International Journal of Production Management and Engineering

https://doi.org/10.4995/ijpme.2025.24341

Received: 2025-07-18 Accepted: 2026-01-30
Available online: 2026-01-31 Published: 2026-01-31

A conceptual model for provider production scheduling in a manufacturing-as-a-service system using deep reinforcement learning

Mateo Del Gallo [a1], Raul Poler [b], Filippo Emanuele Ciarapica [a2]

[a] Department of Industrial Engineering and Mathematical Science, Università Politecnica delle Marche, Ancona, Italy.

[b] Research Centre on Production Management and Engineering, Universitat Politècnica de València, Plaza de Ferrándiz y Carbonell s/n, 03801 Alcoy (Alicante), España.

[a1] m.delgallo@pm.univpm.it, [b] rpoler@cigip.upv.es, [a2] f.e.ciarapica@staff.univpm.it

Abstract:

Manufacturing as a Service (MaaS) is an emerging paradigm in which a provider exposes manufacturing capabilities such as CNC machines, additive manufacturing systems, and other production assets as on-demand services to multiple Consumers. In this setting, Provider side scheduling becomes a critical aspect, as orders arrive dynamically and must be allocated to heterogeneous resources while meeting contractual constraints. This paper proposes an implementable conceptual framework for Provider production scheduling in MaaS, formulating the problem as a Markov Decision Process and enabling Deep Reinforcement Learning based decision making. The proposed Provider Planner models resource allocation as an unrelated parallel machine scheduling problem, where decisions are taken at discrete event-driven decision epochs (e.g., order arrivals and resource releases). The framework explicitly specifies the observation space, action space, and discrete-event transition logic, incorporating practical features such as resource availability and efficiency, operating cost rates, setup states across product families, batch-size constraints, order time windows (earliest/latest start), due dates, and delay penalties. A multi-objective reward formulation is defined to jointly minimise tardiness and associated penalties, overall makespan, and total production cost computed from resource uptimes. The resulting model provides a structured basis for developing and evaluating adaptive scheduling policies for MaaS Providers under demand variability and complex operational constraints.

Key words:

Manufacturing-as-a-service; dynamic scheduling; deep reinforcement learning; unrelated parallel machine scheduling.

1. Introduction

The concept of Manufacturing as a Service (MaaS) was born as a response to the growing need for flexibility and scalability in production processes (Ford et al., 2012). MaaS enables companies to access production resources on-demand through digital platforms, optimising the use of resources and reducing the need for investment in physical infrastructure (Goldhar & Jelinek, 1990; Dutra et al., 2013). Companies needing work requiring advanced technology or a high level of know-how (Consumer) can rent hours of use of production resources, owned by a Provider, to meet demand in real time. In this way, it is possible to have a more agile and adaptable production model, without the need to directly own or manage assets that would require large investments in equipment and training (Fisher et al., 2018). The possibility of having access to a MaaS platform is of considerable interest especially for Small and Medium Enterprises (SME) that can integrate Industry 4.0 paradigms and have access to high-level machinery and resources for a short period of time. A MaaS architecture is based on a cloud platform where Consumers with production needs can choose from a range of Providers who make

To cite this article: Del Gallo, M. (2026). A conceptual model for provider production scheduling in a manufacturing-as-a-service system using deep reinforcement learning. *International Journal of Production Management and Engineering, 14*(1), e24341. https://doi.org/10.4995/ijpme.2026.24341

their resources and expertise available (Karamanli et al., 2025a). Although the possibility of renting the use of expensive and innovative resources (such as CNC machines, 3D printers, etc.) could translate into a considerable advantage for the Consumer, on the other hand it must be considered that the Provider will have to manage and schedule the flow of orders coming from different consumers in a fast and flexible manner. The ability to adapt quickly to changes in demand and resource availability is crucial to ensure a continuous and efficient production flow (Hu et al., 2024; Al-Mahmud et al., 2025). In this context, the dynamic scheduling approach becomes essential.

In this scenario, the paper proposes a conceptual model for dynamic Provider side scheduling in MaaS environments, formulating order allocation as an unrelated parallel machine scheduling problem over a heterogeneous resource pool and enabling event-driven rescheduling through advanced Deep Reinforcement Learning (DRL) techniques. The adoption of DRL allows the system to learn and optimise scheduling decisions in near-real time, adapting to changes in data and production requirements, reducing delays and improving overall process efficiency (M. Zhang et al., 2023).

The proposed framework is based on three main actors: the Consumer, which requests production resources; the MaaS Platform, which manages communication and resource optimisation; and the Provider, which provides the necessary physical resources and manages production scheduling.

The rest of the paper is structured as follows: Section 2 will present an overview of existing dynamic scheduling models and their impact in MaaS contexts. Section 3 will describe the proposed conceptual model in detail, analysing the role of the main actors and how dynamic scheduling works. Finally, in Section 4, we will discuss possible practical implementations and future developments for the improvement of scheduling techniques in MaaS systems. Section 5 summarises and concludes the content of the paper.

2. Literature review

In the following sections, contributions in the literature on MaaS and how models presented by other authors integrate the concepts of dynamic scheduling will be analysed. To conduct this analysis, the contributions in the Scopus database were considered.

2.1. MaaS state of the art

The concept of MaaS in the literature is not a widely addressed topic as can be seen from the number of publications found in Scopus in Table 1. The table shows the queries used for the search (TITLE-ABS-KEY) and the corresponding number of publications. Each query is associated with an ID required for the analysis of publications in common between the various queries. Figure 1 shows the number of publications in common between the various clusters, all of which are contained in the first one.

The bibliometric analysis conducted by Karamanli et al., (2025b) shows that the concept of MaaS is connected to the concepts of Cloud Manufacturing (CM), IoT or Industrial IoT (IIoT), Artificial Intelligence (AI) and blockchain; fundamental concepts of the Industry 4.0 paradigm. The study reports how MaaS fits into the digital transformation of industries, aiming for more agile, scalable and customised manufacturing. However, its adoption

Table 1. Research keywords, number of publications for each query and clusterID.

Query	N° of publications	Cluster ID
"Manufacturing as a Service"	112	1
"Manufacturing as a service" AND "optimization"	19	2
"Manufacturing as a service" AND "scheduling"	4	3
"Manufacturing as a service" AND "scheduling" AND "optimization"	2	4
"Manufacturing as a Service" AND "cloud manufacturing"	39	5
"Manufacturing as a Service" AND "cloud manufacturing" AND "scheduling"	1	6
"Manufacturing as a Service" AND "smart factory"	6	7
"Manufacturing as a Service" AND "smart factory" AND "scheduling"	1	8

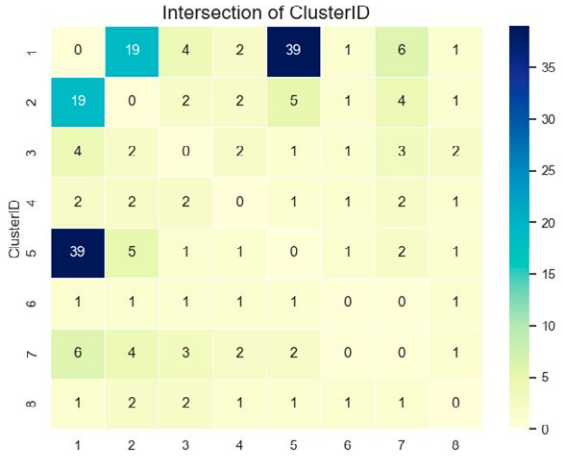

Figure 1. Number of publications in common between the different cluster IDs.

presents significant challenges, including data security, interoperability and regulatory barriers (Xu et al., 2024). As reported by Duran et al. (2024), enabling technologies are essential for the implementation of shared manufacturing. The integration of the above technologies enables a more efficient management of resources and facilitate communication between different actors in the production network. The integration of these technologies allows Providers and Consumers to collaborate in real time, improving production management and service customisation.

Vatankha Barenji et al. (Vatankhah Barenji et al., 2018) explores the use of blockchain in a Cloud Manufacturing context, with a focus on resource and security management at the shop floor and machine level. The study proposes the use of blockchain as a peer-to-peer network for managing the secure sharing of data between various actors, such as service providers and end users. Different is the approach and the technologies proposed by Nicoletti et al. (2024) that explores how extended reality (XR) can be integrated into the MaaS paradigm to optimise the production activities. The work focuses mainly on the use of XR technologies (which include Augmented Reality, Mixed Reality and Virtual Reality) to support the intelligent operator in production, maintenance and training activities. The paper emphasises that XR and MaaS are complementary technologies that, if integrated correctly, can greatly improve productivity and efficiency in manufacturing operations, especially in the context of Industry 4.0. A task service matching solution in a MaaS context is proposed by Chen et al. (2024). The authors propose a bilevel optimisation approach by exploiting genetic algorithms for the coordination of resource scheduling between platform and users. Different is the approach proposed by Hu et al. (2019) who employ chaos optimization algorithms for manufacturer scheduling in a Cloud Manufacturing environment. In the same context, a cloud service that integrates optimal service selection and dynamic resource management, making more flexible and efficient production possible were proposed in (Alinani et al., 2020; Borangiu et al., n.d.). The importance of intelligent and dynamic scheduling which employee Machine Learning (ML) model in a MaaS context (Namjoshi & Rawat, 2022) is an important issue that still finds a gap in the literature of concrete models and case studies.

In this context, the work of Mahmoodi & Fathi (2024) fits in as an attempt to analyse optimisation policies in the MaaS context, using a System Dynamics model. Although the paper does not propose a practical scheduling algorithm, its proposal focuses on optimising platform policies to increase MaaS adoption and improve resource and user management. The conceptual model proposed in this paper aims to bridge the gap on the use of ML models for production order scheduling in a MaaS architecture. In the following subsection, contributions on the use of ML algorithms in the dynamic scheduling framework will be analysed.

2.2. Dynamic scheduling algorithms review

In contrast to MaaS publications, the topic of dynamic scheduling is widespread in the literature. There are various approaches to solving different production scheduling problems (which is classified as NP-hard problem (Garey et al., 1976) that involve exact approaches using mathematical models (Del Gallo et al., 2024), heuristics algorithms which find the local optimum solution (Prata, 2025), metaheuristicas approaches (Bari et al., 2024) or a combination of these methods (Al-Mahmud et al., 2025). In recent years, studies employing the use of AI and ML algorithms to solve various scheduling problems have increased (Serrano-Ruiz et al., 2021). From a previous review conducted in 2023 by Del Gallo et al. (2023), the literature was reviewed in a systematic manner to understand which AI/ML approaches are the most widely used in real production contexts. From the study emerges how Particle Swarm Optimization and Reinforcement Learning (RL) were the two widely

used technologies to solve production scheduling problems in a real manufacturing context. In the last years the approaches which employ RL or Deep Reinforcement Learning (DRL) have been increasingly adopted (X. Zhang & Zhu, 2025). RL or DRL techniques have shown great applicability and flexibility in solving various problems such as: Parallel machine scheduling problem (related and unrelated) (Damodaran et al., 2012; Julaiti et al., 2022), Job-shop Scheduling Problem (JSP) (Wu et al., 2024; W. Zhang & Diettench, 1995), Flow-shop Scheduling Problem (FSP) (Pan et al., 2023). The combination of the parallel scheduling problem with the JSP or FSP, which in the literature is called with the prefix Flexible or Hybrid (Cai et al., 2023; Liu et al., 2023; Said et al., 2021; Song et al., 2023), is considered one of the most difficult scheduling problems.

In a MaaS context, a Consumer requests the hourly use of a given resource (CNC machine, 3D printer, skilled resources, etc.) from a Provider. This, in the simplest case may result in a related or unrelated parallel machine scheduling problem if the Provider must conduct a single process for the Consumer. In this scenario, the scheduling problem translates into an allocation of the Consumer's request to the various resources held by the Provider. In more complex scenarios, the Consumer may need to rent hours of several machines or processes for the realisation of a component, in which case it may fall into the Flexible JSP or Flexible FSP case depending on the production path required. When the scheduling problem has a very large or complex space of states and actions (e.g. in large production instances with many machines, tasks, and stochastic job arrivals), traditional RL algorithms risk not effectively handling the high dimensionality of the inputs (Gil & Lee, 2022). For this reason, DRL algorithms are becoming increasingly popular for approximating the value function or decision policy. Building on these contributions, this paper proposes a DRL-based conceptual model designed to meet the unique challenges of MaaS Provider scheduling plan.

3. Conceptual framework for provider planner

In this section, a conceptual framework is proposed for the implementation of a dynamic scheduling system for orders received from the Consumer to the Provider within a MaaS architecture. This study was developed as part of Horizon Europe's MaaSAI project, which aims to develop a comprehensive MaaS system that exploits AI/ML techniques to automate and simplify interactions between manufacturing system owners (Providers) and manufacturing companies (Consumers).

In particular, this work focuses on a solution of the MaaSAI project called Provider Planner, which is designed to plan the Provider's manufacturing activities in near real time. The goal of the Provider Planner is to optimise resource allocation and production scheduling using advanced ML techniques, including DRL, enabling more efficient, agile and transparent collaboration within the MaaS ecosystem.

In this context, the scheduling problem is explicitly modelled as the assignment of production orders to a heterogeneous set of manufacturing resources R, such as CNC machines, 3D printers and other production technologies, which can be naturally framed as an Unrelated Parallel Machine Scheduling Problem. This modelling choice reflects a typical situation in MaaS environments, where Providers operate diverse resources characterised by different capabilities, processing times, costs and expertise levels.

The system is based on three main actors: the Consumer, who requests production resources; the MaaS platform, which manages data collection, communication and security; and the Provider, which supplies the necessary physical resources (denoted as $R=\{R_1, R_2,\ldots,R_n\}$, where n is the number of available resources) and schedules production activities. Figure 2 illustrates the conceptual framework, where the black arrows represent the flow of information from the Consumer to the Provider. After the calculation of the new schedule, the information will flow back from the Provider to the Consumer (white arrows).

The Consumer sends requests to the MaaS platform, specifying production requirements such as quantities, deadlines and tecnologichal constraints. The MaaS platform, exploiting technologies such as IIoT, Big Data and Blockchain, receives the request, processes the status of the resources and sends the order to the Provider. This actor, through the Provider Planner, intelligently assign tasks to resources R and calculate an optimised schedule for the Consumer orders. In addition, thanks to intelligent sensors, the platform continuously monitors the status of production, enabling dynamic adjustments and rescheduling in real time.

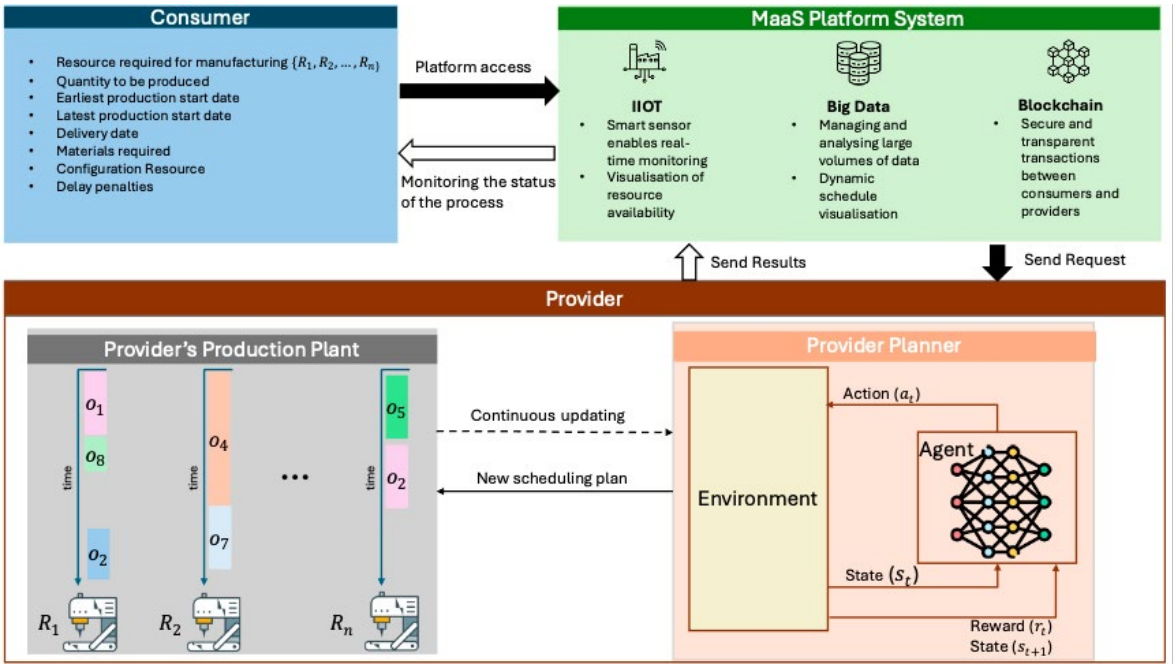

Figure 2. Conceptual framework for Provider planner communication in a MaaS Platform System.

3.1. Consumer

The Consumer represents the entity that makes the machine time rental request within a MaaS system. In a dynamic context such as MaaS, the Consumer plays a crucial role in determining the production needs to be fulfilled by the platform. The information that the Consumer provides is used by the MaaS platform to coordinate resources and optimise the production process. The main information provided by the Consumer is listed in Table 2.

In the proposed conceptual model, the Consumer does not directly select a specific machine, but rather specifies a set of admissible resources or compatible technologies from the set $R=\{R_1, R_2,\ldots,R_n\}$, such as CNC machining, 3D printers or other processing

Table 2. Consumer's information to send into the MaaS Platform system.

Consumer's information	Description
Resource required for manufacturing and configuration	The Consumer specifies the required resources to complete the order from the set $R=\{R_1, R_2,\ldots,R_n\}$, such as specific machinery equipment, or skilled labour. This information is crucial to the scheduling process, as it determines the allocation of physical resources within production. The system must be able to reconcile Consumer demands with the actual availability of resources.
Quantity to be produced	The Consumer specifies the quantity of product required.
Earliest and latest production start time	Production start dates (earliest or latest) indicate the range of when the consumer wants production to start. This data helps determine the timing of the start of the process, and giving the Provider a range of start dates makes scheduling more flexible.
Due dates	Sets the deadline by which the order must be completed and delivered. These time constraints are crucial to optimise the workflow and to ensure that orders are completed on time, meeting customer expectations.
Delay penalties	The Consumer defines the penalties it is prepared to accept in the event of delays in delivery. Penalties may be in the form of financial penalties, loss of trust or other types of commercial agreements.
Material Requirements	The Consumer specifies the materials required for production, which may include specific components or resources to be used during the production process.

capabilities. This allows the system to handle scenarios in which the Provider owns multiple resources capable of fulfilling the same request but characterised by different efficiencies, processing times and operational costs, consistently with an unrelated parallel machine scheduling formulation. After the Consumer specifies the quantity of product required is possible to determine the production capacity required to meet the demand in time. The MaaS platform forwards this information to the Provider Planner which uses this to calculate how much time and resources are needed to complete the order, considering the Provider's production capacity and resource availability. One critical aspect concerns penalties for delays. This information is particularly important for dynamic planning and delay risk management. The Provider Planner must therefore take into account not only meeting deadlines, but also optimising the costs associated with any delays. By evaluating the cost of penalties, the scheduling algorithm will be able to determine how to modify the schedule to avoid delays or minimise the impact of delays. Overall, the information provided by the Consumer contributes directly to defining part of the environment observed by the DRL-agent, influencing both the system state representation and the reward formulation adopted by the Provider Planner.

3.2. MaaS platform system

The MaaS platform system is a cloud-based platform that acts as an intermediary between Consumers and Providers within a MaaS architecture. The platform allows multiple Consumers to log in and select a Provider to request manufacturing services for a specific period. Once the Provider has been selected, the Consumer can submit a request specifying the required production capabilities (held by the Provider) and technological constraints, needed to make specific components or parts. More information useful for processing the order will be delivered to the Provider such as the quantity required, time constraints, materials needed and specific resource requirements. After receiving this information, the Provider processes the new order by returning an updated schedule in near-real time response using the Provider Planner tool. The new schedule will be sent to the Consumer and to the production plant. The latter, equipped with advanced technologies such as Industrial Internet of Things (IIoT) sensors, allows the MaaS platform to continuously update and monitor order status. Moreover, Big Data analysis allows the platform to identify production trends, enabling more informed decisions on resource allocation. Furthermore, blockchain technology ensures secure and transparent transactions between Consumers and Providers, tracking production milestones and increasing trust through compliance with agreements.

In summary, the MaaS platform acts as the backbone of the system, providing crucial functionalities such as order management, provider selection,data exchange, and system monitoring, while integrating IIoT, Big Data and Blockchain technologies. This allows Consumers and Providers to collaborate more effectively, while maintaining flexibility and responsiveness in the face of changing production demands.

3.3. Provider

The Provider is one of the main actors in a MaaS system and plays a key role in the dynamic scheduling of production activities. Within this context, the Provider is responsible not only for providing the physical resources for manufacturing tasks, but also for optimising and managing production based on the set of orders received from several Consumers $O=\{o_1, o_2,\ldots,o_k\}$. To this end, the Provider is characterised in Figure 2 by two main elements: the Provider Planner and the Provider's Production Plant. The Provider Planner is a decision-making unit that manages the resource allocation and the entire scheduling process, while the production plant represents the physical execution layer where manufacturing operations are carried out.

In this work, the Provider Planner is modelled as a decision-making system based on a Markov Decision Process (MDP), in which the scheduling problem is formulated as a sequential decision-making task. An MDP formulation, as introduced by Sutton and Barto (2018), represents a Markovian system through a tuple consisting five components (S, A, P, R, γ) where S is the set of system states, A is the set of actions, P is the state transition probability function, R is the reward function, and $\gamma \in [0,1)$ is the discount factor. At each decision step t, the agent observes the current state $s_t \in S$ and selects an action $a_t \in A$ according to a policy $\pi(a|s)$. After executing the action, the environment evolves to a new state $s_t+\Delta t$ with probability $P(s_{t+\Delta t}|s_t, a_t)$, and provides a reward $r_t=R(s_t, a_t, s_{t+\Delta t})$. The objective is to learn a policy that maximises the expected long-term cumulative reward, discounted by the factor γ, overt the decision horizon.

In the context of this work, at each decision step, the current state of the production system, including the set of active orders and the status of available resources, is combined with newly arrived Consumer requests in order to determine an appropriate scheduling action. More specifically, the scheduling problem is here interpreted as the assignment of production orders to a heterogeneous set of resources $R=\{R_1, R_2,\ldots,R_n\}$, such as CNC machines, 3D printers and other manufacturing technologies, characterised by different processing capabilities, costs and efficiencies. This formulation naturally maps the problem into an unrelated parallel machine scheduling setting, which is typical in MaaS environments where Providers operate diverse and heterogeneous production assets. Within this MDP framework, the Provider Planner is equipped with a DRL agent that learns a policy for dynamically assigning orders to resources so as to optimise production objectives while respecting operational constraints. At each decision step, the agent observes the current system state and selects an action corresponding to a specific order–resource allocation decision. The system then evolves according to a state transition mechanism ($s_t \rightarrow s_{t+\Delta t}$) that reflects the dynamics of the production process, including machine occupation, job completion, and the arrival of new orders. The continuous interaction between the DRL agent and the production environment enables the Provider Planner to progressively improve its scheduling policy, adapting to variations in demand, resource availability and system disturbances. The following subsections describe in detail the modelling of the environment, the action space and transition dynamics, as well as the reward formulation adopted to guide the learning process.

3.3.1. Environment features

In this section, the environment features that constitute the agent's observation space within the Provider Planner are defined. Table 3 summarises the state representation s_t at decision time t, i.e., the minimal yet informative set of variables that systematically describes both the condition of the Provider's resources and the operational and contractual characteristics of the active orders. In the following, the index t denotes variables included in the state observed at decision epoch t. Note that some components (e.g., D_t, Q_t, $\prod_t$) may remain constant over the entire lifecycle of an order and are reported with subscript t only for notational consistency. The aim of this modelling choice is to provide the agent with a coherent snapshot of the production system at every instant in which a resource assignment decision must be taken, explicitly incorporating the elements that most strongly affect scheduling quality in a MaaS context: machine heterogeneity, costs, set-up and batching constraints, time windows, and tardiness penalties.

In particular, T_t denotes the current production time and provides the reference for evaluating system progress and proximity to deadlines. The resource state vector R_t encodes the availability of each schedulable resource and allows one to distinguish between resources that can be allocated immediately ($Resource\_states_n=0$) and resources that are currently busy ($Resource\_states_n=1$); this information is essential to avoid infeasible allocations and to reason about short-term residual capacity. Resource heterogeneity is captured through E_t and C_t: the efficiency vector E_t describes the relative performance of resources (and, when applicable, their capability to execute specific operations), whereas C_t represents the operating cost per unit time, enabling the agent to balance temporal objectives against economic ones. In the presence of batch-oriented processes, or whenever resources impose minimum/maximum lot constraints, the parameters $B_t^{\min}$ and $B_t^{\max}$ model, respectively, the minimum processable batch size and the maximum capacity of each resource; such constraints directly influence assignment feasibility and the convenience of grouping orders or postponing the start of a job.

Aspects related to machine reconfiguration are represented by the set-up state SU_t, defined with respect to the product families F and the resources R. This variable enables, at a conceptual level, the modelling of the dependence of scheduling decisions on the current production context, including family changeover times and implicitly discouraging sequences that require frequent reconfigurations. On the order side, Table 3 includes the admissible start-time windows through ES_t and LS_t, which represent, respectively, the earliest and latest allowable start dates; such constraints are particularly relevant in MaaS services, where Consumers may impose release intervals or preferred start dates for logistical or contractual reasons. The due date D_t completes the temporal representation of the order, allowing the evaluation of urgency and delay risk.

The term P_t represents the expected processing times, specified for each order–resource pair, and constitutes the basis for estimating assignment durations and the impact of decisions on system

e-ISSN: 2340-4876

Table 3. Observation state features, sizes and descriptions.

State Component	Symbol	Size / Type	Description
Current_Time	(T_t)	Integer	Current production time (simulation clock)
Resource_states	(R_t)	$\|R\|$; Binary	Availability status of each resource (0 = idle; 1 = busy).
Resource_efficiency	(E_t)	$\|R\|$; Float	Efficiency coefficient of each resource [0,1]
Resource_cost_per unit	(C_t)	$\|R\|$; Float	Operating cost rate of each resource.
Resource_min_batch size	($B_t^{\min}[i]$)	$\|R\|$; Integer	Minimum batch size required by each resource
Resource_max_batch size	($B_t^{\max}[i]$)	$\|R\|$; Integer	Maximum batch capacity of each resource
Setup_states	(SU_t)	$\|RxF\|$; Float	Setup state of each resource for each product family
Order_earliest_start_date	(ES_t)	$\|O\|$; Integer	Earliest production start date
Order_latest_start_ date	(LS_t)	$\|O\|$; Integer	Latest production start date
Order_due_date	(D_t)	$\|O\|$; Integer	Deadline associated with each order.
Order_quantities	(Q_t)	$\|O\|$; Integer	Quantities required by the Consumer
Order_processing_time	(P_t)	$\|OxR\|$; Float	Baseline estimate processing time of each order on each resource.
Order_penalties	($\prod_t$)	$\|O\|$; Float	Tardiness penalty for each order.
Order_State	(F_t)	$\|O\|$; Binary	Order completion flag (0 = pending, 1 = completed).

evolution; this representation explicitly captures the dependence of processing time on the selected resource, which is typical of heterogeneous machine environments. To make explicit how P_t is exploited within the decision-making process, it is introduce the computation of the effective processing time resulting from a specific allocation. Let i denote the index of the selected resource, with $i \in \{1, \ldots, |R|\}$, and let j denote the index of the order, with $j \in \{1, \ldots, |O|\}$. The matrix P_t contains, for each order–resource pair, a nominal (or estimated) processing time $p_{i,j}$, representing the time required for resource R_i to complete order o_j under standard conditions. Since resources may exhibit different performance levels, the effect of heterogeneity is modelled through the efficiency coefficient $E_t[i]$ associated with resource R_i. In particular, the effective processing time resulting from allocating order o_j to resource R_i is defined as in Equation (1):

$$p_{j,i}^{eff}(t) = \frac{p_{j,i}}{E_t[i]} \tag{1}$$

Here, $p_{i,j}$ denotes a baseline processing-time estimate for the order–resource pair, whereas $E_t[i]$ captures time-varying resource performance (e.g., degradation, condition, or operational efficiency). Therefore, $E_t[i]$ acts as a dynamic scaling factor that adjusts the baseline time to reflect the current shop-floor conditions.

This formulation clarifies how the agent's decision (selection of the resource to allocate R_i) directly affects the duration of the operation and, consequently, the temporal evolution of the system state within the environment transition function.

Finally, $\prod_t$ encodes the penalty associated with tardiness for each order and translates service agreements and Consumer priorities into a numerical form, enabling a planning policy that does not merely "meet deadlines" but also optimises the expected cost of delays. The binary variable F_t tracks the completion status of orders and supports the episode termination logic: when all orders are completed, the environment can close the decision horizon and return the resulting schedule.

This state definition is intentionally modular: while sufficiently specific to describe realistic MaaS scenarios, it allows the level of detail to be extended or reduced depending on the technology (e.g.,

CNC machining, additive manufacturing) and data availability, without altering the Provider Planner logic nor the underlying MDP formulation on which the agent's learning process is based.

3.3.2. Action space and transition function

In the following, the action space of the Provider Planner is defined, consistently with the MDP formulation introduced in Section 3.3 and with the state representation reported in Table 3. Since the system evolves according to a discrete-event logic, the agent's decisions are taken at specific decision epochs t, i.e., time instants at which the system requires an allocation decision (for instance, following the arrival of new orders or the release of one or more resources). Figure 3 shows the process of assigning resources proposed by the agent and the transition function between one event and the next. The figure also refers to the reward assignment process, which will be explained in the next section.

At each decision epoch, the environment performs a preprocessing step to construct the set of schedulable orders O_t^{sched} based on the current state s_t. In particular, an order o_j is included in O_t^{sched} if it is not completed ($F_t(t)=0$) and if the current time T_t falls within its admissible start window ($ES_t(j) \leq T_t \leq LS_t(j)$). The selected orders are then organised into a list O_t^{sched} and sequenced according to parametric priority criterion (e.g., FIFO, SPT, EDD, or hybrid rules), chosen depending on the use case. Once fixed, this criterion induces a deterministic ordering of the schedulable orders as a function of the current state. Given the list $O_t^{sched}=[o_{(1)},\ldots,o_{(m)}]$, the agent's action is modelled as a fixed-length vector of resource selections a_t, where the action's dimension K is chosen as the maximum number of simultaneous assignments that can be handled within a single decision epoch (e.g., $K \leq |R|$). a_t is defined as $a_t=[a_t(1),\ldots,a_t(K)$, where each component $a_t(k)$ denotes the index of the resource proposed for the order in position k of the list O_t^{sched}. For $k \leq m$, the selection $a_t(k)=i$ indicates that the agent proposes to allocate order $o_{(k)}$ to resource R_i; for $k>m$, the component is set to a dummy value (NO-OP), indicating that no assignment is issued for that position.

If the proposed assignment $a_t(k)$ violates operational constraints (e.g., resource capability/enablement, expertise requirements, or batching limits), the assignment is deemed infeasible and rejected; the corresponding action component is converted to NO-OP and the order remains pending to be reconsidered at the next decision epoch.

In this way, the resulting allocation is always consistent (i.e., a resource cannot be assigned simultaneously to multiple orders), while preserving a simple action definition compatible with the discrete-event evolution of the system. Any penalties associated with conflicting selections can be incorporated into the reward function, thereby guiding learning towards policies that generate collision-free assignments.

3.3.3. Reward function

One of the most important aspects in an RL context is reward design; Figure 3 shows several terms related to rewards, which will be explained below. The Provider Planner pursues three objectives: minimisation of makespan, minimisation of tardiness, and minimisation of total costs (delay penalties and operational production costs). In a DRL setting, these objectives are mapped to a scalar reward through a weighted combination of terms computed at event level (step reward r_t) and at the end of the episode (terminal reward r_{term}). Reward is evaluated at completion events to align the signal with realised performance; denser shaping terms (e.g., slack-based penalties) can be incorporated if required by the application. The overall episodic return is defined by Equation 2:

$$R = r_{term} + \sum_{t \in T} \gamma^{n(t)} r_t \tag{2}$$

Where T denotes the set of decision epochs, $n(t)$ is the step index (i.e., counting events in an episode), and $\gamma \in [0,1)$ is the discount factor.

Let C_j denote the completion time of order j and D_j its due date. The tardiness of order j is defined in Equation 3. In addition, a fixed economic penalty $\prod(j)$ is assumed to be incurred only when an order is completed late. Accordingly, the delay penalty associated with order j is defined by Equation 4.

$$Tard_j = \max(0, C_j - D_j) \tag{3}$$

$$Pen_j = \begin{cases} \Pi(j) if C_j > D_j \\ 0 if C_j \leq D_j \end{cases} \tag{4}$$

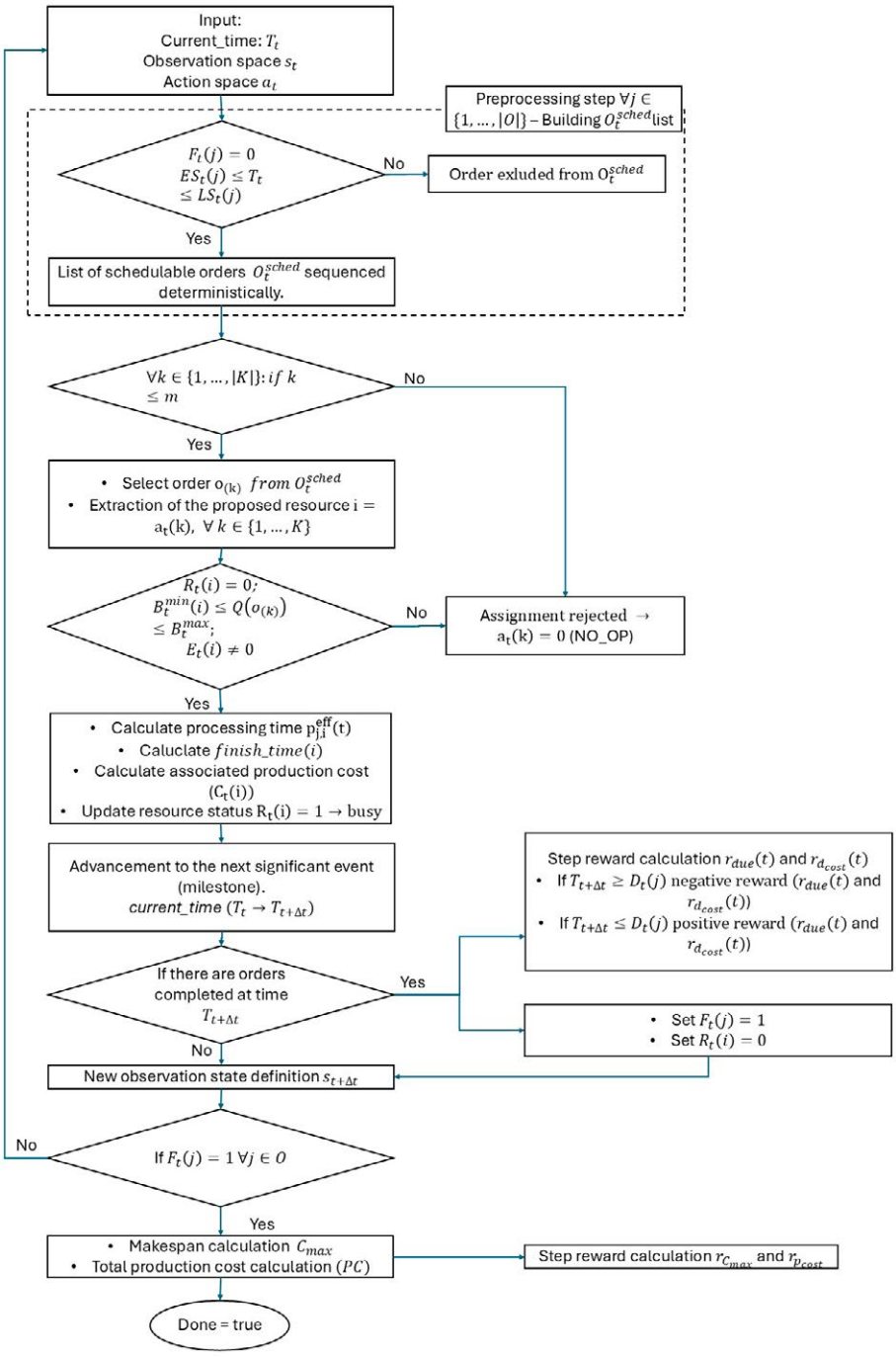

Figure 3. Step function flow-chart, preprocessing and action implementation.

Define O_t^{comp} as the set of orders completed at decision epoch t. The step reward r_t in Equation (2) is computed by aggregating the tardiness and the associated delay penalties over the completed orders, weighted by the non-negative coefficients ω_{tard} and ω_{pen}. As the agent seeks to maximise the return, these cost-like components are included with a negative sign:

$$r_t = -\omega_{tard} \sum_{j \in O_t^{comp}} Tard_j - \omega_{pen} \sum_{j \epsilon O_t^{comp}} Pen_j \quad (5)$$

At the end of the episode, the terminal reward r_{term} captures the remaining two objectives, namely makespan and total production cost. The makespan is defined as the maximum completion time of all the orders that must be processed (Equation 6). Meanwhile to quantify operational production costs, let U_i denote the total productive time (uptime) of resource i over the final schedule, and let $C(i)$ be its hourly operating cost rate. The overall production cost is expressed by Equation 7. Finally, the terminal reward is defined as a weighted sum of these two terminal costs (Equation 8).

$$C_{max} = \max_{j \in O} C_j \tag{6}$$

$$ProdCost = \sum_{i=1}^{|R|} C_t(i) U_i \tag{7}$$

$$r_{term} = -\omega_{C_{max}} C_{max} - \omega_{prod} \sum_{i=1}^{|R|} C_t(i) U_i \tag{8}$$

$\omega_{tard}, \omega_{pen}, \omega_{Cmax}, \omega_{prod} \geq 0$ are tuning parameters that control the relative importance of the corresponding objectives.

3.3.4. Provider production plant

The Provider's Production Plant is where the physical resources needed to produce the required goods are located. The resources and the plant communicate both with the Provider Planner environment to provide continuous updates on their status (e.g. availability, uptime) and allow the system to recalculate the schedule, and with the MaaS Platform System to ensure that the Consumer can monitor production progress in real time. The Provider, using its Provider Planner, recalculates the schedule and decides how to allocate the available resources to fulfil the Consumer order. Feedback on production progress is continuously sent from the platform via smart sensors, which monitor the status of the provider's resources in real time. Once the order is in production, smart sensors placed on the provider's assets provide real-time data on machine availability, task progress and production process efficiency (Pietrangeli et al., 2024). This data is crucial for the MaaS platform, which processes it and provides real-time updates to the Consumer. In addition, should there be any delays or unforeseen events, the platform can send a signal back to the Provider Planner, who can promptly react by modifying the schedule according to the new situation, thereby optimising the entire production process in real time. In conclusion, the Provider plays a crucial role in the MaaS system, not only providing the necessary resources for production, but also dynamically optimising production scheduling through the use of the Provider Planner, which relies on DRL to make optimal decisions. The continuous communication between the Provider's Production Plant, the Provider Planner, and the MaaS Platform System allows the system to adapt quickly to changes, maintaining high production efficiency and ensuring that orders are completed on time, meeting Consumer requirements.

4. Discussion

The proposed conceptual model formalises the provider-side scheduling problem in a MaaS context as a sequential decision-making process with discrete events, in which a DRL agent learns resource allocation policies on a heterogeneous resource set. The explicit definition of the observation space, the action space and the transition function allows us to move from a generic description of the MaaS architecture to an implementable setting, while maintaining sufficient flexibility to represent different technologies and operational constraints (CNC, additive manufacturing, batch-oriented resources, start windows and contractual penalties). In this framework, the use of discrete-event logic allows decisions to be aligned with actual production milestones (order arrival, completion, resource release), avoiding the introduction of artificial time steps and making the modelling of the system's dynamic behaviour more natural.

4.1. Practical implementation

From an implementation perspective, the adoption of DRL is motivated by the inherently dynamic and multi-objective nature of scheduling in MaaS. In these systems, decisions are not isolated but interdependent: the allocation of a resource to an order influences the future state (congestion, availability, setup, delay risk and costs), with effects that manifest themselves over a long time horizon. Simple rules such as FIFO or SPT are effective in stable contexts with limited constraints, but tend to be short-sighted: they optimise local choices without explicitly considering the evolution of the state and the long-term consequences. DRL, on the other

e-ISSN: 2340-4876

Table 4. Conceptual comparison between classic dispatching rules and the proposed DRL-based scheduling approach.

Aspect	Classic dispatch rules (FIFO, SPT, EDD)	DRL-based Approach
Decision horizon	Local, myopic	Global and dynamic
System state awareness	Ignored or minimal	Comprehensive (queues, machines, setup, etc.)
Adaptability	Static	Adaptive through learning
Reaction to congestion	Absent	Possible and proactive
Handling complex constraints	Difficult	Natural and flexible

hand, is designed to learn policies that maximise cumulative return, naturally integrating complex constraints and economic-temporal trade-offs into a single decision-making process.

Table 4 provides a concise conceptual comparison between classic dispatch rules (e.g., FIFO, SPT, EDD) and a DRL-based scheduling approach within the Provider Planner. Dispatch rules typically operate with a local, myopic horizon: decisions are taken based on a limited set of attributes (such as arrival order, processing time, or due date) and do not explicitly account for the full system state. As a result, their behaviour is largely static and they offer limited capability to anticipate congestion or to coordinate decisions across multiple resources and time windows. By contrast, a DRL-based approach conditions decisions on a structured representation of the production state (e.g., queue status, resource availability, setup conditions, and contractual parameters), enabling policies that can adapt to changing demand and shop-floor conditions. This broader state awareness also makes it more natural to incorporate heterogeneous resources and complex operational constraints through feasibility checks and reward design, supporting proactive behaviour under congestion and multi-objective trade-offs (tardiness, cost, and makespan) that are typical in MaaS scenarios.

4.2. Limitations and future developments

Although it provides a structured basis for designing a DRL Provider Planner, the work remains a conceptual model and, as such, has certain limitations. Firstly, the quality of the learned policies depends on how faithfully the environment simulates the dynamics of the plant (estimation of processing times, modelling of setup/changeover, availability and reliability of resources, etc.), as well as on the correct calibration of the weights that combine temporal and economic objectives in the reward. Secondly, defining the action as the selection of resources from an ordered list of schedulable orders introduces a trade-off between simplicity and 'completeness' of the search: the sorting criterion (FIFO/SPT/EDD or hybrids) can influence the quality of allocations and should therefore be evaluated on a case-by-case basis. Finally, the reward signal linked to tardiness and penalties, if calculated mainly upon order completion, may be sparse in some scenarios, requiring reward shaping or normalisation techniques to stabilise training.

As future activities, it is natural to plan an experimental validation campaign through simulation, including:

- training with different environment configurations (number of resources, levels of heterogeneity, order arrival profiles, batching and setup constraints),
- sensitivity analysis on reward weights and DRL parameters,
- ablation studies to quantify the impact of individual state variables (e.g., removing setup state, ES/LS windows, batch constraints, or costs) on final performance.

Such experiments would allow us to understand which components of the model contribute most to the robustness of the policy.

5. Conclusion

This work proposed a conceptual model for dynamic provider-side planning in a Manufacturing-as-a-Service ecosystem, formalising the scheduling problem as a sequential decision-making process based on MDP and addressable through Deep Reinforcement Learning. Unlike descriptive or purely architectural approaches, the main contribution consists in making the framework implementable, explicitly defining the observation space, the action space and the transition function to discrete events, in a manner consistent with realistic MaaS scenarios characterised by heterogeneous resources, operational constraints (setup, batching, time windows) and multiple objectives. Furthermore,

a reward formulation has been introduced that integrates makespan, tardiness, and costs (penalties and operating costs), providing a unified basis for training and evaluating resource allocation policies. Overall, the outlined model represents a structured reference for designing and developing a DRL Provider Planner in MaaS contexts, and constitutes a starting point for future experimental validation activities through simulation and industrial case studies, including ablation analyses and systematic comparisons with traditional dispatching rules.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT as a support tool for language editing (grammar, style, clarity). All ideas, analyses, and conclusions remain entirely their own and comply with the journal's transparency and ethics guidelines. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

Alinani, K., Liu, D., Zhou, D., & Wang, G. (2020). Service Composition and Optimal Selection in Cloud Manufacturing: State-of-the-Art and Research Challenges. *IEEE Access*, *8*, 223988–224005. https://doi.org/10.1109/ACCESS.2020.3045008

Al-Mahmud, S., Cano, J. A., Campo, E. A., & Weyers, S. (2025). Optimizing cut order planning: A comparative study of heuristics, metaheuristics, and MILP algorithms. *International Journal of Production Management and Engineering*, *13*(1), 1–26. https://doi.org/10.4995/ijpme.2025.22196

Bari, P., Karande, P., & Bag, V. (2024). Hybrid genetic algorithm to minimize scheduling cost with unequal and job dependent earliness tardiness cost. *International Journal of Production Management and Engineering*, *12*(1), 19–30. https://doi.org/10.4995/ijpme.2024.19277

Borangiu, T., Trentesaux, D., Leitão, P., Cardin, O., & Lamouri, S. (n.d.). *Studies in Computational Intelligence 952 Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future Proceedings of SOHOMA 2020*. http://www.springer.com/series/7092

Cai, J., Lei, D., Wang, J., & Wang, L. (2023). A novel shuffled frog-leaping algorithm with reinforcement learning for distributed assembly hybrid flow shop scheduling. *International Journal of Production Research*, *61*(4), 1233–1251. https://doi.org/10.1080/00207543.2022.2031331

Chen, W., Feng, P., Luo, X., & Nie, L. (2024). Task-service matching problem for platform-driven manufacturing-as-a-service: A one-leader and multi-follower Stackelberg game with multiple objectives. *Omega (United Kingdom)*, *129*. https://doi.org/10.1016/j.omega.2024.103157

Damodaran, P., Diyadawagamage, D. A., Ghrayeb, O., & Vélez-Gallego, M. C. (2012). A particle swarm optimization algorithm for minimizing makespan of nonidentical parallel batch processing machines. *International Journal of Advanced Manufacturing Technology*, *58*(9–12), 1131–1140. https://doi.org/10.1007/S00170-011-3442-Z

Del Gallo, M., Antomarioni, S., Mazzuto, G., Marcucci, G., & Ciarapica, F. E. (2024). A self-learning framework combining association rules and mathematical models to solve production scheduling programs. *Production and Manufacturing Research*, *12*(1). https://doi.org/10.1080/21693277.2024.2332285

Del Gallo, M., Mazzuto, G., Ciarapica, F. E., & Bevilacqua, M. (2023). Artificial Intelligence to Solve Production Scheduling Problems in Real Industrial Settings: Systematic Literature Review. In *Electronics (Switzerland)* (Vol. 12, Issue 23). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/electronics12234732

Duran, E., Ozturk, C., & O'Sullivan, B. (2024). Planning and scheduling shared manufacturing systems: key characteristics, current developments and future trends. In *International Journal of Production Research*. Taylor and Francis Ltd. https://doi.org/10.1080/00207543.2024.2442549

Dutra, D., Castelhano De Oliveira, V., & Silva, J. R. (2013). *Manufacturing as Service: the challenge of Intelligent Manufacturing*. http://www.sparxsystems.com.au/

Fisher, O., Watson, N., Porcu, L., Bacon, D., Rigley, M., & Gomes, R. L. (2018). Cloud manufacturing as a sustainable process manufacturing route. *Journal of Manufacturing Systems*, *47*, 53–68. https://doi.org/10.1016/j.jmsy.2018.03.005

Ford, S., Rauschecker, U., & Athanassopoulou, N. (2012). System-of-system approaches and challenges for multi-site manufacturing. *Proceedings - 2012 7th International Conference on System of Systems Engineering, SoSE 2012*, 543–548. https://doi.org/10.1109/SYSoSE.2012.6384164

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The Complexity of Flowshop and Jobshop Scheduling. In *Source: Mathematics of Operations Research* (Vol. 1, Issue 2).

Gil, C. B., & Lee, J. H. (2022). Deep Reinforcement Learning Approach for Material Scheduling Considering High-Dimensional Environment of Hybrid Flow-Shop Problem. *Applied Sciences (Switzerland)*, *12*(18). https://doi.org/10.3390/app12189332

Goldhar, J. D., & Jelinek, M. (1990). Manufacturing as a service business: CIM in the 21st century. *Computers in Industry*, *14*(1–3), 225–245. https://doi.org/10.1016/0166-3615(90)90126-A

Hu, Y., Pan, L., & Pan, X. (2024). Dynamic scheduling of workshop resource in cloud manufacturing environment. *Engineering Applications of Artificial Intelligence*, *138*. https://doi.org/10.1016/j.engappai.2024.109405

Hu, Y., Zhu, F., Zhang, L., Lui, Y., & Wang, Z. (2019). Scheduling of manufacturers based on chaos optimization algorithm in cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, *58*, 13–20. https://doi.org/10.1016/j.rcim.2019.01.010

Julaiti, J., Oh, S. C., Das, D., & Kumara, S. (2022). Stochastic parallel machine scheduling using reinforcement learning. *Journal of Advanced Manufacturing and Processing*, *4*(4). https://doi.org/10.1002/amp2.10119

Karamanli, A., Xanthopoulos, A., Gasteratos, A., & Koulouriotis, D. (2025a). A Bibliometric and Systematic Review of Manufacturing-as-a-Service: Literature Insights, Challenges, and Future Trends. In *Applied Sciences (Switzerland)* (Vol. 15, Issue 5). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/app15052440

Karamanli, A., Xanthopoulos, A., Gasteratos, A., & Koulouriotis, D. (2025b). A Bibliometric and Systematic Review of Manufacturing-as-a-Service: Literature Insights, Challenges, and Future Trends. In *Applied Sciences (Switzerland)* (Vol. 15, Issue 5). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/app15052440

Liu, Y., Fan, J., Zhao, L., Shen, W., & Zhang, C. (2023). Integration of deep reinforcement learning and multi-agent system for dynamic scheduling of re-entrant hybrid flow shop considering worker fatigue and skill levels. *Robotics and Computer-Integrated Manufacturing*, *84*, 102605. https://doi.org/10.1016/J.RCIM.2023.102605

Mahmoodi, E., & Fathi, M. (2024). Policy Making to Encourage Platform Thinking in Manufacturing: A System Dynamics-based Multi-Objective Optimization Approach. *IFAC-PapersOnLine*, *58*(27), 1164–1169. https://doi.org/10.1016/j.procir.2024.10.222

Namjoshi, J., & Rawat, M. (2022). Role of smart manufacturing in industry 4.0. *Materials Today: Proceedings*, *63*, 475–478. https://doi.org/10.1016/j.matpr.2022.03.620

Nicoletti, L., Solina, V., Amin, K., Lessi, C., McHard, P., Qiu, R., & Tedeschi, S. (2024). Exploiting Extended Reality under the Manufacturing as a Service paradigm. *Procedia Computer Science*, *232*, 2213–2219. https://doi.org/10.1016/j.procs.2024.02.040

Pan, Z., Wang, L., Wang, J., & Lu, J. (2023). Deep Reinforcement Learning Based Optimization Algorithm for Permutation Flow-Shop Scheduling. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *7*(4), 983–994. https://doi.org/10.1109/TETCI.2021.3098354

Pietrangeli, I., Mazzuto, G., Ciarapica, F. E., Bevilacqua, M., & Ortenzi, M. (2024). Smart Retrofit Solution: An Architecture for Digital Innovation. *Proceedings of the 30th ICE IEEE/ITMC Conference on Engineering, Technology, and Innovation: Digital Transformation on Engineering, Technology and Innovation, ICE 2024*. https://doi.org/10.1109/ICE/ITMC61926.2024.10794309

Prata, B. de A. (2025). An overview of industrial engineering and operations management over the first fifty years of Engineering Optimization. In *Engineering Optimization*. Taylor and Francis Ltd. https://doi.org/10.1080/0305215X.2024.2423181

Said, N. E. D. A., Samaha, Y., Azab, E., Shihata, L. A., & Mashaly, M. (2021). An Online Reinforcement Learning Approach for Solving the Dynamic Flexible Job-Shop Scheduling Problem for Multiple Products and Constraints. *Proceedings - 2021 International Conference on Computational Science and Computational Intelligence, CSCI 2021*, 134–139. https://doi.org/10.1109/CSCI54926.2021.00095

Serrano-Ruiz, J. C., Mula, J., & Poler, R. (2021). Smart manufacturing scheduling: A literature review. In *Journal of Manufacturing Systems* (Vol. 61, pp. 265–287). Elsevier B.V. https://doi.org/10.1016/j.jmsy.2021.09.011

Song, W., Chen, X., Li, Q., & Cao, Z. (2023). Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning. *IEEE Transactions on Industrial Informatics*, *19*(2), 1600–1610. https://doi.org/10.1109/TII.2022.3189725

Sutton, R. S, & Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge: MIT Press.

Vatankhah Barenji, A., Li, Z., & Wang, W. M. (2018). *Smart SysTech 2018 : European Conference on Smart Objects, Systems, and Technologies, June 12-13, 2018, Fraunhofer Institute for Photonic Microsystems (IPMS) in Dresden, Germany*. VDE Verlag GmbH.

Wu, X., Yan, X., Guan, D., & Wei, M. (2024). A deep reinforcement learning model for dynamic job-shop scheduling problem with uncertain processing time. *Engineering Applications of Artificial Intelligence*, *131*. https://doi.org/10.1016/j.engappai.2023.107790

e-ISSN: 2340-4876

Xu, C., Yu, H., Jin, X., Xia, C., Li, D., & Zeng, P. (2024). Industrial Internet for intelligent manufacturing: past, present, and future. In *Frontiers of Information Technology and Electronic Engineering*, *25*(9), 1173–1192. Zhejiang University. https://doi.org/10.1631/FITEE.2300806

Zhang, M., Wang, L., Qiu, F., & Liu, X. (2023). Dynamic scheduling for flexible job shop with insufficient transportation resources via graph neural network and deep reinforcement learning. *Computers and Industrial Engineering*, *186*. https://doi.org/10.1016/j.cie.2023.109718

Zhang, W., & Diettench, T. G. (1995). *A Reinforcement Learning Approach to Job-shop Scheduling*.

Zhang, X., & Zhu, G. Y. (2025). A literature review of reinforcement learning methods applied to job-shop scheduling problems. In *Computers and Operations Research*, *175*. https://doi.org/10.1016/j.cor.2024.106929

e-ISSN: 2340-4876