Scheduling Algorithms: Challenges Towards Smart Manufacturing

Original Scientific Paper

Abebaw Degu Workneh
Euro-Mediterranean University of Fez,
Euromed Research Center, Fes, Morocco
a.deguworkneh@ueuromed.org

Maha Gmira
Euro-Mediterranean University of Fez,
Euromed Research Center, Fes, Morocco
m.gmira@ueuromed.org

Abstract *– Collecting, processing, analyzing, and driving knowledge from large-scale real-time data is now realized with the emergence of Artificial Intelligence (AI) and Deep Learning (DL). The breakthrough of Industry 4.0 lays a foundation for intelligent manufacturing. However, implementation challenges of scheduling algorithms in the context of smart manufacturing are not yet comprehensively studied. The purpose of this study is to show the scheduling No.s that need to be considered in the smart manufacturing paradigm. To attain this objective, the literature review is conducted in five stages using publish or perish tools from different sources such as Scopus, Pubmed, Crossref, and Google Scholar. As a result, the first contribution of this study is a critical analysis of existing production scheduling algorithms' characteristics and limitations from the viewpoint of smart manufacturing. The other contribution is to suggest the best strategies for selecting scheduling algorithms in a real-world scenario.*

Keywords*: Scheduling Algorithm, Smart Manufacturing, Production Scheduling, Industry 4.0*

1. INTRODUCTION

In smart manufacturing industries, huge amounts of data are generated from heterogeneous sources such as sensors, Radio Frequency Identification (RFID), and networked machines [1], [2]. Moreover, inherently stochastic processes exist in industrial processes [3]. The advancement of Industry 4.0 and industrial intelligence leads to increased complexity, dynamics, and uncertainty on the shop floor [4]. This behavior paves the way for production scheduling challenges.

Scheduling algorithms should consider competing requirements to achieve a high-quality solution while remaining computationally efficient. Existing industrial scheduling solutions, such as heuristic algorithms, are efficient but difficult to implement in complex situations [5].

Heuristics, meta-heuristics, and mathematical programming are prominent tools to solve scheduling problems. However, as the complexity and scale of the problem increase, the solution would be unstable or might lead to unacceptable computing overhead [3], [6], [7]. It requires plenty of time to find a new solution [8] or needs manual configuration efforts during changes because of its model-based implementation and static nature [9]. Moreover, it also lacks the adaptability to a stochastic environment and needs a complex design process [10]. For example, as mentioned in [11], genetic algorithm has shown poor local search and slow convergence.

Despite the unavailability of scheduling algorithm challenges review in the smart manufacturing environment, there are an increasing number of review articles about smart manufacturing scheduling.

The purpose of this paper, unlike the previous review articles, is to emphasize the challenges of using different scheduling algorithms in the production environment, to introduce current scheduling strategies and their characteristics from the viewpoint of complex manufacturing and dynamically changing environment in the context of smart manufacturing, and to show the possible future research directions from different perspectives. The paper consists introduction to scheduling algorithms in production scheduling, a review methodology, a literature review, and a discussion on the properties and challenges of current scheduling solutions followed by a conclusion and future work.

2. REVIEW METHODOLOGY

The review is conducted based on the following five criteria: a) semantic areas of the article search; b) repositories used; c) document types; d) subject areas; e) language of the article (English only). The four semantic fields on which the article search was based are: a) field 1: "shop floor scheduling problem"; b) field 2: "smart manufacturing"; c) field 3: "Scheduling technology & tools"; d) field 4: "Scheduling algorithm". Terms for each semantics are selected based on their relevance after individual search and all terms yield a different result in each repository because of their difference in their query system. Four repositories are considered: a) Scopus; b) Crossref; c) PubMed; d) Google Scholar.

The search results were initially obtained using publish or perish tool on Scopus, and the other three repositories are used to complement the search results. Publishers for lots of searched articles are: a) Elsevier; b) IEEE; c) Springer; d) SAGE publications; e) Multidisciplinary Digital Publishing Institute (MDPI); f) Hindawi; g) Wiley Online Library; h) IOP Publishing. The review is conducted based on the following research questions:

a) Which algorithm does the industrial environment need?
b) What has been done so far in the production scheduling field that can contribute to smart manufacturing?
c) What still needs to be done for the practical implementation of scheduling solutions in the smart manufacturing industry?

3. LITERATURE REVIEW

The searched articles, as depicted in Fig. 1., are reviewed based on thematic and content analysis.

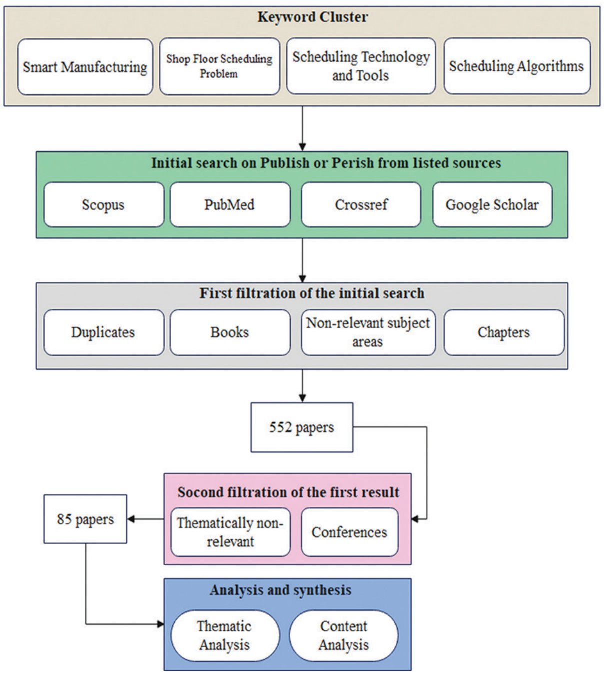

Fig. 1. Searches, collection, analysis, and synthesis methodology

Thematic Analysis: Based on the search term used in the reviewed articles, four thematic areas are identified using ATLAS.ti 9. These themes are smart manufacturing, shop floor scheduling problems, scheduling algorithms, and scheduling technology and tools.

Content Analysis: The content analysis is performed using the following phases: a) grouping of search articles based on the conceptual scheme of the research, b) the focus, c) experimental evaluation techniques used, and d) contributions and shortcomings of reviewed articles.

2.1. SMART MANUFACTURING

The term Smart manufacturing originated in the USA [12] and has no commonly accepted definition. Based on the study in [12], [13], smart manufacturing is a manufacturing operation that manages manufacturing processes with networked data. Likewise, the study in [14] defines the concept as a creation of manufacturing intelligence throughout all parts of the operation. It is a new manufacturing prototype in which manufacturing devices are entirely linked by wireless connections, supervised by sensors, and managed with cutting-edge computational intelligence [15].

The key technologies in smart manufacturing involve IoT, CPS, cloud computing, machine learning, big data, and mobile internet [14], [16], [17].

These technologies are realized through connected sensors, data interoperability, multi-scale dynamic modeling and simulation, smart digitization, and customizable and multi-level network security [18]. Materials, data, production processes and tools, resource sharing and connectivity, predictive engineering, and sustainability are considered the fundamental components [19]. The main idea behind this paradigm is to accumulate and evaluate massive amounts of manufacturing data to drive knowledge and rules [20].

Smart manufacturing involves the deployment of large amounts of sensors and IoTs, which requires the handling of big manufacturing data[15]. Big data is a key component in transforming today's manufacturing into a smart manufacturing paradigm. It helps companies to be competitive using data-driven strategies [21] and satisfy the needs of the manufacturing industry [14]. Deep learning, with its feature learning and large modeling capabilities, is an advanced analytics method for smart manufacturing. Based on the study in [21], smart manufacturing is divided into four modules i.e., "manufacturing module, data driver module, real-time monitoring module, and problem processing module". In the manufacturing module indicated in Fig. 2., the inputs are raw materials, and the outputs are finished goods.

Smart manufacturing has a different definition from different perspectives. For example, from the engineering point of view [22], smart manufacturing is characterized by the application of advanced intelligent systems that enables rapid production of manufacturing products, dynamic response to demand, and real-time optimization of the production and supply chain networks. In other words, the connected manufacturing resources take raw materials as input and produce a finished product for a customer. on the other hand, from (IoT & CPS) as well as interconnection perspective [23], smart manufacturing is defined as the collection of all stages of manufacturing data using sensors and different communication technologies to increase production rate and reduce errors and production waste. However, from the viewpoint of predictive analysis and decision making [24], smart manufacturing is the optimization of planning and control of manufacturing activities such as fault diagnosis, risk assessment, resource utilization, predictive supply, and manufacturing. Based on the aforementioned definition, this study focused on the scheduling approaches in interconnection and decision-making perspectives.

2.2. SHOP FLOOR SCHEDULING PROBLEMS

Scheduling is the process of assigning machines to a set of available jobs to optimize objective functions such as earliness or tardiness of jobs, job completion, and processing times [25]. By its nature, scheduling needs details about tasks to be executed and available resources with a set of constraints [17].

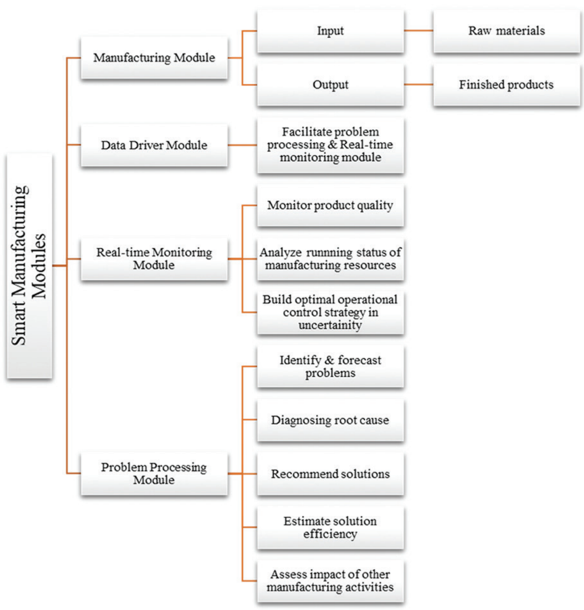

Fig. 2. Smart Manufacturing Modules

On the other hand, re-scheduling or pre-reaction scheduling is a way of scheduling again during the occurrence of new events [26]. It has two stages i.e., the pre-scheduling stage (generating scheduling for actual production) and the re-scheduling stage (re-configure the initial schedule to accommodate dynamic events). Robust scheduling forecasts potential future events based on the existing manufacturing state and generates a pre-schedule that includes different dynamic events. However, its success will be less effective if the dynamic conditions do not happen. Unlike the previous scheduling, the online scheduling is a real-time scheduling technique, that doesn't prepare a schedule in advance and is mainly used after dynamic events happen. This scheduling strategy is mostly used in actual production

In the reactive scheduling approach, decisions at the control and scheduling level don't consider dynamic events. As new events happen in production, a continuous recalculation process will take place, which results to be computationally expensive [27]. Conversely, the preventive scheduling approach considers dynamic events and aims at finding robust scheduling solutions with and without the presence of disruptions. Scheduling accuracy and performance are greatly influenced by the presence of uncertainty [28], [29].

Scheduling can be defined as classical scheduling and dynamic scheduling. In classical scheduling, all machines and constraints such as due date, processing time, etc. are available for scheduling [29]. The production state usually changes with time. However, the predefined states in this scheduling approach cannot address all production states. Moreover, parameters are usually found by statistics or calculations [30].

Investigation of dynamic scheduling is introduced by Jackson in the 1950s [26]. The critical factors in a dynamic manufacturing environment are external disturbances such as failure and production scheduling plan [31]. Despite its advantage over classical scheduling, uncertainties in dynamic scheduling are from direct assumptions, rather than actual production data [30]. As a result, it is not sufficient to provide interactive feedback with the real production. Most of the classical research focused on classical scheduling, in which all system state is known in advance and do not consider changing events [32], [33]. However, in a real production environment, unexpected events may occur at any time. For example, machine breakdown may happen at any time. So, it is also necessary to consider a maintenance plan along with real-time scheduling [34].

Maintenance activities are usually non-separable with production scheduling [32], [33], [35]. Based on [34], there are two main groups of maintenance: corrective maintenance (which involves repair during unexpected machine breakdown) and preventive maintenance (a condition where a planned schedule is executed before machine breakdown happens). On the other hand, the study in [14] classified the industrial maintenance strategies into four: reactive (perform maintenance during complete machine failure); corrective (identify and solve failures when it happens before total machine failure); preventive (performing regular maintenance to prevent partial or complete failure); and predictive (anticipate failures before it happens and guess the remaining useful life of the machine).

2.3. JOB SHOP SCHEDULING

Since the 1960s, Job Shop Scheduling Problems (JSSP) have been considered NP-hard problems [36], [37]. In JSSP, the number of jobs to be scheduled can be processed at a pre-determined set of machines [38]. Machines can also be re-visited by the job more than once. The Re-entrant Job Shop Scheduling Problem (RJSSP) is more complicated than JSSP [39].

To address scheduling problems, many algorithms have been used on JSSP machine environments such as Hybrid Genetic Algorithm (HGA) [40], multi-agent system [41], variable neighborhood search algorithm [42], hybrid particle swarm optimization [39], genetic algorithm [43], and ant colony optimization [44], [45].

2.4. FLEXIBLE JOB SHOP SCHEDULING

Flexible Job Shop Scheduling Problem (FJSSP) is an extension of JSSP, where each operation can be operated beyond a single machine, and each machine is capable of performing distinct tasks [46]. As a result, it is more complicated than traditional JSSP [47]. It is also known as Integrated Process Planning and Scheduling (IPPS) [48]. Because of its applicability in different industrial applications, FJSSP has received wide attention from researchers. Based on the study in [48], flexibility is classified into three: operation flexibility (a situation where a process can be processed by different alternative machines); process flexibility (a condition where it is possible to finish the product by combining different operations); and sequence flexibility (when operations processing sequence is variable for the product). A flexible job shop can respond quickly to market changes and customer demands [49]. In FJSSP, jobs may re-enter or visit the center more than once before completing the process [38]. This feature is widely known in printed circuit board and semiconductor industries.

Some of the previous research's objective functions and used algorithms are illustrated in Table 1.

Table 1. Previous works on FJSSP

References	Objective	Algorithms
[47]	Makespan and energy	Simulated Annealing (SA) and Artificial Immune Algorithm
[49]	Energy	Ant Colony Optimization (ACO)
[50]	Mean tardiness	Greedy randomized adaptive search
[51]	makespan and energy	simulated annealing
[52]	Makespan and energy	Backtracking search algorithm
[53]	Makespan and energy	Genetic algorithm (GA)
[54]	Makespan	Quantum algorithm
[38]	Makespan	Approximation algorithm
[55]	Makespan and due date	Iterated greedy constructive heuristic
[56]	Makespan, machine workload	Clustering search metaheuristics
[57]	Makespan	Jaya algorithm, Monte Carlo
[58]	Makespan	Genetic algorithm
[37]	Makespan & setup time	Non-Dominated Searching Genetic Algorithm (NSGA-II)
[59]	Makespan	Constraint Programming (CP)
[60]	Makespan	Genetic algorithm

2.5. FLOW SHOP SCHEDULING PROBLEMS

The flow shop is composed of multi-stages, and each stage comprises only one machine [61]. In Flow Shop Scheduling Problem (FSSP), machines are assumed to be available during the entire planning setting [33]. It deals with the sequencing of jobs that enters a specified number of machines usually in the same order. Some of the previous studies on the FSSP machine environment are presented in Table 2.

Table 2. Existing studies on FSSP

References	Objective	Algorithms
[62]	Makespan and cost	Mixed Integer Programming (MIP)
[32]	Makespan and cost	NSGA-II and PSO
[35]	Makespan, earliness, and tardiness	Genetic algorithm and Harmony search
[63]	Makespan	MIP
[64]	Makespan and job flow-time	Dispatching rules
[65]	Makespan and total tardiness	Fruit fly optimization algorithm
[66]	Cost	MIP
[67]	Total flow time	MIP
[68]	Makespan	MIP
[61]	Makespan and total tardiness	Evolutionary algorithm
[69]	Makespan	Simulated annealing
[70]	Makespan	Genetic algorithm & tabu search
[25]	Makespan	NSGA-II

2.6. SCHEDULING ALGORITHMS

The most commonly used scheduling algorithms in previous research are meta-heuristics, exact methods, reinforcement learning, deep reinforcement learning, and multi-agent deep reinforcement learning.

2.6.1. Heuristics/ Meta-heuristics

Metaheuristics algorithm usually adopts optimization and approximation methods [44]. An optimization method is used to find solutions in mathematical computation. However, its application is limited in real-time because it takes too much time to find an optimum solution. It also requires mathematically sophisticated uses so that it is computationally intractable [17]. Contrarily, the approximation method is used when it is difficult to apply the optimization method. The approximated optimal solution is found within a specific time for a calculation. The approximation algorithm runs in linear time. As a result, it is computationally efficient [38]. For example, Evolutionary Algorithm (EA) is one of the algorithms that is used to find an approximate solution [52], [71].

Industrial environment scheduling operation needs an efficient algorithm. MIP is an effective approach for solving small-scale instances [62]. This approach is used to find an optimal solution based on the designed constraints. The main weakness of this method is that it tries to solve comprehensive problems by breaking them down into different sub-problems and then using the result of one sub-problem as input to the next sub-problem [72]. As a result, it could be difficult to find the solution in case of different conflicting constraints. For small-scale problems, centralized approaches such as MIP or CPLEX are well suited [73].

In principle, all metaheuristics can be applied to the Flexible Job Shop Problem (FJSP). Due to its fewer parameters, Particle Swarm Optimization (PSO) is much simpler and easier to maintain [46]. Although its convergence speed is fast, PSO will converge to the local optimum and will not be able to jump out with a maximum iteration rate [39]. PSO is known for convenient variable neighborhood search and flexible coding methods for solving some combinatorial optimization problems. Likewise, The Variable Neighborhood Search (VNS) algorithm is a metaheuristic optimization approach for solving combinatorial problems. It finds a solution's neighborhood until a better solution than the existing one is found, and moves to another [74].

2.6.2. Multi-Agent Systems (MAS)

The classical MAS method uses only a single dispatching rule and doesn't consider the impact of environmental changes in selecting dispatching rules [75]. This behavior in turn leads to poor scheduling performance. From the viewpoint of scheduling results, Artificial Intelligence (AI) algorithms perform better than MAS [49].

Multi-Agent System (MAS) is an agent-based system in which distributed agents make their own decisions using available information to ensure the whole system runs smoothly [76]. Another type of MAS approach is one in which agents negotiate while distributed agents make scheduling or production planning decisions. To mention a few, dynamic scheduling algorithm for allocating tasks on MAS with ring structure bidding method and negotiation method [77]; scheduling of distributed machines with negotiation and bidding protocol [78], and agent negotiation protocol to cope with the dynamic manufacturing environment [79].

However, in negotiation and agreement protocol, negotiation between agents is performed through a predetermined rule-based mechanism [80]. As a result, adaptation to the environment remains a challenge.

The combination of decentralized production systems and Industry 4.0 complicates production scheduling optimization. In comparison to the centralized production control system, the decentralized production control system has low complexity, improved scalability, and real-time capability. Implementation of MAS on these problems simplifies the solutions. Despite its solution efficiency, multi-agent systems in this environment tend to show local optimization [81]. To address these challenges, cooperative multi-agents are necessary.

2.6.3. Reinforcement Learning

Reinforcement learning (RL) is concerned with learning from experiences. It describes how agents learn the best policy to achieve the desired objectives by observing an environment, performing possible actions, and obtaining a reward as a result. The agents' goal is to maximize cumulative reward [82].

No algorithm is adaptive enough to address all the wide area of manufacturing problems. Algorithms in previous studies need high computational efforts and failed in the real manufacturing industry where there are dynamic events and uncertainties [83].

Smart manufacturing scheduling differs from job shop scheduling in several ways, including a large number of tasks and services, as well as the dynamic states and uncertainties. Scheduling is a critical process for manufacturing industries to maximize profits while lowering costs. Specifically, in a dynamic and complex manufacturing environment, poor scheduling results in higher costs, longer production times, and higher tardiness [84]. Thereby, to comply with the complexities of a manufacturing site and improve its effectiveness, scheduling must be transformed and enhanced for sustainability and intelligence.

JSSP has been thoroughly researched over the last several decades, and numerous techniques for solving classical JSSP have been developed. Nevertheless, in real manufacturing industries, the environment is mostly dynamic, such as new job arrivals and machine failure [85]. Dynamic systems begin with the jobs that arrive first and are assumed to follow a probabilistic rule [86].

Task scheduling methods are divided into two: precise and approximate scheduling methods [87]. The precise methods search the entire search space for the global optimum solution. consequently, they are computationally complex and are inefficient at solving complex scheduling problems. Conversely, approximate methods have lower complexity and get the appropriate solution faster, while having greater advantages in solving complex scheduling problems. However, approximate methods cannot ensure an optimum solution to the scheduling. A scheduling algorithm's main objective is to use a small number of machines to process a specified number of jobs while optimizing an objective [88].

The high dynamics, difficulty, and unpredictability of the JSSP environment continue to pose significant challenges [4]. Most JSSP methods are implemented as centralized algorithms with complete knowledge of the manufacturing process [88]. In contrast, one of the visions of Industry 4.0 is decentralized, self-learning, self-organizing, and self-optimizing production control [89].

The use of RL in JSSP has huge benefits. First, it is more adaptable than classical priority dispatching rule heuristics. Furthermore, developing such heuristics is tiresome because they require a great deal of expertise in a scheduling instance to be efficient [8]. RL, unlike traditional COP methods like Linear Programming (LP) or Constraint Programming (CP), can model dynamic uncertainties.

The existing research summary on RL-based DJSSP is presented in Table 3.

Table 3. Studies on RL-based DJSSP

References	Machine environment	Objective functions	Algorithms	Uncertainties
[8]	JSSP	Makespan	Actor-critic	Job order and processing time
[90]	JSSP	Robustness to processing time	DQN	Random processing time (RPT)
[91]	JSSP	Lead time	DQN	Machine Failure (MF)
[92]	FMS	Makespan	PNC & DQN	No
[93]	JSSP	Makespan	DQN	Random Job Arrival (RJA)
[85]	FJSSP	Total tardiness	DQN	New job insertion
[94]	FJSSP	Makespan	Q-learning	RPT
[74]	JSSP	Mean flow time	Q-learning	RJA & MF
[87]	FJSSP	Makespan	DQN	RJA

In dynamic JSSP multi-agent configuration, a Markov property which is considered a precondition for convergence will fail because of the independent updating policy by each agent [93]. However, integrating the whole JSSP into a single agent helps to avoid multi-agent interference with each other and convergence to local optimum. As a result, it has the advantage of stability.

2.6.4. Q-learning

Q-learning is characterized as an off-policy method and with its early convergence behavior [82]. In Q-learning, there are different No.s: the learning process could result in a local optimum solution or it could take longer to succeed and generalization problem [86]. Similarly, the presence of a large number of environmental states limits the accuracy of the applied RL approach [95]. SARSA and Q-learning are model-free Temporal Difference (TD) algorithms. In SARSA, the action is chosen at random with a probability, while in Q-learning, the action is the one that increases the value. That means, Q-learning greedily learns state-action value without looking at the policy [96]. If the environment is entirely observable, the DP approach could be used to infer optimum policy. However, usually, it is unknown, and no precise understanding of the environment exists. Under these scenarios, RL finds the optimum strategy using an iterative process [82].

One of the main challenges of Q-learning in scheduling is its limitation on continuous state space. In the practical industrial environment, where there is a continuous state feature, the total number of states is potentially infinite, establishing a massive Q table is unrealistic [8].

2.6.5. Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) has been recently applied from Traveling Salesman Problem (TSP) in a graph optimization to Satisfiability problem [97]. DRL solutions to scheduling problems, on the other hand, are more recent and limited. DRL has the following features which are suitable for intelligent scheduling: (1) Ability to communicate with its surroundings and utilization of feedback data to optimize its strategy. (2) DRL, like different machine learning algorithms, requires intensive offline training; however, becomes efficient while executed. (3) The synchronization of DNN parameters takes advantage of the scheduling policy consistency between the simulation model and a real factory [92]. Solving dynamic scheduling problems requires the environment to satisfy the MDP requirement.

In DRL, the neural network is used to pick a candidate action. The main advantage of DRL is the ability to demonstrate the complex model in a comparatively simple manner than RL [84]. Furthermore, the agent learns the optimal strategy by trial and error, and this strategy helps the agent to decide in a dynamic environment.

2.6.6. Deep Q Network

When DNN is utilized to approximate the Q-value, it is known as a Deep Q Network (DQN). The problem of RL is its inability to converge because of the correlation between the expected value and Q-value [84]. DQN uses experience replay memory, which stores encountered data, to choose the data at random during learning to eliminate the correlation. The target network's weight is also iteratively updated for optimal convergence of the anticipated Q-value. The only distinction between DQN and Q-learning is; that in DQN, the agent's brain is DNN, whereas, in Q-learning, it is Q-table.

In complex job shop settings, breaking down and adjusting global objectives to local Key Performance Indicators (KPIs) is difficult. The DQN agent optimizes globally rather than locally. This implies that manually breaking down production objectives is not essential [89]. Regardless of its benefits, DQN has also drawbacks. First, training is time-consuming. Second, due to the black-box nature of the neural network, it is difficult to anticipate how DQN agents will behave in uncertain situations. It is also incapable to deal with continuous action spaces. Because of the continuous nature of the training process, each agent's policy changes regularly. This prevents straightforward usage of experience replay, which enables DQN to learn stability. Moreover, Deep Policy Gradient (DPG) also suffers from high variance [98]. Experience replay is the agent's huge experience data pool in which the experience, at each step, will be stored [8].

2.6.7. Drawbacks of DRL

There are two types of RL: model-free and model-based. The latter forecast the future state and understand the entire MDP transition model. Conversely, the majority of JSSP states are usually huge, if not infinite, which makes it impossible to understand the entire changing scenario [8].

The major challenge in model-free DRL is the absence of robustness when the environment changes [99]. It has less potential for reacting to huge environmental uncertainty. These constraints can be solved by retraining the model on the different distributions before deployment. Likewise, combining this approach with a model-based strategy enables solving the problem. Furthermore, DRL involves a large sample, which could be obtained by interacting with the simulator, to learn an optimum strategy. This simulation model should be efficient, but usually hard to construct.

In the case of Policy Gradients (PG) methods, there is no guarantee of optimality [99]. The best strategy to train model-free algorithms is by making a deliberate mistake early on and then determining which actions result in the maximum long-term rewards using a series of Monte Carlo simulations of the scheduling environment. Designing a reward is also a big challenge in DRL [8].

2.7. Scheduling Technology and Tools

The emergence of CPS has led to the development of digital twin technology. Digital twins are a digital copy of the physical machine and are modeled based on different dimensions such as geometric, physical, behavioral, rule, and data modeling.

2.7.1. Digital Twin

Digital Twin (DT) was first introduced by prof. Michael Grieves in 2003 [100]. The main idea of DT is the realization of interoperability and interconnection between virtual and physical elements of the shop floor [30]. There is no commonly agreed definition of a digital twin. However, the general definition of DT is a simulation model of a real-world system that is linked to a physical twin [101]. This linkage aids in the collection of actual data for simulation, and forward responses to the physical environment to fine-tune the behavior of the actual component [102].

DT can be used in a variety of settings, including production and manufacturing processes [31], and in all stages of product lifecycle [102], digital product development, process planning, lean manufacturing, construction of smart cities, energy, and mining solutions [103]. Nevertheless, it is not yet extensively implemented in the production stage [102]. The main advantage of using DT includes a reflection of the real-time working process and direction for the subsequent operational process of the physical model. Apart from simulation, DT is used to showcase unknown problems by predictions [103]. DT enables cyber-physical integration and real-time management between physical objects and digital representation [20].

DT is composed of four levels i.e., geometry, physics, behavior, and rules [104], and it helps not only to show the dynamic and geometric features but also to define the physical attributes and rules [102]. Using DT in production has also some challenges. To monitor composite twin data and extract insights it represents, an effective technique is required [31]. In addition to this, it is time-consuming and costly and requires experts in different areas, for the construction of complete and detailed DT [29]. Accurate and highly efficient communication between physical and digital spaces is needed [101]. Moreover, security No.s are also a critical component that needs to be considered before applying it to a larger scale.

In previous research, the digital twin has been used to assist with a scheduling problems. Machine failure detection and performance evaluation [27], [29], analysis of transportation and production processing stages [60], process simulation and production scheduling [17], and production scheduling for defense weapon systems [83] are among the studies. The reason why the simulation package becomes better than the stochastic Petri-net package is, because of its convenience, timely, and easier to operate nature [102].

Based on the analysis, existing manufacturing paradigms have the following limitations. Interconnection between physical machines and virtual models, the interconnection between the virtual model and physical production, generation of accurate data by converging the data from virtual and physical spaces, a realization of intelligent production simulation and optimization [30]; and lack of consideration of actual transportation condition in shop floors [60] are among the challenges. Most of the existing studies on digital twins focus on individual machines [83] and remain a challenge on how to construct and when to apply them on the shop floor [102].

2.7.2. Petri Net

A Petri net (place/transition net) is a directed connected graph that represents a finite set of arcs and used as a tool for process transitions. Topologically structured graphs or nets, which can represent regulations and connections are more capable of modeling production processes than standard tensors [92]. It is a popular method of process modeling not for searching for optimal scheduling. For example, in heuristic strategies, Petri nets design the manufacturing process, while heuristic rules focus on resolving scheduling conflicts [95]. The main drawback of PN-based scheduling is state space explosion.

4. DISCUSSION

In the era of smart manufacturing, a vast amount of data is being generated from different smart products and resources, which always provide feedback about their status to the system. Despite the extraction of the enormous amount of data, machine interoperability between shop floor environments is still a challenge.

The future of IoT objects will be standardized towards everything-as-a-service, which will bring better interoperability, re-usability, lower complexity, and higher scalability options. However, it will also incur high costs, have a lack of standards, lack of knowledge, and other limitations. The research findings based on the expert ideas in [105] show that service-oriented architecture will be the core component of smart manufacturing. So, this will help to solve the interoperability problems.

The most challenges found in the study are shop floor environment challenges related to CPS and handling of large amounts of information in adaptive manufacturing, machine pro-activeness (suggesting changes by themselves) and scheduling, decentralized and flexible decision making, human-robot collaboration, and constant evolution of new technologies.

Moreover, the choice of algorithm for the industrial environment is still vague. Usually, academic research algorithms' performance is evaluated with existing algorithms on the same setting, parameters, and constraints. This strategy will not help to implement the solution in the real environment. Algorithmic scheduling solutions should be evaluated not only with the existing algorithm but also with the Key Performance Indicators (KPI) of the particular factory.

However, if the solution has to deal with the real industrial environment, then adaptive scheduling such as RL and DRL can be the best fit. Dynamic Programming (DP) operates in fully observable MDP. In other words, DP can only be applied in environments with fully known transition probability. But in the real world, it is difficult to anticipate the entire environment and it is also computationally expensive. Similarly, Monte Carlo (MC) methods cannot be applied in an expensive critical industrial environment. The backup or update is performed at the terminal state. To update the value function, this approach waits for something to happen. In this case, if the machine is broken down, or if it explodes, it is difficult to reverse the initial working state.

Multi-agent Deep Reinforcement Learning (MADRL) scheduling algorithms are used to deal with dynamic uncertainty and a huge environment. However, the social dilemma is the main challenge to implement the solution. In another word, if each agent is competing with each other in a multi-agent environment, then they will waste resources. So, to make them synchronized and achieve a common goal, an appropriate reward function is needed. In MADRL, crafting a reward function is the most difficult task.

5. CONCLUSION AND FUTURE WORK

Scheduling tasks requires a comprehensive accounting of jobs and resources which are available with possible limitations in their use [17]. Scheduling problems are not only NP-hard but also computationally difficult combinatorial problems.

The common bottlenecks in dynamic scheduling include prediction of machine availability, disruption detection, and performance evaluation [29], [31]. Dynamic events and uncertainties are the main cause of scheduling performance deterioration and production disruption. The widely used approach of disturbance detection is, setting predefined constraints as a benchmark to evaluate the change between actual production and the anticipated plan. However, manufacturing states always change with time so the predefined benchmarks cannot correctly visualize currently anticipated production states. The other limitation of existing dynamic scheduling research is that dynamic events are considered from direct assumptions or derived by statistics rather than actual production data. As a result, it fails to provide interactive feedback and is limited in solving real-time problems.

Smart manufacturing system usually fails to achieve the desired objective because of non-reasonable design [106]. Incorporating AI techniques with a digital twin-based design approach can be a solution to such problems. In the majority of existing scheduling solutions, the machine states are modeled as a binary state i.e., up or down. However, it could also be interesting to consider the rate of machine performance degradation and the time to go to an intermediate state before its failure.

States in a job shop environment are infinite. As a result, applying model-based RL methods that know the entire MDP transition model is not recommended. An infinite number of states makes it difficult to understand the entire transition situation. Moreover, the challenging issues in model-based RL scheduling is the exhaustive computation of Q values. When the number of machines and jobs is more than twenty, the agent will find it hard to find an optimum policy and difficult to converge to the global optimum. Because the value has to be computed for every possible state. However, improving the policy directly using the policy-based approach leads to convergence and an optimum policy.

6. REFERENCES:

[1] N. Stricker, A. Kuhnle, R. Sturm, S. Friess, "Reinforcement learning for adaptive order dispatching in the semiconductor industry", CIRP Annals, Vol. 67, No. 1, 2018, pp. 511-514.

[2] Y.-R. Shiue, K.-C. Lee, C.-T. Su, "Real-time scheduling for a smart factory using a reinforcement learning approach", Computers & Industrial Engineering, Vol. 125, 2018, pp. 604-614.

[3] T. J. Ikonen, K. Heljanko, I. Harjunkoski, "Reinforcement learning of adaptive online rescheduling timing and computing time allocation", Computers & Chemical Engineering, Vol. 141, 2020., p. 106994

[4] H. Hu, X. Jia, Q. He, S. Fu, K. Liu, "Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0", Computers & Industrial Engineering, Vol. 149, 2020, p. 106749.

[5] L. Li et al. "Bilevel Learning Model Towards Industrial Scheduling", arXiv Prepr. arXiv2008.04130, 2020.

[6] Y. Du, T. Wang, B. Xin, L. Wang, Y. Chen, L. Xing, "A data-driven parallel scheduling approach for multiple agile earth observation satellites", IEEE Transactions on Evolutionary Computation, Vol. 24, No. 4, 2019, pp. 679-693.

[7] H. Lu, X. Zhang, S. Yang, "A learning-based iterative method for solving vehicle routing problems", Proceedings of the International Conference on Learning Representations, 2019.

[8] C. L. Liu, C. C. Chang, C. J. Tseng, "Actor-critic deep reinforcement learning for solving job shop scheduling problems", IEEE Access, Vol. 8, 2020, pp. 71752-71762.

[9] A. Kuhnle, L. Schäfer, N. Stricker, G. Lanza, "Design, implementation and evaluation of reinforcement learning for an adaptive order dispatching in job shop manufacturing systems", Procedia CIRP, Vol. 81, 2019, pp. 234-239.

[10] Y. Liu, L. Zhang, L. Wang, Y. Xiao, X. Xu, M. Wang, "A framework for scheduling in cloud manufacturing with deep reinforcement learning", Proceedings of the IEEE 17th International Conference on Industrial Informatics, 2019, Vol. 1, pp. 1775-1780.

[11] H. Zhu, M. Li, Y. Tang, Y. Sun, "A deep-reinforcement-learning-based optimization approach for real-time scheduling in cloud manufacturing", IEEE Access, Vol. 8, pp. 9987-9997, 2020.

[12] S. Mittal, M. A. Khan, D. Romero, T. Wuest, "Smart manufacturing: Characteristics, technologies and enabling factors", Proceedings of the Institution of Mechanical Engineers, Part B, Vol. 233, No. 5, pp. 1342-1361, 2019.

[13] J. Davis et al. "Smart manufacturing", Annual Review of Chemical and Biomolecular Engineering, Vol. 6, 2015, pp. 141-160.

[14] P. O'Donovan, K. Leahy, K. Bruton, D. T. J. O'Sullivan, "An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities", Journal of Big Data, Vol. 2, No. 1, 2015, pp. 1-26.

[15] J. Wang, Y. Ma, L. Zhang, R. X. Gao, D. Wu, "Deep learning for smart manufacturing: Methods and applications", Journal of Manufacturing Systems, Vol. 48, 2018, pp. 144-156.

[16] F. Tao and Q. Qi, "New IT driven service-oriented smart manufacturing: framework and characteristics", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 49, No. 1, 2017, pp. 81-91.

[17] A. Koulouris, N. Misailidis, D. Petrides, "Applications of process and digital twin models for production simulation and scheduling in the manufacturing of food ingredients and products", Food and Bioproducts Processing, Vol. 126, 2021, pp. 317-333.

[18] H. S. Kang et al. "Smart manufacturing: Past research, present findings, future directions", International Journal of Precision Engineering and Manufacturing-Green Technology, Vol. 3, No. 1, 2016, pp. 111-128.

[19] A. Kusiak, "Smart manufacturing",International Journal of Production Research, Vol. 56, No. 1-2, 2018, pp. 508-517.

[20] Q. Qi, F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison", IEEE Access, Vol. 6, 2018, pp. 3585-3593.

[21] F. Tao, Q. Qi, A. Liu, A. Kusiak, "Data-driven smart manufacturing", Journal of Manufacturing Systems, Vol. 48, 2018, pp. 157-169.

[22] S. Manufacturing and others, "Implementing 21st Century Smart Manufacturing", Smart Manufacturing Leadership Coalition, 2011.

[23] P. Zheng et al. "Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, future perspectives", Frontiers in Mechanical Engineering, Vol. 13, No. 2, 2018, pp. 137-150.

[24] J. Lee, E. Lapira, B. Bagheri, H. Kao, "Recent advances and trends in predictive manufacturing systems in big data environment", Manufacturing Letters, Vol. 1, No. 1, 2013, pp. 38-41.

[25] M. Zandieh, S. M. Sajadi, R. Behnoud, "Integrated production scheduling and maintenance planning in a hybrid flow shop system: a multi-objective approach", International Journal of System Assurance Engineering and Management, Vol. 8, No. 2, 2017, pp. 1630-1642.

[26] C. Zhang, Y. Zhou, K. Peng, X. Li, K. Lian, S. Zhang, "Dynamic flexible job shop scheduling method based on improved gene expression programming", Measurement and Control, Vol. 54, No. 7-8, 2021, pp. 1136-1146.

[27] E. Negri, V. Pandhare, L. Cattaneo, J. Singh, M. Macchi, J. Lee, "Field-synchronized Digital Twin framework for production scheduling with uncertainty", Journal of Intelligent Manufacturing, Vol. 32, No. 4, 2021, pp. 1207-1228.

[28] Y. Wang, Z. Wu, "Model construction of planning and scheduling system based on digital twin", The International Journal of Advanced Manufacturing Technology, Vol. 109, No. 7-8, 2020, pp. 2189-2203.

[29] M. Zhang, F. Tao, A. Y. C. Nee, "Digital twin enhanced dynamic job-shop scheduling", Journal of Manufacturing Systems, Vol. 58, 2021, pp. 146-156.

[30] H. Zhang, G. Zhang, Q. Yan, "Digital twin-driven cyber-physical production system towards smart shop-floor", Journal of Ambient Intelligence and Humanized Computing, Vol. 10, No. 11, 2019, pp. 4439-4453.

[31] Z. Liu, W. Chen, C. Zhang, C. Yang, Q. Cheng, "Intelligent scheduling of a feature-process-machine tool supernetwork based on digital twin workshop", Journal of Manufacturing Systems, Vol. 58, 2021, pp. 157-167.

[32] R. Boufellouh, F. Belkaid, "Bi-objective optimization algorithms for joint production and maintenance scheduling under a global resource constraint: Application to the permutation flow shop problem", Computers & Operations Research, Vol. 122, 2020, p. 104943.

[33] A. Branda, D. Castellano, G. Guizzi, V. Popolo, "Metaheuristics for the flow shop scheduling problem with maintenance activities integrated", Computers & Industrial Engineering, Vol. 151, 2021., p. 106989

[34] H. Ye, X. Wang, K. Liu, "Adaptive Preventive Maintenance for Flow Shop Scheduling With Resumable Processing", IEEE Transactions on Automation Science and Engineering, Vol. 18, No. 1, 2021., pp. 106-113

[35] W. Cui, Z. Lu, C. Li, X. Han, "A proactive approach to solve integrated production scheduling and maintenance planning problem in flow shops", Computers & Industrial Engineering, Vol. 115, 2018, pp. 342-353.

[36] C. Anand Deva Durai, M. Azath, J. S. C. Jeniffer, "Integrated Search Method for Flexible Job Shop Scheduling Problem Using HHS--ALNS Algorithm", SN Computer Science, Vol. 1, No. 2, 2020, pp. 1-6.

[37] X. Wu, J. Peng, X. Xiao, S. Wu, "An effective approach for the dual-resource flexible job shop scheduling problem considering loading and unloading", Journal of Intelligent Manufacturing, Vol. 32, No. 3, 2021, pp. 707-728.

[38] W. Kubiak, Y. Feng, G. Li, S. P. Sethi, C. Sriskandarajah, "Efficient algorithms for flexible job shop scheduling with parallel machines", Naval Research Logistics, Vol. 67, No. 4, 2020, pp. 272-288.

[39] R. Li, H. Ma, "Integrating preventive maintenance planning and production scheduling under reen-

trant job shop", Mathematical Problems in Engineering, Vol. 2017, 2017.

[40] M. Ghaleb, H. Zolfagharinia, S. Taghipour, "Real-time production scheduling in the Industry-4.0 context: Addressing uncertainties in job arrivals and machine breakdowns", Computers & Operations Research, Vol. 123, 2020, p. 105031.

[41] O. J. Shukla, G. Soni, R. Kumar, A. Sujil, "An agent-based architecture for production scheduling in dynamic job-shop manufacturing system", Automatisierungstechnik, Vol. 66, No. 6, 2018, pp. 492-502.

[42] M. Abderrahim, A. Bekrar, D. Trentesaux, N. Aissani, K. Bouamrane, "Bi-local search based variable neighborhood search for job-shop scheduling problem with transport constraints", Optimization Letters, 2020, pp. 1-26.

[43] B. Grosch, T. Kohne, M. Weigold, "Multi-objective hybrid genetic algorithm for energy adaptive production scheduling in job shops", Procedia CIRP, Vol. 98, 2021, pp. 294-299.

[44] W.-R. Jong, H.-T. Chen, Y.-H. Lin, Y.-W. Chen, T.-C. Li, "The multi-layered job-shop automatic scheduling system of mould manufacturing for Industry 3.5", Computers & Industrial Engineering, Vol. 149, 2020, p. 106797.

[45] E. K. A. Pakpahan, S. Kristina, A. Setiawan, "Proposed algorithm to improve job shop production scheduling using ant colony optimization method", in IOP Conference Series: Materials Science and Engineering, Vol. 277, No. 1, 2017, p. 12050.

[46] R. Zarrouk, I. E. Bennour, A. Jemai, "A two-level particle swarm optimization algorithm for the flexible job shop scheduling problem", Swarm Intelligence, Vol. 13, No. 2, 2019, pp. 145-168.

[47] X. Chen, J. Li, Y. Han, H. Sang, "Improved artificial immune algorithm for the flexible job shop problem with transportation time", Measurement and Control, Vol. 53, No. 9-10, 2020, pp. 2111-2128.

[48] S. Zhang, S. Wang, "Flexible assembly job-shop scheduling with sequence-dependent setup times and part sharing in a dynamic environment: Constraint programming model, mixed-integer programming model, dispatching rules", IEEE Transactions on Engineering Management, Vol. 65, No. 3, 2018, pp. 487-504.

[49] S. Tian, T. Wang, L. Zhang, X. Wu, "An energy-efficient scheduling approach for flexible job shop problem in an Internet of manufacturing things environment", IEEE Access, Vol. 7, 2019, pp. 62695-62704.

[50] A. Baykasouglu, F. S. Madenouglu, A. Hamzadayi, A. Baykasoğlu, F. S. Madenoğlu, A. Hamzaday, "Greedy randomized adaptive search for dynamic flexible job-shop scheduling", Journal of Manufacturing Systems, Vol. 56, 2020, pp. 425-451.

[51] X. Wu, X. Shen, C. Li, "The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously", Computers & Industrial Engineering, Vol. 135, 2019, pp. 1004-1024.

[52] R. H. Caldeira, A. Gnanavelbabu, T. Vaidyanathan, "An effective backtracking search algorithm for multi-objective flexible job shop scheduling considering new job arrivals and energy consumption", Computers & Industrial Engineering, Vol. 149, 2020, p. 106863.

[53] M. Dai, D. Tang, A. Giret, M. A. Salido, "Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints", Robotics and Computer-Integrated Manufacturing, Vol. 59, 2019, pp. 143-157.

[54] B. Denkena, F. Schinkel, J. Pirnay, S. Wilmsmeier, "Quantum algorithms for process parallel flexible job shop scheduling", CIRP Journal of Manufacturing Science and Technology, Vol. 33, 2021, pp. 100-114.

[55] J. L. Andrade-Pineda, D. Canca, P. L. Gonzalez-R, M. Calle, "Scheduling a dual-resource flexible job shop with makespan and due date-related criteria", Annals of Operations Research, Vol. 291, No. 1, 2020, pp. 5-35.

[56] D. C. Bissoli, N. Zufferey, A. R. S. Amaral, "Lexicographic optimization-based clustering search metaheuristic for the multiobjective flexible job shop scheduling problem", International Transactions in Operational Research, Vol. 28, No. 5, 2021, pp. 2733-2758.

[57] R. H. Caldeira, A. Gnanavelbabu, "A simheuristic approach for the flexible job shop scheduling problem with stochastic processing times", Simulation, Vol. 97, No. 3, 2021, pp. 215-236.

[58] S. L. Aquinaldo, N. R. Cucuk, others, "Optimization in job shop scheduling problem using Genetic Algorithm (study case in furniture industry)", IOP Conference Series: Materials Science and Engineering, Vol. 1072, No. 1, 2021, p. 12019.

[59] J. M. Novas, "Production scheduling and lot streaming at flexible job-shops environments using constraint programming", Computers & Industrial Engineering, Vol. 136, 2019, pp. 252-264.

[60] J. Yan, Z. Liu, C. Zhang, T. Zhang, Y. Zhang, C. Yang, "Research on flexible job shop scheduling under finite transportation conditions for digital twin workshop", Robotics and Computer-Integrated Manufacturing, Vol. 72, No. June, 2021, p. 102198.

[61] Y. Fu, M. Zhou, X. Guo, L. Qi, "Scheduling dual-objective stochastic hybrid flow shop with deteriorating jobs via bi-population evolutionary algorithm", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 50, No. 12, 2019, pp. 5037-5048.

[62] Z. Zhang, Q. Tang, M. Chica, "Maintenance costs and makespan minimization for assembly permutation flow shop scheduling by considering preventive and corrective maintenance", Journal of Manufacturing Systems, Vol. 59, 2021, pp. 549-564.

[63] S. Aghighi, S. T. A. Niaki, E. Mehdizadeh, A. A. Najafi, "Open-shop production scheduling with reverse flows", Computers & Industrial Engineering, Vol. 153, 2021, p. 107077.

[64] T.-S. Yu, M. Pinedo, "Flow shops with reentry: Reversibility properties and makespan optimal schedules", European Journal of Operational Research, Vol. 282, No. 2, 2020, pp. 478-490.

[65] Y. Fu, M. C. Zhou, X. Guo, L. Qi, "Stochastic multi-objective integrated disassembly-reprocessing-reassembly scheduling via fruit fly optimization algorithm", Journal of Cleaner Production, Vol. 278, 2021, p. 123364.

[66] J. Seif, M. Dehghanimohammadabadi, A. J. Yu, "Integrated preventive maintenance and flow shop scheduling under uncertainty", Flexible Services and Manufacturing Journal, Vol. 32, No. 4, 2020, pp. 852-887.

[67] P. P. Suryadhini, S. Sukoyo, S. Suprayogi, A. H. Halim, "A batch scheduling model for a three-stage flow shop with job and batch processors considering a sampling inspection to minimize expected total actual flow time", Journal of Industrial Engineering and Management, Vol. 14, No. 3, 2021, pp. 520-537.

[68] J. Bautista-Valhondo, R. Alfaro-Pozo, "Mixed integer linear programming models for Flow Shop Scheduling with a demand plan of job types", Cent. European Journal of Operational Research, Vol. 28, No. 1, 2020, pp. 5-23.

[69] J. Krishnaraj and S. Thiagarajan, "A two-phase simulated annealing algorithm to minimise the completion time variance of jobs in flowshops", International Journal of Process Management and Benchmarking, Vol. 10, No. 2, 2020, pp. 261-281.

[70] W. Li, D. Han, L. Gao, X. Li, Y. Li, "Integrated production and transportation scheduling method in hybrid flow shop", Chinese Journal of Mechanical Engineering, Vol. 35, No. 1, 2022, pp. 1-20.

[71] L. Sun, L. Lin, M. Gen, H. Li, "A hybrid cooperative coevolution algorithm for fuzzy flexible job shop scheduling", IEEE Transactions on Fuzzy Systems, Vol. 27, No. 5, 2019, pp. 1008-1022.

[72] A. Al-Shayea, E. Fararah, E. A. Nasr, H. A. Mahmoud, "Model for Integrating Production Scheduling and Maintenance Planning of Flow Shop Production System", IEEE Access, Vol. 8, 2020, pp. 208826-208835.

[73] W. Bouazza, Y. Sallez, B. Beldjilali, "A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect", IFAC-Papers OnLine, Vol. 50, No. 1, 2017, pp. 15890-15895.

[74] J. Shahrabi, M. A. Adibi, M. Mahootchi, "A reinforcement learning approach to parameter estimation in dynamic job shop scheduling", Computers & Industrial Engineering, Vol. 110, 2017, pp. 75-82.

[75] H. Zhu, M. Chen, Z. Zhang, D. Tang, "An adaptive real-time scheduling method for flexible job shop scheduling problem with combined processing

constraint", IEEE Access, Vol. 7, 2019, pp. 125113-125121.

[76] S. Karnouskos, P. Leitao, "Key contributing factors to the acceptance of agents in industrial environments", IEEE Transactions on Industrial Informatics, Vol. 13, No. 2, 2016, pp. 696-703.

[77] M. Owliya, M. Saadat, R. Anane, M. Goharian, "A new agents-based model for dynamic job allocation in manufacturing shopfloors", IEEE Systems Journal, Vol. 6, No. 2, 2012, pp. 353-361.

[78] C.-J. Huang, L.-M. Liao, "A multi-agent-based negotiation approach for parallel machine scheduling with multi-objectives in an electro-etching process", International Journal of Production Research, Vol. 50, No. 20, 2012, pp. 5719-5733.

[79] S. Wang, J. Wan, D. Zhang, D. Li, C. Zhang, "Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination", Computer Networks, Vol. 101, 2016, pp. 158-168.

[80] Y. G. Kim, S. Lee, J. Son, H. Bae, B. Do Chung, "Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system", Journal of Manufacturing Systems, Vol. 57, 2020, pp. 440-450.

[81] L. Monostori, J. Váncza, S. R. T. Kumara, "Agent-based systems for manufacturing", CIRP Annals, Vol. 55, No. 2, 2006., pp. 697-720

[82] M. H. Moghadam, S. M. Babamir, "Makespan reduction for dynamic workloads in cluster-based data grids using reinforcement-learning based scheduling", Vol. 24, 2018, pp. 402-412.

[83] H. Latif, B. Starly, "A simulation algorithm of a digital twin for manual assembly process", Procedia Manufacturing, Vol. 48, No. 2019, 2020, pp. 932-939.

[84] S. Lee, Y. Cho, Y. H. Lee, "Injection Mold Production Sustainable Scheduling Using Deep Reinforcement Learning", Sustainability, Vol. 12, No. 20, 2020, p. 8718.

[85] S. Luo, "Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning", Applied Soft Computing, Vol. 91, 2020, p. 106208.

[86] M. E. Aydin, E. Öztemel, "Dynamic job-shop scheduling using reinforcement learning agents", Robotics and autonomous systems, Vol. 33, No. 2-3, 2000, pp. 169-178.

[87] L. Zhou, L. Zhang, B. K. P. Horn, "Deep reinforcement learning-based dynamic scheduling in smart manufacturing", Procedia CIRP, Vol. 93, 2020, pp. 383-388.

[88] M. S. A. Hameed, A. Schwung, "Reinforcement learning on job shop scheduling problems using graph networks", arXiv Prepr. arXiv2009.03836, 2020.

[89] B. Waschneck et al. "Deep reinforcement learning for semiconductor production scheduling", Institute of Electrical and Electronics Engineers Inc., 2018, pp. 301-306.

[90] D. Shi, W. Fan, Y. Xiao, T. Lin, C. Xing, "Intelligent scheduling of discrete automated production line via deep reinforcement learning", International Journal of Production Research, Vol. 58, No. 11, 2020, pp. 3362-3380.

[91] T. Altenmüller, T. Stüker, B. Waschneck, A. Kuhnle, G. Lanza, "Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints", Production Engineering, Vol. 14, 2020, pp. 319-328.

[92] L. Hu, Z. Liu, W. Hu, Y. Wang, J. Tan, F. Wu, "Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network", Journal of Manufacturing Systems, Vol. 55, 2020, pp. 1-14.

[93] B. Luo, S. Wang, B. Yang, L. Yi, "An improved deep reinforcement learning approach for the dynamic job shop scheduling problem with random job arrivals", Journal of Physics: Conference Series, Vol. 1848, No. 1, 2021, p. 12029.

[94] I.-B. Park, J. Huh, J. Kim, J. Park, "A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities", IEEE Transactions on Automation Science and Engineering, Vol. 17, No. 3, 2019, pp. 1420-1431.

[95] M. Drakaki, P. Tzionas, "Manufacturing Scheduling Using Colored Petri Nets and Reinforcement Learning", Applied Sciences, Vol. 7, No. 2, 2017, p. 136.

[96] N. Chauhan, N. Choudhary, K. George, A comparison of reinforcement learning based approaches to appliance scheduling. Institute of Electrical and Electronics Engineers Inc., 2016, pp. 253-258.

[97] P. Tassel, M. Gebser, K. Schekotihin, "A reinforcement learning environment for job-shop scheduling", arXiv Prepr. arXiv2104.03760, 2021.

[98] S. Li, Y. Wu, X. Cui, H. Dong, F. Fang, S. Russell, "Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient", Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 01, 2019, pp. 4213-4220.

[99] C. D. Hubbs, C. Li, N. V. Sahinidis, I. E. Grossmann, J. M. Wassick, "A deep reinforcement learning approach for chemical production scheduling", Computers & Chemical Engineering, Vol. 141, 2020, p. 106982.

[100] Y. Wang, Z. Wu, "Digital twin-based production scheduling system for heavy truck frame shop", Proceedings of the Institution of Mechanical Engineers, Part C, Vol. 236, No. 4, 2022, pp. 1931-1942.

[101] A. Villalonga et al. "A decision-making framework for dynamic scheduling of cyber-physical production systems based on digital twins", Annual Reviews in Control, 2021.

[102] C. Zhuang, T. Miao, J. Liu, H. Xiong, "The connotation of digital twin, the construction and application method of shop-floor digital twin", Robotics and Computer-Integrated Manufacturing, Vol. 68, 2021, p. 102075.

[103] H. Yu, S. Han, D. Yang, Z. Wang, W. Feng, "Job Shop Scheduling Based on Digital Twin Technology: A Survey and an Intelligent Platform", Complexity, Vol. 2021, 2021.

[104] F. Tao, M. Zhang, "Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing", IEEE Access, Vol. 5, 2017, pp. 20418-20427.

[105] M. Moghaddam, M. N. Cadavid, C. R. Kenley, A. V Deshmukh, "Reference architectures for smart manufacturing: A critical review", Journal of Manufacturing Systems, Vol. 49, 2018, pp. 215-225.

[106] Q. Liu et al. "Digital twin-based designing of the configuration, motion, control, optimization model of a flow-type smart manufacturing system", Journal of Manufacturing Systems, Vol. 58, 2021, pp. 52-64.