




## Article

# Modeling of Digital Twin Workshop in Planning via a Graph Neural Network: The Case of an Ocean Engineering Manufacturing Intelligent Workshop

Jinghua Li <sup>1,2</sup>, Wenhao Yin <sup>3,\*</sup> , Boxin Yang <sup>1,\*</sup>, Li Chen <sup>4</sup>, Ruipu Dong <sup>3</sup>, Yidong Chen <sup>3</sup>  and Hanchen Yang <sup>2</sup> <sup>1</sup> College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin 150001, China<sup>2</sup> Sanya Nanhai Innovation and Development Base of Harbin Engineering University, Harbin Engineering University, Sanya 572024, China<sup>3</sup> College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China<sup>4</sup> Tianjin Construction Company, Offshore Oil Engineering Co., Ltd., Tianjin 300461, China

\* Correspondence: yinwenhao1995@hrbeu.edu.cn (W.Y.); yangboxin@foxmail.com (B.Y.)

**Abstract:** In the era of Industry 4.0 to 5.0, the manufacturing industry is dedicated to improving its production efficiency, control capability and competitiveness with intelligent enhancement. As a typical discrete manufacturing industry, it is difficult for ocean engineering (OE) manufacturers to accurately control the entire production process, and the establishment of an integrated system supported by digital twin (DT) technology is a better solution. This paper proposes a comprehensive set of system architectures for the DT workshop. It focuses on planning, which is the main line of control, to establish a model based on graph neural networks (GNNs) and suggests five decision-support approaches associated with the model from a practical application perspective. The utilization of complete twin data for prediction and visual simulation effectively eliminates the problem of unexpected factors interfering with scheduling in enterprise production planning and achieves the goals of rapid processing and just-in-time completion. The planning model is based on the attention mechanism, which characterizes the disjunctive graph, extracts the input GNN, and outputs the scheduling decision by constructing the multi-attention network of operations and machines to deal with the complicated “operation–machine” combination relationship. The proposed method has been verified in the case of structural assembly and welding workshops, has validity and reliability, and is superior to the traditional priority scheduling rules and heuristics in terms of precision rate and rapidity. Furthermore, the DT system completes the production line application, and its proven reliability supports its full-scale application in future smart factories.

**Keywords:** digital twin; graph neural network; intelligent workshop modeling; ocean engineering manufacturing; decision support



**Citation:** Li, J.; Yin, W.; Yang, B.; Chen, L.; Dong, R.; Chen, Y.; Yang, H. Modeling of Digital Twin Workshop in Planning via a Graph Neural Network: The Case of an Ocean Engineering Manufacturing Intelligent Workshop. *Appl. Sci.* **2023**, *13*, 10134. <https://doi.org/10.3390/app131810134>

Academic Editor: Kiril Tenekedjiev, D.Sc.

Received: 28 July 2023

Revised: 2 September 2023

Accepted: 5 September 2023

Published: 8 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The synchronization of rapid Artificial Intelligence innovation and the transformation and upgrading of the manufacturing industry is underway. Numerous companies are eliminating outdated capacity and creating a new generation of smart factories by introducing a large number of automated equipment, establishing production process simulations, and developing manufacturing execution systems. The Digital Twin (DT) workshop is driven by real-time data [1], is the representative of the subsequent generation of manufacturing and information technology integration [2] and is steadily evolving into the best intelligent solution. A DT system can integrate the entire range of production chain components efficiently in order to better address the management challenges of discrete manufacturing industries, such as a large amount of production data, high data maintenance workload, extensive product non-standardization, and reliance on manual experience. The productivity improvements have been wholly proven for the mainstream DT systems, such as

Siemens Tecnomatix [3] and Dassault’s 3D Experience [4], piloted in civil engineering, robotics manufacturing, aerospace, and other fields. Prior to full commercial application, it is necessary to establish a reliable data processing model for the interaction of simulated and actual measurement data to achieve iterative optimization of the DT system.

Ocean Engineering (OE) manufacturing, a giant system engineering in discrete manufacturing mode, is developing late in the field of digitalization and has completed the networking and upgrading work of crucial equipment in recent years, lacking the ability to process and apply a large amount of production data to guide the optimization of production processes, and is in urgent need of a systematic, intelligent control framework for the whole production process. The OE manufacturing process is complicated and involves the intersection of multiple disciplinary fields, specifically material pretreatment, cutting, plate assembly, welding, and assembly, as shown in Figure 1. Considering two modes of the production process, human–machine collaboration and fully automated operation of equipment, the architecture must be designed according to the actual conditions and process routes when establishing the DT workshop. In this paper, based on the five-dimensional model proposed by Tao, Fei, et al. [5], we refine the operational behavior state criterion and real-time mapping under the finite state mechanism according to the integrated operation mode of structural assembly and welding workshop, and combine the feasible approaches in practice from several research papers [6–9] to design the architecture of OE intelligent workshop implementation, covering equipment layer, acquisition layer, data layer, application layer and presentation layer.

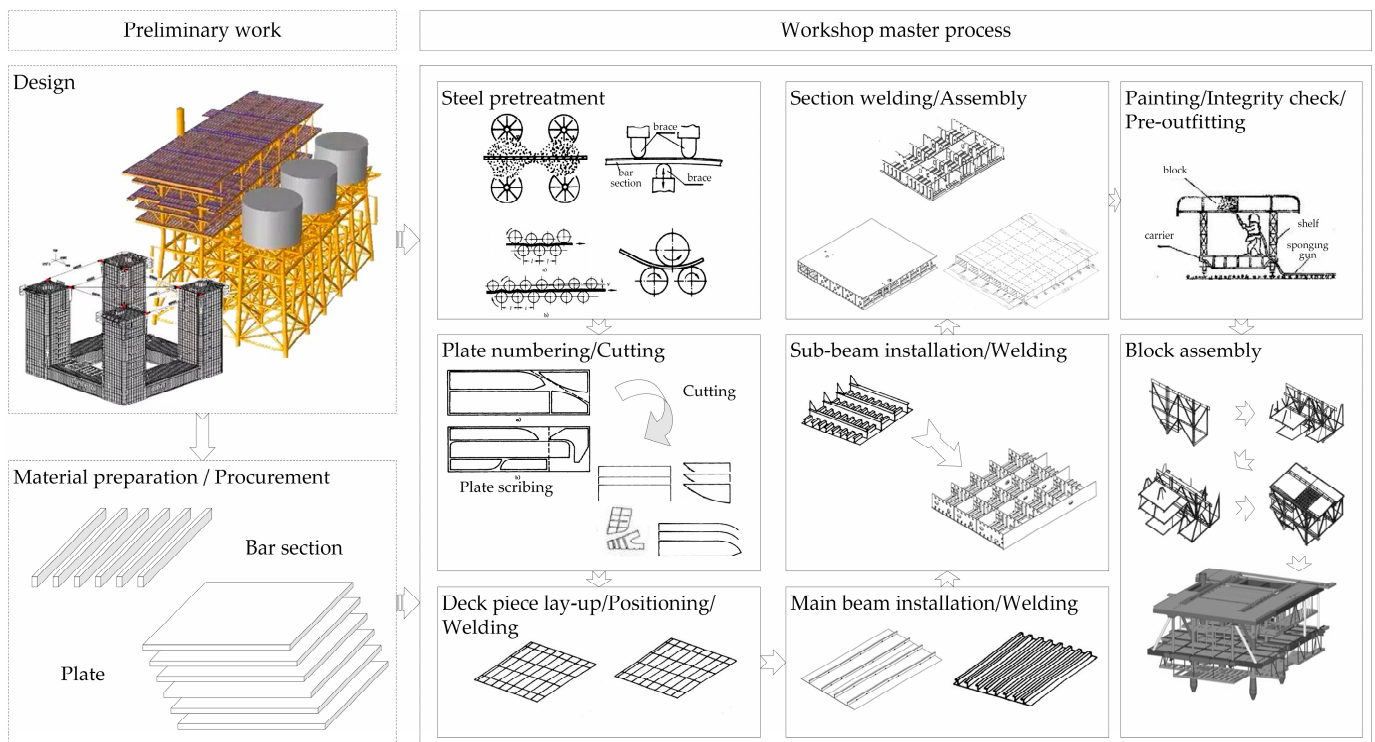


Figure 1. Ocean engineering platform manufacturing process.

Compared with the DT system architecture utilized in the conventional manufacturing industry, this paper considers the data transformation process among multiple systems and the compatibility of multiple types of equipment due to the lack of production line flexibility in large-scale construction projects. Independent design for various degrees of digitization in the production process. For example, in the pre-processing, cutting, and part of the welding stages of full automation, the DT system is arranged to participate directly in the production process through the inter-system output instructions to organize the production, and the production status is presented with direct access to the system. In

the manual phase, the DT system is responsible for issuing tasks, and each station receives the request to complete the work via a variety of terminals. The implementation of DT has had a revolutionary impact on the optimization of process workflows in OE assembly and construction. The core of DT is full-factor data-driven production process prediction and optimization to guide the operation execution and collect actual data for operation analysis to fulfill the timely adjustment in the execution process, not just a simple visualization display. The work in this paper is geared towards developing companies with large-scale engineering, where all-around compatibility and utilization of existing resources are the crucial roles of the proposed DT architecture.

Job planning, as the lifeblood of production control, determines production efficiency and capacity. The primary emphasis of establishing a DT workshop is to implement a twin data-driven planning methodology. In this paper, we propose an algorithmic model for job planning based on Graph Attention Networks (GATs), an improved version of GNN. Different from the traditional Priority Dispatch Rules (PDRs) and heuristic algorithms, a GAT extracts features from the disjunctive graph model in the scheduling problem, and outputs constraint-compliant “operation–machine” combinations from the set of operations and the set of machines based on the Markov Decision Process (MDP) to form the decision sequence. Among them, the weight parameters in a GAT are obtained by deep reinforcement learning (DRL) performing small-sample training. This is one of the latest mainstream algorithms in recent years, and most studies [10–16] have developed solutions for standard arithmetic cases, on the basis of which this paper completes customized development for the characteristics of the OE workshop and case validation based on actual production data. Its high performance, excellent computational effect, and ability to adapt to maintain rapidity computation under a large amount of data input from the DT system are the main reasons why it is selected instead of PDRs and heuristic algorithms in this paper, as compared with the traditional algorithms. Moreover, the GAT model enables effective interfacing between knowledge graphs and expert systems in the future, making it a superior optional method.

The DT workshop planning model is trialed under the scenario in the standardized test datasets and the OE manufacturing workshop. The scenario, as the critical node in the production chain of an OE smart factory, is capable of covering the production elements of the whole factory. The case validation is centered around ensuring the high availability of both the DT system and its associated model. There are prerequisites for this industry in the scenario, and the case validation can give a part of the reference of executable solutions to the peers.

In general, this paper makes the following contributions:

- We design the DT workshop system architecture in detail, form a targeted five-dimensional model for OE intelligent manufacturing workshops, and create an integrated discrete operation workshop management model with “virtual-real co-drive”.
- We present a novel model for the DT workshop planning that utilizes GNN. Our method realizes generating the workshop’s master plan that exhibits good generalization capabilities. This is achieved through node embedding, graph attention network structure building, and a deep reinforcement learning-based algorithm for model training.
- Drawing upon the proposed GNN model, our research expands upon the discourse surrounding five distinct decision support techniques aimed at resolving production-related issues, with the ultimate objective of augmenting the efficacy of the planning model.
- The feasibility of the model is verified, and a demand-oriented DT system is created for workshop job execution. We optimize the limitations of the current production mode by focusing on the critical development of DT technology and the efficient use of twin data—not just the simple interaction of visual simulation models.

The remainder of this paper is structured as follows: We detail previous theoretical literature reviews on DTs and GNNs in Section 2. The architecture of the DT Workshop

System is briefly defined in Section 3. Section 4 proposes the graph attention network structure to solve the modeling problem of the DT workshop planning and extends the discussion of five decision support methods in Section 5. Section 6 summarizes the case validation results and discussion. Finally, conclusions and future work are presented in Section 7.

## 2. Literature Reviews

### 2.1. Digital Twin

The concept of Digital Twin (DT) was first proposed by Michael Grieves [17] when discussing the digital representation of physical objects or processes throughout the product lifecycle management. The theory originated in the aerospace field, where NASA introduced the “flying twin” for flight training and simulating spacecraft status [18]. Subsequent research further clarified the implementation of multidimensional and multi-attribute simulation models for monitoring and prediction functions by integrating multiple systems and high-precision sensors [19]. As simulation and Model-based Systems Engineering (MBSE) continue to develop, the DT field has expanded into industrial production and become a widely researched hotspot in recent years [20]. Establishing a digital twin workshop is the most fundamental and important research direction for manufacturing. F. Tao and M. Zhang [21] proposed a completely new workshop operation model. They summarized five essential elements of DT workshop composition, focusing on the perception access of heterogeneous multi-source data and the fusion ability of “human–machine–thing–environment”, i.e., all elements. Guo et al. [22] proposed a modular approach to construct a digital twin of the factory, considering frequent changes in the factory design stage. Designers can rapidly evaluate different designs and positively identify design defects. In the production stage, DT can improve the predictability and controllability of the production process. Zhang et al. [23] designed a control framework based on DT to address the problem of plan-based decision-making and control mechanisms that are inapplicable to uncertain production states by introducing optimal state control theory into the virtual layer and ensuring the production system is in the best state through dynamic matching mechanisms and hierarchical goal-cascading methods.

Workshop-oriented digital twin modeling is a systematic implementation of current researchers’ achievements in the DT key technologies and has formed a set of methodologies. In 2013, Lee [24] designed a coupled model for monitoring machine conditions, performing predictive analysis based on industrial big data, and improving management transparency. As a physical product twin for design and manufacturing, B. Schleich et al. [25] provided a thorough reference model based on the surface model form idea. They described the conception, representation, implementation, and application of the model throughout the product’s lifetime. Sun et al. [26] used DT technology to anticipate assembly process dependability and optimize assembly accuracy by using twin assembly data, the assembly and debugging full-element model, and iteration between virtual and actual assembly. Current research has concentrated on data collecting, heterogeneous data conversion, and model data matching in the data processing layer. There is still a large study area for data application and service of DT. This paper spotlights the data application modeling of intelligent services after the completion of data processing in the DT system.

### 2.2. Graph Neural Network

With the advancement of deep learning in recent years, the Graph Neural Network (GNN), a novel deep learning model, has drawn more and more attention. The GNN aims to map nodes and edges on the graph structure to low-dimensional vectors, i.e., output low-dimensional vectors for both the graph and nodes through artificial neural networks [27]. Unlike traditional deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), GNNs can handle the graph structure’s interaction information of nodes and edges. A GNN usually has a multi-layer structure, and each layer can calculate different features, thus being adaptable to different application scenar-

ios [28]. The GNN was first proposed by Gori et al. [29] and was extended using RNN. In subsequent studies, two branches were developed: spectral domain and spatial domain. To solve the irregularity of spatial neighborhoods, J. Bruna et al. [30] introduced the Spectral Graph Theory to decompose the Laplacian matrix of the graph, obtained eigenvalues and eigenvectors, and then performed convolution operation. This is the starting point of the spectral-based GNN.

To deal with the high complexity, M. Defferrard et al. [31] proposed the ChebyNet, which defined the convolution kernel as a polynomial and used Chebyshev expansions to approximate the convolution kernel for efficiency improvement. Based on this, T.N. Kipf and M. Welling [32] further simplified the model by using a first-order approximation convolution kernel, forming the Graph Convolutional Network (GCN). In the spatial-based GNN, which is a recent research hotspot, there are mainly the Gate Graph Neural Network (GGNN) [33] that replaces the recursive neural network node update method, the Graph Attention Network (GAT) [34] that defines graph convolution with attention mechanism, the GraphSAGE [35] which inductive learning, and the message passing neural network (MPNN) [36], which unifies the message passing mode. Considering the outstanding cross-domain properties of graph neural networks and the fact that the current server computing power has reached the basic requirements based on the generalization ability of GNNs, introducing GNNs into the data application process of the Digital Twin workshops can effectively achieve decision support.

### 3. Architecture of DT Workshop System

An extensive analysis of physical entities, operational behavior, information transfer, and interaction rules is necessary for the development of the Digital Twin (DT) system for the Ocean Engineering (OE) manufacturing execution process, which is a complex system engineering involving many disciplines, resources, and technologies. The current OE production process generally adopts an integrated construction model, which collects huge amounts of materials, equipment, personnel, storage, and other elements. On the basis of satisfying the production line simulation, designing a DT workshop architecture covering the “Humans–Machines–Materials–Rules–Environments” is the first step of production data application modeling.

#### 3.1. Intelligent Workshop Integrated Manufacturing Execution Mode

As the basic unit of offshore platform construction, the assembly and welding workshop is based on the principle of grouping technology, with the intermediate product as the guide, and production is organized by area, with the platform deck as the base and outfitting as the core, and the “structure–outfitting–painting” operations are separated in space and ordered in time to realize the integration of design, production, and management [37]. The integrated operation mode is balanced and continuous, avoiding the traditional mode of independent and uncoordinated production links, reducing the simultaneous cross-work of multi-disciplinary construction, and simplifying the work packages interface and the interference between different types of work.

The integration manufacturing execution mode has extremely strict requirements on the production data chain. As shown in Figure 2, the five major elements surrounding the operation of the workshop are associated with dozens of forms. They are intertwined with design, planning, materials, equipment, storage, quality control, and other professions. In the intelligent workshop, in addition to cranes, welding torches, material pallets, and other manually operated equipment, there are automatic plate-cutting machines, gantry-type welding robots, CNC pipe-cutting machines, automatic plate winders, material conveyors, uncrewed transportation vehicles, and other automated uncrewed equipment. In general, more than 60% of the equipment is networked. In the construction process, the operators adopt the workshop-level plan as the main line. Each workstation completes the processes of material collection, plate assembly, welding, and inspection according to the work order and provides progress and quality information in real time. Structural and quality experts

simultaneously verify the weld quality and global strength properties of completed assemblies across multiple stations, in hours, to form production master data under a time series. With the addition of digital devices, data-aware collection devices, and simulation models in the smart shop, the conversion and integration of multidimensional heterogeneous data demand to be realized under a unified and customized architecture. A standardized production database relying on the model proposed in Section 4 enables optimizing production quantities, balancing operational tasks, improving equipment utilization, reducing energy consumption, etc.

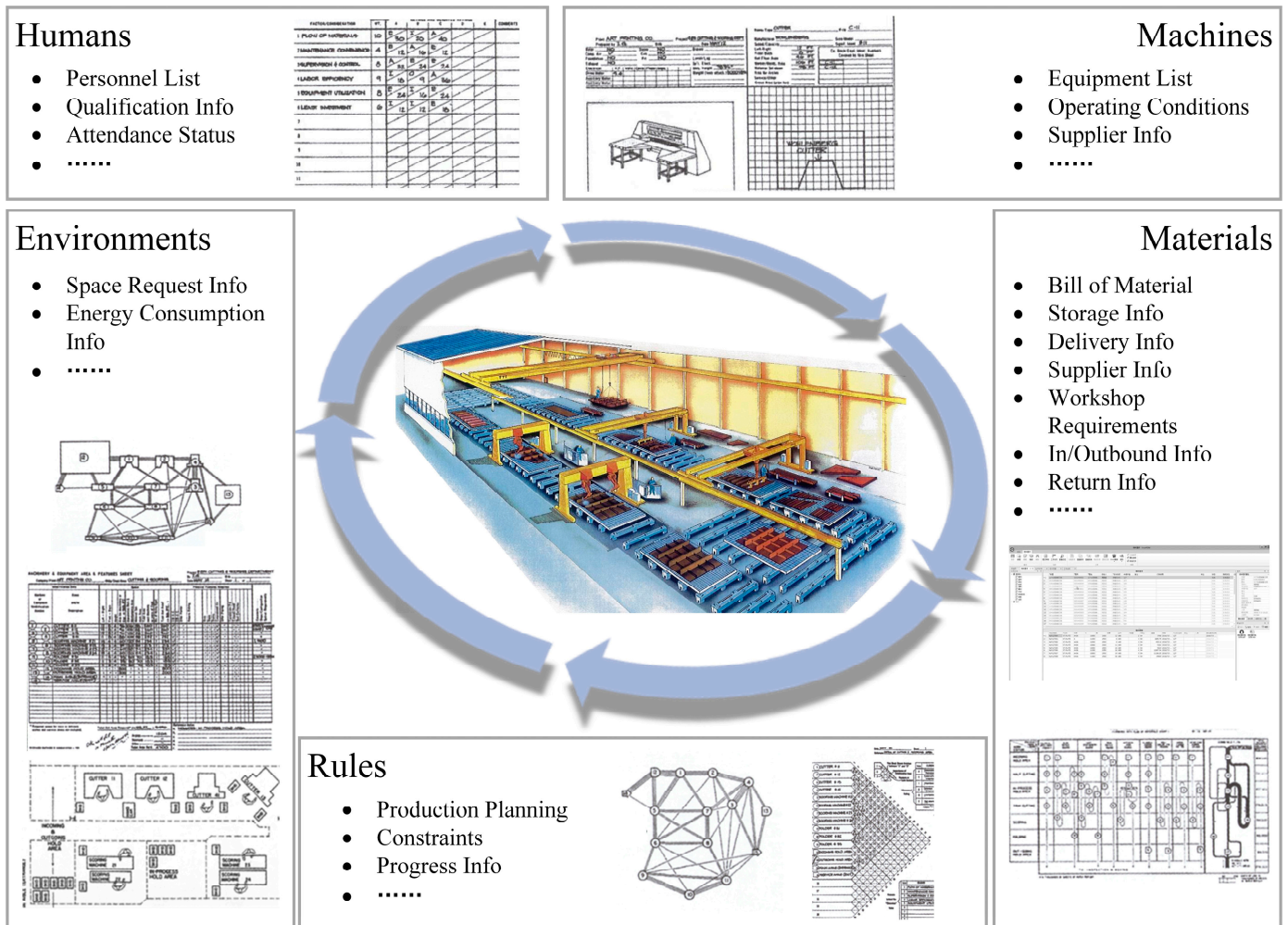


Figure 2. Integrated manufacturing execution mode for “Humans–Machines–Materials–Rules–Environments”.

### 3.2. Design of DT Workshop Architecture Based on Five-Dimensional Model

Based on the five-dimensional model concept proposed in the literature [5], this paper designs the ocean engineering manufacturing DT workshop architecture as shown in Figure 3. The architecture unfolds from the 5D model, including Physical Entities (PE), Virtual Objects (VO), DT Data (DD), Connections (CN), and Application Services (AS), whose vector format can be expressed as Equation (1).

$$M_{DT\_workshop} = (PE, VO, DD, CN, AS)^T \tag{1}$$

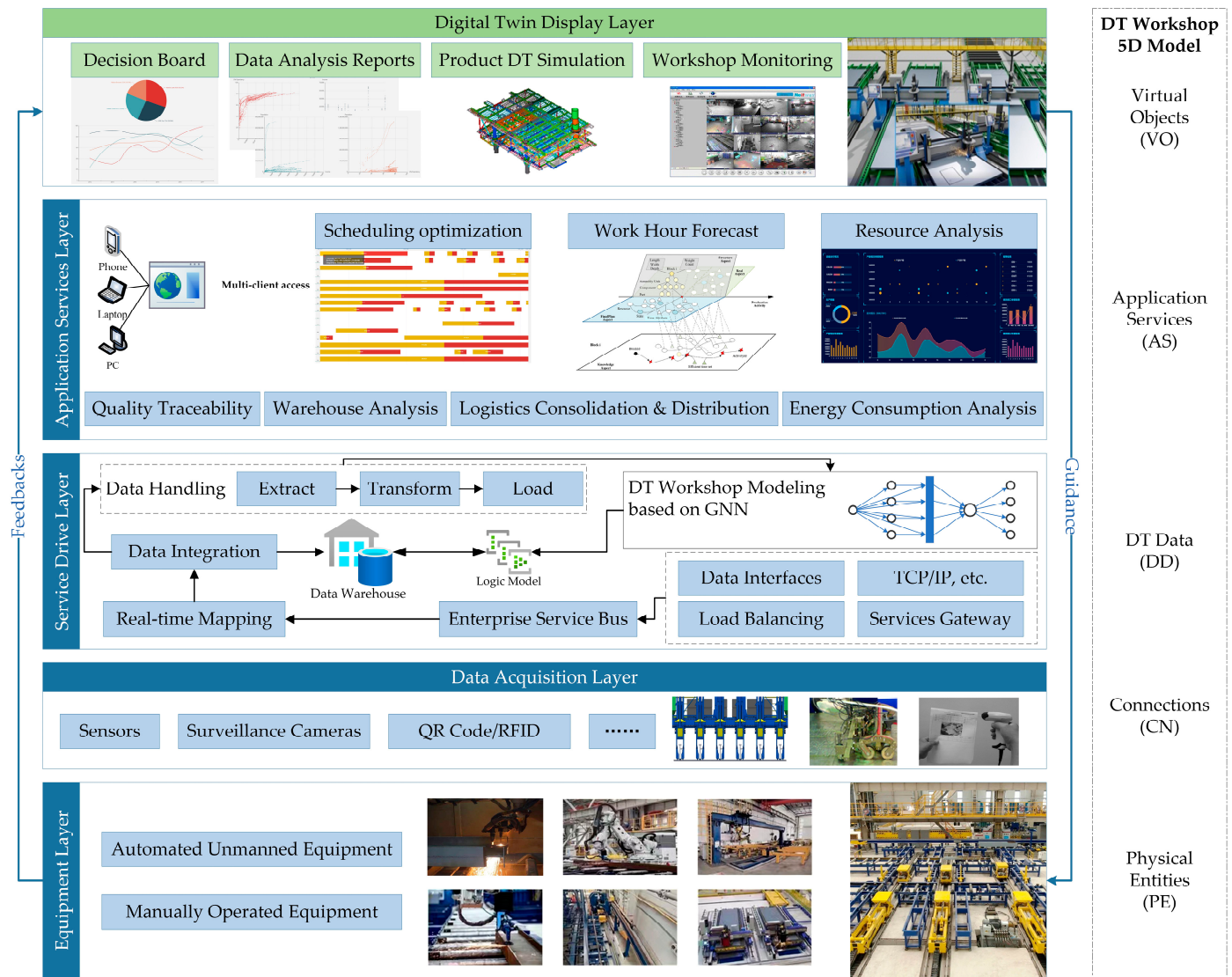


Figure 3. Ocean engineering manufacturing DT workshop architecture.

In the implementation of the digital twin workshop, five dimensions correspond to the Equipment Layer, the Data Acquisition Layer, the Service Drive Layer, the Application Service Layer, and the Digital Twin Display Layer, with each layer corresponding to handling various complex tasks. The architecture from the bottom up specifically includes:

- Equipment Layer

This is the collection of physical entities. By establishing data tables for all production elements in the workshop, distinguishing fully automated uncrewed equipment from manually operated equipment by categories such as equipment, supplies, and buffers, forming information labels for corresponding entities, establishing fields for equipment status, operators, supplier information, consumables spare capacity, etc. These allow real-time mapping with subsequent operational processes to complete DT data input to equipment operation as well as actual measurement data feedback to DT console. The vectorized representation of *PE* is shown in Equations (2)–(5).

$$PE = (E, M, A, \dots)^T \tag{2}$$

$$E = (No., EquipmentName, Status, Operator, Supplier, Amount, \dots)^T \tag{3}$$

$$\mathbf{M} = (\text{No.}, \text{MaterialName}, \text{Amount}, \text{Supplier}, \dots)^T \quad (4)$$

$$\mathbf{A} = (\text{No.}, \text{AreaName}, \text{Location}, \text{Size}, \dots)^T \quad (5)$$

where  $E$  represents the equipment,  $M$  is the material, and  $A$  is the area, each containing their respective detailed properties.

- Data Acquisition Layer

This layer collects or uploads shop operation execution data through sensors, surveillance cameras, QR codes/RFID, etc., and develops heterogeneous data interfaces to link the equipment layer with the service driver layer. The scope of data collection includes, but is not limited to, equipment operation data, transportation process awareness, material information tracking, equipment status feedback, environmental monitoring, etc. The data interface development workload is enormous and requires one-to-one customization to be completed based on multiple models, protocols, and standards of the equipment to form a data chain for real-time transmission. An enormous quantity of basic work is a necessary component of constructing the DT workshop. The integrity and standardization of the collected data determine the operational efficiency and calculation accuracy of the service driver layer. The vectorized representation of the data connection process is exemplified by Equation (6).

As the central layer of the digital twin system, the service drive layer is responsible for data reception, conversion, and application. The data is accessed by the equipment layer feedback and digital twin emulation via interfaces to the Enterprise Services Bus and Data Warehouse. The data integration component standardizes the multi-source heterogeneous content in the database and updates the master database based on the information provided by the real-time mapping. The standardized production data is available to be input into the GNN-based DT workshop model to obtain the manufacturing execution plan with workshop schedule and optimal allocation of multiple resources. All data are uniformly stored in the data warehouse within this layer, and the logical model serves as the control core to integrate the DT model with the actual data, providing a unified description and encapsulation of geometric properties, physical properties, operational behaviors, and simulation rules for each production-related physical entity. When instantiated virtual objects receive iterative data updates, their information updates can be quickly completed in the logical model. The service drive layer of DT data support can be expressed as follows:

$$\mathbf{DD} = (\mathbf{D}_{PE}, \mathbf{D}_{VO}, \mathbf{D}_{AS}, \mathbf{D}_K, \mathbf{D}_\Delta)^T \quad (6)$$

where  $D_{PE}$  represents the physical workshop's actual measurement data,  $D_{VO}$  is the virtual object simulation data,  $D_{AS}$  means the application service output data,  $D_K$  represents the process knowledge base data, and  $D_\Delta$  represents the data fusion-derived data.

- Application Service Layer

This layer supports the operation and maintenance of digital twin system applications. Based on the twin data and service drive layer, it is oriented to the workshop planning, work time, progress, resources, quality, storage, and energy control needs, extracting critical data in the logic model and supporting the realization of decision support functions. Intelligent workshop managers and workers can access application services through multiple clients. Taking the specific business of the welding workshop as an example, the application service provides block assembly and welding scheduling optimization, welding work time forecasting, steel resource demand analysis, sub-section quality traceability, storage availability analysis, logistics consolidation and distribution, energy consumption analysis, and other intelligent algorithm solutions. In the future, functions such as the welding



process knowledge base and high-precision control of workshop equipment can be further expanded. The application services can be expressed as follows:

$$AS = (S_{Scheduling}, S_{Workhours}, S_{Resources}, \dots)^T \quad (7)$$

where  $S_*$  means the data set corresponding to each service.

- Digital Twin Display Layer

Visualization of system functions is the significant feature of this layer, which is the only layer that the users of the digital twin system visualize. The application module proceeds from the simulation model of the virtual workshop and enters the business module by selecting the workshop components. The business module displays the functional indicators of the workshop and the product information based on the data analysis results of the application service layer, specifically including the decision board, data analysis reports, the product DT simulation, workshop monitoring, etc. The functionalities of the digital twin display layer are realized entirely based on the integrity data of the virtual objects, whose vectorized form is:

$$VO = (G_{VO}, P_{VO}, B_{VO}, R_{VO})^T \quad (8)$$

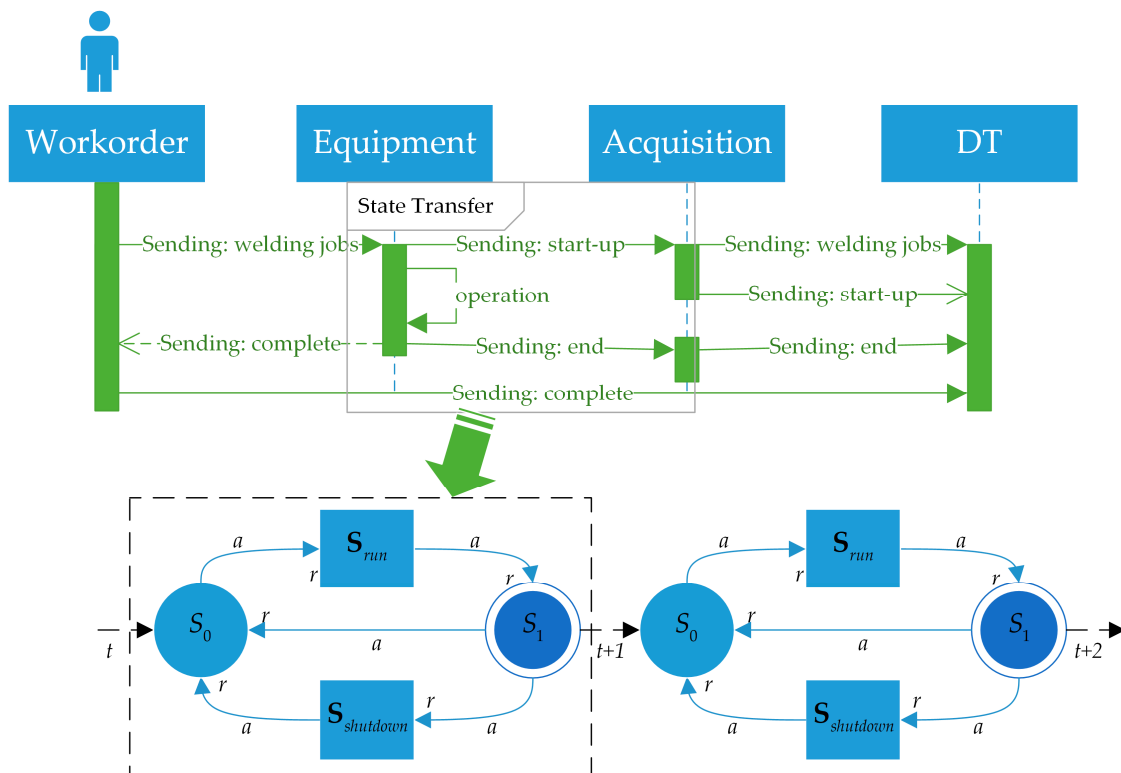
where  $G_{VO}$  corresponds to the geometric model,  $P_{VO}$  is the set of physical attributes,  $B_{VO}$  means the quantitative representation of the operational behavior, and  $R_{VO}$  represents the rule model.

In a nutshell, the demand-oriented digital twin workshop architecture is based on the theory of the five-dimensional model, which is theoretically feasible and practically usable and sufficiently satisfies all elements of workshop production control. It realizes the comprehensive digital in the whole cycle of assembly and welding operation. With the logical model as the core and all production elements interlinked, the five-layer architecture materializes the iterative cycle of “DT data simulation guidance—actual measurement data optimization” based on the data chain.

### 3.3. Transfer Rules for Manufacturing Execution Behaviors under Finite State Mechanism

In contrast to a traditional workshop, the DT workshop enables the synchronization of the presentation of dynamic executions on virtual objects. The various types of platform assembly and welding operation behaviors require uniform quantitative transfer rules to complete the abstraction of production actions. In accordance with prior research [38,39] on Finite State Machines (FSM), which can guarantee the integrity of operational behaviors and state transfer in the workshop, the behaviors of workers and machines during assembly and welding operations could be categorized into a finite set of states.

In the instance of welding operations, there are standby, operational, and shutdown states for a single welding robot. The standby state is usually at the beginning or end of the task, as in Figure 4,  $S_0$  and  $S_1$ , which are transient states, while the continuous processes of operation and shutdown states are expressed as  $S_{run}$  and  $S_{shutdown}$ . The transition of the state depends on the trigger event  $a$ , which returns the event response information  $r$  after the completion of the state transition. Specifically, when the workstation receives a work order, it first confirms that the welder is in the  $S_0$  available state. The DT system issues a welding start command to the instantiated object according to the task trigger  $a$ , enters the  $S_{run}$  welding operation state, and confirms the receipt of  $r$ . After the task is completed, the equipment enters the  $S_1$  completion state. If there is equipment downtime, insufficient material and other termination conditions, enter  $S_{shutdown}$  until the repair is complete to restore the equipment to  $S_0$ .



**Figure 4.** Jobs information and status transfer between DT and equipment.

The states and commands of the equipment at each workstation in the workshop are integrated into the database in the form of a time series, enabling simultaneous simulation at the data twin display layer and direct issuance of commands to the machine at the equipment layer. Behavioral information is transmitted between the device layer and the digital twin model in order of task priority, and the “First-In-First-Out” principle is adopted when the same level is available. When conflicting data occurs, operation instructions are suspended on the equipment side, and simulation data are backed up and pushed to the system operator for confirmation. Transmission rules are the standard for interaction between the equipment layer and the digital twin model. Simplistic rules decrease the frequency of conflicts and provide robustness for the system’s daily operation, but standardized operation training for system participants is required to reduce the risk of interaction errors.

### 3.4. Real-Time Mapping of Workshop Processes

Multidimensional data fusion is the cornerstone of ensuring the reliability of virtual objects. The authenticity of the production process simulation relies on the comprehensive description of the actual environment. The multidimensional dynamic data is associated with physical entity properties via real-time mapping to support the operation of the service drive layer. Real-time mapping subjects are predominantly workpieces, equipment, workers, production environment, and production knowledge. At the physical entity side, multiple types of unstructured data (e.g., contracts, drawings, notifications, etc.), multi-device collection data, multi-resource structured forms, and existing systems (e.g., MES/ERP/WMS, etc.) are blended in each production stage, which standardization data based on association mapping algorithms is critical.

The mapping logic utilizes tagging. The workpiece/intermediate product forms a set of virtual tags, including product size, processing information, work order information, processing worker information, and other collections based on their self-encoding. During the work execution, operation behavior information and state transfer data are recorded synchronously to provide a process traceability interface. According to the rules

of operation behavior transfer, the real-time mapping of equipment is stored with each piece of equipment corresponding to an independent database, and the digital twin system enables the virtual processing process to be launched according to the data so as to monitor the production quality. Workshop operations-oriented environment-aware data take the form of setting sensitive parameter thresholds in the mapping process due to the excessive amount of data and only convert and store records of environmental change periods. The knowledge data are scattered in the existing systems of the enterprise, where interfaces are established with the digital twin system for labeling transformation and storage by process category.

The real-time mapping process consists of three phases. Firstly, the initialization is completed, the corresponding interface is launched, the form information is verified, and the information on virtual workshop equipment, environment, and workers is adjusted to be consistent with the physical entity. Subsequently, the real-time mapping operates, and the background activates data mapping transmission monitoring to promptly adjust and make up the data for the interfaces with package loss. Finally, completing the data handling and the massive production data are used as the input of the service drive layer and application service layer to support production decision-making, scheduling optimization, abnormality warning, etc., which are specifically researched in Sections 4 and 5. The schematic diagram of the real-time mapping of workshop operations is shown in Figure 5.

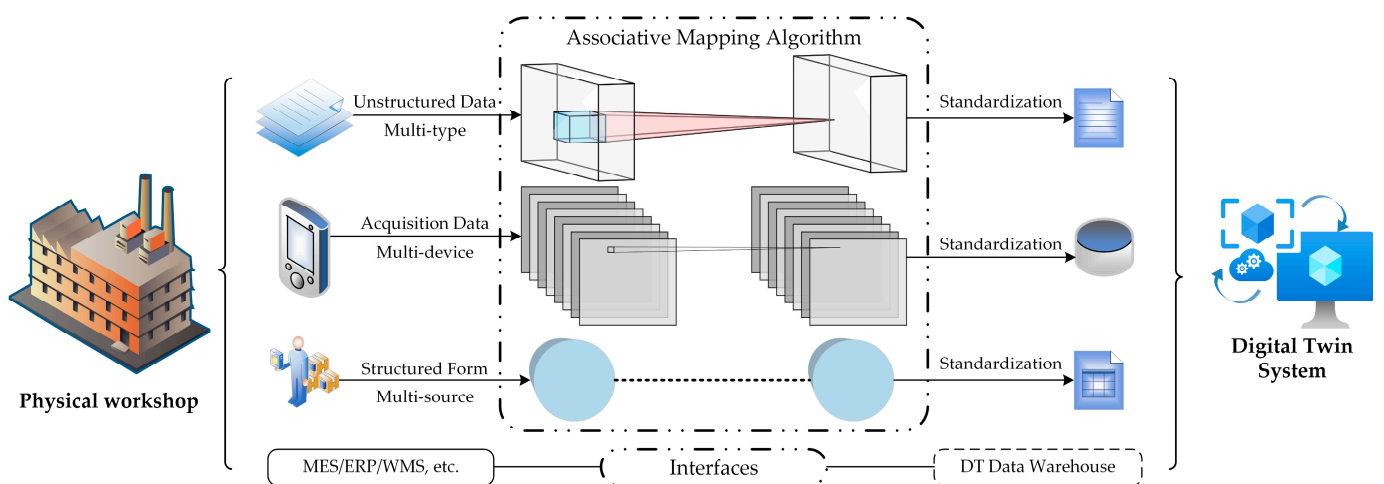


Figure 5. Real-time mapping of the physical workshop to DT.

The manufacturing execution behavior transfer rules and the workshop process real-time mapping are the fundamental operations in the DT system development, which are indispensable links to support the simulation operation and provide feasibility guarantees for the introduction of subsequent mathematical models and the application of decision support methods.

#### 4. DT Workshop Planning Modeling Based on GNN

This section proposes a DT workshop Planning model based on a Graph Neural Network, which aims to address the problem of twin data-based master production schedule generation and dynamic optimization. Planning is the pivotal segment of offshore platforms from design to fabrication, with the primary task being calculating the critical path of the assembly and welding process by combining the real-time multidimensional attribute data provided by the DT system, which obtains the full job duration and forms a schedule Gantt chart. Considering the high performance of GNNs in processing graph structure data, it is capable of solving the NP-hard problem effectively in the face of flexible process route mesh diagrams with plan, equipment, and resource integration.

#### 4.1. Traditional Mathematical Model

Traditional planning methods perform mathematical modeling based on business requirements, integrating optimization objectives and constraints that are solved using, for example, heuristic algorithms. Considering that GNN models are trained to adapt the network parameters to the strategy of the target, the purpose of traditional mathematical modeling is to provide a planning reference benchmark to the neural network. The operational task of the Ocean Engineering manufacturing workshop is mainly the assembly and welding of blocks. The operational processes are material pre-treatment, cutting, coiling, plate assembly, welding, and others, which correspond to multiple networked pieces of equipment with different functions but capable of participating in multiple processes. Considering that the workshop has the characteristics of process flexibility and equipment flexibility, as well as the requirements for efficiency and energy cost, the planning of assembly and welding operations should be defined as an optimization problem for the flexible job shop.

Specifically, shop floor operations can be summarized as a process plan of size  $N \times M$ , where  $N$  is the number of workpieces and  $M$  is the number of machines. There are multiple process routes for each part being machined, and the critical path needs to be solved to meet the makespan optimal objective, that is, to minimize the maximum completion time. The mathematical model containing the objective function is shown in Equation (9).

$$F = \min\{c_{max}\} \tag{9}$$

where  $c_{max}$  represents the makespan. It is the time difference between the start and completion of all processes, with each process's completion time being non-negative, illustrated in Equations (10) and (11).

$$c_{max} = \max_{1 \leq i \leq N} \{c_{i,j=P_{il},k}\} \quad k \in [1, M] \tag{10}$$

$$c_{i,0} = 0, \quad c_{i,j} > 0 \quad \forall i, j \tag{11}$$

the constraints include:

$$\sum_l X_{il} = 1 \quad \forall i \in [1, N] \tag{12}$$

$$\sum_{k=1}^M Z_{ijkl} = 1 \quad \forall i \in [1, N], \forall j \in [1, P_{il}], \forall l \in [1, G_i] \tag{13}$$

$$(X_{il} \times Z_{i(j')lk})s_{i(j')lk} \geq (X_{il} \times Z_{ijkl})(s_{ijkl} + t_{ijkl}) \tag{14}$$

$$(X_{il} \times Z_{i(j+1)l(k')})s_{i(j+1)l(k')} \geq (X_{il} \times Z_{ijkl})(s_{ijkl} + t_{ijkl}) \tag{15}$$

where Equation (12) represents the multi-flexible process routes of a workpiece that is uniquely selectable, Equation (13) restricts a process to allow only one selectable device to be selected, Equation (14) indicates a machine constraint that only one process is performed at a time and Equation (15) acts as a process constraint, expressing that multiple process operations cannot be completed on the same workpiece at the same time. The definitions and descriptions of the symbols are listed in Table 1.

Prior to solving the model, the following assumptions are required to ensure that:

1. One machine operates only one process of one workpiece at the same time;
2. All the equipment is available at the starting moment;
3. The locations of the segments in the workshop are close to each other without considering the transfer time between different processes;
4. The equipment is available at all times, regardless of breakdowns;
5. The process has no other influence to interrupt the processing.

**Table 1.** The definitions and descriptions of the symbols in the mathematical models.

Symbols	Definition	Description
$N$	Total number of workpieces	$\{1, 2, \dots, i, \dots, N\}$
$i$	Workpiece number	$i \in [1, N]$
$j$	Process number	$j \in [1, P_{il}]$
$k$	Equipment number	$k \in [1, M]$
$M$	Total number of equipment	$\{m_1, m_2, \dots, m_k, \dots, m_M\}$
$G_i$	Number of all process routes for $i$	$G_i = \{O, C, M\}$
$l$	Flexible process route number	$l \in [1, G_i]$
$P_{il}$	Total number of processes planning for the process plan $l$ of $i$	$P_{il} \in G_i$
$o_{ijl}$	The operation $j$ of the process plan $l$ for $i$	$o_{ijl} \in P_{il}$
$s_{ijkl}$	Workpiece manufacturing start time	Time
$c_i$	Workpiece delivery time	Time
$c'_i$	Workpiece completion time	Time
$t_{ijkl}$	Workpiece processing time	$t_{ijkl} = [s_{ijkl}, c'_i]$
$X_{il}$	Process route selection decision variable	$X_{il} = \begin{cases} 1 & \text{if } i \text{ select process plan } l \\ 0 & \text{else} \end{cases}$
$Z_{ijkl}$	Equipment selection decision variable	$Z_{ijkl} = \begin{cases} 1 & \text{if } o_{ijl} \text{ select equipment } k \\ 0 & \text{else} \end{cases}$

#### 4.2. Graph Neural Network Structure

Recently, there has been a significant surge in scholarly inquiry regarding the utilization of Graph Neural Networks (GNNs) in conjunction with Deep Reinforcement Learning (DRL) for the purpose of resolving workshop job scheduling problems [10–16,40]. This approach is regarded as one of the most efficient computational methods for optimizing scheduling. In contrast to traditional heuristic rules and metaheuristic algorithms, which require computation under the objective function and constraints of the previous section, the priority scheduling rules (PDRs)-based sequential decision problem is turned into a Markov decision process (MDP) by training GNN parameters through DRL. This approach incorporates the optimization objectives and constraints into the reward mechanism, thereby achieving the generalization of the scheduler. This methodology can effectively decrease the computational burden of each scheduling computation while maintaining solution precision and enabling the repeated utilization of GNNs with stability subsequent to a solitary training iteration. Consequently, this methodology has the potential to significantly mitigate the substantial burden on the computing capacity of the planners’ personal computers during the plant’s planning phase and delegate the computationally intensive tasks to the server for execution. Furthermore, a server-side interface might be devised to retrieve real-time progress and work time information from jobs gathered in the DT system and incorporate it into the training sets. These can serve as an effective foundation for updating status or action values in subsequent rounds of training.

Combining the DT workshop foundation and job planning requirements, this paper proposes a planning framework based on a graph attention network (GAT) and DRL, as shown in Figure 6.

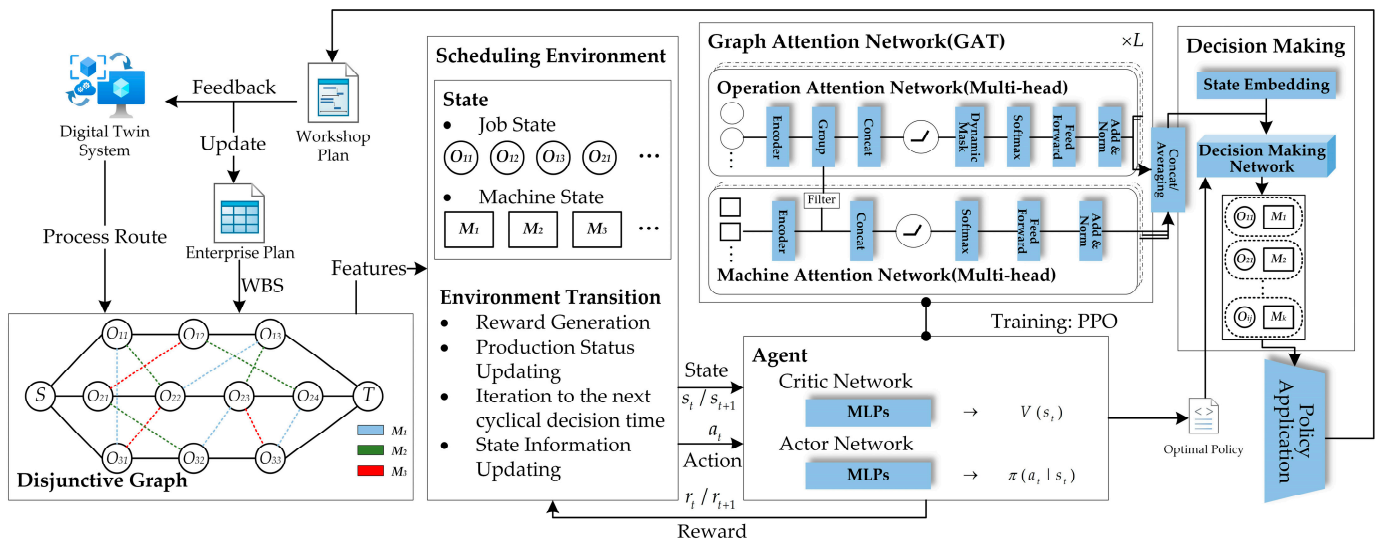


Figure 6. The planning framework based on the GAT and DRL.

#### 4.2.1. Disjunctive Graph

The use of a disjunctive graph is an intuitive method for describing workshop planning, which expresses the operations and machine attributes in a flexible job shop using graph-structured data. For a workshop scheduling environment with  $m$  machines and  $n$  workpieces, the disjunctive graph is represented as  $\mathcal{G} = (\mathcal{O}, \mathcal{E}, \mathcal{D})$ , and where  $\mathcal{G}$  represents the directed graph, and  $\mathcal{O} = \{O_{ij} | \forall i, j\} \cup \{S, T\}$  denotes the set of nodes corresponding to the operation time and the start/end time points of each job.  $\mathcal{E}$  is a subset of  $n$  edges, indicated by solid lines representing the processing path from  $S$  to  $T$  for all operations.  $\mathcal{D}$  is composed of multiple undirected disjunctive arcs, representing that a certain process can be completed on a certain machine. Under the flexible feature, an operation can be completed by any available machine. The plan-solving process involves determining a disjunction arc at each node, fixing the direction, and forming the workpiece processing sequence for a certain machine. After the scheduling calculation is completed, the disjunctive graph is converted into a directed acyclic graph, and the scheduling target is to select the least total operation time among multiple directed acyclic graphs.

#### 4.2.2. Markov Decision Process

The MDP is a method that transforms the conventional workshop scheduling process into a five-tuple  $\langle \mathcal{S}, \mathcal{A}, P(s'|s, a), r, \gamma \rangle$  to describe it. The five elements within the tuple correspond to the state, action, state transfer function, reward, and discount factor of the scheduling calculation, which are important parameters for the realization of the DRL-based scheduling decision process. The decision-making process is performed by discerning whether there is a combination of process  $o_{ij}$  that needs to be scheduled and idle machine  $m_k$  at each time point  $t$  within the planning time  $T_p$ . The  $(o_{ij}, m_k)$  is transformed into  $\mathcal{A}(t)$  input to the agent and obtains the reward  $r(t)$  from the environment at the current  $\mathcal{S}(t)$ , which based on the makespan value. Finally, a complete plan scheme with the total operation time is obtained after several iterations. The detailed MDP definition is as follows.

(1) *State*. State definition is the pivotal aspect of solving scheduling problems by graph representation learning, and it is also the crucial step for feature extraction in a large-scale schedule optimization problem. The scheduling process state encompasses three types: completed operation, operation in progress and operation to be scheduled. Let the operation, device and action correspond to  $\mathcal{O}(t)$ ,  $\mathcal{M}(t)$  and  $\mathcal{A}(t)$ , respectively, and determine  $\mathcal{S}(t)$  according to the decision step  $t$  together with the scheduling environment. By extracting the feature values of each element, the state  $\mathcal{S}(t)$  forms a set of multidimensional feature

vectors, including but not limited to operation features  $\mathcal{R}_{o_{ij}} \in \mathbb{R}^8, o_{ij} \in \mathcal{O}(t)$ , machine features  $\mathcal{R}_{m_k} \in \mathbb{R}^5, m_k \in \mathcal{M}(t)$ , and action features  $\mathcal{R}_{(o_{ij}, m_k)} \in \mathbb{R}, (o_{ij}, m_k) \in \mathcal{A}(t)$ .

To be specific, the operation features consist of eight-dimensional parameters, which are the operation status label (Whether to complete scheduling, 0—no, 1—yes), process time  $p_{ij}^k$  (the value of an unscheduled operation is the average of the completion times of that operation on all available machines, i.e.,  $\overline{p_{ij}^k} = \sum_{m_k \in \mathcal{M}_{ij}(t)} p_{ij}^k / |\mathcal{M}_{ij}(t)|$ ), start time  $st_{ij}(t)$  (the start time is  $st_{ij}(t) = st_{il} + p_{ilk}$  when the preorder operation has been scheduled, otherwise it is estimated recursively according to  $st_{ij}(t) = st_{il}(t) + \overline{p_{ilk}}$ ), remaining process time (Unscheduled operation is 0), completion time, latest deadline, operation waiting time and the number of remaining operations  $J_i$ . The 5-dimensional machine features contain machine status (Available or not, 0—no, 1—yes), machine operating time, number of operations to be processed, waiting time and remaining processing time. The action feature corresponds to the operation-machine combination feature, taking the value of the process time  $p_{ij}^k$  for selecting machine  $k$  for the  $o_{ij}$ .  $\mathcal{S}(t)$  integrates the static and dynamic attributes of the scheduling process, which can be updated in a timely manner by the data acquired by the DT system, enabling a genuine realization of adaptive scheduling in the actual business of the workshop.

(2) *Action.* The action space  $\mathcal{A}(t)$ , which is in a dynamic change process at each decision time, contains all feasible combinations of  $(o_{ij}, m_k)$ . Each action execution process first selects an eligible action  $o_{ij}$  for which the predecessor operation has been scheduled, and then randomly picks a set of compatible combinations with available machines under this operation. During the candidate action set confirming action  $a(t)$ , the  $(o_{ij}, m_k)$  search space is narrowed down according to the operations priority and machine capacity constraints in conjunction with the invalid action masking [10] technique to achieve a more efficient output.

(3) *State transition.* At the time of  $T_p(t)$ , when the agent executes the action  $a_t$ , it determines  $(o_{ij}, m_k)$  as well as completes sampling, updating the state  $s_{t+1} \leftarrow s_t$  based on the scheduling environment. During the state transition, the disjunctive arc ascertains the direction depending on the selected machine  $m_k$ ; meanwhile, the feature values of  $\mathcal{O}(t)$ ,  $\mathcal{M}(t)$  and  $\mathcal{A}(t)$  are updated. The decision-making time moves to  $T_p(t + 1)$ , which is the moment when the operation is concluded.

(4) *Reward.* The setting of the reward value  $r_t$  depends on the scheduling objective, i.e., makespan denoted by  $c_{max}$ , and takes the value of the difference between the completion time of the current state  $s_t$  and the next state  $s_{t+1}$  in Equation (16). As shown in Equation (17), maximizing the cumulative reward when  $\gamma = 1$  is equivalent to minimizing the completion time.

$$r_t = c_{max}(s_t) - c_{max}(s_{t+1}) \tag{16}$$

$$\sum_{t=0}^{|\mathcal{O}|-1} r_t = c_{max}(s_0) - c_{max}(s_{|\mathcal{O}|}) \tag{17}$$

where  $c_{max}(s_0)$  is a constant when  $t = 0$ .

In the case of  $t > 0$ , the scheduled  $o_{ij}$  is available to calculate the reward value based on the completion time  $c_{ij}$ , while the unscheduled operation replaces  $c_{ij}$  approximately by calculating the sum of the completion time of the previous process and the work hours of the current process according to the method proposed in the literature [40], as shown in Equations (18) and (19).

$$C_{ij} = C_{i(j-1)} + \min_{k \in \mathcal{M}(t)} p_{ij}^k \tag{18}$$

$$C_{i0} = 0 \tag{19}$$

(5) *Policy.* The policy of the scheduling agent belongs to the category of stochastic policy, and the policy  $\pi(a_t|s_t)$  is the probability distribution of the scheduling action set  $\mathcal{A}(t)$ , derived by the actor-critic network before the training process of policy parameters

$\pi$  for achieving optimal cumulative rewards, which later in this section, guiding the probability of the agent to choose  $a_t \in \mathcal{A}(t)$  in a certain state  $s_t$ .

#### 4.2.3. Graph Attention Network

The core of a GNN in solving scheduling problems is to embed operations and machine features, fix undirected arcs in the disjunction graph to form a job decision sequence, and ultimately obtain the maximum completion time. As shown in Table 2, current mainstream network structures mainly include traditional multi-layer GNNs, graph isomorphism networks (GINs), and heterogeneous graph neural networks (HGNNs), each of which has been adapted and innovated for the specific scheduling environment. Considering that the OE workshop requires dynamic adjustment of the number of operations and machines in each scheduling cycle based on the resource and machine information fed back from the DT system, the GAT eliminates the concern for multiple graph structures compared to GIN & HGNN, has simple data structures, and has the redundancy to handle large amounts of input data, adopting a GAT can effectively avoid the situation that the neural network is misaligned by the training parameters due to the non-uniform scheduling scale.

**Table 2.** Comparison of the Graph Neural Network structures.

Network Structure	GNN [12]	GIN [10]	HGNN [11]	GAT
Characteristics	The GNN is constructed by stacking $k$ embedding layers, and the embedding graph with node features and its domain is computed from the input graph after $k$ embedding iterations.	By encoding the dis-junction graph using GIN with discriminative ability, the preprocessing of receiving data is performed, and utilizing “add arc scheme” which ignores the un-directed arc to reduce the computational complexity.	Adopting a heterogeneous graph structure, preserving the traditional set of operation node set and conjunctive arcs, adding machine nodes, and replacing the disjunction arc set with the “operation–machine” arc set.	Based on the attention mechanism, neighbor node aggregation is implemented to select high-priority “operation–machine” combinations based on adaptive weights to solve FJSP instances of different scales.
Contributions	By learning general JSP properties, the GNN scheduler delivers better computational performance in the arbitrary instances.	Defining strategies to predict the probability distributions of operations and machines through the encoder-decoder component.	Efficient integration of operational and machine information with superior graph density and computational efficiency.	Explicit exploration of operation scheduling and machine competition relationships in disjunction graphs is attempted.
Training method	Proximal Policy Optimization (PPO)			

A GAT aggregate neighboring node is featured through the attention mechanism, and performs weight updates for each node and its first-order neighbor weight information. Where, the Graph Attention Layer (GAL) aggregates the graph containing the node feature vectors  $\mathcal{Z} = \{\mathbf{z}_i \in \mathbb{R}^d | i \in [1, N]\}$  and outputs the new feature vectors  $\mathbf{z}'_i$  for each node [34]. The correlation between neighboring nodes, as a scalar, can be obtained using a fully connected layer with an activation function of LeakyReLU, as shown in Equation (20).

$$e_{ij} = \text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}\mathbf{z}_i \parallel \mathbf{W}\mathbf{z}_j]) \tag{20}$$

where the weight parameter is  $\mathbf{a} \in \mathbb{R}^{2d^{(l+1)}}$ ,  $\mathbf{W} \in \mathbb{R}^{d^{(l+1)} \times d^{(l)}}$  represents the weight parameter of the node feature transformation in the layer and  $\cdot \parallel \cdot$  is the splicing operation. The correlation of node  $v_i$  and all first-order neighbor nodes  $\mathcal{N}(v_i)$  are normalized based on



Softmax function as shown in Equation (21), meanwhile, the new node feature vector  $\mathbf{z}'_i$  is derived from Equation (22).

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{v_k \in N(v_i)} \exp(e_{ij})} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}\mathbf{z}_i \parallel \mathbf{W}\mathbf{z}_j]))}{\sum_{v_k \in \mathcal{N}(v_i)} \exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}\mathbf{z}_i \parallel \mathbf{W}\mathbf{z}_j]))} \tag{21}$$

$$\mathbf{z}'_i = \sigma \left( \sum_{v_j \in N(v_i)} \alpha_{ij} \mathbf{W}\mathbf{z}_j \right) \tag{22}$$

From the perspective of avoiding model performance degradation associated with different scheduling scales during the OE workshop planning, as well as efficiently selecting the appropriate machine among the multi-competitive relationships, this paper adopts a similar scheme as in [40], as shown in the GAT section of Figure 6, to construct a multi-head operation attention network and a multi-head machine attention network, respectively. The operation network receives inputs of operation features and outputs a new feature sequence of operations to be scheduled based on finite constraints. The machine network assigns available devices to each operation based on each machine’s operating status and process requirements with sufficient consideration of priority relationships. Collectively, the two networks form a complete GAT, and the output operation and machine features are stacked and pooled in  $L$ -layers to generate global features for the entire scheduling instance. The following describes in detail the computational process of the single-layer network at decision step  $t$ .

(1) *Operation Node Embedding.* In the same job, the operations are restricted by the existence of pre/post constraints on the process requirements, and the determination of the execution time for each operation depends on the node as well as the inherent attributes of the neighboring nodes, such as the priority level. Based on the aforementioned attention coefficient calculation method, for each operation  $o_{ij}$ , it is necessary to analyze their attribute values for both the pre operation  $o_{i,j-1}$  and the post-operation  $o_{i,j+1}$  in the operation attention network to obtain the attention coefficients for that operation. Taking the current operation  $o_{ij}$  with its post-operation  $o_{i,j+1}$  as an example, the coefficient  $e_{i,j,j+1}$  calculation process is shown in Equation (23). Over the entire job, which usually has start/end nodes as well as some operations with no pre- or post-operation, a dynamic mask is used uniformly to cover their attention coefficient. As in Equation (21), the attention weights are obtained by normalizing all coefficients utilizing the Softmax function. Eventually, by the weighted linear combination of the input features  $\mathbf{W}\mathbf{z}_{o_{il}}$  as well as the feed-forward updating of some weights and following the computation of the nonlinear activation function  $\sigma$ , the feature vector  $\mathbf{z}'_{o_{ij}}$  of operation  $o_{ij}$  is obtained as shown in Equation (24).

$$e_{i,j,j+1} = \text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}\mathbf{z}_{o_{ij}} \parallel \mathbf{W}\mathbf{z}_{o_{i,j+1}}]) \tag{23}$$

$$\mathbf{z}'_{o_{ij}} = \sigma \left( \sum_{l=j-1}^{j+1} \alpha_{i,j,l} \mathbf{W}\mathbf{z}_{o_{il}} \right) \tag{24}$$

where  $l$  denotes the neighboring process of process  $j$ , and  $|l - j| \leq 1$ .

Leveraging the data interaction between the multilayer networks, the operation  $o_{ij}$  related parameters transferable to each related operation in this job.

(2) *Machine Node Embedding.* Machine node embedding is more complicated than operation nodes, considering that one operation can correspond to multiple machine executions. In the case of unscheduled operations  $\forall o_{ij} \in \mathcal{N}_t(m_k)$  ( $\mathcal{N}_t(m_k)$  denotes the set of operations that can be processed by the idle machine  $m_k$ ); the first step is to filter the machines based on their status, operation processing time, etc. to acquire a unique “operation–machine” combination. Hence, prior to computing the attention coefficients of each machine, the original feature vectors with their associated operations and “operation–

machine” combination are expanded to obtain  $\mathbf{v}_{ijk} = [\mathbf{h}_{o_{ij}} \parallel \mathbf{h}_{(o_{ij}, m_k)}] \in \mathbb{R}^9$  and the linear transformations  $\mathbf{W}_{\mathcal{M}} \in \mathbb{R}^{d \times 5}$  and  $\mathbf{W}_{\mathcal{O}} \in \mathbb{R}^{d \times 9}$  are set up in synchronization. It can be concluded that for a machine  $m_k$  attention coefficient is calculated as Equation (25).

$$e_{ijk} = \text{LeakyReLU}(\mathbf{b}^T [\mathbf{W}_{\mathcal{M}} \mathbf{h}_{m_k} \parallel \mathbf{W}_{\mathcal{O}} \mathbf{v}_{ijk}]) \tag{25}$$

For each machine, the effect of the self-attention coefficient additionally be considered before computing its new feature vectors, i.e., Equation (26).

$$e_{kk} = \text{LeakyReLU}(\mathbf{b}^T [\mathbf{W}_{\mathcal{M}} \mathbf{h}_{m_k} \parallel \mathbf{W}_{\mathcal{M}} \mathbf{h}_{m_k}]) \tag{26}$$

Eventually, the attention weights  $\alpha_{ijk}$  and  $\alpha_{kk}$  are obtained by normalizing  $e_{ijk}$  and  $e_{kk}$  based on the Softmax function and the feature vectors of the machine are successively solved, as shown in Equation (27).

$$\mathbf{h}'_{m_k} = \sigma \left( \alpha_{kk} \mathbf{W}_{\mathcal{M}} \mathbf{h}_{m_k} + \sum_{o_{ij} \in \mathcal{N}_i(m_k)} \alpha_{ijk} \mathbf{W}_{\mathcal{O}} \mathbf{v}_{ijk} \right) \tag{27}$$

(3) *Multi-head Attention Mechanism.* The multi-head attention mechanism provides an effective enhancement of the feature representation as well as the model self-learning capability of operation and machine attention networks. In this paper, this method is invoked in GAT, which is part of the procedure to realize the superior generalization ability of GNNs for different scale scheduling problems, and its computational principle is shown in Figure 7.

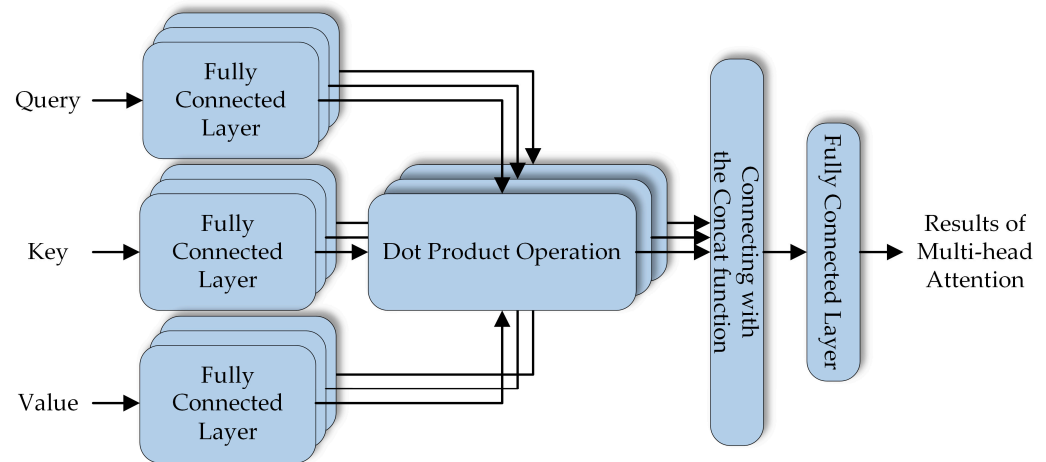


Figure 7. Multi-head attention mechanism.

Specifically, by passing operation or machine features to the query, key and value,  $K$  attention heads are set up to individually form attention blocks with different parameters, which are able to compute the attention coefficients in parallel. The output is fed back to the aggregation operator for splicing, as shown in Equation (28). The averaging operator with  $\sigma$  activation is executed after the last layer of network computation completes to obtain efficient feature expression results. The method ensures that localization is preserved without affecting the global features.

$$\mathbf{h}'_{o_{ij}} = \parallel_{k=1}^K \sigma \left( \sum_{o_{ij} \in \mathcal{O}} \alpha_{o_{ij}} \mathbf{W} \mathbf{h}_{o_{ij}} \right) \tag{28}$$

(4) *Pooling.* The mentioned GAT exports a set of features  $\mathbf{h}_{o_{ij}}^{(l)}$  and  $\mathbf{h}_{m_k}^{(l)}$  at each single layer. When  $l = 1$ , the operation and machine feature are determined by the original feature values. In addition, the “operation–machine” combination is consistently input

to the original features during calculating in each layer. After stacking the  $L$  layers, the ultimate outputs  $\mathcal{R}_{o_{ij}}^{(L)}$  and  $\mathcal{R}_{m_k}^{(L)}$  are used in the subsequent decision network. The whole scheduling features are spliced based on the mean pool of operation and machine features as in Equation (29).

$$\mathcal{R}_{Total}^{(L)} = \left[ \frac{\sum_{o_{ij} \in \mathcal{O}} \mathcal{R}_{o_{ij}}^{(L)}}{|\mathcal{O}|} \parallel \frac{\sum_{m_k \in \mathcal{M}} \mathcal{R}_{m_k}^{(L)}}{|\mathcal{M}|} \right] \tag{29}$$

(5) *Decision Making: Actor-Critic Algorithm.* Following the acquisition of the graph representation, policy network development is the culminating step in the fulfillment of the decision support. The main purpose of the policy network is to emit a feasible action  $a(t)$  and obtain the reward value  $r_t$  of that action to form an optimal sequence of actions. The actor-critic algorithm is compatible with variable-size attention models and is a superior method for achieving scheduling decisions. While the actor network and the critic network each correspond to a Multi-Layer Perceptron (MLP), the former task is interacting with the scheduling environment to generate a priority index  $\mu(a_t | s_t)$  in the first step, and in the second step, to randomize the policies  $\pi_\theta(a_t | s_t)$  based on the expectation distribution solved by the Softmax function, described in Equations (30) and (31). For the action  $a_t = (o_{ij}, m_k)$  operation features, machine features, global features, and “operation-machine” combination feature information have been fused.

$$\mu(a_t | s_t) = \text{MLP}_\theta \left[ \mathcal{R}_{o_{ij}}^{(L)} \parallel \mathcal{R}_{m_k}^{(L)} \parallel \mathcal{R}_{Total}^{(L)} \parallel \mathcal{R}_{(o_{ij}, m_k)} \right] \tag{30}$$

$$\pi_\theta(a_t | s_t) = \frac{\exp(\mu(a_t | s_t))}{\sum_{b_t \in \mathcal{A}(t)} \exp(\mu(b_t | s_t))} \tag{31}$$

The latter network objective aims to collect the value function  $v_\phi(s_t)$  during the actor network’s interaction with the environment to recognize the action’s merits and demerits. The actor-critic advances the state transition process through the continuous execution of actions by the agents until the total operation scheduling has been completed, with the objective function and the total loss function for each agent as in Equations (32) and (33).

$$L_t^{CILP}(\theta) = \hat{\mathbb{E}}_t \left[ \min(\mathcal{R}_t(\theta) \hat{A}_t, \text{clip}(\mathcal{R}_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t) \right] \tag{32}$$

$$\mathcal{L}_t(\theta, \phi) = c_p L_t^{CILP}(\theta) + c_e L_t^E(\theta) - c_v L_t^V(\theta) \tag{33}$$

where  $\mathcal{R}_t(\theta) = \pi_\theta(a_t | s_t) / \pi_{\theta_{old}}(a_t | s_t)$  stands for the ratio of the probability distributions between the old and new policy,  $\hat{A}_t$  represents the dominance estimation function, and  $\epsilon$  denotes the amount of fluctuation control. The total loss function is the weighted sum of the objective function with its entropy and mean square error,  $c_p, c_e, c_v$  represent the loss coefficients, respectively.

### 4.3. Model Training

The proposed model employs the MLP to implement both state representation learning and strategy learning. In order to better the GAT fit the optimal solution with superior generalization, the training process is completed using the Proximal Policy Optimization (PPO) [41] algorithm based on the actor-critic structure to achieve network parameter updates. PPO, as an easy-to-implement policy gradient algorithm, is the current solution to similar problems in the mainstream approach. The fast convergence and output strategy stability compared to other deep reinforcement learning algorithms is proven in [41], adopting the easy-to-implement PPO algorithm is far more suitable for other algorithms.

The pseudo-code of the training algorithm is illustrated in Algorithm 1. Both the actor network and the critic network adopt MLPs, which represent policies and states, respectively.  $v_\phi$  is obtained by inputting state embedding values passed to GAT. Training operations are executed after  $I$  iterations, in which the loss function is solved and the model

parameters are updated in PPO update steps  $K$ . In addition, a set of policy validations is fulfilled every ten iterations in the test sets.

---

**Algorithm 1:** The PPO-based training procedure

---

**Input:** GAT network, actor network  $\pi_\theta$  with trainable parameter  $\theta$  and critic network  $v_\phi$  with trainable parameter  $\phi$ , total iterations  $I$ , decision-making steps  $T$ , PPO update steps  $K$ , batch size  $B$ .

**Output:** Trained parameter sets of  $\theta$ ,  $\phi$ .

```

1  Initialization: Initialize parameters and sample  $B$  planning instances;
2  for  $i = 1, \dots, I$  do
3    for  $b = 1, \dots, B$  do
4      Initialize  $s_t$  state with instance  $b$ ;
5      for  $t = 1, \dots, T - 1$  do
6        while  $s_t$  is not terminal then
7          Obtain embeddings by GAT;
8          Sample action  $a$  based on  $\pi_\theta$ ;
9          Observe the reward  $r_t$  and next state  $s_{t+1}$ ;
10          $s_t \leftarrow s_{t+1}$ ;
11        end
12        if  $s_t$  is terminal then
13          break;
14        end
15      end for
16    end for
17    for  $k = 1, \dots, K$  do
18      Calculate the PPO loss value  $\mathcal{L}_t(\theta, \phi)$ , Iterative update parameters  $\theta$ ,  $\phi$  for  $\mathcal{R}$  epochs;
19      Update network parameters;
20    end for
21    if  $i \% 10 = 0$  then
22      Validate policy model performance;
23    end
24  end for

```

---

## 5. DT Workshop Planning Model-Based Decision Support Methodology

In a broad sense, the workshop Master Production Schedule (MPS) management includes the production forecast and execution decision of the next stage of the workshop. On the basis of the GNN-based planning model, this section expands data availability and supports dynamic decision-making of the whole production process by combining the large amount of time series data formed by the DT workshop. Specifically, in the planning aspect, the job schedule is dynamically optimized according to the real-time progress of the DT system, and in the resource aspect, the job bill of materials (BOMs) is associated with the enterprise resource plan, and the resource procurement, storage, and distribution plan is optimized in the reverse direction from the workstation demand, to realize the workshop operations' work time accuracy and resource equalization.

### 5.1. Work Hours Forecasting

The colossal assembly and welding workloads are determined by the intricate design of OE platforms. The operators' estimation of their working hours is inaccurate due to the twin effects of the learning effect, including operation proficiency, and the deterioration effect, including fatigue degree. The errors' cumulative effect severely hampers the execution rate of the workshop plan. Hence, it is essential to anticipate labor hours using DT.

Based on the previous study [42], the measured data access and the forecasted data output are additional elements in the work hours forecasting framework, as shown in

Figure 8. The measured data from the data acquisition layer of the DT framework is input to the work hours quota database and the process route database, and K-means-based clustering analysis of welding processes is performed according to the operational characteristics to evaluate the work hours quotas according to the classification of different types of welding processes with various difficulties. The evaluation method uses a slack-based measure model in data envelopment analysis to update the time quota  $OP_i$  for the process type  $i$  according to the planned data and execution data and to evaluate the validity. Finally, according to the same type of process over time iteration of work time rules, the formation of dynamic standard work time allows MPS to achieve highly reliable data input. Equation (34) shows the prediction model  $WHFM(POT_i, ft_i)$  for work hours of welding operations with welding characteristics as the independent variable.

$$WHFM(POT_i, ft_i) = f(OP_i) \tag{34}$$

subject to:

$$\min(ft_i) \forall i \in [1, N] \tag{35}$$

$$\min(POT_i + ft_i - AOT_i) \forall i \in [1, N] \tag{36}$$

$$\min \sum_{i=1}^N (POT_i + ft_i - AOT_i) \tag{37}$$

where  $POT_i$  is planned operating time,  $AOT_i$  represents actual operating time, and  $ft_i$  stands for floating time.  $f(OP_i)$  is a recursive equation based on Graph Long Short-Term Memory (Graph LSTM):

$$f(OP_i) = \sigma \left( \mathbf{W}^f op_i^{(t)} + \mathbf{U}_{m(i,k)}^f \mathbf{h}_{ik}^{(t-1)} + \mathbf{b}^f \right) \tag{38}$$

where  $\mathbf{W}$  stands for the weights,  $\mathbf{U}$  for the weight matrix to be trained,  $m(i, k)$  indicates the relationship type label between nodes,  $\mathbf{h}$  for the hidden vector, and  $\mathbf{b}$  for the training bias.

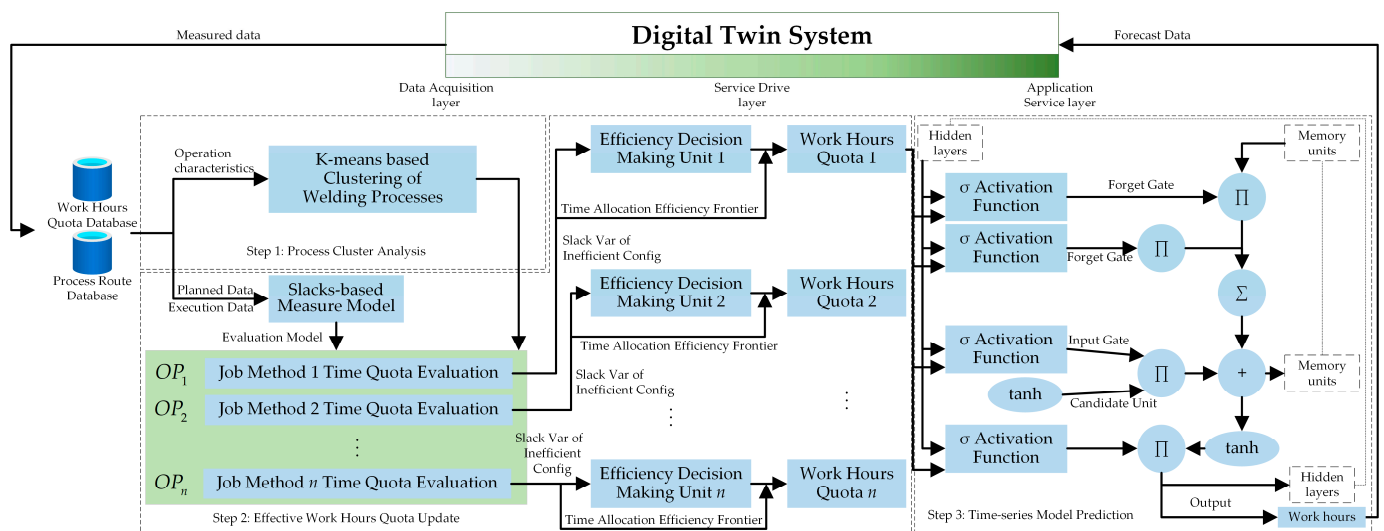


Figure 8. Work hours forecasting architecture.

### 5.2. Resource Demand Forecasting

With the granularity refinement of the DT system, the resource demand forecast originates from the demand side of the operators, and the workshop resource plan and the enterprise resource plan are backed up in time with the progress of each workstation’s assembly and welding operations. In contrast to the traditional resource supply mode,

in which the top level conducts the whole resource plan and breaks down BOMs to the basic operation units, the real-time demand and buffer capacity of each workstation will be dynamically adjusted along with the manufacturing execution status to ensure the quantity and period of material supply, such as plate, steel, and welding material. The limitation of demand forecasting is the uncertainty of manufacturing progress and the uncoordinated procurement/storage plan with the workshop schedule.

There are two stages of demand forecasting from the operation execution side: material demand clustering and quantity forecasting. Firstly, the material demand clustering is based on the steel, welding material and equipment necessary for each workstation in the process of job execution to establish the attribute class rules to complete the clustering analysis. Given that  $R = \{r_1, r_2, \dots, r_i, \dots, r_n\}$  is the set of material resource demand, and  $k$  attributes exist under each material, the material attribute matrix is obtained as follows:

$$(r_i)_{n \times k} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1k} \\ r_{21} & r_{22} & \cdots & r_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nk} \end{bmatrix} \tag{39}$$

$$ct = \frac{c - \min(c)}{\max(c) - \min(c)} \times 100\% \tag{40}$$

$$S(c_i, c_j) = \text{dist}(c_i, c_j) = \sqrt{\sum_{l=1}^k (c_{il} - c_{jl})^2} \tag{41}$$

To reduce the impact brought by different quantiles, the material attribute matrix is normalized by Equation (40) and then solved for the similarity of demand at different workstations. As in Equation (41), the higher the similarity, the smaller the Euclidean distance value, representing the high demand for certain types of resources at certain workstations and the need to increase the advanced procurement and supply of this type of resource.

The second step is quantity forecasting based on the work orders and BOMs of each workstation to count the demand of each type of resource combined with the real-time capacity of the material buffer monitored by the DT system. In addition, the material ratio must be outputted according to the clustering results of the dependence of each workstation on different materials, and the relevant information must be adjusted in MPS and Enterprise Resource Planning (ERP) in time.

### 5.3. Resource Bottleneck Identification

Resource bottlenecks are an important factor affecting the continuity of operations. Dynamic resource bottleneck identification methods and optimization of manufacturing resources [43] have been explored in many studies, and the feasibility and practicality of bottleneck identification analysis are further enhanced with the support of the Digital Twin. In the event that the sensor identifies fluctuations in the amount of material in the operation area or buffer, the bottleneck identification method timely pushes the operation blocking node to the DT system and triggers the rapid adjustment of MPS and ERP.

The bottleneck identification calculation mainly includes resource state, device state, and buffer state characteristics. In addition, further consideration is given to the interactions between operational links. The real-time bottleneck calculation methodology can be summarized as follows:

$$RBI(i) = \omega f_{res}(i) + \mu f_{eq}(i) + \lambda f_{buffer}(i) \tag{42}$$

$$f(i) = \frac{v_{measure}}{\max\{v_{measure}^{(1)}, v_{measure}^{(2)}, \dots, v_{measure}^{(t)}\}} \tag{43}$$

$$\omega + \mu + \lambda = 1 \tag{44}$$

where  $i$  is the workstation number,  $f_{res}(i)$  is the resource state function,  $f_{eq}(i)$  represents the device state function, and the buffer state function is  $f_{buffer}(i)$ . The state calculation function is shown specifically in Equation (43) as the ratio of the real-time measured value to the maximum value in time  $t$ .  $\omega, \mu,$  and  $\lambda$  are the weight coefficients, whose sum is 1 as shown in Equation (44).

Demand forecasting and bottleneck identification are the two most significant aspects of workshop resource management, which are intimately related to the planning model. Based on the real-time update of resource information by the DT system, managers are able to analyze the operation plan and resource plan collaboratively and construct an effective “Plan-Resource” management framework, as shown in Figure 9.

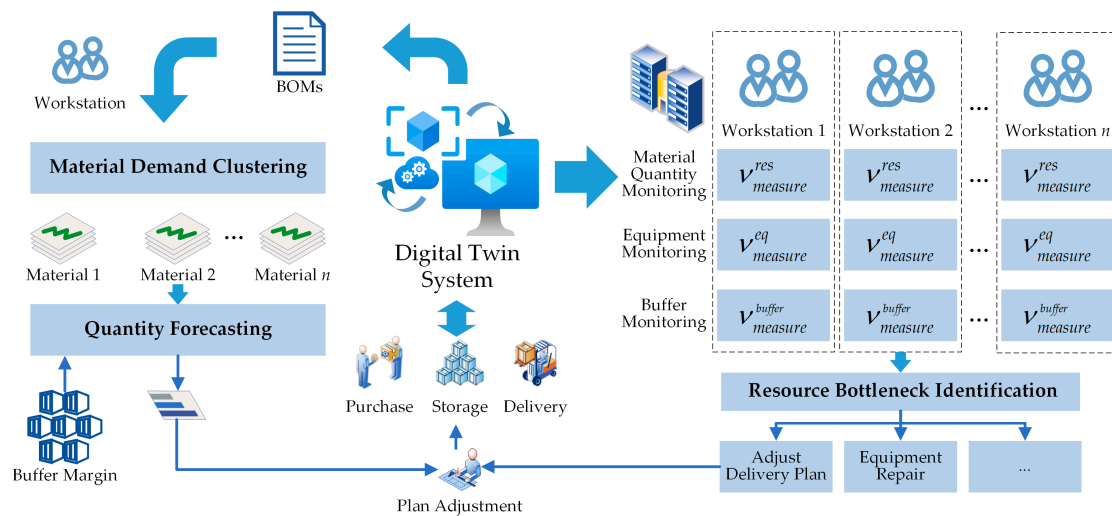


Figure 9. “Plan-Resource” management framework.

#### 5.4. Dynamic Job Shop Scheduling

The dynamic environment-oriented planning method has been proposed in the planning model study in Section 4. However, during the execution of the plan, if there are uncertain events, such as equipment malfunction, production insertion order, material blockage, etc., it will greatly interfere with the plan and cause the manufacturing tasks to be unable to be completed within the scheduled time. Accordingly, it is essential to design a dynamic scheduling strategy for uncertain events and to reduce the impact of unexpected occurrences on job execution by making small-scale process start time adjustments. Meanwhile, multiple optimization objectives to obtain optimal or non-dominated solutions require consideration in the dynamic adjustment process.

In the previous studies [44,45], focusing on the multi-objective optimization strategy of dynamic scheduling problems in the presence of uncertain events and proposing optimization algorithms for the makespan and optimal energy consumption, the ability to deal with event occurrences was theoretically developed. However, the prior algorithm research is based on definite assumptions, and the uncertain events appear randomly with probability, which is still partially deviated from the actual working conditions. Based on the real-time monitoring of progress, materials, equipment, and other attributes of the DT system to achieve effective identification of uncertain events, which completely responds to the actual situation of the workshop, there is a higher requirement for dynamic scheduling algorithm triggering and rapid calculation. By constructing a hybrid cycle and event-driven strategy, the workshop problem node production plan is optimized in time for different types of events, and the adjustment of the total plan is minimized. As shown in Figure 10, after an uncertain event such as equipment damage at workstation 2, the remaining workload of the current work order is first recalculated, then the three affected

subsequent tasks with floating time are statistically counted, and finally, the fine-tuning of the plan is accomplished. The dynamic scheduling process is shown in Figure 11. The efficiency of the dynamic scheduling trigger and plan iteration process based on the DT system is significantly improved compared with the traditional manual adjustment, with a saving of about 17.1% or more according to statistics. The dynamic scheduling model that makespan and minimizes energy consumption for production can be expressed as follows:

$$\min F(x) = [f_1, f_2] \tag{45}$$

$$f_1 = \min(C_{max}) \tag{46}$$

$$f_2 = \min(E) \tag{47}$$

where  $F(x)$  denotes the target vector,  $F = (f_1, f_2) \in Y$ ,  $Y$  is the target space formed by multiple target vectors,  $x$  is the decision vector,  $C$  is the completion time,  $E$  is the energy consumption parameter,  $f_1$  represents the makespan, and  $f_2$  is the minimum production energy consumption. The constraints are the same as in Equations (12)–(15).

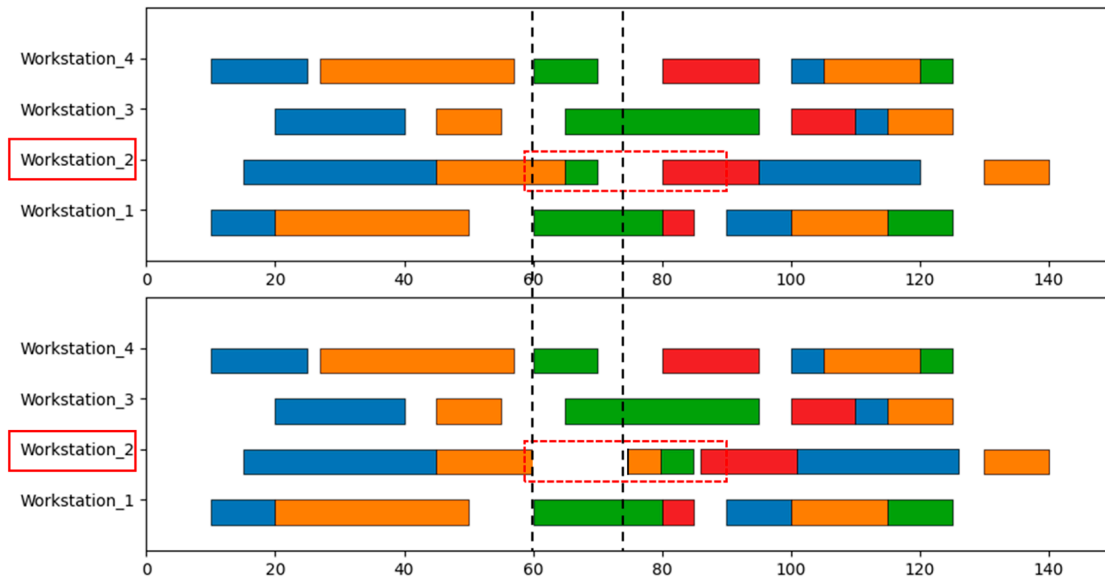


Figure 10. Dynamic job shop scheduling example.

### 5.5. Multi-Station Collaborative Distribution

The delivery of materials and intermediate products within the workshop has been a relatively neglected part of production planning. In contrast to the high priority given to supply chain and in-factory logistics distribution, the scheduling of logistics distribution between workstations is autonomously done by operators, which easily causes blockages in the flow between processes. Based on the digital twin workshop planning model, expanding the multi-station collaborative distribution method enables the effective improvement of the inter-operational material and intermediate product turnover efficiency. Starting from the assignment plan of each workstation, based on the work sequential relationship between multiple workstations, the distribution requests are received and associated with BOMs to transport the target objects in the form of pallets to the specified workstations or buffer zones, and the idle vehicles and optimal paths within the effective turnaround time are calculated, and the distribution plan is output, as shown in Figure 12. The collaborative distribution model with the minimum empty time of transportation equipment as the optimization objective is:



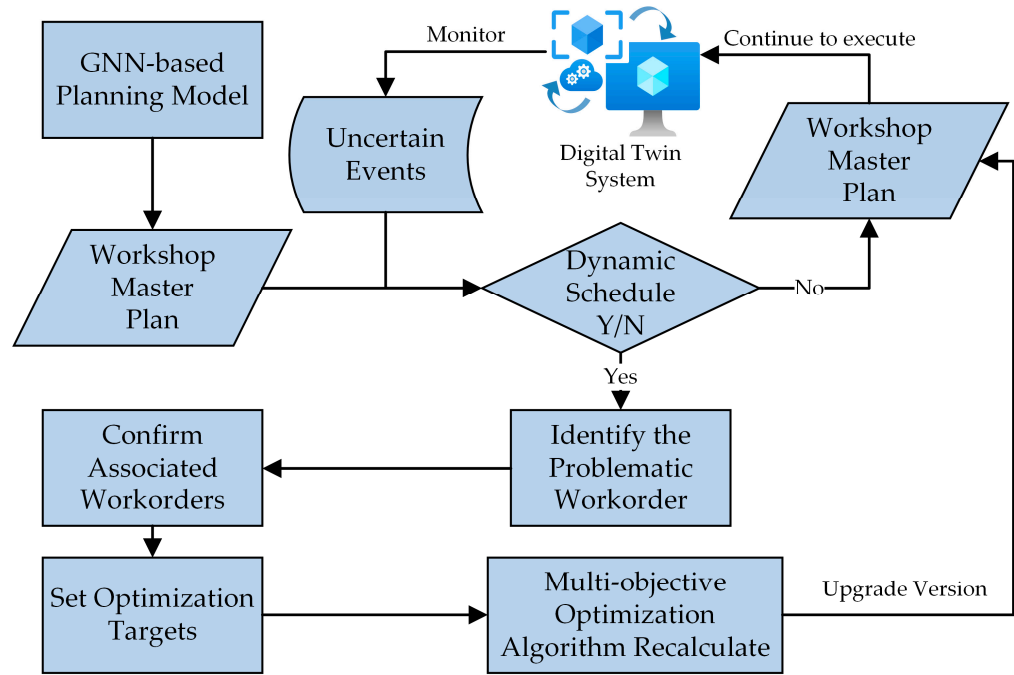


Figure 11. Rescheduling process.

$$f = \sum_{m=1}^M \sum_{j=1, j \neq i}^n \sum_{i=1}^n y_{imj} \cdot Nt_{ij} + \sum_{m=1}^M \sum_{j=1}^n y_{0mj} \cdot Nt_{0j} + \sum_{m=1}^M \sum_{i=1}^n y_{im0} \cdot Nt_{i0} \quad (48)$$

$$y_{imj} = \begin{cases} 1 & \text{Vehicle } m \text{ transport task } i \text{ after transport task } j \\ 0 & \text{else} \end{cases} \quad (49)$$

the constraints include:

$$Rt_i - Rt_j + Ut_i + LT_i + y_{imj} \cdot Nt_{ij} \leq R \cdot (1 - \sum_{m \in M} y_{imj}) \forall i, m, j \quad (50)$$

$$y_{0mj} = 1 \quad (51)$$

$$St_i \leq Rt_i \leq Et_i \quad (52)$$

$$\sum_{i \in n, i \neq j}^n y_{imj} + y_{0mj} = x_{jm} \forall m, j \quad (53)$$

$$\sum_{j \in n, j \neq i}^n y_{imj} + y_{im0} = x_{im} \forall i, m \quad (54)$$

$$\sum_{j \in n, j \neq i}^n y_{imj} \leq x_{jm} \forall i, m \quad (55)$$

$$x_{im} = \begin{cases} 1 & \text{task } i \text{ is transported by vehicle } m \\ 0 & \text{else} \end{cases} \quad (56)$$

where  $i$  and  $j$  are task numbers,  $m$  represents the vehicle number,  $x_{im}$  and  $y_{imj}$  are decision variables for task assignment,  $St_i$ ,  $Rt_i$ ,  $Et_i$  denote the earliest start time, actual start time and latest end time respectively,  $LT_i$  is the load run time, and  $Nt_{ij}$  represents the no-load time.

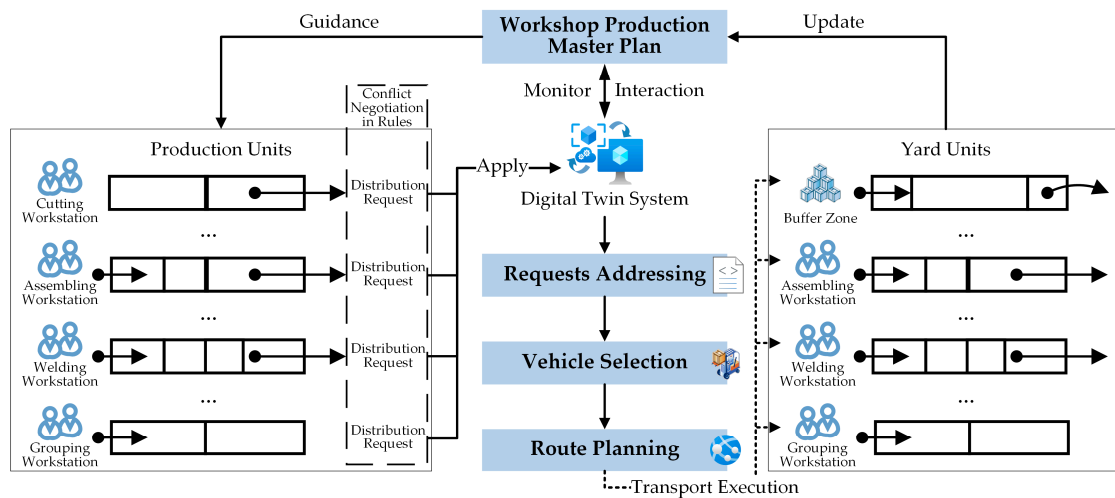


Figure 12. Multi-station collaborative distribution schematic.

In summary, the five decision support methods proposed in this section are all application services tightly associated with the DT workshop planning model, which enrich the practicality of the DT architecture and enormously enhance the value of utilization and service-driven importance of the GNN-based planning model. This section maintains immense significance as it is dedicated to studying the feasibility of expanding the application of the DT service-driven layer. According to the development trend of intelligent workshops, the existing quality management and traceability methods [46], as well as inspection, safety, green production, and other workshop control businesses, may be explored in the future to integrate into the intelligent management solutions of “virtual-real co-drive” to assist enterprises achieve full digital coverage.

### 6. Case Validation and Discussion

This section concentrates on validating the effectiveness of the aforementioned GNN-based planning model and the application of the digital twin system by ocean engineering enterprises. We analyze the proposed model in terms of the computational effect and generalization ability under the attention mechanism, as well as discuss the feasibility and practicability of the DT system for actual execution in enterprises.

#### 6.1. Effectiveness of GNN-Based Planning Model

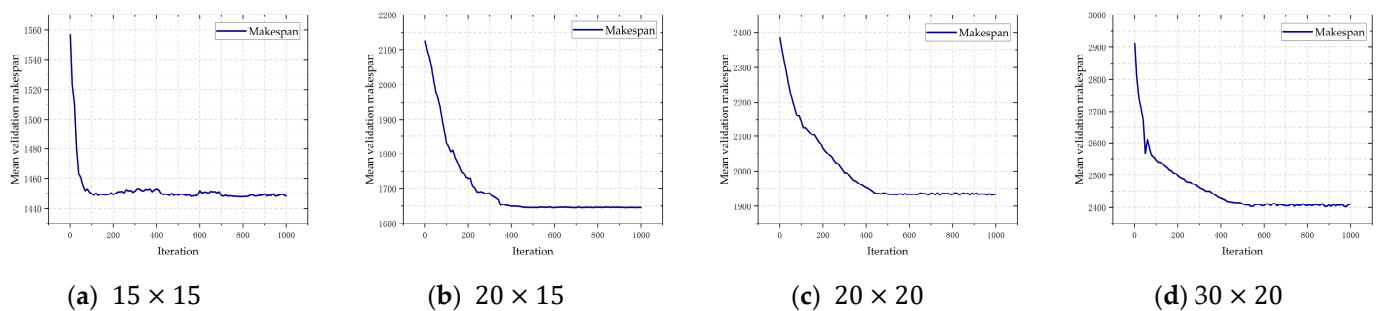
The model proposed in this paper for solving the workshop planning problem is a combination of Graph Attention Networks and Deep Reinforcement Learning, i.e., GAT-DRL. In order to validate the computational effectiveness of the planning model in the actual scheduling environment, a synthetic dataset consisting of standard arithmetic examples and workshop planning samples are employed as the GNN training, validation and testing dataset. The training dataset has a total of four scales corresponding to the number of jobs  $n = \{15, 20, 30\}$  and machines  $m = \{15, 20\}$ , where each operation  $o_{ij}$  contains available machine  $|\mathcal{M}|$  and workhour values  $p_{ij}^k$  sampled integrally from the set of machines  $U(1, m)$  and the set of workhours  $U(1, 99)$ , respectively.

The PPO hyperparameters of the model, in accordance with the optimal scheme of comparable research [10–16,40], are configured with the total number of iterations  $I = 1000$ , the instance batch size (training environment)  $B = 4$ , the step size of each iteration of policy update  $K = 10$ , the learning rate  $lr = 3 \times 10^{-4}$ , the discount rate  $\gamma = 0.98$ , and the clipping parameter  $\epsilon = 0.2$ . The policy loss coefficients, the entropy coefficients as well as the critics’ loss coefficients correlate to 2, 0.01, and 1. Regarding the GAT itself, the number of attention layers  $L = 2$ , and the operation attention network and the machine attention network are both set up with four attention heads per layer each, while ELU is used as the activation function  $\sigma$ . The sizes of the MLPs involved are uniformly 64. The model with its

parameter settings has been applied in practical engineering deployed in the server of the OE manufacturing enterprise with the DT system, which has been packaged as a module and constitutes one of the crucial functions in the application service layer. The entire test is done in a test computer with an Intel i9-12900H CPU and NVIDIA RTX 3070Ti GPU to verify that the model has efficient computational performance and generalizability under the initial conditions, ignoring the influence of the parameters that have been trained in the actual project.

The benchmark analyzes the gap in the optimal solution, stability, and speed against GAT-DRL, with simultaneous PDRs [11] and Genetic Algorithms (GA) [45] solving the makespan of scheduling instances as well as the computation time. In particular, the PDRs are evaluated against four rules, namely, First-in-First-out (FIFO), Shortest Processing Time First (SPT), Longest Processing Time First (LPT), and Most Work Remaining (MWKR), which have the best overall performance. The mathematical model of the comparison algorithm is subject to Section 4.1.

The training curves of GAT-DRL on  $15 \times 15$ ,  $20 \times 15$ ,  $20 \times 20$  and  $30 \times 20$  instances are illustrated in Figure 13, which reveals that the model accommodates different sizes of datasets and converges rapidly, verifying it acclimates to varying sizes of scheduling problems and obtains high-quality solutions. In addition, with the support of IAM, the converged mean validated makespan is stable with minor fluctuations.



**Figure 13.** Training curves of the GAT-DRL on four scheduling instances.

For the trained GAT-DRL model, multiscale performance tests have been conducted in this paper based on the TA dataset and a set of synthetic datasets with equivalent actual computational sizes in the workshop. The tests are executed for a total of eight sizes, corresponding to solving small-scale ( $15 \times 15$  &  $20 \times 15$ ), medium-scale ( $30 \times 15$ ,  $20 \times 20$  &  $30 \times 20$ ), and large-scale ( $50 \times 20$ ,  $100 \times 20$  &  $100 \times 12$ ) scheduling problems. Each algorithm in each scale carries out 100 group tests, records the makespan minimum in each group, and statistics the average computation time along with the error fluctuation gap  $\Delta = (c_{max}/c_{max}^* - 1) \times 100\%$ .

Table 3 shows that GAT-DRL has excellent solving ability and the optimal or sub-optimal solution obtained in each size, which indicates that the model trained based on small and medium-sized datasets has superior generalization ability to adaptively deal with complex actual working conditions, whereby the model outputs the optimal results when the number of jobs and the number of machines change dynamically. In comparison with the proposed method in this paper, the four PDRs have the best computational speed, which is more than 50%, yet the solution results are not satisfactory, and the randomness of acquiring the minimum value is considerable. In terms of stability of results, the genetic algorithm, as a representative of heuristic algorithms, has prominent performance in the process of calculating workshop scale instances ( $100 \times 12$ ). Nevertheless, restricted by the high requirements of GA on data size, the performance has enormous constraints in calculating the standard test dataset, which demands targeted improvement. In addition, the computation time of GA is substantially higher than that of GAT-DRL and PDRs, thus it is suitable for pursuing a highly customized scheduling system with great reliability. Overall, GAT-DRL is the optimal generalized scheduling solution with high accuracy, high

performance, and high reliability suitable for DT systems. It should be supplemented that the action selection in this model takes a random sampling approach, and the convergence speed may be further improved by switching to a greedy strategy with a fixed number of machines. The gap analysis graph for the eight sets of comparative computation results is shown in Figure 14.

**Table 3.** Comparison of models' performance in different scale data.

	Size ( $n \times m$ )	GAT-DRL	FIFO	SPT	LPT	MWKR	GA
$c_{max}$	15 × 15	1448	1513	1546	1547	1489	1433
Time		1.11	0.60	0.74	0.71	0.71	46.17
Gap		3.14%	5.58%	7.89%	7.98%	4.31%	0.56%
$c_{max}$	20 × 15	1646	1778	1814	1775	1677	1702
Time		1.46	0.81	0.99	0.93	0.96	60.66
Gap		2.52%	8.00%	10.31%	7.82%	3.55%	3.37%
$c_{max}$	30 × 15	2189	2330	2419	2379	2216	2275
Time		2.31	1.48	1.87	1.79	1.78	96.20
Gap		3.60%	7.55%	10.83%	9.12%	3.69%	3.91%
$c_{max}$	20 × 20	1933	2065	2067	2128	2005	2012
Time		1.98	1.09	1.31	1.18	1.26	82.29
Gap		2.17%	6.80%	6.93%	10.09%	3.94%	4.06%
$c_{max}$	30 × 20	2403	2549	2619	2604	2479	2580
Time		3.57	1.97	2.46	2.31	2.34	148.80
Gap		3.70%	6.09%	8.99%	8.36%	4.61%	7.34%
$c_{max}$	50 × 20	3338	3504	3571	3501	3368	3754
Time		9.09	3.14	4.23	3.86	3.89	378.77
Gap		4.58%	6.32%	7.02%	5.25%	4.00%	12.46%
$c_{max}$	100 × 20	5845	6052	6139	6093	5840	6374
Time		43.42	13.20	21.14	19.14	19.97	1808.81
Gap		3.04%	4.81%	5.12%	4.82%	2.95%	9.14%
$c_{max}$	100 × 12	6782	6934	7117	7004	6811	6845
Time		25.76	7.93	12.66	11.52	12.04	1073.09
Gap		6.75%	7.72%	9.20%	6.09%	5.88%	1.56%

## 6.2. Application of the Digital Twin System in Workshop

For testing and validation, the proposed digital twin system architecture in this paper is implemented at a Chinese ocean engineering manufacturer. Each business module in the system may function both independently and efficiently together to assist in the fulfillment of the digital twin service via the system's adoption of a micro-service architecture. The service middleware and management backend use Java and Oracle, the system front-end uses Vue.js technology, and the visualization board is finished using Unity 3, and other open-source board technology. As indicated in Figure 15, it completes the creation of 5 modules and 33 functions. The system function module is designed to complete data management in phases of the ocean engineering construction process, integrate corresponding business data according to design, manufacturing, and basic data, provide prediction and optimization services, and eventually provide a visualization board to fulfill decision support.

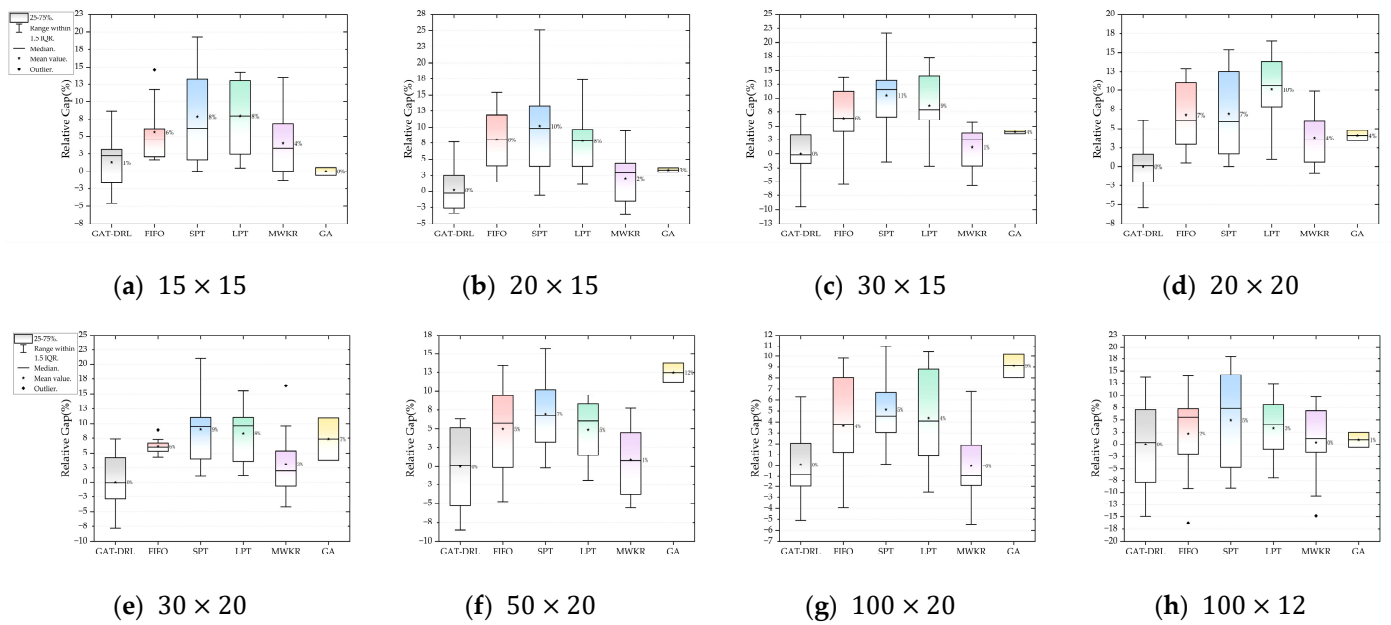


Figure 14. Relative gaps of algorithm results for different size datasets.

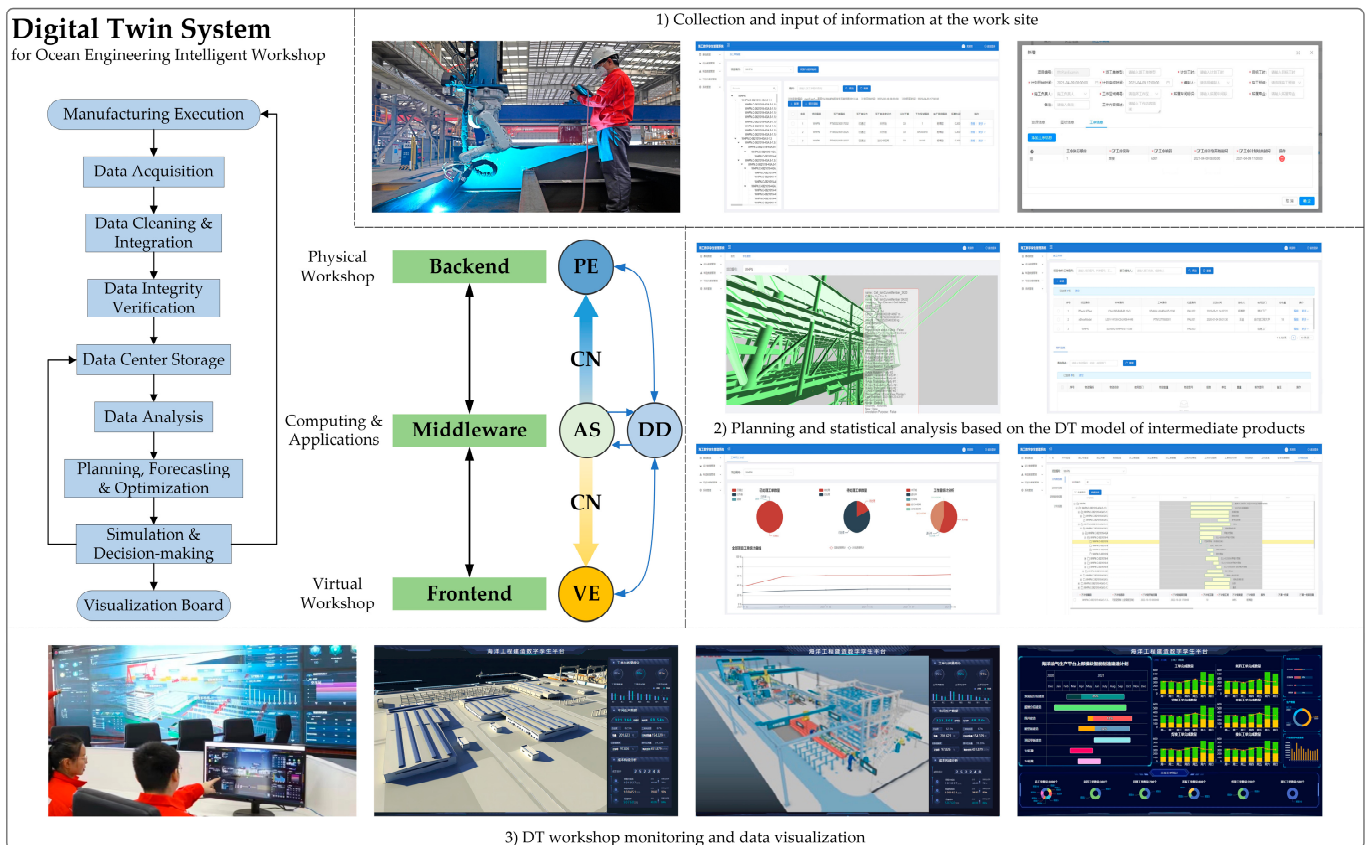


Figure 15. Application of the DT system for OE intelligent workshop.

By contrasting the DT system with the in-service MES in the block assembly and welding workshop, system validation of multiple project construction scheduling has been carried out. The test results are listed in Table 4. According to the findings, the DT system’s scheduling function compared to the MES can be completed in 5.08% less time for the total duration, and calculation speed can be increased by 16.30%. The visual interface of this system not only makes it convenient to monitor the job plan and progress in the workshop,

but it also emphasizes the optimization of the data-driven job execution process and makes use of the five-dimensional logic model in production management.

**Table 4.** Comparison of the DT system performance with the existing MES.

Project	Total Project Duration (Days)		Calculation Time (s)	
	DT	MES	DT	MES
A	246	261	783.28	905.45
B	242	254	780.10	903.89
C	239	251	757.61	967.77

Data collection in the system export results. Commercially sensitive information is not shown, and only system-exported test data are provided.

## 7. Conclusions and Future Work

This study proposes a five-dimensional architecture of the digital twin system for the ocean engineering intelligent workshop and concentrates on introducing a graph neural network in the service-driven layer to transform the workshop disjunctive graph model into multi-dimensional data, and relying on the attention mechanism, analyze the operation and equipment information to complete node embedding and realize feature extraction. The PPO algorithm training is performed to establish an end-to-end learning framework for solving FJSP, and the GAT network parameters are optimized so that the planning model has excellent generalization after the training of small-sized data, which resolves the problems of long computation time and poor optimization effect in the current workshop planning management practice. With the support of the DT system, a large amount of data input ensures that the model is adequately trained and systematically robust in the face of complex conditions in the shop floor scheduling environment. The numerical results show that the model proposed in this paper can complete the output of workshop planning, and the output results and computation time are significantly better than the traditional heuristic methods.

In future research work, the DT system must fulfill the functional iterations following the continuous upgrading of the intelligent workshop and transform itself from the current “real-time status feedback and decision optimization in parallel” mode to the “prediction data-based virtual-real co-drive” mode to materialize the workshop production process that enables pre-awareness, advance decision-making, and rapid optimization. In addition, the GNN-based planning model is capable of handling production conflicts quickly, i.e., optimizing the model response speed in the dynamic scheduling procedure, further exploring offline reinforcement learning methods, reducing training costs, and improving the generalization ability of the model for online applications.

**Author Contributions:** Conceptualization, J.L. and W.Y.; methodology, W.Y.; software, B.Y.; validation, W.Y., B.Y., L.C. and H.Y.; formal analysis, W.Y.; investigation, Y.C. and H.Y.; resources, J.L. and L.C.; data curation, B.Y.; writing—original draft preparation, W.Y.; writing—review and editing, W.Y. and B.Y.; visualization, R.D. and H.Y.; supervision, J.L.; project administration, J.L., B.Y. and L.C.; funding acquisition, J.L. and L.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research program is funded by the projects of Research on Intelligent Manufacturing Solutions and Key Technologies for the Upper Module of Offshore Oil and Gas Production Platforms [Grant number 2018473] and Research on Collaborative Design Technology of High-Tech Ocean-going Passenger Ships [Grant number 2019331].

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors are responsible for the contents of this publication. In addition, the authors would like to thank lab classmates for their contribution to the writing quality.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Tao, F.; Qi, Q. Make more digital twins. *Nature* **2019**, *573*, 490–491. [[CrossRef](#)] [[PubMed](#)]
2. Tao, F.; Zhang, H.; Liu, A.; Nee, A. Digital twin in industry: State-of-the-art. *IEEE Trans. Ind. Inform.* **2018**, *15*, 2405–2415. [[CrossRef](#)]
3. Bangsow, S. *Tecnomatix Plant Simulation*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 103–180.
4. Fourgeau, E.; Gomez, E.; Adli, H.; Fernandes, C.; Hagege, M. System Engineering Workbench for Multi-views Systems Methodology with 3DEXPERIENCE Platform. In Proceedings of the Second Asia-Pacific Conference on Complex Systems Design & Management, CSD&M Asia 2016, Singapore University of Technology, Singapore, 24–26 February 2016.
5. Tao, F.; Liu, W.; Zhang, M.; Hu, T.; Qi, Q.; Zhang, H.; Sui, F.; Wang, T.; Xu, H.; Huang, Z.; et al. Five-dimension digital twin model and its ten applications. *Comput. Integr. Manuf. Syst.* **2019**, *25*, 1–18.
6. Wei, Y.; Guo, L.; Chen, L.; Zhang, H.; Hu, X.; Zhou, H.; Li, G. Research and implementation of digital twin workshop based on real-time data driven. *Comput. Integr. Manuf. Syst.* **2021**, *27*, 352–363.
7. Kong, T.; Hu, T.; Zhou, T.; Ye, Y. Data construction method for the applications of workshop digital twin system. *J. Manuf. Syst.* **2021**, *58*, 323–328. [[CrossRef](#)]
8. Leng, J.; Zhang, H.; Yan, D.; Liu, Q.; Chen, X.; Zhang, D. Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1155–1166. [[CrossRef](#)]
9. Zheng, Y.; Yang, S.; Cheng, H. An application framework of digital twin and its case study. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1141–1153. [[CrossRef](#)]
10. Yuan, E.; Cheng, S.; Wang, L.; Song, S.; Wu, F. Solving job shop scheduling problems via deep reinforcement learning. *Appl. Soft Comput.* **2023**, *143*, 110436. [[CrossRef](#)]
11. Song, W.; Chen, X.; Li, Q.; Cao, Z. Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1600–1610. [[CrossRef](#)]
12. Park, J.; Chun, J.; Kim, S.; Kim, Y.; Park, J. Learning to schedule job-shop problems: Representation and policy learning using graph neural network and reinforcement learning. *Int. J. Prod. Res.* **2021**, *59*, 3360–3377. [[CrossRef](#)]
13. Lei, K.; Guo, P.; Zhao, W.; Wang, Y.; Qian, L.; Meng, X.; Tang, L. A multi-action deep reinforcement learning framework for flexible Job-shop scheduling problem. *Expert Syst. Appl.* **2022**, *205*, 117796. [[CrossRef](#)]
14. Lei, K.; Guo, P.; Wang, Y.; Xiong, J.; Zhao, W. An End-to-end Hierarchical Reinforcement Learning Framework for Large-scale Dynamic Flexible Job-shop Scheduling Problem. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022.
15. Chen, R.; Li, W.; Yang, H. A Deep Reinforcement Learning Framework Based on an Attention Mechanism and Disjunctive Graph Embedding for the Job-Shop Scheduling Problem. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1322–1331. [[CrossRef](#)]
16. Han, B.-A.; Yang, J.-J. Research on adaptive job shop scheduling problems based on dueling double DQN. *IEEE Access* **2020**, *8*, 186474–186495. [[CrossRef](#)]
17. Grieves, M.W. PLM—beyond lean manufacturing. *Manuf. Eng.* **2003**, *130*, 23.
18. Glaessgen, E.; Stargel, D. The digital twin paradigm for future NASA and US Air Force vehicles. In Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, HI, USA, 23–26 April 2012.
19. Liu, X.; Furrer, D.; Kusters, J.; Holmes, J. *Vision 2040: A Roadmap for Integrated, Multiscale Modeling and Simulation of Materials and Systems*; No. E-19477; NASA: Washington, DC, USA, 2018; pp. 1–168.
20. Tao, F.; Xiao, B.; Qi, Q.; Cheng, J.; Ji, P. Digital twin modeling. *J. Manuf. Syst.* **2022**, *64*, 372–389. [[CrossRef](#)]
21. Tao, F.; Zhang, M. Digital twin shop-floor: A new shop-floor paradigm towards smart manufacturing. *IEEE Access* **2017**, *5*, 20418–20427. [[CrossRef](#)]
22. Guo, J.; Zhao, N.; Sun, L.; Zhang, S. Modular based flexible digital twin for factory design. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1189–1200. [[CrossRef](#)]
23. Zhang, K.; Qu, T.; Zhou, D.; Jiang, H.; Lin, Y.; Li, P.; Guo, H.; Liu, Y.; Li, C.; Huang, G. Digital twin-based opti-state control method for a synchronized production operation system. *Robot. Comput. Integr. Manuf.* **2020**, *63*, 101892. [[CrossRef](#)]
24. Lee, J.; Lapira, E.; Yang, S.; Kao, A. Predictive manufacturing system-Trends of next-generation production systems. *Ifac Proc. Vol.* **2013**, *46*, 150–156. [[CrossRef](#)]
25. Schleich, B.; Anwer, N.; Mathieu, L.; Wartzak, S. Shaping the digital twin for design and production engineering. *CIRP Ann.* **2017**, *66*, 141–144. [[CrossRef](#)]
26. Sun, X.; Bao, J.; Li, J.; Zhang, Y.; Liu, S.; Zhou, B. A digital twin-driven approach for the assembly-commissioning of high precision products. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101839. [[CrossRef](#)]
27. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [[CrossRef](#)]
28. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [[CrossRef](#)]

29. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005.
30. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv* **2013**, arXiv:1312.6203.
31. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3844–3852.
32. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
33. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated graph sequence neural networks. *arXiv* **2015**, arXiv:1511.05493.
34. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
35. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1024–1034.
36. Gilmer, J.; Schoenholz, S.; Riley, P.; Vinyals, O.; Dahl, G. Neural Message Passing for Quantum Chemistry. In Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017.
37. Zhu, J.; Wu, M.; Liu, C. Research on the Application Mode of Blockchain Technology in the Field of Shipbuilding. In Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 27–29 June 2020.
38. Shi, J.; Liu, X.; Liu, T. Method of digital twin logic model oriented to production line simulation. *Comput. Integr. Manuf. Syst.* **2022**, *28*, 442–454.
39. Rocha, M.; Simão, A.; Sousa, T. Model-based test case generation from UML sequence diagrams using extended finite state machines. *Softw. Qual. J.* **2021**, *29*, 597–627. [[CrossRef](#)]
40. Wang, R.; Wang, G.; Sun, J.; Deng, F.; Chen, J. Flexible Job Shop Scheduling via Dual Attention Network Based Reinforcement Learning. *arXiv* **2023**, arXiv:2305.05119.
41. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
42. Li, J.; M Sun, M.; Han, D.; Wang, J.; Mao, X.; Wu, X. A knowledge discovery and reuse method for time estimation in ship block manufacturing planning using DEA. *Adv. Eng. Inform.* **2019**, *39*, 25–40. [[CrossRef](#)]
43. Su, X.; Lu, J.; Chen, C.; Yu, J.; Ji, W. Dynamic Bottleneck Identification of Manufacturing Resources in Complex Manufacturing System. *Appl. Sci.* **2022**, *12*, 4195. [[CrossRef](#)]
44. Guo, H.; Li, J.; Yang, B.; Mao, X.; Zhou, Q. Green scheduling optimization of ship plane block flow line considering carbon emission and noise. *Comput. Ind. Eng.* **2020**, *148*, 106680. [[CrossRef](#)]
45. Yang, B.; Yin, W.; Li, J.; Zhou, Q. A Multitime Window Parallel Scheduling System for Large-Scale Offshore Platform Project. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 2352651. [[CrossRef](#)]
46. Li, J.; Yin, W.; Yang, B.; Zhou, Q. Research on Welding Quality Traceability Model of Offshore Platform Block Construction Process. *CMES Comput. Model. Eng. Sci.* **2023**, *134*, 699–730. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.