



Reinforcement learning applied to production planning and control

Ana Estesó, David Peidro, Josefa Mula & Manuel Díaz-Madroño

To cite this article: Ana Estesó, David Peidro, Josefa Mula & Manuel Díaz-Madroño (2022): Reinforcement learning applied to production planning and control, International Journal of Production Research, DOI: [10.1080/00207543.2022.2104180](https://doi.org/10.1080/00207543.2022.2104180)

To link to this article: <https://doi.org/10.1080/00207543.2022.2104180>



© 2022 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



[View supplementary material](#)



Published online: 06 Aug 2022.



[Submit your article to this journal](#)



Article views: 548




[View related articles](#)



[View Crossmark data](#)

Reinforcement learning applied to production planning and control

Ana Esteso ^a, David Peidro ^b, Josefa Mula ^b and Manuel Díaz-Madroñero ^b

^aResearch Centre of Production Management and Engineering (CIGIP), Universitat Politècnica de València, Valencia, Spain; ^bResearch Centre on Production Management and Engineering (CIGIP), Universitat Politècnica de València, Alicante, Spain

ABSTRACT

The objective of this paper is to examine the use and applications of reinforcement learning (RL) techniques in the production planning and control (PPC) field addressing the following PPC areas: facility resource planning, capacity planning, purchase and supply management, production scheduling and inventory management. The main RL characteristics, such as method, context, states, actions, reward and highlights, were analysed. The considered number of agents, applications and RL software tools, specifically, programming language, platforms, application programming interfaces and RL frameworks, among others, were identified, and 181 articles were reviewed. The results showed that RL was applied mainly to production scheduling problems, followed by purchase and supply management. The most revised RL algorithms were model-free and single-agent and were applied to simplified PPC environments. Nevertheless, their results seem to be promising compared to traditional mathematical programming and heuristics/metaheuristics solution methods, and even more so when they incorporate uncertainty or non-linear properties. Finally, RL value-based approaches are the most widely used, specifically Q-learning and its variants and for deep RL, deep Q-networks. In recent years however, the most widely used approach has been the actor-critic method, such as the advantage actor critic, proximal policy optimisation, deep deterministic policy gradient and trust region policy optimisation.

ARTICLE HISTORY

Received 29 July 2021
Accepted 11 July 2022

KEYWORDS



Artificial intelligence;
machine learning;
reinforcement learning; deep
reinforcement learning;
production planning and
control; industry 4.0


1. Introduction

Reinforcement learning (RL) is similar to the way humans and animals learn. In fact many RL algorithms are inspired in biological learning systems (Sutton and Barto 2018). RL is a branch of machine learning (ML) where an agent interacts with an environment by performing actions and perceiving environmental states and has to learn a 'correct behaviour' (the optimal policy) by means of a feedback rewarding signal. Unlike a stationary database, the environment has its own internal memory (a state), which the agent alters with its actions (Sutton and Barto 2018; Russell and Norvig 2003; Briegel and Cuevas 2012). In RL, all the agents have explicit goals and learn decisions by interacting with their environment to achieve these goals (Han 2018).

Production management is the set of activities that determines the start and end times of tasks that have to be performed in a production system to fulfil customer orders and demand forecasts (Vollmann et al. 2005; Mehra 1995). Some of the basic subsystems of the

production management system are production planning and control (PPC). PPC activities are developed based on not only the constraints and objectives defined by a company or supply chain's strategic level, but also on demand forecasts, customer orders, inventory holding, and work orders in progress. It should be noted that markets, technologies and competitive pressures are constantly changing, which may require changes in the processes and design of PPC systems and their algorithms. In the twenty-first century, PPC systems must possess characteristics like agility, intelligence and rapid response, and must favour the production of high-quality products, small production batches, customisation requirements, customer commitment and environmental friendliness (Qiao and Zhu 2000). In addition, the company's production systems can no longer be managed independently, but must be modelled from the perspective of belonging to one or several supply chains (Lambert and Cooper 2000). In this context, the supply chain is characterised by concurrent engineering based intensively on information

CONTACT Josefa Mula  fmula@cigip.upv.es  Research Centre on Production Management and Engineering (CIGIP), Universitat Politècnica de València, C/Alarcón 1. Alcoy, Alicante, Spain

 Supplemental data for this article can be accessed here. <https://doi.org/10.1080/00207543.2022.2104180>

and communication technology (ICT), such as digitalisation, computer networks, artificial intelligence, among others.

From a PPC view, Industry 4.0 (I4.0) is a connection of digital technologies, organisational concepts and management principles to provide cost-effective, responsive, resilient and sustainable operations (Ivanov et al. 2021; Cañas, Mula, and Campuzano-Bolarín 2020; Mula and Bogataj 2021; Bueno, Filho, and Frank 2020). The PPC area is evolving, according to design principles, digital enabling technologies and implementation principles of the I4.0 initiative (Cañas et al. 2021), towards self-management PPC models based on automation and autonomy. Serrano-Ruiz et al. (2021) present a classification of I4.0 enabling technologies based on seven categories: (i) infrastructure; (ii) technical assistance based on software; (iii) technical assistance in manufacturing; (iv) technical assistance at the interface; (v) technical assistance in data management/decision making; (vi) technical assistance in communications and connectivity; (vii) associated with management models. Here ML and RL are enabling technologies in the relevant trend of the I4.0 category of technical assistance in data management and decision making. RL algorithms can provide solutions to many real-world applications from artificial intelligence to operations research or control engineering (Szepesvári 2010). Previously, Usuga et al. (2020) conducted a systematic literature review of 93 articles to consider ML or deep learning, and the PPC areas of production scheduling, production planning, production control and line balancing in the searching strategy. Their findings showed that the PPC area with more ML applications was smart scheduling and planning. Furthermore, RL techniques have been widespread, which confirms the interest shown in agent-based models. The aim of other reviews has been deep RL (DRL) in the job shop scheduling area (Cunha et al. 2020) or RL in health care (Yu, Liu, and Nemati 2020). Nevertheless, as far as we know, we have not identified other states of the art that address RL in PPC.

Here we perform a literature review of 181 articles that focus on RL techniques and PPC by considering the following PPC areas based on the proposal of Jeon and Kim (2016): facility resource planning, capacity planning, purchase and supply management, production scheduling and inventory management. The main contributions of this paper are to: (i) classify the main RL research and applications in the PPC area; (ii) define the main highlights and limits from the revised RL-PPC literature; (iii) identify the main characteristics of the RL software tools used in the PPC area; (iv) discuss and propose the main trends and further research.

The rest of the paper is arranged as follows. Section 2 revises the main RL and DRL foundations. Section 3

presents the review methodology. Section 4 offers the literature review. Section 5 discusses the main RL-PPC trends and further research. Section 6 ends with the main reached conclusions and new research lines for the future.

2. Reinforcement learning foundations

Artificial intelligence is a discipline concerned about creating computer programmes that exhibit human-like intelligence. ML is defined as an artificial intelligence sub-field that confers computers the ability to automatically learn from data without being explicitly programmed indicating rules. RL is one of the three main branches of ML techniques: (a) supervised learning, which is the task of apprenticeship from tagged data and its goal is to generalise; (b) unsupervised learning, where training data do not contain labels, but it is the algorithm that will attempt to classify information; (c) RL is the task of learning through trial and error and its goal is to act to obtain the highest reward in the long run (Torres 2020).

According to Sutton and Barto (2018), three main elements are required for the RL process: a policy, which defines the actions to choose in a given state of the environment; a reward signal, which divides these actions into good or bad according to the immediate return received by the transition between states; and a value function to evaluate which actions have positive long-term effects by considering not only the immediate reward of a state, but also the reward that is likely to follow in the long run.

The agent's overall goal is to maximise the total reward and, therefore, the reward signal is the basis for adjusting the policy and the value function. There is a fourth optional element in some RL systems, namely a model of the environment. It mimics the environment's behaviour or allows inferences to be more generally made as to how the environment will behave. Models are employed for planning, which means any way to decide a course of action by contemplating what future situations may take place before they are experienced. Model-based methods solve the RL problems that utilise models and plan conversely to simpler model-free methods. They act explicitly as trial-and-error learners and are considered virtually the opposite of planning (Sutton and Barto 2018).

RL processes can be modelled as a Markov decision process (MDP) (Sutton and Barto 2018; Vasilev et al. 2019). A 5-tuple (S, A, P, R, γ) can formalise this stochastic mathematical model: S is the finite set of all the feasible environment states; s_t is the state at time t ; A is the set of all the possible actions; a_t is action at time t ; P is the environment's dynamics or the transition probabilities matrix, which can define the conditional probability of transitioning to a new state s' with reward r given

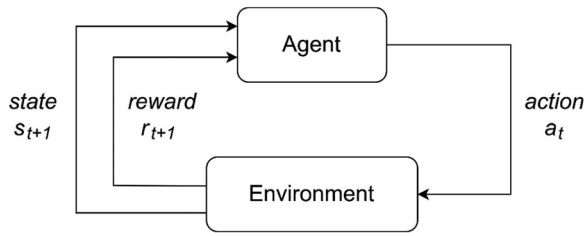


Figure 1. RL cycle.

the existing state, s , and an action, a (for all states and actions).

$$P_{ss'}^a = Pr(s_{t+1} = s' \vee s_t = s \vee a_t = a) \quad (1)$$

The transition probabilities between states actually represent the model of the environment; i.e. how it is likely to change with its current state and an action, a . Model-based agents possess an internal representation of P to forecast the results of their actions. In an MDP, the Markov property guarantees that the new state will depend on only the current state rather than on any previous states. This implies that the state totally characterises the environment's total state to make MDP a memory-less process. So R is the reward function that describes the reward which the agent will receive when it carries out action a and transitions from s to s' .

$$R_{ss'}^a = E[r_{t+1} \vee s_{t+1} = s', s_t = s, a_t = a] \quad (2)$$

Lastly, γ is the discount factor, a value within the $[0:1]$ range that determines how much the algorithm values immediate rewards as opposed to future rewards.

Figure 1 shows the RL cycle (Sutton and Barto 2018).

One of the main differences of RL in relation to other ML branches is the presence of delayed rewards. An agent may receive insignificant rewards during a relatively long sequence of actions until it reaches a particular state with a very significant reward value for RL (i.e. finding the way out of a maze). The agent must be able to learn which actions are desirable based on a reward that may arbitrarily take place in the future. This is not the case with supervised learning, where the data used for training already contain the desired solution (label). In this case, the algorithm must learn a model or function to map an input to an output that is already known in advance. Thus RL deals with dynamic decisions while traditional optimisation usually focus on static ones.

In RL, an agent's policy, denoted as $\pi(a \vee s)$, is the strategy that the agent uses to determine the next action a based on current state s (a probability distribution that maps an action a to state s). During the learning process, the policy may change because the agent acquires more experience. Then we need a method that automatically

helps to learn and find optimal policies for the agent. This is why a value function is needed that determines what is 'good' for the agent in the long run (it differs from the immediate reward concept). There are two types of value functions: state-value function that informs us about the total return we can expect in the future if we start from that state (usually denoted by $v_\pi(s)$) and action-value function, which refers to the expected return when the agent is in state s and performs action a following policy π (usually denoted by $q_\pi(s, a)$).

The state-value and action-value functions can be defined recursively using the Bellman equation (Sutton and Barto 2018), which is one of the central elements of many RL algorithms. It decomposes the value function into two parts: the immediate reward, plus discounted future values. This equation simplifies the computation of the value function in such a way that rather than summing up many time steps, the optimal solution of a complex problem can be found by breaking it down into simpler recursive subproblems and finding their optimal solutions (Torres 2020). To find the optimal policy, which maximises the expected reward in the long run, Bellman optimality equations supply the basis for iterative approaches as follows:

- (1) Dynamic programming (DP) (Bellman 1957) refers to a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment like MDP (model-based method). DP includes two main different approaches: value and policy iteration. Value iteration specifies the optimal policy in terms of a value function by iterating through Bellman equations. The value function is updated until the difference between two iterations is smaller than a low threshold. The optimal policy is then derived. Policy iteration directly learns the policy. It starts with an initial random policy, and iteratively updates the policy by first computing the associated value function (policy evaluation or prediction) and then improving the policy (policy improvement or control) (Yu, Liu, and Nemati 2020).

The utility of classic DP algorithms is limited in RL because them assuming a perfect model (which is unrealistic in most applications) and their high computational expense as we need to loop through all the states in every iteration (they can grow exponentially in size and the state space can be very large, or even infinite).

- (2) Monte Carlo (MC) methods are all about learning from experience (model-free). Any expected value can be approximated by sample means. So

it is required to play a bunch of episodes, gather the returns and average them. MC methods are for episodic tasks, where the interaction stops when an agent encounters a terminal state. Thus experience is divided into episodes, which eventually terminate no matter what actions are selected. If a model is unavailable, then it is particularly useful to estimate action-values (the values of state-action pairs) rather than state values. With a model, state-values alone do suffice to determine a policy; it is a matter of simply looking one step ahead and choosing whatever action leads to the best combination of reward and the next state. Without a model, however, state values alone are not sufficient. It is necessary to explicitly estimate the value of each action for the values to be useful for suggesting a policy (Sutton and Barto 2018). MC methods only provide us with a value for found states and actions, and if we never encounter a state, its value remains unknown. This will undermine our efforts in estimating the optimal action-value function. While performing an action, the agent ought to strike a balance between exploiting any knowledge acquired to date by optimally acting and exploring the unknown space to discover new efficacious actions. This exploration-exploitation trade-off dilemma is one of the most basic theoretical matters in RL. The commonest approach to achieve this trade-off is the ϵ -greedy strategy. According to this strategy, the agent will perform random action with probability ϵ . When training begins, ϵ is set at 1 and ensures that the environment is explored by the agent. With time, ϵ is reduced using a decay rate (a hyperparameter) to make a trade-off between exploring and exploiting.

- (3) Temporal-difference (TD) methods, like MC methods, can directly learn from raw experience without using a model of the environment's dynamics. However, like DP, TD methods can update estimates (step-by-step), partly based on other learned estimates (bootstrapping), without having to waiting for any final outcome. Q-learning is one of the widespread off-policy TD approaches in RL (Watkins and Dayan 1992). Likewise, the SARSA algorithm (Rummery and Niranjan 1994) is a representation for the on-policy TD approaches. As with MC methods, in Q-learning and SARSA action-value function $q(s_t, a_t)$ is estimated. Here off-policy means that the action of current state and the action of the next state do not come from the same policy. On the contrary in an on-policy approach, the action selection in the next state follows the same policy that enables the agent to take the action in the

current state (Farazi et al. 2020). Each experienced sample brings the current estimate $q(s_t, a_t)$ closer to optimal value. Q-learning starts with an initial estimate for each state-action pair. When an action a_t is taken in state s_t , which results in the next state s_{t+1} , the corresponding Q-value is updated with a combination of its current value and the TD error. The TD error is the difference between the current estimate $q(s_t, a_t)$ and the expected discounted return based on the experienced sample (Yu, Liu, and Nemati 2020). The Q-value of each state-action pair is stored in a table for a discrete state-action space. This tabular Q-learning converges to the optimal when all the state-action pairs are visited infinitely often and an appropriate exploration strategy and learning rate are chosen (Watkins and Dayan 1992).

- (4) Policy-based approach. Unlike the value-based approach, the policy-based approach does not require having to estimate the value of a certain state or a state-action pair, but directly searches for an optimal policy π . In policy-based approaches, typically a parameterised policy $\pi(a \vee s, \theta)$ is chosen and this parameter θ of policy $\pi(a \vee s, \theta)$ is gradually updated to maximise the expected return. This parameter update can either be done by a gradient-free or gradient-based approach (Deisenroth, Neumann, and Peters 2013). Gradient-based (PG) approaches are mostly used in existing RL algorithms. REINFORCE (Williams 1992) is the main MC policy gradient algorithm on which almost all more advanced and modern ones are based. Policy-based methods are very suitable for continuous action spaces because they can learn true stochastic policies unlike value-based methods (Torres 2020; Sutton and Barto 2018).
- (5) Actor-critic (AC) methods. Value-based algorithms are unable to handle those problems involving continuous (real-valued) and high-dimensional action spaces. In addition, as the agent learns to approximate the solution by the Bellman equations, the agent can resort to a near-optimal policy. In policy-based algorithms, gradient estimators can possess very wide variances (Konda and Tsitsiklis 2000). Furthermore with policy changes, the new gradient can be estimated regardless of earlier policies. Thus agents do not learn in relation to previously accumulated information. To cope with this limitation, the existing literature suggests adopting the actor-critic approach that combines both policy-based and value-based algorithms (Konda and Tsitsiklis 2000; Grondman et al. 2012). In the actor-critic approach, the agent is trained using two estimators: the critic function, which approximates and updates

the value function; the actor function, which controls the agent's policy-based behaviour. Depending on the value function derived from the critic function, the actor function's policy parameter is updated in the performance improvement direction. While the actor function controls the agent's policy-based behaviour, the critic function assesses the selected action according to the value function.

Furthermore, we obtain DRL methods when we use deep neural networks (DNN) to represent the state or observation, and/or to approximate any of the following RL components: value function, $\hat{v}(s, \theta)$ or $\hat{q}(s, a, \theta)$, policy $\pi(a \mid s, \theta)$, and model (state transition function and reward function). Here parameters θ are the weights in deep neural networks (Li 2018). For example, in value-methods like Q-learning, the Q value of each state-action pair is stored in a table. Each additional feature added to the state space leads to the number of Q-values that needs to be stored in the table to exponentially grow (Sutton and Barto 2018). To mitigate this curse of dimensionality, DNN and RL can be integrated to act as a function approximator. DRL is able to automatically and directly abstract and extract high-level features and semantic interpretations from input data to, hence, avoid delicate feature hand-crafting selection or complicated feature engineering for an individual task (Sze et al. 2017). Some very popular DRL algorithms that can be applied to PPC problems are identified as follows. Regarding value-based methods, we find: deep Q-network (DQN) (Mnih et al. 2015); double DQN (DDQN) (Van Hasselt, Guez, and Silver 2016); duelling DQN (Wang et al. 2016); prioritised experience replay for DQN (Schaul et al. 2016); rainbow (Hessel et al. 2018); recurrent experience replay in distributed RL (R2D2) (Kapturowski et al. 2019), among others. Some DRL actor-critics methods are deep deterministic policy gradient (DDPG) (Lillicrap et al. 2015); advantage actor critic (A2C, A3C) (Mnih et al. 2016); trust region policy optimisation (TRPO) (Schulman et al. 2015); proximal policy optimisation (PPO) (Schulman et al. 2017); decentralised distributed proximal policy optimisation (DD-PPO) (Wijmans et al. 2020) and soft actor critic (SAC) (Haarnoja et al. 2018).

Having introduced the main characteristics of the RL and DRL approaches, this paper offers a literature review to analyse and classify the RL research produced in the PPC context to highlight the main trends and research gaps in the respect.

3. Review methodology

The search for scientific articles applying RL to PPC was carried out in Scopus. The search was done with 'Article

Table 1. Number of publications per source.

Source	References
International Journal of Production Research	10
Expert Systems with Applications	7
Computers and Industrial Engineering	5
IEEE Access	5
Journal of Manufacturing Systems	5
CIRP Annals	4
IIE Transactions	4
Computers and Chemical Engineering	3
Investigación Operacional	3
Journal of Intelligent Manufacturing	3
Production Engineering	3
The International Journal of Advanced Manufacturing Technology	3
Applied Intelligence	2
Engineering Applications of Artificial Intelligence	2
IFAC-PapersOnLine	2
Simulation Modelling Practice and Theory	2
Others	118
Total	181

title, Abstract, Keywords' by combining the following keywords as follows: *reinforcement learning* AND (*manufacturing* OR *production planning and control* OR *supply chain* OR *inventory*). The time window is not defined. Published from 1994 to 2021, 610 scientific articles were initially collected during this process.

As a second step, three exclusion processes were conducted on the results. Firstly, duplicated articles were eliminated, which left 596 articles. Secondly, the title, abstract and keywords of the remaining articles were read to eliminate those that neither applied RL nor addressed PPC problems, which left 278. Thirdly, these articles were completely read to determine if they applied RL to PPC problems, and those that did not meet this requirement were ruled out. Finally, 181 papers (see Appendix 1) were selected for the literature review, of which 101 were scientific journal publications, 76 conference proceedings, and four book chapters. Table 1 presents, in frequency order, the scientific journals in which more than one article was published. The articles published in other journals, conference proceedings or book chapters are grouped as 'Others'. The results show that research into the application of RL to PPC problems is very dispersed in different conferences and scientific journals, where the International Journal of Production Research was that with more published research works in this area.

Note that 65% of the articles were published in the last 10 years, while 40% of publications appeared in the last 4 years (Figure 2). This denotes that researchers' interest in applying RL to PPC is considerably growing.

In order to analyse and classify the selected 181 references, a conceptual framework for the application of RL to PPC problems is proposed (Figure 3).

The first dimension of the conceptual framework identifies five relevant PPC areas based on the classification

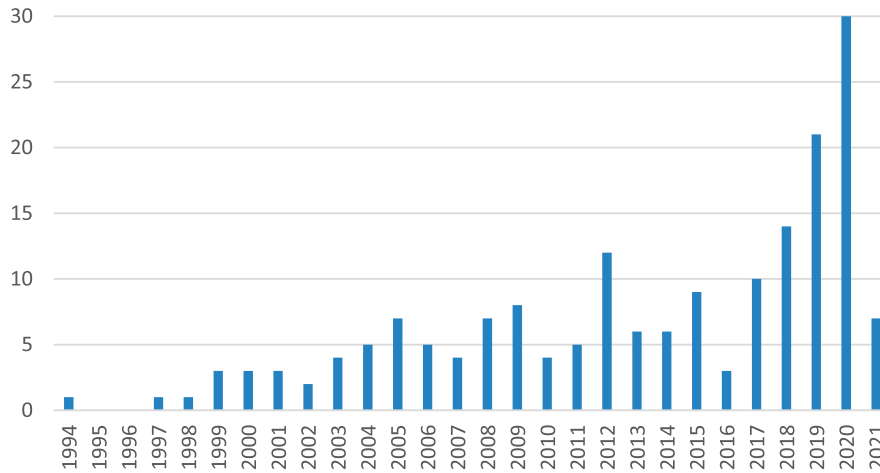


Figure 2. Number of publications per year.

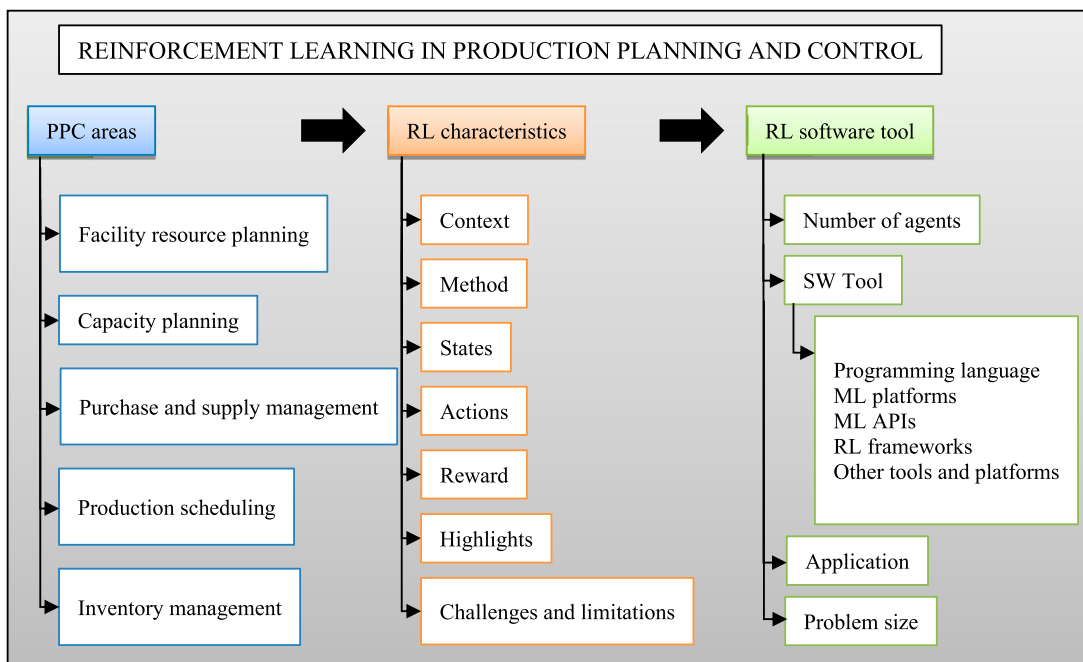


Figure 3. Conceptual framework for the application of RL to PPC.

by Jeon and Kim (2016). The main problems that can be addressed in all these areas are detailed below:

- Facility resource planning: determining location, resource management, layout design, maintenance planning and machine (re)configuration
- Capacity planning: establishing optimal quantity to generate products on a planning horizon by forecasting problems for demand uncertainties, and optimal capacity selections to set the total cost and product revenues
- Purchase and supply management: making decisions for suitable order times, order quantities, lot sizing, supply chain management network and transportation planning
- Production scheduling: shop floor scheduling, schedule management (queuing, slack time, due date, process sequence planning, bottleneck problems, equipment planning, machine routing, material processing planning, job-shop planning and management, and machine job sequence planning)
- Inventory management: product shortages or overstocks, lost orders and the forecast inventory turnover ratio problem

The second dimension identifies the main elements that define RL approaches, which are:

- Context: it defines the type of productive system addressed by the revised papers and/or the industrial sector

- Method: training algorithms (Q-learning, SARSA, DQN, A3C, PPO, among others) can be used to support models to learn which actions are more beneficial for the system based on its state
- States: identify the states considered by the proposed RL approaches. These states can be discrete or continuous depending on the employed training algorithm and the definition of the problem under study
- Actions: they identify the possible decisions to be made by the agent in existing RL approaches. Actions can also be discrete or continuous
- Rewards: they determine whether the selected action is good or bad by evaluating its impact on the state
- Highlights: they define the main contribution of each paper as identified by the authors of the revised papers
- Challenges and limitations: they identify the main limitations as well as the future research lines pointed out, in most cases, by the authors of the revised papers

Finally, the third dimension comprises four elements related to the software used to solve the RL algorithm:

- Number of agents: it identifies if the RL approach is developed for a single-agent or a multi-agent environment, understood as a system composed of multiple distributed entities that autonomously make decisions and interact in a shared environment (Weiss 1999)
- SW tool: it identifies whether RL has been implemented into high-level programming language (Python, Java, C++, Delphi C# or Visual Basic, NET, among others) or has been extracted from an existing library. It also identifies the rest of the software involved in solving RL: ML platforms (Tensorflow, Pythorch, DL4J); ML APIs (Keras, Google, Microsoft, Amazon); RL frameworks (TensorForce, Keras-RL, RLlib, Stable Baselines); and other tools and platforms or simulation software, such as Simpy, ARENA,

MATLAB, CSIM, Weka, Minitab, GAMBIT, and multi-agent platforms like JADE and MADKit

- Application: it defines whether the proposed RL algorithms are applied to a numerical study, a benchmark, a case study or a real-world problem
- Problem size: it identifies the number of each element considered in the problem addressed by the proposed RL models during experimentation

4. Literature review

This section examines the conceptual framework put forward to be used to analyse and classify the previously identified and selected RL proposals based on PPC areas, RL characteristics and RL software tools.

4.1. PPC areas

On the PPC areas dimension, the publication frequency of RL approaches to address the problems related to each area is analysed (Figure 4). The results show that RL approaches applied to PPC have focused mainly on production scheduling (60% of the papers). This problem is followed by purchase and supply management, which has been addressed by 18% of the papers. The remaining PPC has hardly been addressed with RL in the literature: 17 papers have dealt with facility resource planning, 20 papers have addressed inventory management problems, and four proposals have covered the capacity planning problem.

4.2. RL characteristics

In this section, the main RL characteristics of the revised papers are analysed and classified for each identified PPC area, namely: method, context, states, actions, reward and highlights.

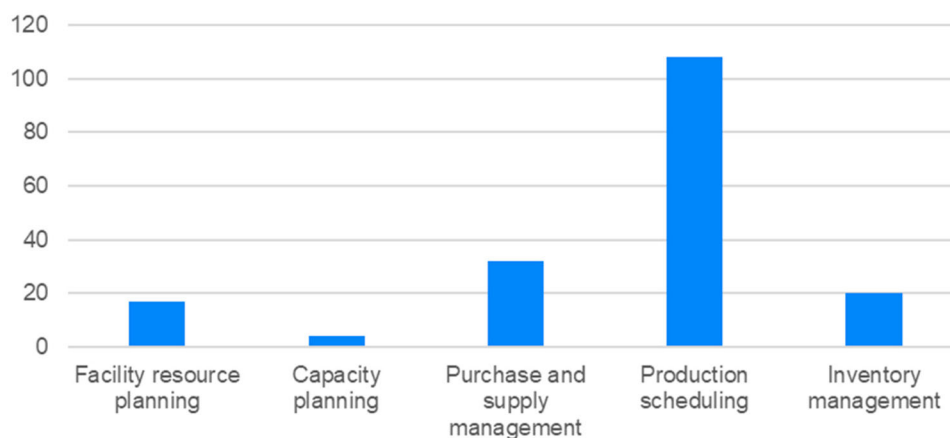


Figure 4. Number of RL approaches per PPC area.

On the facility resource planning problem (see Appendix 2), we identified that value-based approaches are the main RL algorithms used, and in this order of use: Q-learning, Q-batch learning, SMART, iSMART, Q-P learning and R-learning. Regarding the DRL value-based approaches, D convolutionary Q-network and DDQN are employed to a lesser extent. Only one article uses an actor-critic approach, and more concretely the PPO. In addition, one article follows the MC control algorithm.

As expected, the manufacturing system is the application context for the reviewed facility resource planning problems. Here problems are generally similar and do not address very complex machine maintenance and configuration problems. Thus the number of machines considered in the reviewed problems ranges from 1 to 10. Most of them address operational maintenance and tactical preventive maintenance problems (repair or replacement strategies), the majority of RL algorithms contemplate the states related to machines (e.g. machine age, remaining maintenance duration, machine's deterioration state). The second type of problem is operational machine reconfiguration, which contemplates the states related to productive cycle times or the machine's current configuration. Actions are mostly related to maintain machines (repair, corrective or preventive maintenance) or to do nothing about machine maintenance problems and select machine configuration (reset, align barcode reader, increase or reduce limits, increase or reduce pressure) or do nothing about machine reconfiguration problems. Revenue and costs are the main feedback rewarding signals, while inventory position and capacity loss are promptly considered. Rewards based on specific rules or performance measures are identified.

About capacity planning (Table 2), different authors follow several RL value-based methods to address tactical and operational decision problems. Here the oldest articles utilised RL algorithms (Q-learning and least square policy iteration) to define the production capacity of manpower or machines in a manufacturing system to minimise the related costs. About manpower hiring, workers with different knowledge levels (new-hired, semi-experience, fully-experienced) can be contracted, which affects inventory levels (states) and hiring costs (reward) (Karimi-Majd, Mahootchi, and Zakery 2017). About controlling operations, machines are switched on/off (actions) according to inventory levels (states) and costs (reward) (Zheng, Lei, and Chang 2017a, 2017b). The latest one was based on a DRL algorithm, DQN, applied to select the demand forecasting method (action) in a semiconductor supply chain: single exponential smooth, simple moving average, naïve forecast, Syntetos-Boylan approximation, artificial neuronal network, recurrent neuronal network and support vector

Table 2. RL characteristics for capacity planning.

Context	Method	Author/s	States	Actions	Reward	Highlights	Challenges and limitations
Manufacturing system	Q-learning (TD)	Karimi-Majd, Mahootchi, and Zakery (2017)	Surplus level or inventory level during the last period, number of workers at the second knowledge level, number of workers at the third knowledge level	Define the number of workers to be hired	Costs (workers depending on the knowledge level)	An optimisation model to hire workers that contemplates production and inventory levels, and demand uncertainty	A two-stage manufacturing system is considered. The extension to a multistage manufacturing system is indicated as further research. The proposed methodology can also be applied to call centres
	Least square policy iteration (PB)	Zheng, Lei, and Chang (2017a, b)	A machine showing a fault or not, at the buffer level	Turn on/off machines	Costs (production, permanent production loss, inventory level)	Two RL-based policies are proposed to control the operation of two machines and an intermediate buffer based on switching them on/off at the beginning of each time slot to reduce production costs, the penalty of permanent production loss and the inventory level cost	Extending the model to more machines and intermediate buffers is considered further research, as is improving the proposal to facilitate the control of real-time information
Supply chain	DQN (TD)	Chien, Lin, and Lin (2020)	Inventory level during the last period, shortage of supply, last historic demand, length of successive zero demand	Select a forecasting model	Forecasting error	A decision-making framework based on DRL for dynamically selecting the best demand forecasting model in a semiconductor supply chain with 157 products with widely varying demand data	Including big data in the forecasting model is a challenge. Developing artificial intelligence algorithms to self-tune the adjustment mechanism to incorporate business insights and domain knowledge is also proposed

regression. The selection of the forecasting model is dynamic and is based on the agent learning about inventory levels, shortage of supply, last historic demand, length of successive zero demand (states) and forecasting errors (reward received) (Chien, Lin, and Lin 2020).

Regarding purchase and supply planning (Appendix 3), and similarly to the facility resource planning problems, value-based approaches are the main applied RL algorithms and in this order of use: Q-learning, SARSA, SMART, average reward learning (ARL), FFRL and profit sharing. As for DRL approaches, the A2C actor-critic method is applied in one article (Barat et al. 2019) and two articles use the PPO method (Alves and Mateus 2020; Vanvuchelen, Gijsbrechts, and Boute 2020). Only one article follows the MC tree search algorithm.

In these cases, the supply chain is the most addressed application context, followed by the manufacturing systems, for purchase and supply planning problems. As expected, the tactical and operational decision levels are the most contemplated ones in the reviewed papers. Only one article considers the strategical decision level. For tactical decisions, production planning, inventory management, procurement and truck assignment and routes planning and bullwhip effect problems are addressed. With operational decisions, inventory control, vendor management inventory (VMI), production scheduling, production and procurement problems are modelled. The strategical decision level is related to a global supply chain management problem.

States are related mainly to inventory level, inventory position, demand, backorders, prices, among others. Actions focus mostly on defining the order quantity and/or the order up to level at the tactical and operational decision levels, while the strategical decision level considers the supplier selection and transportation modes (Pontrandolfo et al. 2002).

With feedback rewarding signals, economic aspects like costs, followed by revenues and profits, are those that come over more in the revised articles. Only one article considers the environmental aspect of sustainability for minimising waste in a grocery supply chain (Barat et al. 2019). Once again, only one article considers the social aspect of sustainability to define the service level as a reward (Jiang and Sheng 2009).

In production scheduling problems (Appendix 4), value-based approaches are the most frequently employed RL algorithms and are from the most to the least used as follows: Q-learning, approximate Q-learning, TD(λ) algorithm, SARSA, ARL, informed Q-learning, dual Q-learning, gradient descent TD(λ) algorithm, profit sharing, Q-III learning, relational RL, relaxed SMART, and TD(λ)-learning. For DRL, value-based approaches like DQN, deep Q-learning,

loosely-coupled DRL, multiclass DQN and Q-network algorithm have been utilised.

Three papers consider a policy gradient approach (Zheng, Gupta, and Serita 2020; Qu, Wang, and Jasperneite 2018; Gabel and Riedmiller 2012). Four articles apply actor-critic RL approaches, such as adapted Q-learning (Schneckenreither and Haeussler 2019), A2C (Hubbs et al. 2020a), TRPO (Kuhnle et al. 2019, 2021) and PPO (Park et al. 2021). One paper follows an MC approach (Tuncel, Zeid, and Kamarthi 2014).

In this case, scheduling and order acceptance operational problems are addressed in these contexts ordered from the most to the least frequently occurring ones: manufacturing system, job shop, flow shop, shop floor, work cell, single machine, multisite company, semiconductor industry, supply chain, workflow, automotive industry, cloud manufacturing, discrete automated production line, injection mould industry, parallel machines and wafer manufacturing shop.

States are related mainly to the number of jobs in the system or await to be processed per machine, job attributes, machine attributes, set up of machines, processing times, inventory level, availability of machines, mean tardiness and capacity of resources. Actions centre mostly on selecting a schedule rule, allocating jobs to machines, selecting the job to be processed, accepting/rejecting orders, among others. Feedback rewarding signals in scheduling problems do not only include economic measures like profits or costs, but most articles include rewards or penalties related to production time indicators, such as makespan, lateness and tardiness.

Inventory management problems (Appendix 5) are addressed mostly by value-based approaches, specifically and in order of use as follows: ARL, Q-learning, CMRL, fitted Q-iteration, R-learning, and TD(λ) algorithm. Regarding DRL approaches, the A3C actor-critic method is applied in one article (Kim, Jeong, and Shin 2020), two articles apply the PPO method, and another uses the DDPG and clipped PPO algorithms. On DRL value-based algorithms DQN, duelling DDQN and deep Q-learning are applied to solve inventory management problems.

In this case, the tactical and operational inventory management problem is addressed in manufacturing systems and supply chain contexts. Only one article addresses this problem in a different context: blood banks (Elkan 2012). Regarding states, most proposals include an inventory level or position, while others include information like customer demand or in-transit inventory. Actions focus mainly on defining replenishment or reorder quantities or defining certain inventory safety factors like safety stock, safety factor or safety lead time. In the blood bank-related proposal, not only

replenishment quantity is defined, but the allocation of the blood type to blood demand is also defined according to compatibilities among blood types.

For feedback rewarding signals, two main types of rewards are envisaged: rewards that promote the participants' economic sustainability, such as profits or inventory management costs, and rewards that promote social sustainability, such as service level.

4.3. RL software tools

In this section, the number of agents, software tools, and the application of each proposal, are summarised in Appendix 6. In application terms, a distinction is made in a numerical example that consists of using a small invented dataset to validate the proposed approach, a benchmark consisting of using an instance of data published in other sources (research articles or benchmark instances) to compare its performance and results, a case study that consists of using realistic simulated data, and a real-world problem in which real data from a real company are used. Additionally, the quality solution is identified in problem dataset size terms, i.e. number of products, machines or suppliers, among others. Regarding runtimes, agents' training times, which are the most important part of the solution time (Valluri, North, and MacAl 2009; Shiue, Lee, and Su 2018; Park et al. 2020), are not normally specified by the reviewed articles. As an illustrative example, Huang, Chang, and Chakraborty (2019) provide training times of around 7.5 h. Similarly, although execution times are not indicated, they are almost negligible compared to training times.

Almost 80% of the consulted works use a single agent in learning algorithms. With multi-agent approaches, agent training can be classified into two large groups: distributed, where each agent learns independently of other agents and no explicit information exchange takes place; centralised, where agents' policies are updated based on shared mutual information (Weiß 1995). Apart from training type, agents can be classified according to the way they select actions. There are two execution schemes: centralised, where agents are guided from a central unit that sets the joint actions for them all; decentralised, where agents determine actions according to their own individual policy. For a more detailed use of RL and DRL in multi-agent systems, readers are referred to Gronauer and Diepold (2021) and to Canese et al. (2021). Here multi-agent approaches are mainly developed with the JAVA programming language using multi-agent frameworks like JADE. Python has also recently appeared to support multi-agent approaches.

Continuing with programming languages, it is worth noting that until the current main ML platforms like

Tensorflow (Abadi et al. 2016) and Pytorch (Paszke et al. 2017, 2019) appeared, most approaches used general languages, such as C++, MATLAB, JAVA, Visual Basic.NET and Delphi, or simulation applications like ARENA. After these platforms were released and their ease of use with Python API, this language seems to be predominate for developing agents, although we ought not to forget the presence of MATLAB and JAVA. The use of Python is supported by the OpenAI Gym toolkit appearing (Brockman et al. 2016), which allows the development in this language of simulation environments for agents' autonomous learning. Before this tool emerged, it was once again necessary to build custom environments for learning agents. Even so, other tools are also used in different programming languages to simulate the learning environment, such as Simpy, MATLAB, SimTalk, Anylogic, among others.

The application of RL is no straightforward process as the different algorithms to be applied to the agent and the environment have to be tested and fine-tuned. For this purpose, integrated frameworks that facilitate this entire development and testing process are employed. These frameworks facilitate the creation of new algorithms, or the application and adjustment of algorithms that already exist in the literature (no need to programme from scratch) to new problem environments. For this reason, these frameworks have begun to appear in the last 5 years, and include Tensorforce (Kuhnle, Schaarschmidt, and Fricke 2017), Keras-RL (Plappert 2016) and Stable Baselines (Hill et al. 2018). It is noteworthy that there are now more frameworks than those indicated above, such as TF-Agents (Guadarrama et al. 2018), RL-Coach (Caspi et al. 2017), Acme (Hoffman et al. 2020), Dopamine (Castro et al. 2018), RLlib (Liang et al. 2017) and Stable Baselines3 (Raffin et al. 2021), among others. It is also worth highlighting the RL frameworks selection methodology proposed by Winder (2020), which contemplates quantitative classification criteria associated with GitHub repository statistics (stars, number of commits, number of committers, time since the project was launched, etc.) and criteria related to modularity, ease of use, flexibility and maturity. It also analyzes 15 frameworks, but does not contemplate others like Acme and Stable Baselines3. According to the methodology of Winder (2020), and by extending it to other frameworks, it is worth pointing out that if ML library Tensorflow is employed, using the Tf-Agents framework could be suitable. If ML library Pytorch is utilised, the Stable Baselines3 framework might also be a good initial option, an easy one to apply and one that comes with the available documentation. Yet despite the easy use of DNN API Keras, the frameworks that it uses do not arouse much interest. Finally, it is worth stressing the selection

of two frameworks, Acme and Rllib, via the ray project. Acme is quite a new framework (and does not come with complete documentation) that is mostly research-oriented. The design of its architecture is good and is used by DeepMind (one of the leading firms to invest in RL and AI). Rllib is an industrial-grade library for RL built on top of Ray. Ray is a distributed execution framework that allows programs in Python to be scaled thanks to Rllib via Ray, which is a relatively affordable form by means of developed documentation. Besides, it can work with both Tensorflow and Pytorch. In Ray, there are other libraries apart from Rllib that allow the scalable adjustment of hyperparameters and RaySGD to support distributed training algorithms.

Finally, it is worth indicating that the vast majority of the RL applications to PPC are carried out on numerical examples or case studies, which are not normally very large in size. Fewer than 9% of the contributions are applied to real-world problems. Thus the use of RL, and more currently DRL, leaves plenty of room for development in PPC, and generally in the operations research and industrial engineering areas. Indeed the work of Hubbs et al. (2020) develops OR-Gym, an open-source library for developing RL algorithms to address operations research problems.

5. Discussion

One of the first steps in applying RL is to properly define states and actions. States ought to be defined so that they come as closely as possible to the behavioural policy that generates data. It is preferable to have a detailed design of those data to help the agent learn the most appropriate policy. Yet enlarging the state space makes the model a more complex one to be solved. Therefore, deciding about good state representation is fundamental, and one that can include only those factors with a causal relation to the actions to be taken and outcomes. In order to reduce the state space size, most articles assume the discretisation of their values based on very simplified input data, which can generate accuracy losses in the solution. This makes it difficult to apply RL algorithms to more complex and realistic PPC problems with satisfactory results.

As regards actions, the vast majority of the analysed articles have also focused mainly on the discretisation of the action space and using discrete decision variables. This discrete formulation is quite reasonable in PPC when related to binary decisions (e.g. whether or not to activate a location of a resource or facility), combinatorial decisions (which product to sequence from those available in the next period on a machine), or integer decisions (purchase or production batches). However,

provided that the underlying problem contemplates continuous decision variables, the action space should also adopt this format to obtain quality results. Once again, although some DRL algorithms work with continuous spaces, such as actor-critic or policy-based methods, if the action space is large/infinite, proper action selection becomes more complicated. Moreover, one disadvantage of DRL is there is no optimality guarantee in these algorithms, which tend to converge to the local optimal instead of the global optimal. Furthermore, state and action spaces are becoming larger as the problem's complexity grows. This makes convergence less efficient in times and solution quality terms. This is one of the main challenges to be faced with RL algorithms.

Therefore, the RL algorithms in our literature review have been applied to adapted and simplified environments. RL is data-intensive, and it is necessary to have a history of data samples to feed agents' training to obtain a good policy. With no such real or representative samples, these types of solutions cannot be applied to enterprise environments. Moreover, if that data history contains biases, they will be learned by the agent itself. As previously mentioned, only a few analysed works have been applied to real environments. It is worth noting that in the I4.0 concept, cyber physical systems, Internet of things (IoT) and/or Internet of everything (IoE), together with traditional MES (manufacturing execution systems) and WMS (warehouse management systems), can help to capture the necessary samples to train agents in the PPC area.

The learning process typically takes place in the simulation environment with which the agent interacts. Appendix 5 identifies the different RL software tools applied in the literature to create these simulation environments: Anylogic, ARENA, MATLAB, Simpy, JADE, OpenAI Gym, among others. It should be noted that the vast majority of the analysed algorithms are free of models. This means that the agent which interacts with the (simulated) environment has no *a priori* knowledge about what will happen when it performs a certain action in that environment. Therefore, it learns to maximise its reward by trial-and-error. As the agent may have to make many trial-and-error attempts while learning, the simulator has to be efficient since, depending on the problem, the model may be difficult to build. Hence an important challenge to be addressed in RL algorithms is related to hyperparameter tuning, where it is necessary to investigate and develop a methodology to set the initial values so that algorithms learn more quickly and satisfactorily. So this structured approach to adjust hyperparameters is needed to reduce the number of trial-and-error attempts, which condition convergence speed.

One of the great advantages of RL and DRL is that once the agent is trained (although it may take time to train it), the time required to obtain a solution is practically instantaneous. The solution will be obtained from a sequence of accesses to a Q-table (in the case, tabular Q-learning) or will pass through a DNN (with a DRL algorithm). This feature is very useful for those use cases in the PPC area that are operational or real-time based. In these situations, the possibility of making a quick, or even a real-time decision, is a feature that should be taken into account. This partly explains why there are more contributions in the production scheduling area (Appendix 4).

The works herein shown present RL and DRL as a viable promising alternative in the PPC area as they have compared the results of the proposed algorithms to traditional resolution models (heuristics, metaheuristics and mathematical programming optimisation) and obtained similar, or even better, results in different scenarios. This is because of another advantage of DRL: it is easier to incorporate uncertainty or non-linear relations into simulation than, for example, into alternative mathematical resolution models. So it is necessary to bear in mind that if there are solutions based on traditional operations research models that are efficient in time, money or computational capacity terms, it is not necessary to resort to other types of solution approaches like those based on RL. This is because these models have better solution guarantees than many RL algorithms and their behaviour is well-known. However, if uncertainty and/or non-linear relations have to be incorporated into a problem, traditional stochastic and DP models quickly produce the curse of dimensionality, which can considerably prolong resolution times. Additionally, many problems in the PPC area can be very diverse and/or complex and are much easier to simulate than formulating a mathematical programming model. Therefore, we suggest readers using traditional mathematical programming models for PPC problems in static and determinist contexts when they can obtain satisfactory solutions in a reasonable computational time. RL algorithms are recommended for PPC scenarios in dynamic and uncertain contexts that contemplate Industry 4.0 issues such as: online applications, real-time data, cyber physical production systems, the combination of production systems and automated guided vehicles (AGVs), among others.

Notwithstanding, there are limitations to the use of RL algorithms which, being mainly model-free and with no reference model, have difficulties in responding to changes in the environment. For example, if the demand pattern changes for an agent trained in the purchase and supply chain management area, the quality of these decisions can drastically decrease. In some cases, retraining

agents is necessary to avoid loss of quality decisions. Here self-tuning approaches based on AI algorithms are proposed as a challenge by several reviewed articles.

It is noteworthy that no more than 20% of the studied articles in the literature adopt the multi-agent approach, whose training requires higher computational capacity because the space of actions and states substantially increases with number of agents. All the multi-agent approaches herein analysed apply the centralised training type, and most of the analysed works adopt a decentralised execution scheme. According to Stone and Veloso (2000), it is not possible to completely solve many real-world problems by only one active agent interacting with the environment. Indeed several problems in the purchase and supply management area adopt multi-agent approaches because they involve different companies that belong to one same supply chain. This makes it difficult to use a single agent with all the necessary information to both model and overcome them.

Over the years, RL algorithms' complexity and scale have significantly grown. In this scenario, programming an RL implementation from scratch can be a tedious job and might pose a high risk of programming errors. To address this, the RL community started building frameworks and libraries to simplify the development of RL algorithms. These libraries offer a number of built-in algorithms (both single-agent and multi-agent) and also facilitate the creation of custom algorithms. These frameworks are also prepared to take advantage of the high computational capacity that can be achieved only with parallel and distributed systems composed of heterogeneous servers with multicore CPUs and hardware accelerators GPU and TPU (Torres 2020). In this literature review, the use of these libraries and frameworks has started being identified (especially in recent years). In the present and immediate future, the use of these tools is expected to become more widespread for both the accomplished time savings and meeting the increased computational demands of most current RL problems. Hence the difficulty of deciding which of all the available RL frameworks is the most suitable one to be used in the PPC domain. This is because there are many available options (some of which have been abandoned) and new frameworks have been constantly created in recent years. Besides, selecting a specific framework can mean having to invest learning time, which can later complicate the possibility of changing to another different (albeit assumedly better) framework. Based on the selection methodology of Wan et al. (2020), Ray via RLlib enables rapid scaling, facilitates the adjustment of hyperparameters and applies distributed training. Thus it is considered to be one of the most suitable ones for the real implementation of RL applications in the PPC domain.

Thus the following research gaps were identified: (i) more RL approaches should be applied and tested in facility resource planning, capacity planning and inventory management problems, and generally in more real-world PPC problems; (ii) tactical and strategical decision levels can be addressed with RL and DRL approaches; for instance, supply chain design, facility location, aggregate planning, master production scheduling and material requirement planning are problems that can be focussed on; (iii) environmental and social aspects of sustainability have hardly been dealt with in many of the revised RL approaches, which have been oriented mostly on the economical aspect. Therefore, RL-PPC models should contemplate the triple bottom line of sustainability; (iv) comparative studies between the different RL and DRL methods applied to similar real-world problems are scarce. Hence the need to compare distinct RL and RDL methods in computational efficiency terms per PPC area; (v) the performance of RL and DRL solutions in more generic and comprehensive environments should be evaluated to identify the best solutions, which might also be robust in different environments; (vi) lack of robustness of model-free algorithms to face changes in the environment, which can be solved by retraining the agent or applying the transfer learning concept. If the agent is retrained from the beginning, all the accumulated knowledge will be discarded. However, the human brain does not work like that because it does not discard previously obtained knowledge when solving a new

task. Transfer learning attempts to mimic this behaviour to retrain agents that have already been applied to similar situations to obtain a much faster convergence when learning the new underlying policy (Garnier et al. 2021); (vii) agent training can be done by means of hybrid systems that can combine RL with optimisation methods. In this way, while learning agents can consult the output of a mathematical programming solver to learn how to use certain actions in the future without having to run it again (Hubbs et al. 2020b); (viii) the practitioner and academic use of RL frameworks and libraries should become more general due to both the savings made in modelling times and to face the greater computing demands generated by the most current real-world RL problems.

Finally, Figure 5 summarises the main limitations and challenges of RL applied to PPC problems.

6. Conclusions

This paper presents a thorough literature review on RL applications in the PPC field. The collected, selected and analysed papers were classified according to the following PPC areas: facility resource planning, capacity planning, purchase and supply management, production scheduling and inventory management. RL approaches have focused mainly on production scheduling and order acceptance operational problems followed by the purchase and supply management area, where tactical problems related to production planning, inventory

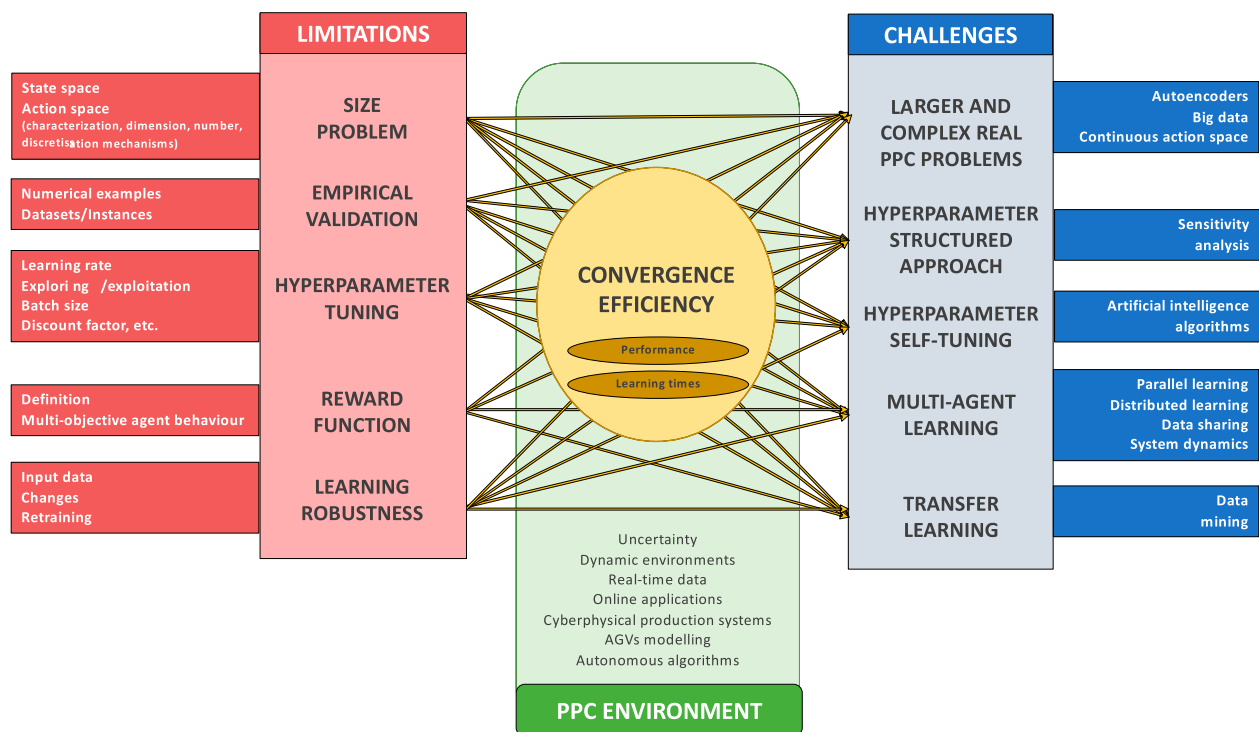


Figure 5. Main limitations and challenges of RL applied to PPC.

management, procurement and truck assignment and routes planning and bullwhip effect problems, and operational problems related to inventory control, VMI, production scheduling and production and procurement planning, are mostly contemplated. Moreover, the strategic decision level has been considered in a global supply chain management problem.

Tactical and operational inventory management problems have been addressed in manufacturing systems and supply chains, although one article has addressed this problem for blood banks. Regarding facility resource planning, operational maintenance, tactical preventive maintenance and operational machine reconfiguration are the main problems to have been addressed. Very few RL approaches have dealt with capacity planning, and from the tactical and operational decision levels, they have modelled and solved manpower hiring, machine control and forecasting method selection problems.

In relation to the RL method, RL value-based approaches are the most frequently used specifically for Q-learning and their variants, SMART and their variants, ARL, SARSA and profit sharing. For DRL approaches, DQN and their variants, Q-network and policy gradient methods have been utilised. Regarding actor-critic based approaches, PPO is the most widespread followed by A2C, A3C, DDPG and TRPO approaches. Finally, a few MC approaches have been adopted in the facility resource planning, purchase and supply management and production scheduling areas.

Managerial implications aim to provide an overview of the possible applications of RL and DRL approaches for PPC practitioners and academics. The usefulness of this literature review for PPC managers is twofold: on the one hand, to serve as a general guide to find similar PPC problems to be solved with RL and DRL approaches and, on the other hand, to also provide a guide that proposes new approaches to support PPC problems with a view to select RL software tools, which have been extensively discussed, and to define states, actions and rewards in similar problems to be addressed.

It should be highlighted that this literature review has some limitations. The consulted database is Scopus, which is constantly being updated and the provided data correspond to those obtained at the time when the research was conducted. Here we review the literature published until February, 2021. In the meantime, several new studies on RL in PPC problems have appeared (Kuhnle et al. 2021; Panzer and Bender, 2022; Yang and Xu, 2021, among others) which corroborates the great interest in this research area. Furthermore, despite having followed a systematic search process, some valuable papers could have been overlooked for this review. For instance, we avoided specific searching

words like ‘scheduling’ because it contemplates an extensive research area (around 2,000 papers when combining ‘reinforcement learning’ and ‘scheduling’), which requires more specific reviews, and even addresses fields with production control regarding computers or robotics scheduling tasks, which have been addressed by other specific reviews. In any case, some limitations that were revealed while conducting the study are an opportunity for new research lines and forthcoming works. Thus, we consider it necessary to develop more specific literature reviews oriented to RL and scheduling (Kayhan and Yildiz 2021). Furthermore, it would be desirable to extend RL and DRL approaches to new real-world strategic and tactical problems in the PPC areas of facility resource design specifically for supply chain design and facility and warehouse location problems, and also in the capacity planning and purchase and supply management areas to model and solve aggregate planning, lot-sizing and scheduling (Rummukainen and Nurminen 2019; Lang et al. 2020; Zhang et al. 2020), and even logistics problems (Rabe and Dross 2015). Then these RL and DRL approaches could be automatically and autonomously connected to operational production scheduling and inventory management problems to look for self-managed PPC systems as claimed in the I4.0 era.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The funding for the research work that has led to the obtained results came from the following grants: CADSA4.0 (Ref. RTI2018-101344-B-I00) and NIOTOME (Ref. RTI2018-102020-B-I00), financed by MCIN/AEI/10.13039/501100011033 and ‘ERDF A way of making Europe’; the EU H2020 research and innovation programme with grant numbers 825631 ‘Zero-Defect Manufacturing Platform (ZDMP)’ and 958205 ‘Industrial Data Services for Quality Control in Smart Manufacturing (i4Q)’; ‘Industrial Production and Logistics Optimization in Industry 4.0’ (i4OPT) (Ref. PROM-ETEO/2021/065) and ‘Resilient, Sustainable and People-Oriented Supply Chain 5.0 Optimization Using Hybrid Intelligence’ (RESPECT) (Ref. CIGE/2021/159) Projects were funded by the Generalitat Valenciana (Valencian Regional Government).

Notes on contributors



Ana Esteso is Assistant Professor in the Department of Business Management of the Universitat Politècnica de València (UPV), Spain. Her teaching focuses on subjects related to Operations Management, Business Management and Operations Research at the UPV’s School of Industrial Engineering. She is member of

the Research Centre on Production Management and Engineering (CIGIP) of the UPV. Her research focuses on the development of operational research tools to support sustainable supply chain design and management and production planning, mainly in the agri-food and ceramics sectors. She has participated in research projects funded by the European Commission, the Spanish Government, the Generalitat Valenciana. As a result, she has published 16 papers in international journals and 20 papers in Conferences in the last five years.



David Peidro is Associate Professor in the Department of Business Management of the Universitat Politècnica de València (UPV), Spain. His research interest focuses on machine learning and intelligent optimisation applied to supply chain management, logistics, production planning and inventory management. He has

participated in different research projects funded by the European Commission, the Spanish Government, the Valencian Regional Government and the UPV. He is member of the Research Centre on Production Management and Engineering (CIGIP). He has published more than forty articles in international conferences and journals such as International Journal of Production Research, European Journal of Operations Research, International Journal of Production Economics, Computers and Industrial Engineering, Applied Soft Computing, Computers & Mathematics with Applications Applied Mathematical Modelling and Fuzzy Sets and Systems.



Josefa Mula is Professor in the Department of Business Management of the Universitat Politècnica de València (UPV), Spain. She is a member of the Research Centre on Production Management and Engineering (CIGIP) of the UPV. Her teaching and principal research interests concern production engineering and management, operations research and supply chain simulation. She is editor in chief of the International Journal of Production Management and Engineering. She regularly acts as associate editor, guest editor and member of scientific boards of international journals and conferences, and as referee for more than 50 scientific journals. She is author of more than 120 papers mostly published in international books and high-quality journals, among which International Journal of Production Research, Fuzzy Sets and Systems, International Journal of Production Economics, European Journal of Operational Research, Computers and Industrial Engineering and Journal of Manufacturing Systems, among others.

She is editor in chief of the International Journal of Production Management and Engineering. She regularly acts as associate editor, guest editor and member of scientific boards of international journals and conferences, and as referee for more than 50 scientific journals. She is author of more than 120 papers mostly published in international books and high-quality journals, among which International Journal of Production Research, Fuzzy Sets and Systems, International Journal of Production Economics, European Journal of Operational Research, Computers and Industrial Engineering and Journal of Manufacturing Systems, among others.



Manuel Díaz-Madroñero is Associate Professor in the Department of Business Management of the Universitat Politècnica de València (UPV), Spain. He teaches subjects related to Information Systems, Operational Research and Operations Management and Logistics. He is member of the Research Centre on Production Management and Engineering (CIGIP) of the UPV. He has participated in different research projects funded by the European Commission, the Spanish Government, the Valencian Regional Government and the UPV. As a result, he has published (in collaboration) more than forty articles in different indexed journals and international conferences. He is co-author of the

book *Operations Research Problems: Statements and Solutions* (Springer, 2014). His research areas include production planning and transportation, fuzzy mathematical programming and robust optimisation, multicriteria decision making and sustainable operations management.

book *Operations Research Problems: Statements and Solutions* (Springer, 2014). His research areas include production planning and transportation, fuzzy mathematical programming and robust optimisation, multicriteria decision making and sustainable operations management.

Data availability statement

The authors confirm that the data supporting the findings of this study are available within the article and its supplementary materials.

ORCID

Ana Esteso  <http://orcid.org/0000-0003-0379-8786>

David Peidro  <http://orcid.org/0000-0001-8678-6881>

Josefa Mula  <http://orcid.org/0000-0002-8447-3387>

Manuel Díaz-Madroñero  <http://orcid.org/0000-0003-1693-2876>

References

- Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al. 2016. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." <https://www.tensorflow.org/>.
- Alves, Júlio César, and Geraldo Robson Mateus. 2020. "Deep Reinforcement Learning and Optimization Approach for Multi-Echelon Supply Chain with Uncertain Demands." *Lecture Notes in Computer Science*, 584–599. doi:10.1007/978-3-030-59747-4_38.
- Barat, Souvik, Harshad Khadilkar, Hardik Meisheri, Vinay Kulkarni, Vinita Baniwal, Prashant Kumar, and Monika Gajrani. 2019. "Actor Based Simulation for Closed Loop Control of Supply Chain Using Reinforcement Learning." *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 3*, 1802–1804.
- Bellman, Richard. 1957. *Dynamic Programming*. Princeton, New Jersey: Princeton University Press.
- Briegel, Hans J, and Gemma las Cuevas. 2012. "Projective Simulation for Artificial Intelligence." *Scientific Reports* 2 (1): 1–16.
- Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. "OpenAI Gym." <http://arxiv.org/abs/1606.01540>.
- Bueno, Adauto, Moacir Godinho Filho, and Alejandro G. Frank. 2020. "Smart Production Planning and Control in the Industry 4.0 Context: A Systematic Literature Review." *Computers & Industrial Engineering* 149 (November): 106774. doi:10.1016/j.cie.2020.106774.
- Cañas, Héctor, Josefa Mula, and Francisco Campuzano-Bolarín. 2020. "A General Outline of a Sustainable Supply Chain 4.0." *Sustainability* 12 (19): 7978. doi:10.3390/su12197978.
- Cañas, Héctor, Josefa Mula, Manuel Díaz-Madroñero, and Francisco Campuzano-Bolarín. 2021. "Implementing Industry 4.0 Principles." *Computers & Industrial Engineering* 158 (August): 107379. doi:10.1016/j.cie.2021.107379.

- Canese, Lorenzo, Gian Carlo Cardarilli, Luca Di Nunzio, Rocco Fazzolari, Daniele Giardino, Marco Re, and Sergio Spanò. 2021. "Multi-Agent Reinforcement Learning: A Review of Challenges and Applications." *Applied Sciences* 11 (11): 4948.
- Caspi, Itai, Gal Leibovich, Gal Novik, and Shadi Endrawis. 2017. "Reinforcement Learning Coach." doi:10.5281/zenodo.1134899.
- Castro, Pablo Samuel, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G Bellemare. 2018. "Dopamine: A Research Framework for Deep Reinforcement Learning." <http://arxiv.org/abs/1812.06110>.
- Chien, Chen Fu, Yun Siang Lin, and Sheng Kai Lin. 2020. "Deep Reinforcement Learning for Selecting Demand Forecast Models to Empower Industry 3.5 and an Empirical Study for a Semiconductor Component Distributor." *International Journal of Production Research* 58 (9): 2784–2804. doi:10.1080/00207543.2020.1733125.
- Cunha, Bruno, Ana M. Madureira, Benjamim Fonseca, and Duarte Coelho. 2020. "Deep Reinforcement Learning as a Job Shop Scheduling Solver: A Literature Review." *Intelligent Decision Support Systems – A Journey to Smarter Healthcare*, 350–359. doi:10.1007/978-3-030-14347-3_34.
- Deisenroth, Marc Peter, Gerhard Neumann, and Jan Peters. 2013. *A Survey on Policy Search for Robotics*. Now Publishers.
- Elkan, Charles. 2012. "Reinforcement Learning with a Bilinear Q Function." *Lecture Notes in Computer Science*, 78–88. doi:10.1007/978-3-642-29946-9_11.
- Farazi, Nahid Parvez, Tanvir Ahamed, Limon Barua, and Bo Zou. 2020. "Deep Reinforcement Learning and Transportation Research: A Comprehensive Review." *ArXiv Preprint ArXiv:2010.06187*.
- Gabel, Thomas, and Martin Riedmiller. 2012. "Distributed Policy Search Reinforcement Learning for Job-Shop Scheduling Tasks." *International Journal of Production Research* 50 (1): 41–61. doi:10.1080/00207543.2011.571443.
- Garnier, Paul, Jonathan Viquerat, Jean Rabault, Aurélien Larcher, Alexander Kuhnle, and Elie Hachem. 2021. "A Review on Deep Reinforcement Learning for Fluid Mechanics." *Computers & Fluids* 225: 104973. doi:10.1016/j.compfluid.2021.104973.
- Gronauer, Sven, and Klaus Diepold. 2021. "Multi-Agent Deep Reinforcement Learning: A Survey." *Artificial Intelligence Review* 55 (2): 895–943.
- Grondman, Ivo, Lucian Busoniu, Gabriel A D Lopes, and Robert Babuska. 2012. "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42 (6): 1291–1307.
- Guadarrama, Sergio, Anoop Korattikara, Oscar Ramirez, Pablo Castro, Ethan Holly, Sam Fishman, Ke Wang, et al. 2018. "TF-Agents: A Library for Reinforcement Learning in TensorFlow." <https://github.com/tensorflow/agents>.
- Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor." *ArXiv Preprint ArXiv:1801.01290*.
- Han, Miyoung. 2018. "Reinforcement Learning Approaches in Dynamic Environments."
- Hessel, Matteo, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. "Rainbow: Combining Improvements in Deep Reinforcement Learning." In *Proceedings of the AAAI Conference on Artificial Intelligence*. 32.
- Hill, Ashley, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, et al. 2018. "Stable Baselines." *GitHub Repository*. GitHub.
- Hoffman, Matt, Bobak Shahriari, John Aslanides, Gabriel Barth-Maron, Feryal Behbahani, Tamara Norman, Abbas Abdolmaleki, et al. 2020. "Acme: A Research Framework for Distributed Reinforcement Learning." <https://arxiv.org/abs/2006.00979>.
- Huang, Jing, Qing Chang, and Nilanjan Chakraborty. 2019. "Machine Preventive Replacement Policy for Serial Production Lines Based on Reinforcement Learning." In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 523–528. IEEE. doi:10.1109/COASE.2019.8843338.
- Hubbs, Christian D., Can Li, Nikolaos V. Sahinidis, Ignacio E. Grossmann, and John M. Wassick. 2020a. "A Deep Reinforcement Learning Approach for Chemical Production Scheduling." *Computers and Chemical Engineering* 141: 106982. doi:10.1016/j.compchemeng.2020.106982.
- Hubbs, Christian D, Hector D Perez, Owais Sarwar, Nikolaos V Sahinidis, Ignacio E Grossmann, and John M Wassick. 2020b. "OR-Gym: A Reinforcement Learning Library for Operations Research Problems."
- Ivanov, Dmitry, Christopher S. Tang, Alexandre Dolgui, Daria Battini, and Ajay Das. 2021. "Researchers' Perspectives on Industry 4.0: Multi-Disciplinary Analysis and Opportunities for Operations Management." *International Journal of Production Research* 59 (7): 2055–2078. doi:10.1080/00207543.2020.1798035.
- Jeon, Su Min, and Gitae Kim. 2016. "A Survey of Simulation Modeling Techniques in Production Planning and Control (PPC)." *Production Planning & Control* 27 (5): 360–377. doi:10.1080/09537287.2015.1128010.
- Jiang, Chengzhi, and Zhaohan Sheng. 2009. "Case-Based Reinforcement Learning for Dynamic Inventory Control in a Multi-Agent Supply-Chain System." *Expert Systems with Applications* 36: 6520–6526. doi:10.1016/j.eswa.2008.07.036.
- Kapturovski, Steven, Georg Ostrovski, Will Dabney, John Quan, and Remi Munos. 2019. "Recurrent Experience Replay in Distributed Reinforcement Learning." In *International Conference on Learning Representations*. <https://openreview.net/forum?id=r1lyTjAqYX>.
- Karimi-Majd, Amir Mohsen, Masoud Mahootchi, and Amir Zakery. 2017. "A Reinforcement Learning Methodology for a Human Resource Planning Problem Considering Knowledge-Based Promotion." *Simulation Modelling Practice and Theory* 79: 87–99. doi:10.1016/j.simpat.2015.07.004.
- Kayhan, Behice Meltem, and Gokalp Yildiz. 2021. "Reinforcement Learning Applications to Machine Scheduling Problems: A Comprehensive Literature Review." *Journal of Intelligent Manufacturing*, doi:10.1007/s10845-021-01847-3.
- Kim, Byeongseop, Yongkuk Jeong, and Jong Gye Shin. 2020. "Spatial Arrangement Using Deep Reinforcement Learning to Minimise Rearrangement in Ship Block Stockyards." *International Journal of Production Research* 58 (16): 5062–5076. doi:10.1080/00207543.2020.1748247.
- Konda, Vijay R, and John N Tsitsiklis. 2000. "Actor-Critic Algorithms." *Advances in Neural Information Processing Systems* 12: 1008–1014.

- Kuhnle, Andreas, Johannes Jakubik, and Gisela Lanza. 2019. "Reinforcement learning for opportunistic maintenance optimization." *Production Engineering* 13 (1): 33–41. <http://dx.doi.org/10.1007/s11740-018-0855-7>.
- Kuhnle, Andreas, Jan-Philipp Kaiser, Felix Theiß, Nicole Stricker, and Gisela Lanza. 2021. "Designing an adaptive production control system using reinforcement learning." *Journal of Intelligent Manufacturing* 32 (3): 855–876. <http://dx.doi.org/10.1007/s10845-020-01612-y>.
- Kuhnle, Andreas, Marvin Carl May, Louis Schäfer, and Gisela Lanza. 2021. "Explainable reinforcement learning in production control of job shop manufacturing system." *International Journal of Production Research*: 1–23. <http://dx.doi.org/10.1080/00207543.2021.1972179>.
- Kuhnle, Alexander, Michael Schaarschmidt, and Kai Fricke. 2017. "Tensorforce: A TensorFlow Library for Applied Reinforcement Learning." <https://github.com/tensorforce/tensorforce>.
- Lambert, Douglas M, and Martha C Cooper. 2000. "Issues in Supply Chain Management." *Industrial Marketing Management* 29 (1): 65–83. doi:10.1016/S0019-8501(99)00113-3.
- Lang, Sebastian, Fabian Behrendt, Nico Lanzerath, Tobias Reggelin, and Marcel Muller. 2020. "Integration of Deep Reinforcement Learning and Discrete-Event Simulation for Real-Time Scheduling of a Flexible Job Shop Production." In *2020 Winter Simulation Conference (WSC)*, 3057–3068. IEEE. doi:10.1109/WSC48552.2020.9383997.
- Li, Yuxi. 2018. "Deep Reinforcement Learning." *ArXiv Preprint ArXiv:1810.06339*.
- Liang, Eric, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Joseph Gonzalez, Ken Goldberg, and I. Stoica. 2017. "Ray RLlib: A Composable and Scalable Reinforcement Learning Library." *ArXiv abs/1712.0*.
- Lillicrap, Timothy P, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. "Continuous Control with Deep Reinforcement Learning." *ArXiv Preprint ArXiv:1509.02971*.
- Mehra, A. 1995. *Hierarchical Production Planning for Job Shops*. College Park: University of Maryland, Harvard University and Industry.
- Mnih, Volodymyr, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. "Asynchronous Methods for Deep Reinforcement Learning." In *International Conference on Machine Learning*, 1928–1937.
- Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, et al. 2015. "Human-Level Control Through Deep Reinforcement Learning." *Nature* 518 (7540): 529–533.
- Mula, Josefa, and Marija Bogataj. 2021. "OR in the Industrial Engineering of Industry 4.0: Experiences from the Iberian Peninsula Mirrored in CJOR." *Central European Journal of Operations Research*. doi:10.1007/s10100-021-00740-x.
- Panzer, Marcel, and Benedict Bender. 2022. "Deep reinforcement learning in production systems: a systematic literature review." *International Journal of Production Research* 60 (13): 4316–4341. <http://dx.doi.org/10.1080/00207543.2021.1973138>.
- Park, Junyoung, Jaehyeong Chun, Sang Hun Kim, Youngkook Kim, and Jinkyoo Park. 2021. "Learning to Schedule Job-Shop Problems: Representation and Policy Learning Using Graph Neural Network and Reinforcement Learning." *International Journal of Production Research* 59 (11): 3360–3377. doi:10.1080/00207543.2020.1870013.
- Park, In Beom, Jaeseok Huh, Joongkyun Kim, and Jonghun Park. 2020. "A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities." *IEEE Transactions on Automation Science and Engineering* 17 (3): 1420–1431. doi:10.1109/TASE.2019.2956762.
- Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. "Automatic Differentiation in PyTorch." In *31st Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, USA.
- Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In *Advances in Neural Information Processing Systems* 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, 8024–8035. Curran Associates. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- Plappert, Matthias. 2016. "Keras-RL." *GitHub Repository*. GitHub.
- Pontrandolfo, P., A. Gosavi, O. G. Okogbaa, and T. K. Das. 2002. "Global Supply Chain Management: A Reinforcement Learning Approach." *International Journal of Production Research* 40 (6): 1299–1317. doi:10.1080/00207540110118640.
- Qiao, B., and J. Zhu. 2000. *Agent-Based Intelligent Manufacturing System for the 21st Century*. Nanjing: Mechatronic Engineering Institute, Nanjing University of Aeronautics and Astronautics.
- Qu, Shuhui, Jie Wang, and Juergen Jasperneite. 2018. "Dynamic Scheduling in Large-Scale Stochastic Processing Networks for Demand-Driven Manufacturing Using Distributed Reinforcement Learning." *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2018-Sept (1)*. IEEE, 433–440. doi:10.1109/ETFA.2018.8502508.
- Rabe, Markus, and Felix Dross. 2015. "A Reinforcement Learning Approach for a Decision Support System for Logistics Networks." In *2015 Winter Simulation Conference (WSC)*, 2020–2032. IEEE. doi:10.1109/WSC.2015.7408317.
- Raffin, Antonin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. "Stable-Baselines3: Reliable Reinforcement Learning Implementations." *Journal of Machine Learning Research* 22 (268): 1–8.
- Rummery, Gavin A, and Mahesan Niranjan. 1994. *On-Line Q-Learning Using Connectionist Systems*. Cambridge: University of Cambridge, Department of Engineering Cambridge.
- Rummukainen, Hannu, and Jukka K. Nurminen. 2019. "Practical Reinforcement Learning – Experiences in Lot Scheduling Application." *IFAC-PapersOnLine* 52 (13): 1415–1420. doi:10.1016/j.ifacol.2019.11.397.
- Russell, Stuart J, and Peter Norvig. 2003. *Artificial Intelligence: A Modern Approach*. New Jersey: Pearson Education.
- Schaul, Tom, John Quan, Ioannis Antonoglou, and David Silver. 2016. "Prioritized Experience Replay." *ArXiv Preprint ArXiv:1511.05952*.

- Schneckenreither, Manuel, and Stefan Haeussler. 2019. "Reinforcement Learning Methods for Operations Research Applications: The Order Release Problem." *Lecture Notes in Computer Science* 11331: 545–559. doi:10.1007/978-3-030-13709-0_46.
- Schulman, John, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. "Trust Region Policy Optimization." In *International Conference on Machine Learning*, 1889–1897.
- Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. "Proximal Policy Optimization Algorithms." *ArXiv Preprint ArXiv:1707.06347*.
- Serrano-Ruiz, J. C., J. Mula, D. Peidro, and M. Díaz-Madroño. 2021. "A Metamodel for the Supply Chain 4.0." *Journal of Industrial Integration Information*. Under Review.
- Shiue, Yeou Ren, Ken Chuan Lee, and Chao Ton Su. 2018. "Real-Time Scheduling for a Smart Factory Using a Reinforcement Learning Approach." *Computers and Industrial Engineering* 125 (101): 604–614. doi:10.1016/j.cie.2018.03.039.
- Stone, Peter, and Manuela Veloso. 2000. "Multiagent Systems: A Survey from a Machine Learning Perspective." *Autonomous Robots* 8 (3): 345–383.
- Sutton, Richard S, and Andrew G Barto. 2018. *Reinforcement Learning: An Introduction*. Cambridge: MIT Press.
- Sze, Vivienne, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. 2017. "Efficient Processing of Deep Neural Networks: A Tutorial and Survey." *Proceedings of the IEEE* 105 (12): 2295–2329.
- Szepesvári, Csaba. 2010. "Algorithms for Reinforcement Learning." *Synthesis Lectures on Artificial Intelligence and Machine Learning* 4 (1): 1–103. doi:10.2200/S00268ED1V01Y201005AIM009.
- Torres, Jordi. 2020. "Deep Reinforcement Learning Explained." <https://torres.ai/deep-reinforcement-learning-explained-series/>.
- Tuncel, Emre, Abe Zeid, and Sagar Kamarthi. 2014. "Solving Large Scale Disassembly Line Balancing Problem with Uncertainty Using Reinforcement Learning." *Journal of Intelligent Manufacturing* 25 (4): 647–659. doi:10.1007/s10845-012-0711-0.
- Usuga, Cadavid, Juan Pablo, Samir Lamouri, Bernard Grabot, Robert Pellerin, and Arnaud Fortin. 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* 31 (6): 1531–1558. doi:10.1007/s10845-019-01531-7.
- Valluri, Annapurna, Michael J. North, and Charles M. MacAl. 2009. "Reinforcement Learning in Supply Chains." *International Journal of Neural Systems* 19 (5): 331–344. doi:10.1142/S0129065709002063.
- Van Hasselt, Hado, Arthur Guez, and David Silver. 2016. "Deep Reinforcement Learning with Double Q-Learning." In *Proceedings of the AAAI Conference on Artificial Intelligence*. 30.
- Vanvuchelen, Nathalie, Joren Gijsbrechts, and Robert Boute. 2020. "Use of Proximal Policy Optimization for the Joint Replenishment Problem." *Computers in Industry* 119: 103239. doi:10.1016/j.compind.2020.103239.
- Vasilev, Ivan, Daniel Slater, Gianmario Spacagna, Peter Roelants, and Valentino Zocca. 2019. *Python Deep Learning: Exploring Deep Learning Techniques and Neural Network Architectures with Pytorch, Keras, and TensorFlow*. Birmingham: Packt Publishing Ltd.
- Vollmann, T. E., W. L. Berry, D. C. Whybark, and F. R. Jacobs. 2005. *Manufacturing Planning and Control for Supply Chain Management*. New York: McGraw Hill.
- Wan, Xing, Xingquan Zuo, Xiaodong Li, and Xinchao Zhao. 2020. "A Hybrid Multiobjective GRASP for a Multi-Row Facility Layout Problem with Extra Clearances." *International Journal of Production Research* 60 (3): 1–20. doi:10.1080/00207543.2020.1847342.
- Wang, Ziyu, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. 2016. "Dueling Network Architectures for Deep Reinforcement Learning." In *International Conference on Machine Learning*, 1995–2003.
- Watkins, Christopher J C H, and Peter Dayan. 1992. "Q-Learning." *Machine Learning* 8 (3–4): 279–292.
- Weiß, Gerhard. 1995. "Distributed Reinforcement Learning." In *The Biology and Technology of Intelligent Autonomous Agents*, edited by L. Steels, 415–428. Berlin, Heidelberg: Springer.
- Weiss, Gerhard. 1999. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge: MIT Press.
- Wijmans, Erik, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. 2020. "DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames."
- Williams, Ronald J. 1992. "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning." *Machine Learning* 8 (3–4): 229–256.
- Winder, Phil. 2020. *Reinforcement Learning: Industrial Applications of Intelligent Agents*.
- Yang, Shengluo, and Zhigang Xu. 2021. "Intelligent scheduling and reconfiguration via deep reinforcement learning in smart manufacturing." *International Journal of Production Research* 89: 1–18. <http://dx.doi.org/10.1080/00207543.2021.1943037>.
- Yu, Chao, Jiming Liu, and Shamim Nemati. 2020. "Reinforcement Learning in Healthcare: A Survey." *ArXiv Preprint ArXiv:1908.08796*.
- Zhang, Cong, Wen Song, Zhiguang Cao, Jie Zhang, Puay Siew Tan, and Chi Xu. 2020. "Learning to Dispatch for Job Shop Scheduling via Deep Reinforcement Learning." <http://arxiv.org/abs/2010.12367>.
- Zheng, Shuai, Chetan Gupta, and Susumu Serita. 2020. "Manufacturing Dispatching Using Reinforcement and Transfer Learning." *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, October. <http://arxiv.org/abs/1910.02035>.
- Zheng, Wei, Yong Lei, and Qing Chang. 2017a. "Comparison Study of Two Reinforcement Learning Based Real-Time Control Policies for Two-Machine-One-Buffer Production System." *IEEE International Conference on Automation Science and Engineering*, 1163–1168. doi:10.1109/COASE.2017.8256260.
- Zheng, Wei, Yong Lei, and Qing Chang. 2017b. "Reinforcement Learning Based Real-Time Control Policy for Two-Machine-One-Buffer Production System." In *Volume 3: Manufacturing Equipment and Systems*. American Society of Mechanical Engineers. doi:10.1115/MSEC2017-2771.