2019) SCARL: attentive reinforcement learning-based scheduling in a multi-resource heterogeneous cluster. IEEE Access 7:153432–153444. https://doi.org/10.1109/ACCESS.2019.2948150
2. Fonseca-Reyna YC, Martinez Y, Dervecia-Cabrera A, Rodriguez EA (2019) Optimization of heavily constrained hybrid-flexible flowshop problems using a multi-agent reinforcement learning approach. Investigacion Oper 40(1):100–111
3. Fonseca-Reyna YC, Martínez-Jiménez Y, Nowé A (2017) Q-learning algorithm performance for m-machine, n-jobs flow shop scheduling problems to minimize makespan. Investigacion Oper 38(3):281–290
4. Fonseca Reyna YC, Martinez Y, Bermudez JM, Mendez B (2015) A reinforcement learning approach for scheduling problems. Investigacion Oper 36(3):225–231
5. Gyulai D, Pfeiffer A, Nick G, Gallina V, Sihn W, Monostori L (2018) Lead time prediction in a flowshop environment with analytical and machine learning approaches. IFAC-PapersOnLine 51(11):1029–1034. https://doi.org/10.1016/j.ifacol.2018.08.472
6. Han W, Guo F (2019) A reinforcement learning method for a hybrid flow-shop scheduling problem. Algorithms 12(11):222. https://doi.org/10.3390/a12110222
7. Hao L, Tieke L, Bailin W, Zhiwei L (2017) Estimation of distribution algorithm with machine learning for permutation flowshop scheduling with total tardiness criterion. Proc 10th Int Symposium Comput Intell Des ISCID 2017(2):283–286. https://doi.org/10.1109/ISCID.2017.193
8. Ihsan, Sabuncuoglu (1998) Scheduling with neural networks: A review of the literature and new research directtions. Production Planning & Control 9(1) 2–12 10 (2010). https://doi.org/10.1080/095372898234460
9. Li F, Hu B (2019) DeepJS: Job scheduling based on deep reinforcement learning in cloud data center. In: ACM international conference proceeding series, pp 48–53. https://doi.org/10.1145/3335484.3335513
10. Lin C, Deng D, Chilh Y, Chiu H (2019) Smart Manufacturing scheduling with edge computing using multiclass deep q network. IEEE Trans Indus Inf 15(7):4276–4284. https://doi.org/10.1109/TII.2019.2908210
11. Mao H, Schwarzkopf M, Venkatakrishnan B, Meng Z, Alizadeh M (2019) Learning scheduling algorithms for data processing clusters. SIGCOMM. 10(1145/3341302):3342080
12. Palacio JC, Jiménez YM, Nowé A (2019) Multi-agent reinforcement learning tool for job shop scheduling problems. In: CEUR workshop proceedings, vol 2491, pp 1–3
13. Pavelski L, Delgado M, Kessaci ME (2018) Meta-learning for optimization : a case study on the flowshop problem using decision trees. In: IEEE international conference auto science Engineering. https://doi.org/10.1109/CEC.2018.8477664
14. Russell SJ, Norvig P (2010) Artificial intelligence—a modern approach. Third international edition. Pearson Education
15. Sadel B, Snieezynski B (2016) Online supervised learning approach for machine scheduling. Theoretical Found Mach Learn Kraków 25(December):165–176. https://doi.org/10.4467/20838476SI.16.013.6194
16. Setiawan K (2012) Progress in Business Innovation and technology management reinforcement learning combined with radial basis function neural network to solve jobshop scheduling problem. APBITM Society 002:31–38
17. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press.
18. Tao OZ, Shufang X, Rose O (2018) Real-time batching in jobshops based on simulation and reinforcement learning. In: 2018 winter simulation conference (WSC). IEEE, pp 3331–3339
19. Thomas TE, Koo J, Chaterji S, Bagchi S (2018) MINERVA: a reinforcement learning-based technique for optimal scheduling and bottleneck detection in distributed factory operations1. In: IEEE international conference on automation science and engineering, pp 129–136. https://doi.org/10.1109/COMSNETS.2018.8328189

20. Tselios D, Savvas I, Kechadi MT (2013) RNN modelling for bi-objective MPM Job shop scheduling problem. IEEE Comput Intell Magaz 13–18. https://doi.org/10.1109/CICSYN.2013.38
21. Wang C, Jiang P (2019) Deep neural networks based order completion time prediction by using real-time job shop RFID data. J Intell Manuf Springer, US 30(3):1303–1318. https://doi.org/10.1007/s10845-017-1325-3
22. Wang X (2015) Tang L (2017) A machine-learning based memetic algorithm for the multi-objective permutation flowshop scheduling problem. Comput Oper Res Elsevier 79:60–77. https://doi.org/10.1016/j.cor.2016.10.003
23. Wang Y (2018) Adaptive job shop scheduling strategy based on weighted Q-learning algorithm. J Intell Manuf Springer US (1997). https://doi.org/10.1007/s10845-018-1454
24. Waschneck B, Reichstaller A, Belzner L, Altermüller T, Bauernhansl T, Knapp A, Kyek A (2018) Optimization of of global production scheduling deep reinforcement reinforcement learning. Elsevier. https://doi.org/10.1016/j.procir.2018.03.212