Production Management and Process Control, Vol. 36, 2022, 106–115
https://doi.org/10.54941/ahfe1001625

Dynamic Scheduling Techniques in Cloud Manufacturing - An Exploration of Deep Reinforcement Learning as a Critical Opportunity for Future Research

David Chambers, Thorsten Lammers, and Kun Yu

University of Technology Sydney 15 Broadway, Ultimo, NSW, Australia

ABSTRACT

For many years, metaheuristic algorithms have represented the state of the art in manufacturing scheduling techniques, proving to be exceptionally reliable for optimising schedules. However, metaheuristics suffer from inherent weaknesses that inhibit their ability to be applied to dynamic cloud manufacturing (CMfg) scheduling problems in practice. Thanks to the very recent and rapidly accelerating development in deep reinforcement learning (DRL), a small sample of studies have described how those approaches have thoroughly outperformed metaheuristic algorithms in dynamic manufacturing scheduling problems, establishing a new state of the art. However, a significant lag in maturity exists between the algorithms used in CMfg and state-of-the-art DRL. This paper systematically reviews the CMfg scheduling literature published between 2010 and 2020, summarises the development of deep reinforcement learning in this context and offers valuable directions for future research.

Keywords: Cloud manufacturing, Dynamic scheduling, Deep reinforcement learning, Optimization, Smart manufacturing

INTRODUCTION

Enabled by advancements in automated manufacturing technologies, cloud computing, and the Internet of Things (IoT), Cloud Manufacturing (CMfg) is a new, decentralised smart manufacturing paradigm. First proposed by Li et al. (2010), CMfg transforms the resources and capabilities of a network of manufacturers into on-demand manufacturing services to suit user requirements.

In the decade since, researchers have proposed several frameworks, architectures, and operational models for CMfg (Wang and Xu, 2013, Liu et al. 2018). As a result, the core features required to execute CMfg have been thoroughly explored, with multi-agent systems emerging as the dominant architecture. In many cases, however, these frameworks overlook the complex scheduling realities of implementing such networks in practice (Liu et al. 2018). Only recently has a technology emerged that demonstrates the potential to deliver a practical solution to CMfg scheduling – deep reinforcement learning (DRL) (Dong et al. 2020, Liang et al. 2020, Zhu et al. 2020).

Researchers have demonstrated the superiority of early DRL algorithms compared to applications of state-of-the-art metaheuristics methods in CMfg (Zhu, 2020). However, DRL is a field that is advancing rapidly with algorithms regularly setting new benchmarks in performance (Mnih et al. 2013, Schulman et al. 2017, Barth-Maron et al. 2018, Hafner et al. 2020). The resulting lag between advancements in DRL and the literature related to CMfg scheduling presents a unique opportunity for future research. This paper analyses and synthesises the research conducted on scheduling in CMfg since the inception of the concept in 2010, describes the current state-of-the-art and identifies gaps that researchers should investigate in the future. The following sections outline the processes involved with CMfg scheduling, introduce RL and how the field has developed, explore the limitations of metaheuristic algorithms and merit of RL as a state-of-the-art technique for dynamic CMfg scheduling, and offers directions for future research.

THE FOUR PHASES OF CMFG SCHEDULING

Scheduling in CMfg involves four discrete phases (see Figure 1). 1) Job Decomposition (JD), wherein jobs which are submitted by users to the CMfg system are broken down into the individual tasks to be executed. This preliminary step also defines the resources and capabilities which the CMfg network must find to complete jobs. 2) Service Matching (SM) which is the process of first determining whether resources and capabilities are suitable for tasks and then matching the available CMfg services with tasks. 3) Service Composition, or the process of optimally combining potential services selected via the matching process to complete jobs. 4) Task Scheduling (TS), which follows from SC and is the process of determining when tasks should be executed. (Liu et al. 2019).

Historically, manufacturing scheduling problems have predominantly been considered as static problems (where completion times are known, and disruptions do not occur). This concession of practicality has largely been determined by the NP-hard nature of manufacturing scheduling problems as proven by Sotkov (1995). As a result, solutions to such problems are derived by determining an approximation of the optimal schedule for all work to be executed. Conversely, dynamic approaches consider scheduling environments to be stochastic and rely on *online* scheduling policies where decisions are made reactively to the dynamics of the environment. For a realistic and robust representation of the CMfg scheduling problem, factors such as processing times, machine availability, staff availability, and logistics times must be stochastic. The core objective of this paper is to conduct a literature review to investigate how applications of popular metaheuristics and recent RL algorithms perform across the four CMfg scheduling phases in dynamic formulations.

AN INTRODUCTION TO REINFORCEMENT LEARNING

Reinforcement learning (RL) is a framework for learning how to map situations to actions such that the action taken maximises a numerical reward signal (Sutton and Barto, 2018). Using trial and error, RL algorithms learn

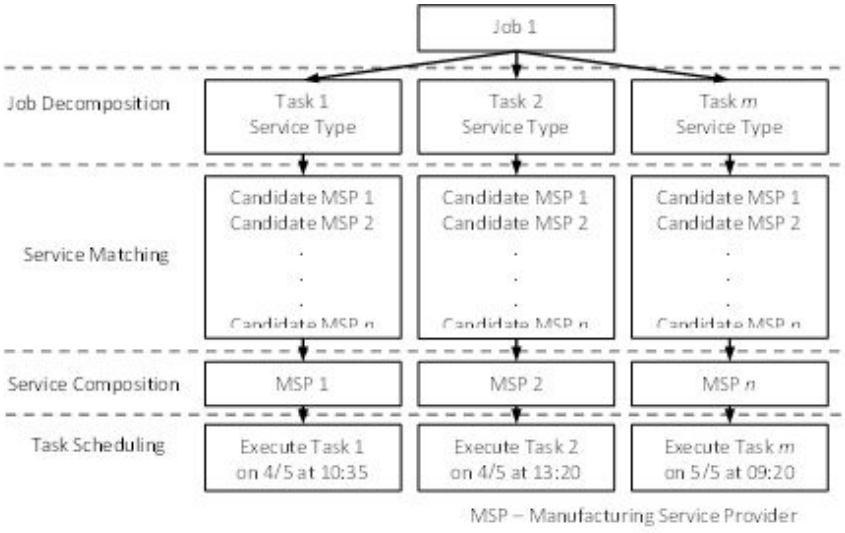

Figure 1: Visualisation of the CMfg scheduling phases.

to estimate the value of decisions, enabling these techniques to be deployed in stochastic problems. The structure of RL problems follows the Markov Decision Process framework (Howard, 1960), defined by an agent, a state, actions, and a reward signal (see Figure 3). The agent is the decision-maker and observes a representation of the decision problem, represented by the state. In each state, the agent selects an action, moves to the next state, and receives a reward signal which indicates the quality of the chosen action (Sutton and Barto, 2018).

RL algorithms use this framework to estimate the value of actions and/or develop an agent's policy. An agent's policy is a mapping of states to probabilities of selecting each possible action. Several approaches to algorithm development have been devised, categorised by their focus on learning. In value-based methods, the algorithm exclusively focuses on estimating the expected value of actions and generally utilises a greedy policy, where the action with the highest value is selected. Policy-based methods learn a policy directly and use the reward signal to refine the policy iteratively. Actor-critic methods also learn a policy directly (actor) but also estimate the value of actions (critic) to create a target for the policy (Sutton and Barto, 2018).

Each of these methods may also employ models. Model-based algorithms learn an independent understanding of the environment. The model is then used to predict how an environment will respond to the agent's actions, rather than exclusively focussing on interacting directly with the environment to learn a value function and/or policy (Sutton and Barto, 2018).

Tabular algorithms Q-learning (QL) (Watkins and Dayan, 1992) and Policy Gradient (PG) (Sutton et al. 1999) remained the state-of-the-art in RL up until the adoption of deep neural networks (DNNs) in 2013. DNNs are universal, non-linear function approximators and have dramatically expanded the capabilities of RL algorithms (Sutton and Barto, 2018). Mnih et al. (2013) produced a ground-breaking algorithm Deep Q-Network (DQN),

Table 1. Chronological summary of state-of-the-art RL algorithm development

Algorithm	RL Method				DNNs
	Value-based	Policy-based	Actor-critic	Model-based	
Q-Learning (Watkins 1989)	•				
Policy Gradient (Sutton et al. 1999)		•			
DQN (Mnih et al 2013)	•				•
A3C (Mnih et al. 2016)			•		•
DDPG (Lillicrap et al. 2016)			•		•
PPO (Schulman et al. 2017)			•		•
SAC (Haarnoja et al. 2018)			•		•
D4PG (Barth-Maron 2018)			•		•
Dreamer (Hafner et al. 2020			•	•	•

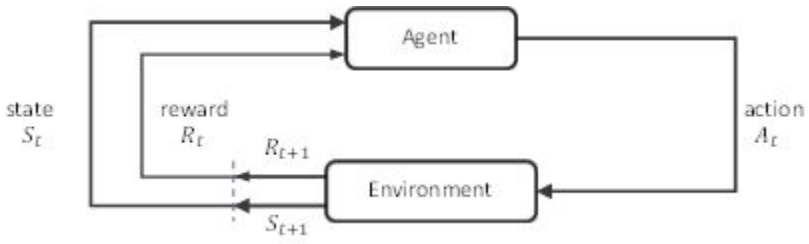

Figure 2: The agent-environment interaction in a Markov decision process.

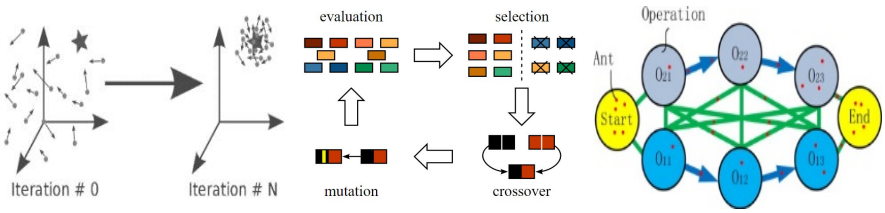

Figure 3: Metaheuristics (Sholtz, 2019, Becker, 2013, Wang et al. 2014).

applying DNNs as a value function approximator to the QL algorithm. Several RL algorithms have been developed after the introduction of DNNs (see Table 1), with actor-critic architectures dominating the field, each setting new benchmarks in overall performance and computation efficiency.

LITERATURE REVIEW METHODOLOGY

CMfg as a term has only recently been coined, limiting the overall breadth of research available. On the Scopus database, a search for the keywords 'cloud manufacturing' and 'scheduling' produces only 163 documents (114 journal articles from 57 journals, and 49 conference papers). For this literature review a concept-centric approach to classifying works and critical keywords was adopted (Webster and Watson, 2002), followed by a filtering process to reduce the review to only the most relevant papers (Levy and Ellis, 2006). Concept analysis revealed three important research streams for metaheuristics-based approaches, namely Genetic Algorithm, Particle Swarm Optimisation and Ant Colony Optimisation. The set of papers to be included in this review was filtered to only those that are concerned with 'cloud manufacturing', 'genetic algorithms', 'particle swarm optimisation', 'ant colony optimisation', and 'reinforcement learning'. The resulting set of papers were further condensed. Some papers, while appearing on the Scopus database, were not accessible or were written in languages other than English. Other papers did not publish the results of their algorithm implementations and were also excluded. The search and filtering methodology outlined above produced 31 papers relevant to the review. The distribution of their themes is as follows: 'Genetic algorithm' (13), 'particle swarm optimisation' (6), 'reinforcement learning' (5), 'ant colony optimisation' (4), and both 'genetic algorithm' and 'ant colony optimisation' (3).

LITERATURE REVIEW RESULTS

The following sections introduce the metaheuristic and RL techniques applied to CMfg scheduling phases in static and dynamic formulations, summarise the development of RL algorithms, describe the current state-of-the-art and identify critical research gaps.

Metaheuristics in CMfg

In response to the NP-hard nature of CMfg scheduling, metaheuristics have long represented state-of-the-art CMfg scheduling techniques. In CMfg research, three metaheuristics represent the core focus of the literature: Genetic Algorithms (GA), Particle Swarm Optimisation (PSO) and Ant Colony Optimisation (ACO). These algorithms use processes inspired by biology to guide a search of possible solutions. Chromosomes in GA to evolve solutions (Reeves, 2003), flocking behaviour in PSO allows a population of solutions to share their relative quality and the trajectory toward better solutions (Gass and Fu 2013), and candidate solutions in ACO use pheromones to guide future solutions (Dorigo and Stutzle, 2010) (see Figure 2). Metaheuristics have demonstrated a strong ability to produce effective global solutions to optimization problems in many domains (Sorensen and Glover, 2013). However, in a CMfg context, their practicality is limited. A key weakness of metaheuristics is their need to thoroughly search a solution space to approximate an optimal solution. The large search spaces inherit to CMfg scheduling require significant amounts of computation time before a global solution can be found (Cao et al. 2016, Li, Zhang and Ren, 2017). In response, several

Table 2. Summary of applications of metaheuristics to CMfg scheduling.

	CMfg process							Problem	
	JD	SM	SC	TS	SM + SC	SM + SC + TS	SC + TS	Static	Dynamic
Papers	1	0	12	7	2	1	4	26	1

Job Decomposition (JD) Service Matching (SM) Service Composition (SC) Task Scheduling (TS).

CMfg researchers have developed techniques to modify metaheuristics and limit the search space (Ding et al. 2019, Ghomi et al. 2019). However, even in static conditions, metaheuristics still struggle to produce solutions with the computational efficiency necessary when reacting to disturbances in a production scale CMfg network (Liu et al. 2019).

In turn, a significant research gap has emerged, as research conducted on scheduling in CMfg is almost exclusively conducted with static problem formulations where machine processing times do not vary, a full set of jobs is known in advance, and jobs do not suffer from interruptions. Manufacturers in practice do not operate in static conditions; instead, their circumstances are dynamic (Qu, Jie and Shivani, 2016). The dynamic nature of practical CMfg scheduling problems is therefore incompatible with state-of-the-art static solutions. Only one paper from the literature search considers dynamic conditions but requires up to 12.5 minutes to create a new schedule, with a new schedule required at any point where production deviates from the original solution (Zhang et al. 2019). This processing time is infeasible when solving dynamic, production scale scheduling problems in real-time where deviations may take place multiple times per minute (Park et al. 2020).

A second gap has also emerged as metaheuristics-based research in CMfg scheduling is predominantly focussed on individual phases of the CMfg scheduling process (see Table 2). A comprehensive literature review revealed only seven examples of researchers attempting to produce solutions that integrate multiple phases of the CMfg scheduling process.

Reinforcement Learning in CMfg

Very few researchers have investigated RL algorithms in CMfg. However, their results, particularly after the adoption of deep learning techniques, suggest that an exciting field is emerging. As highlighted in Table 3, researchers have only recently begun to apply algorithms from RL to CMfg scheduling. Of notable interest is the lag between the publication of the RL algorithms and the publication of their application to CMfg scheduling. Results from CMfg research are encouraging, though, with three of the five studies successfully finding solutions to dynamic scheduling. The state-of-the-art in RL has moved far beyond the DQN and PG algorithms implemented in papers published in 2020. However, these early algorithms have established a new state-of-the-art in CMfg, most notably in the example of SHARER (Zhu et al. 2020), where both GA and PSO are thoroughly outmatched. SHARER was able to achieve 40% greater resource utilisation and 30% shorter completion times. These results were achieved by modifying a PG algorithm which was first proposed at the turn of the millennium. These results beg the question:

Table 3. Applications of reinforcement learning in CMfg scheduling.

Paper	CMfg process				Problem		RL
	JD	SM	SC	TS	Static	Dynamic	Algorithm
Li et al. 2019			•	•		•	QL
Chen et al. 2019				•		•	PG
Dong et al. 2020				•	•		DQN
Liang et al. 2020			•		•		DQN
Zhu et al. 2020				•		•	PG + DNN

what kind of performance gains may be found through other, more mature actor-critic RL algorithms?

DISCUSSION AND CONCLUSION

To summarise, this literature review has revealed three critical gaps in the literature with each offering valuable opportunities for future research:

- Very few researchers have considered dynamic CMfg scheduling problems, limiting the effectiveness of their solutions in real-world applications.
- DRL techniques have demonstrated an ability to outperform metaheuristics in CMfg scheduling, both in solution quality and computational efficiency. However, a significant lag exists between the state-of-the-arts in CMfg and DRL with no applications of actor-critic or model-based architectures in CMfg.
- Integrated approaches to SC and TS have not been thoroughly explored in the literature, with only 6 examples found of such a problem formulation (5 using metaheuristics and 1 using RL)

In the decade since its inception, Cloud manufacturing (CMfg) has attracted a significant and growing amount of research attention with well-established scheduling functions and processes. However, researchers investigating the realities of optimally scheduling such a complex, dynamic, distributed physical network are yet to produce solutions capable of executing these operating models in practice.

Metaheuristic methods excelled at finding global solutions to optimisation problems in many fields. CMfg researchers who have adopted metaheuristics have improved the efficiency and performance of their solutions, albeit heavily skewed towards the service composition function. Applying these approaches to CMfg practice, however, is infeasible, due to the computational cost of rescheduling when network dynamics inevitably demand it.

CMfg researchers have also focussed intently on breaking the CMfg process into manageable steps, aiming to produce solutions to parts of the scheduling problem. This focus has resulted in siloed solutions that produce results in experiments but lack the ability to integrate with approaches in other steps. Segmentation of the problem has also left elements of the CMfg process underserved, limiting the potential of future implementations. Future research that focuses on integrating scheduling processes is needed.

The small sample of RL-based CMfg papers available is a notable limitation of this review, expanding the search to neighbouring field should be considered.

Developments in reinforcement learning have offered new and exciting opportunities to CMfg scheduling researchers. Algorithms have been developed that are capable of decision making across a large variety of tasks with excellent computational efficiency. In 2020, 3 different papers have shown the advantage that DRL algorithms have over metaheuristics in CMfg scheduling. The rate of improvement in DRL is so dramatic that the approaches applied to scheduling have been surpassed by many new benchmarks in performance, achieving efficiency gains of several orders of magnitude. A significant lag exists between state-of-the-art CMfg scheduling methods and state-of-the-art RL algorithms. The application of more mature RL algorithms to the CMfg scheduling problem is a clear and vital area for future research.

REFERENCES

Barth-Maron, G., Hoffman, M.W., Budden, D., Dabney, W. Horgan, D. Dhruva, TB., Muldal, A. Heess, N. and Lillicrap, T. (2018) "Distributed distributional deterministic policy gradients", arXiv:1804.08617v1.

Becker, Jonathan. (June 24, 2013) An Introduction to Particle Swarm Optimization (PSO) with Applications to Antenna Optimization Problems. Wireless Technology Website: http://wirelesstechthoughts.blogspot.com/2013/06/an-introduction-to-particle-swarm.html

Cao, Y., Wang, S., Kang, L. and Gao, Y. (2016) "A TQCS-based service selection and scheduling strategy in cloud manufacturing", International Journal of Advanced Manufacturing Technology, vol. 82, no. 43922, pp. 235–251.

Chen, S., Fang, S. and Tang, R. (2018) "A reinforcement learning based approach for multi-projects scheduling in cloud manufacturing", International Journal of Production Research, vol. 57 no. 8, pp. 1–19.

Ding, J., Wang, Y., Zhang, S., Zhang, W. and Xiong, Z. (2019), "Robust and stable multi-task manufacturing scheduling with uncertainties using a two-stage extended genetic algorithm", Enterprise Information Systems, vol. 13, no. 10, pp. 1442–1470.

Dong, T., Xue, F., Xiao, C. and Li, J. (2020), "Task scheduling based on deep reinforcement learning in a cloud manufacturing environment", Concurrency Computation, vol. 32, no.11.

Dorigo, M. and Stutzle, T. (2010) In: Gendreau, M. and Potvin, J.Y. (eds.), "Handbook of Metaheuristics", International Series in Operations Research and Management Science, vol. 146.

Gass, S.I. and Fu, M. C. (2013), "Encyclopedia of Operations Research and Management Science", 3rd edn, Springer, New York.

Ghomi, E., Rahmani, A.M. and Qader N.N. (2019) "Service load balancing, scheduling, and logistics optimization in cloud manufacturing by using genetic algorithm", Concurrency Computation, vol. 31, no. 20.

Haarnoja, T., Zhou, A., Abbeel, P. and Levin, S. (2018) "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with stochastic actor", arXiv:1801.01290.

Hafner, D., Lillicrap, T., Norouzi, M. and Ba, J. (2021) "Mastering atari with discrete world models", arXiv:2010.02193.

Howard, R. A. (1960). Dynamic programming and Markov processes. Cambridge, Massachusetts: The M.I.T. Press.

Levy, Y. and Ellis, T.J. (2006), "A systems approach to conduct an effective literature review in support of information systems research", *Informing Science Journal*, vol. 9. pp. 181–212.

Li, B., Zhang, L., Wang, S., Tao, F., Cao, J.X. and Song, C.X (2010) "Cloud manufacturing - A new service-oriented networked manufacturing model", Computer Integrated Manufacturing Systems, vol. 16, no. 1.

Li, F., Zhang, L. and Lu, H. (2019) "A reinforcement learning based scheduling for cross enterprises collaboration in cloud manufacturing", Proceedings of International Conference on Computers and Industrial Engineering, CIE, vol. 2019-October.

Li, F., Zhang, L. and Ren, L. (2017) "A Production-Based Scheduling Model for Complex Products in Cloud Environment", Proceedings - 2017 5th International Conference on Enterprise Systems: Industrial Digitalization by Enterprise Systems, vol. ES 2017, pp. 113–118.

Liang, H., Wen, X., Liu, Y., Zhang, H., Zhang, L. and Wang, L. (2020) "Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning", Robotics and Computer-Integrated Manufacturing, vol. 67.

Lillicrap, T.P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. and Wierstra. D. (2016), "Continuous control with deep reinforcement learning", arXiv:1509.02971v6.

Liu, Y., Wang, L., Wang, Y., Wang, X.V. and Zhang, L. (2018) "Multi-agent-based scheduling in cloud manufacturing with dynamic task arrivals", 51st CIRP Conference on Manufacturing Systems.

Liu, Y., Wang, L., Wang, X.V., Xu, X. and Zhang, L. (2019) "Scheduling in cloud manufacturing: state-of-the-art and research challenges", International Journal of Production Research, vol. 57, no. 15-16, pp. 4854–4879.

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Harley, T., Lillicrap, T.P., Silver, D. and Kavukcuoglu, K. (2016) "Asynchronous methods for deep reinforcement learning", Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA. JMLR: WandCP vol. 48.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M. (2013) "Playing atari with deep reinforcement learning", arXiv:1312.5602v1.

Park, I.B., Huh, J., Kim, J. and Park J. (2020) "A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities", IEEE Transactions on Automation Science and Engineering, vol. 17, no. 3, pp. 1420–1431.

Qu, S., Jie, W. and Shivani, G. (2016) "Learning adaptive dispatching rules for a manufacturing process system by using reinforcement learning approach", IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, vol. 2016-November.

Reeves, C. (2003) "Genetic Algorithms". In: Glover F. and Kochenberger G.A. (eds) Handbook of Metaheuristics. International Series in Operations Research and Management Science, vol. 57. Springer, Boston, MA.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017) "Proximal policy optimization algorithms", arXiv:1707.06347v2.

Sholz, J. (2019) "Genetic Algorithms and the Traveling Salesman Problem a historical Review", arXiv:1901.05737v1.

Sorensen, K. and Glover F.W. (2013) "Metaheuristics". In: Gass S.I., and Fu M.C. (eds) Encyclopedia of Operations Research and Management Science. Springer, Boston, MA.

Sotkov, Y.N. and Shakhlevich, N.V. (1995) "NP-hardness of shop-scheduling problems with three jobs", Discreet Applied Mathematics, vol. 59, no. 3, pp. 237–266.

Sutton, R.S. and Barto, A.G. (2018), Reinforcement learning: an introduction, 2nd edn. The MIT Press, Cambridge, Massachusetts.

Sutton, R.S., McAllester, D., Singh, S. and Mansour, Y. (1999) "Policy gradient methods for reinforcement learning with function approximation", Neural Information Processing Systems, Vol. 12, pp. 1057–1063.

Webster, J. and Watson, R.T. (2002), 'Analyzing the past to prepare for the future: writing a literature review', *MIS Quarterly*, vol. 26, no. 2, pp. xiii-xxiii.

Wang, W., Fan, X., Zhang, C. and Wan, S. (2014) "A Graph-based Ant Colony Optimization Approach for Integrated Process Planning and Scheduling", Chinese Journal of Chemical Engineering, vol. 22, no. 7, pp. 748–753.

Wang, X.V., and Xu, X. 2013, "Virtualise manufacturing capabilities in the cloud: Requirements, architecture and implementation", ASME 2013 International Manufacturing Science and Engineering Conference Collocated with the 41st North American Manufacturing Research Conference, MSEC 2013.

Watkins, C.J.C.H. and Dayan, P. (1992), 'Q-learning', Machine learning vol. 8, pp. 279–292.

Zhang, W., Ding, J., Wang, Y., Zhang, S. and Xiong, Z. (2019) "Multi-perspective collaborative scheduling using extended genetic algorithm with interval-valued intuitionistic fuzzy entropy weight method", Journal of Manufacturing Systems, vol. 53, pp. 249–260.

Zhu, H., Li, M., Tang, Y. and Sun, Y. (2020) "A Deep-Reinforcement-Learning-Based Optimization Approach for Real-Time Scheduling in Cloud Manufacturing", IEEE Access, vol. 8, pp. 9987–9997.