applied sciences

Article

A Multiproject and Multilevel Plan Management Model Based on a Hybrid Program Evaluation and Review Technique and Reinforcement Learning Mechanism

Long Wang [1,2], Haibin Liu [1,*], Minghao Xia [2], Yu Wang [2] and Mingfei Li [1]

[1] College of Mechanical and Energy Engineering, Beijing University of Technology, Beijing 100124, China; wanglong@emails.bjut.edu.cn (L.W.); limingfei@emails.bjut.edu.cn (M.L.)
[2] Capital Aerospace Machinery Co., Ltd., Beijing 100076, China; xiamh_camc@hotmail.com (M.X.); wangyu_camc@hotmail.com (Y.W.)
* Correspondence: liuhb@bjut.edu.cn

Abstract: It is very difficult for manufacturing enterprises to achieve automatic coordination of multiproject and multilevel planning when they are unable to make large-scale resource adjustments. In addition, planning and coordination work mostly relies on human experience, and inaccurate planning often occurs. This article innovatively proposes the PERT-RP-DDPGAO algorithm, which effectively combines the program evaluation and review technique (PERT) and deep deterministic policy gradient (DDPG) technology. Innovatively using matrix computing, the resource plan (RP) itself is used for the first time as an intelligent agent for reinforcement learning, achieving automatic coordination of multilevel plans. Through experiments, this algorithm can achieve automatic planning and has interpretability in management theory. To solve the problem of continuous control, the second half of the new algorithm adopts the DDPG algorithm, which has advantages in convergence and response speed compared to traditional reinforcement learning algorithms and heuristic algorithms. The response time of this algorithm is 3.0% lower than the traditional deep Q-network (DQN) algorithm and more than 8.4% shorter than the heuristic algorithm.

Keywords: multiproject; multilevel plan; PERT-RP-DDPGAO; resource planning; automatic coordination

check for updates

Citation: Wang, L.; Liu, H.; Xia, M.; Wang, Y.; Li, M. A Multiproject and Multilevel Plan Management Model Based on a Hybrid Program Evaluation and Review Technique and Reinforcement Learning Mechanism. *Appl. Sci.* **2024**, *14*, 7435. https://doi.org/10.3390/app14177435

Academic Editor: Arkadiusz Gola

Received: 1 August 2024
Revised: 15 August 2024
Accepted: 19 August 2024
Published: 23 August 2024

1. Introduction

Vigorously developing manufacturing and the real economy is conducive to increasing social wealth and better meeting people's material and spiritual needs. It is also beneficial for providing more job opportunities and maintaining social stability. Furthermore, it is beneficial for increasing government fiscal revenue, meeting the needs of the public, and ensuring social welfare and public safety. Manufacturing companies need to constantly acquire new projects and orders. Enterprises ensure sustainable development by undertaking multiple projects. A large manufacturing enterprise often undertakes dozens or even hundreds of different projects, and how to manage the production of all projects is a huge challenge faced by the enterprise.

Every production project can generate more income, win more markets, or assume more social responsibility. Therefore, from the perspective of the project management department, there are almost no unimportant projects. However, production resources are always limited. When the same resource is allocated to multiple project plans and the necessary production resources cannot be obtained as expected, a resource conflict occurs. When there are disputes and conflicts among multiple project departments over the same critical resource, coordination can be realized only by higher-level managers. Under normal circumstances, coordinating multiple projects is extremely difficult. On the one hand, it is difficult for managers to evaluate the impact on the overall project plan after some project plans have been compromised. On the other hand, only the project management department knows the time margin of a project. However, to obtain more resources, the actual margin is generally not disclosed to personnel outside the project team.

Some new manufacturing enterprises or production workshops will try their best to avoid the mixed-line production of production projects when implementing the industrial layout. By constructing independent production units and lines, the abovementioned problems can be effectively avoided. However, most manufacturing enterprises have already formed enormous fixed assets. To undertake more production projects and respond to diversified market demands, mixed-line production must be carried out, and the products of different projects must be processed on each piece of processing equipment. Moreover, considering the cost of fixed asset investment, enterprises generally find it difficult to carry out large-scale transformation and upgrading of existing production resources. In such a situation, it is necessary for enterprise managers to organize the overall arrangement of project production plans and pursue the overall interests of the enterprise. Objective management is necessary for project surplus, project schedule impact, and project priority. In reality, multiproject management relies on the experience of managers, resulting in a lack of lean management in enterprises and a loss of overall benefits.

Based on the current situation of multiproject and multilevel planning management in manufacturing firms, this paper comes up with a planning model that combines PERT and reinforcement learning algorithms, namely, PERT-RP-DDPGAO. PERT is a technique that utilizes network analysis to develop and quantitatively evaluate plans. This paper decomposes the project plan through the PERT model and incorporates a feedback mechanism, which enables the PERT model to have dynamic optimization capabilities, complete the decomposition of the project plan, and distribute the decomposed enterprise project-level plan to various production units. After these plans arrive at the production unit, they are preliminarily decomposed into various pieces of processing equipment based on process documents, equipment resources, and working hours. This paper uses a manufacturing execution system (MES) to extract resource demand plans from processing equipment plans at 7-day intervals. The resource demand plans are used as intelligent agents, cleverly using matrix calculation methods to achieve intelligent agent actions. In PERT-RP-DDPG, RP refers to resource planning. Finally, based on the results of the PERT model, the DDPG learning model is applied to achieve automatic optimization of resource demand planning, which is called DDPGAO in this paper. In addition, by using the total time difference parameter, the calculation results of the PERT model are used as inputs for optimizing the resource demand plan. The optimization results of the resource demand plan are also fed back to the PERT model, which achieves multilevel planning management from the enterprise to the workshop and from the workshop to the section. Moreover, the optimization results and response times of the DDPG model are compared with those of traditional reinforcement learning models and greedy algorithms, and the advantages of the DDPG model in dealing with discrete production problems are discussed. The DDPG algorithm has a response time 3.0% lower than that of the DQN algorithm. The DDPG algorithm has a response time that is 8.4% shorter than that of the gray search algorithm. The DDPG algorithm has a response time reduced by 19.7% compared to that of the random search algorithm.

2. Related Work

With the ever-changing market, planning and scheduling management technology is constantly evolving, from economic order quantity models, material demand planning, and manufacturing resource planning to just-in-time production, enterprise resource planning, and load-based production control theory. The development of these theories has greatly promoted planning and scheduling development and improvement. Considering that planning is the carrier of resources and costs, people pay great attention to the management of planning. It is the main line of enterprise operations. If the accuracy of the plan continues to improve, the operational capabilities of the enterprise will also improve accordingly.

The research content of this paper is the multiproject and multilevel planning management of manufacturing enterprises. Mixed production is the greatest production feature and the most prominent problem for such enterprises. Based on production practices, the main mixed production problems can be divided into two categories: mixed production problems in a specific workshop and mixed production problems in multiple workshops.

In terms of solving mixed-line production problems in a specific workshop, Pereira studied the issue of mixed-line production in assembly workshops, optimized the fluctuation in the product output rate, and successfully developed a precise branch definition algorithm [1]. Abdul Nazar and Madhusudanan Pillai also studied the mixed-line production problem in assembly workshops [2]. Their research subjects were larger in size. Therefore, scholars have developed optimization solutions based on mutation algorithms. Siala conducted classification research on heuristic models and found that heuristic algorithms for branching and selecting classes have better feedback mechanisms [3]. Sun and Fan proposed a scheduling model based on the ant colony algorithm to address the problem of mixed assembly of multiple orders in automotive assembly workshops, considering the impact of switching between orders [4]. The ant colony algorithm was used to optimize the minimization of rule breaking times and target switching situations. An integrated model based on balanced production scheduling and buffer allocation was proposed by Lopes [5]. An iterative decomposition method was used to solve the assembly mixed-line production model. A multiobjective algorithm based on free time, total duration, and idle time was proposed by Rauf to overcome multiobjective production scheduling problems [6]. A new mixed-line scheduling model based on a simulated annealing algorithm combined with total duration minimization and idle time weighting was proposed by Mosadegh et al. [7,8]. The authors used the Q-learning algorithm to optimize heuristic rules. A mixed-line planning model considering preparation time was proposed by Nazar [9]. This model focuses on the operation of the equipment. A multiobjective optimization planning algorithm was proposed by Wang [10], focusing on maximizing net profit and reducing preparation time and turnover. On account of a better particle swarm optimization algorithm, a multiobjective optimization algorithm focusing on the plan completion rate and plan change rate was proposed by Zhong [11]. Zhang used a genetic algorithm based on a cellular strategy to optimize the energy consumption and adjustment rate of production systems [12]. Manavizadeh innovatively focused on the scheduling problem of mixed linear and U-shaped assembly lines and proposed a new heuristic algorithm [13]. A new algorithm based on an integer linear programming algorithm and a hybrid genetic model considering assembly line length and the number of terminals was proposed by Defersha [14].

In terms of solving mixed production problems in multiple workshops, an accelerated dynamic programming algorithm was used by Hong to minimize switching costs for solving the painting workshop scheduling problem [15]. Leng transformed the color model of a surface treatment workshop into a Markov decision process and solved it [16]. A taboo search algorithm that considers work and cache costs was proposed by Kampker by considering both the final assembly workshop and the assembly workshop together [17]. A multiobjective integer linear programming model based on color batching load balancing and raw material balancing was proposed by Taube [18]. A hybrid weighted model and integer programming algorithm scheduling model was proposed by Wu for the multistage planning problem of a surface treatment workshop, turnover workshop, and final assembly workshop [19].

Based on the research above, special issues considered in this paper are introduced. This paper addresses another type of mixed-production situation. This situation does not consider a single workshop or a few workshops but, rather, all the production units of the enterprise. For a large enterprise, there may be more than ten or even dozens of production units. For the top management of the enterprise, the goal is to overcome the mixed production issue of all production units in the enterprise. At present, few people are conducting research in this field.

The research above reveals that scholars studying planning and scheduling algorithms mainly use heuristic algorithms, artificial intelligence algorithms, and hybrid algorithms. Heuristic algorithms include genetic algorithms [20], taboo search algorithms [21–24], particle swarm optimization algorithms [25,26], and ant colony algorithms [27].

Chen used a genetic algorithm to solve the fuzzy assembly line workshop planning problem considering resource occupancy in mixed flow shop scheduling [28]. A genetic composite algorithm was proposed by Liu to minimize energy consumption and delay [29]. A solution based on a genetic algorithm was proposed by Yu to solve the mixed-line scheduling problem of unrelated parallel machines in a workshop [30]. A two-stage hybrid scheduling model considering energy conservation was proposed by Wang [31]. Jamrus came up with the idea of combining two different heuristic models [32]. They improved the particle swarm optimization algorithm based on Cauchy distribution and incorporated the concept of a genetic algorithm. This algorithm has made significant improvements in overcoming mixed-line problems. Robotic equipment is crucial for flexible and hybrid production, and the ant colony optimization algorithm was used by Elmi to solve the scheduling issue of multi-robot, hybrid production lines [33].

An increasing number of scholars are applying artificial intelligence algorithms to solve planning and scheduling problems. Sun et al. used machine learning methods to schedule robot resources [34]. Asghari et al. combined artificial intelligence computing models with genetic algorithms for scheduling cloud computing resources [35]. Luo considered the impact of plan insertion and implemented dynamic scheduling in the workshop through machine learning [36]. Zhang et al. used graph neural networks for workshop planning and control [37]. Swarup et al. achieved results in saving computational costs by dynamically arranging cloud computing resources through machine learning [38].

Among the numerous artificial intelligence algorithms, reinforcement learning models are highly favored. Reinforcement learning (RL) can enable intelligent agents to interact with the environment and achieve automatic scheduling of plans or resources through reward and punishment mechanisms [39]. In recent years, some scholars have begun to pay attention to the management of multi-level planning systems. Zhao et al. regarded the workshop and logistics as two levels and used priority algorithms for planning optimization [40]. Wan et al. divided cloud computing resource scheduling into user-level scheduling and sub-level scheduling [41]. Manna and Bhunia treated inventory as an additional level of scheduling [42]. Meanwhile, we have also noticed that no scholars have conducted multi-level planning and scheduling research on project management level and workshop resource level planning in manufacturing enterprises.

In summary, solving the multilevel planned mixed-line production problem of enterprise production unit equipment resources is highly important, but little related research has been conducted. Additionally, we note that the main target audience for scheduling problems is managers. Managers have another set of research ideas. Tripathi and Jha have focused on the management role of performance tools [43]. Kadri and Boctor ingeniously combined time parameter calculation methods with genetic models [44]. Olivieri et al. improved workflow and resource utilization through location management methods [45]. Tripathi and Jha used success factors to model management models [46]. Habibi established a mathematical model for supply chain management [47]. These inspire us to combine management tools and models with artificial intelligence technology. This composite approach makes the new algorithm more in line with management activities, allowing artificial intelligence technology to leverage its advantages and assist managers in making decisions.

This article is based on the above discussion, combining advanced artificial intelligence technology and management tools (resource planning and project management models) to transform the behavior of managers using management tools into the behavior of machines constantly making decisions using management tools. From the research results, it can be seen that the interdisciplinary integration of computer algorithms and management tools has significant innovation in the field of scheduling algorithm research. The following will elaborate on the methods, experiments, discussions, and conclusions.

3. Method

The PERT-RP-DDPGAO algorithm includes a module framework and data acquisition, a PERT optimization model, a resource plan processing method, and an automatic optimization model based on DDPG.

3.1. Module Framework and Data Acquisition

The research object of this article is the most common machining processes in manufacturing enterprises. Mechanical processing is generally divided into small product mechanical processing and large product mechanical processing. The ordinary mechanical processing of small products has a short time cycle and can be performed using multiple pieces of equipment, generally without causing resource conflicts. Large-scale product machining generally involves medium-to-large-scale machining centers, which have high difficulty in product processing, high equipment value, and long production cycles and are prone to resource conflicts during mixed-line production. To solve practical production problems, this paper focuses on the mechanical processing of medium- and large-sized products and explains the content of the PERT-RP-DDPGAO algorithm model.

Figure 1 shows the PERT-RP-DDPGAO algorithm framework, which is divided into an enterprise planning layer, a production unit planning layer, and an equipment planning layer based on the application scenarios. At the enterprise planning level, the model obtains product structure tree information and standard operating time information from the manufacturing execution system. The PERT model takes a structural tree model and standard operating time information as inputs to form a product plan and the total time difference for each product plan. Through the PERT model, project plan decomposition is achieved, resulting in a planned dispatch from the enterprise to the production unit. After receiving the plan, the production unit forms a resource plan for equipment or workstations based on process information and work hour quota information. Through this mechanism, a planned dispatch from the production unit to the equipment is formed. The last layer is the equipment planning layer. At the device planning level, the DDPGAO model takes resource planning on mechanical processing equipment as an intelligent agent and optimizes resource planning autonomously through reinforcement learning. During the optimization of the DDPGAO model, the operation processing plan is adjusted. After changes in the process processing plan, the process processing plan is fed back to the product plan, which has an impact on the total time difference. This paper adds a feedback algorithm to the PERT model so that the results of the equipment planning layer can be fed back to the enterprise planning layer, enhancing the robustness of the entire planning system.

According to the theory of project management, manufacturing enterprises determine the composition structure of products based on the product structure tree when designing the product project organization. The composition structure of a product represents the logical sequence of processing from parts to components and from components to the final finished product. In addition, most manufacturing enterprises have established information systems that can conveniently determine the processing cycle of each product. Therefore, with knowledge of the product processing logic and processing cycle, the project can be decomposed into work. Usually, companies add standard operating times to the product structure tree. The product structure tree information is represented by B, as shown in Equation (1). In Equation (1), M represents the product name, L represents the product-level code, and the product processing logic relationship can be obtained through the product-level number. t_1 represents the standard operating time, t_2 represents the loose operating time, and t_3 represents the emergency operating time. σ_1 is the loose coefficient, usually taken as 1.3. σ_2 is emergency coefficient, usually taken as 0.8.

$$B = (M, L, t_1, t_2, t_3) \tag{1}$$

$$t_2 = \sigma_1 * t_1 \tag{2}$$

$$t_3 = \sigma_2 * t_1 \tag{3}$$

The calculation equations for the loose operation time and emergency operation time are shown in Equation (2) and Equation (3), respectively.

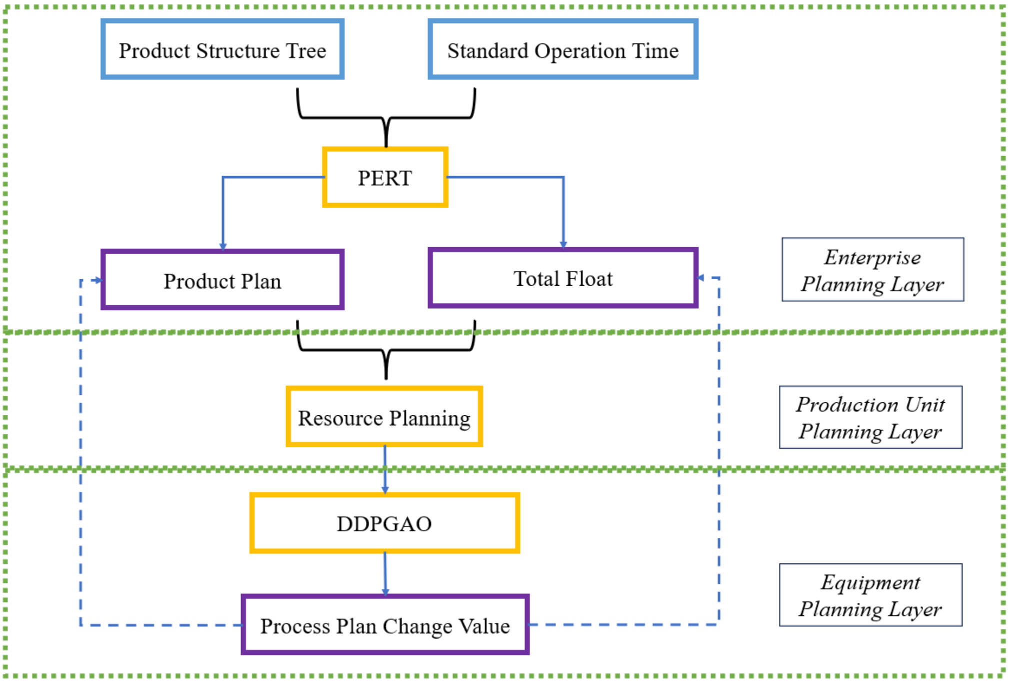

Figure 1. PERT-RP-DDPGAO algorithm framework.

3.2. PERT Optimization Model

Current product projects are very complex. If the final product is unfolded in the form of a product tree, a very complex tree-like structure is obtained, and the number of branches can reach thousands or tens of thousands. To cope with complex project management, manufacturing enterprises have developed the PERT model based on operations research theory. This model calculates key time parameters based on project task decomposition and the operation time of the task. By analyzing time arguments, elements such as the planned time, critical work, critical path, and total duration are obtained to support managers in better project management. The setting of the model needs to consider the field requirements of the information system. C must include the logical relationship expression of the task and related time parameters, as detailed in Equation (4). The nodes before and after the task are represented by d_i, and d_j represents the logical relationship between related tasks. The duration of the task is represented by m. The pre-task and post-task work nodes are assigned by P_i and p_j represents. Total float T_{ij} and task completion status w are key control parameters. w can be determined through the task handover procedure.

$$C = (d_i, d_j, T_{ij}, m, P_i, p_j, w) \tag{4}$$

The difference between the earliest start time and the latest start time is T_{ij}. For a one-to-one link, the earliest start time of point P on the left side of node i is denoted by

$t_{ES}(P,i)$ represents. The duration between node i and node j is represented by $t(P,i)$. For many-to-one links, it is ES'_{ij}. The calculation methods are shown in Equations (5) and (6).

$$ES_{ij}= t_{ES}(P,i)+t(P,i) \tag{5}$$

$$ES'_{ij}= max(t_{ES}(P,i)+t(P,i)) \tag{6}$$

Compared with Equation (6), the main logic for the latest start time is to change from the maximum value to the minimum value. Duration between node j and node k is represented by $t(j,p)$. Latest start time of the node on the right side of node j is represented by $t_{LS}(j,p)$.

$$LS'_{ij}= min(t_{LS}(j,p)-t(j,p)) \tag{7}$$

Using the principle of PERT technology, Equations (8) and (9) can be derived through Equations (5)–(7).

$$T_{ij} = ES'_{ij}-LS'_{ij} \tag{8}$$

$$T_{ij} = max(t_{ES}(P,i)+t(P,i)) - min(t_{LS}(p,k)-t(p,k)) \tag{9}$$

The backend of the algorithm program uses Python 3.5 language and the NetwordX 1.11 module. In the front-end design, the paper adopts SpringCloud. SpringCloud is able to utilize the development of Spring Boot to simplify the development of distributed system infrastructure.

To increase the robustness of the PERT model, t_3 and t_1 are subtracted from Equation (3), as shown in Equation (10).

$$t' = t_1 - t_3 \tag{10}$$

This paper incorporates a feedback mechanism into the PERT model, as shown in Figure 2.

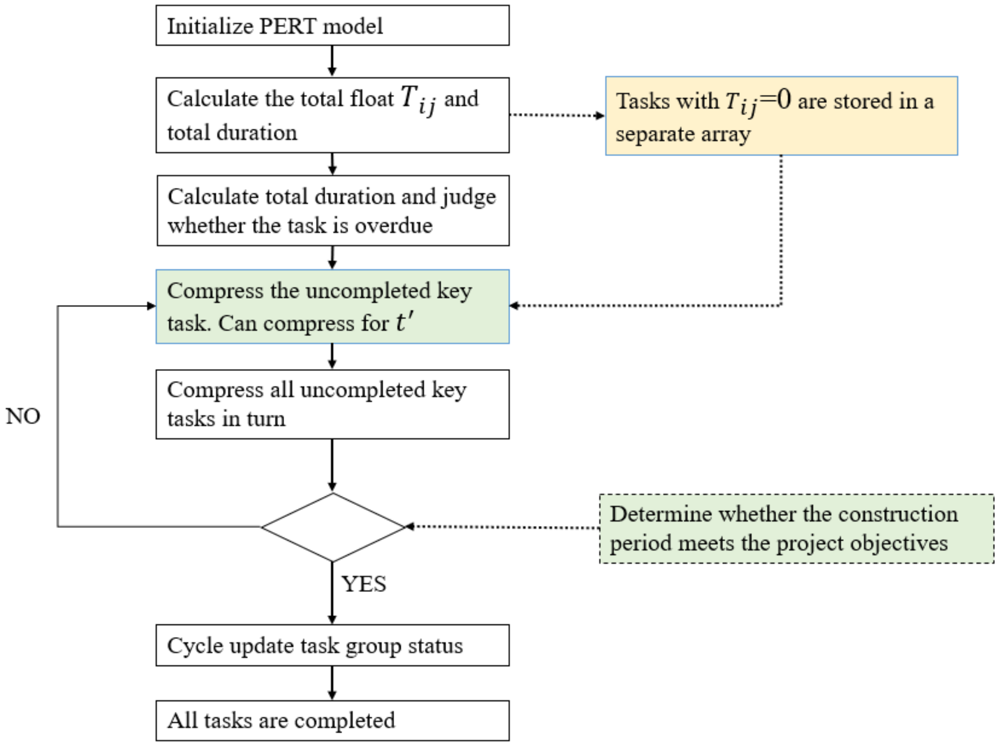
Initialize PERT model
Calculate the total float T_{ij} and total duration
Tasks with T_{ij}=0 are stored in a separate array
Calculate total duration and judge whether the task is overdue
Compress the uncompleted key task. Can compress for t'
Compress all uncompleted key tasks in turn
NO
Determine whether the construction period meets the project objectives
YES
Cycle update task group status
All tasks are completed

Figure 2. PERT model with added feedback mechanism.

The focus of the feedback mechanism is to record and monitor the total time difference. In addition, this paper introduces the emergency time in Equation (3). By using the emergency time, the compressible time of the task can be calculated. By compressing the time of key tasks, the total project duration can be shortened, ultimately achieving the goal of controlling the total project duration.

3.3. Resource Plan Processing Method

This paper achieves project plan decomposition through PERT technology. These decomposed plans can be called enterprise-level plans or production unit plans. After receiving the plan, the production unit can derive the process-level plan based on the working hour quota of each process of the product. The working hour quota for a certain product is set to T, as shown in Equation (11).

$$T = (p_1, p_2, \ldots p_n) \tag{11}$$

Although the working hour quota cannot fully represent the actual processing time of the product, it can accurately identify which processes have longer and shorter processing times and determine the proportion of time needed for each process. Therefore, through Equation (12), this paper can obtain the plan for process j. T'_j represents the planned time of process j. T_f represents the project plan time calculated by PERT, usually the end time. T_p represents the processing cycle of the product in the production unit.

$$T'_j = T_f - \left(\frac{\sum_{i=j}^{n} p_n}{\sum_{i=1}^{n} p_n}\right) \times T_p \tag{12}$$

Through T'_j, this paper can obtain the production plan of the product process. After associating production resources or equipment in the product process production plan, the resource demand plan for a certain resource or equipment can be obtained.

This paper presents the resource plan in the form of matrix X, as shown in Equation (13). Based on the actual work situation, the model retrieves equipment resource plans on a weekly basis. The horizontal column of X represents the daily processing quantity of project a on this equipment within a week. The vertical column of X represents the types of projects undertaken within a week.

$$X = \begin{bmatrix} a_1 & \cdots & a_7 \\ \vdots & \ddots & \vdots \\ z_1 & \cdots & z_7 \end{bmatrix} \tag{13}$$

Through matrix processing of resource planning, the resource plan can act as an intelligent agent for reinforcement learning, achieving automatic coordination and arrangement of resource planning. To carry out resource planning and coordination, the manager needs to calculate the total number of tasks undertaken on the equipment every day and compare the processing capacity of the equipment. The current X matrix still lacks these elements, so it cannot be used for subsequent calculations. To achieve the subsequent calculation goals, it is necessary to extend matrix X to matrix X', as shown in Equation (14). In Equation (14), A represents the total planned quantity of vertical projects, and B represents the task-carrying capacity of the equipment (the maximum quantity that can be processed on the same day). C represents the difference between the task-carrying capacity and the total number of assigned tasks.

$$X' = \begin{bmatrix} a_1 & \cdots & a_7 \\ \vdots & \ddots & \vdots \\ z_1 & & z_7 \\ A_1 & & A_7 \\ B_1 & \cdots & B_7 \\ C_1 & & C_7 \end{bmatrix} \tag{14}$$

3.4. Automatic Optimization Model Based on DDPG

Through matrix processing, resource planning can serve as an intelligent agent for reinforcement learning. Another advantage of the resource planning matrix is that it can achieve overall planning actions through matrix operations.

Therefore, this paper improves matrix X' by adding a new column and initializing it to 0 to form final resource matrix M, as shown in Equation (15). Additionally, a new action matrix N is established. The first few rows of the action matrix correspond to the planned quantity rows of resource matrix M, and each row has only one pair (+1/−1), as shown in Equation (16).

$$M = \begin{bmatrix} a_1 & \cdots & a_7 & 0 \\ \vdots & \ddots & \vdots & \\ z_1 & & z_7 & 0 \\ A_1 & & A_7 & 0 \\ B_1 & \cdots & B_7 & 0 \\ C_1 & & C_7 & 0 \end{bmatrix} \tag{15}$$

$$N = \begin{bmatrix} -1 & \cdots & +1 & 0 \\ \vdots & \ddots & \vdots & \\ 0 & & -1 & +1 \\ 0 & & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ 0 & & 0 & 0 \end{bmatrix} \tag{16}$$

Matrix N enables resource planning agents to take action through paired (+1/−1) operations. We simulate the situation where the plan for this week is called out through the eighth column of the matrix. This paper incorporates a plan adjustment constraint, which means that the plan for this week should be completed as much as possible. If it cannot be completed during the week, it must be arranged on the first day of the next week. The paper adds matrix M to matrix N_i to obtain M_i, forming an adjusted resource plan.

$$M_i = M + N_i = \begin{bmatrix} a_1 & \cdots & a_7 & 0 \\ \vdots & \ddots & \vdots & \\ z_1 & & z_7 & 0 \\ A_1 & & A_7 & 0 \\ B_1 & \cdots & B_7 & 0 \\ C_1 & & C_7 & 0 \end{bmatrix} + \begin{bmatrix} -1 & \cdots & +1 & 0 \\ \vdots & \ddots & \vdots & \\ 0 & & -1 & +1 \\ 0 & & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ 0 & & 0 & 0 \end{bmatrix} \tag{17}$$

Our goal is to train a strategy to automatically coordinate and balance resource plans. However, from matrix N, it can be seen that the action space of the resource planning matrix is relatively complex and diverse, and the combination of +1/−1 can randomly appear at any position in matrix N. It is difficult to obtain the optimal strategy using only deep reinforcement learning models. This paper proposes the use of the DDPG (deterministic policy gradient) algorithm, with the actor–critic algorithm as its basic framework, deep neural networks as approximations of the policy network and action value function, and the use of the stochastic gradient method to train the parameters of the policy network and value network models.

The DDPG algorithm framework is displayed in Figure 3. The value of proxy operations is accurately evaluated by the critical network. There will be continuous interaction between agents and the environment. This interaction is also an iterative and trial and error process. Reward r, observed state s, selected action a, and new state s' in replay buffer D are saved in the network. The agent trains the critic network from small-batch sampled data in the replay buffer. By using this training method, the difference in output from the target neural network is reduced.

$$L_i(\theta_i)= \mathbb{E}_{(s,a,r,s')\sim u(D)}\left[(r+\gamma \mathcal{Q}(s',\mu(s';\varphi^-);\theta^- - \mathcal{Q}(s,a;\theta_i))^2\right] \tag{18}$$

$L_i(\theta_i)$ is the expected value of the difference between the $\mathcal{Q}$-value of the target critic network and the $\mathcal{Q}$-value of the training critic network. $(s,a,r,s') \sim u(D)$ is the sampling through mini-batch data from replay buffer D. γ is the discount factor. θ_i is the training network parameter, and θ^- is the target critic network parameter; these parameters give the weights and bias. i is the parameter index. μ is the policy of the actor network, and φ^- is the parameter of the target actor network.

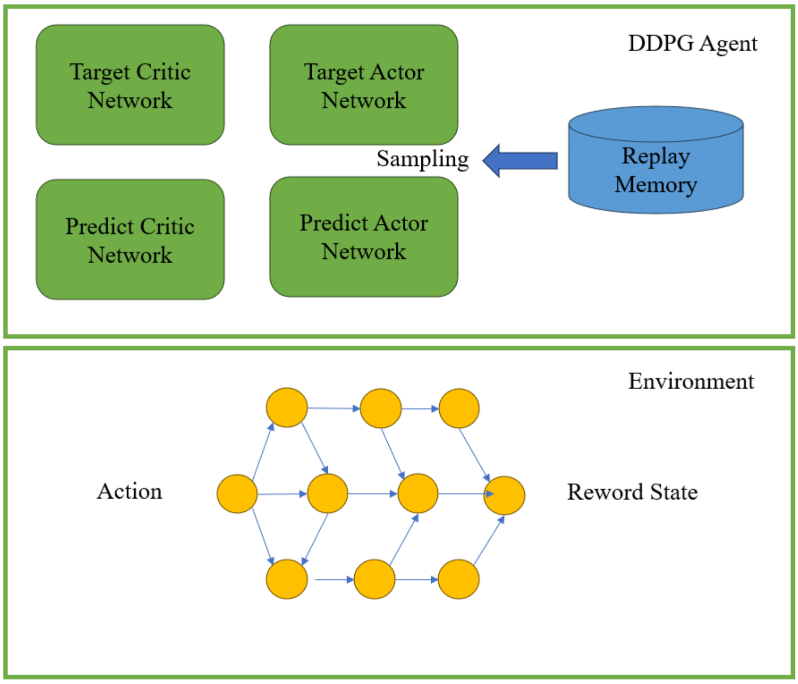

Figure 3. DDPG algorithm framework.

The actor network uses the state of the environment as the input and actions as the output to calculate the policy. The method of evaluating the policy of the actor network is to use the $\mathcal{Q}$-value, which is the output of the critic network.

$$J_i(\varphi_i)= \mathbb{E}_{s\sim u(D)}[\mathcal{Q}(s,\mu(s;\varphi);\theta)] \tag{19}$$

$J_i(\varphi_i)$ is the expected $\mathcal{Q}$-value through an action selected according to the policy, and the actor network trains to increase $J_i(\varphi_i)$. $S \sim u(D)$ denotes the states sampled from replay buffer D. In fact, even if the model uses a determined participant network learning strategy, the accuracy of the results is still questioned. An exploration process still needs to be added to the model to determine the appropriate strategy. The Gaussian noise method was used in the detection process in the paper. During the training process, Gaussian noise is added to the actions generated by the actor network, further allowing for the exploration of various actions. The reward function mainly considers four elements. First, the total sum of the vertical columns of the resource plan matrix does not exceed the capacity, indicating task completion. Second, there is a collision when negative elements or the total time difference of the task is negative. Third, the more nonzero values there are in the eighth column, the greater the penalty. Fourth, the greater the total time difference between the priority tasks, the greater the reward.

The parameters of the target network for critics and actors are updated according to a certain cycle. This update is based on the level of network training, and when the training level is determined, the new target value is also determined. In the subsequent calculation process, this value can be fixed.

4. Experimental Evaluation and Discussion

4.1. Experimental Environment Design

The multilevel planning system includes an enterprise-level plan, a workshop-level plan, and an equipment-level plan. The enterprise-level plan mainly revolves around project management. The PERT method is used in project management to decompose plans. The project management environment is mainly based on the enterprise resource planning (ERP) system. The ERP system distributes the plan to the production workshop. The workshop receives tasks through the MES and dispatches them to equipment to form a resource plan. The management of resource plans is mainly based on MES. ERP systems and MES collect, transmit, and control equipment data through industrial control networks. This paper officially runs the PERT-RP-DDPGAO algorithm through the industrial information implementation framework shown in Figure 4.

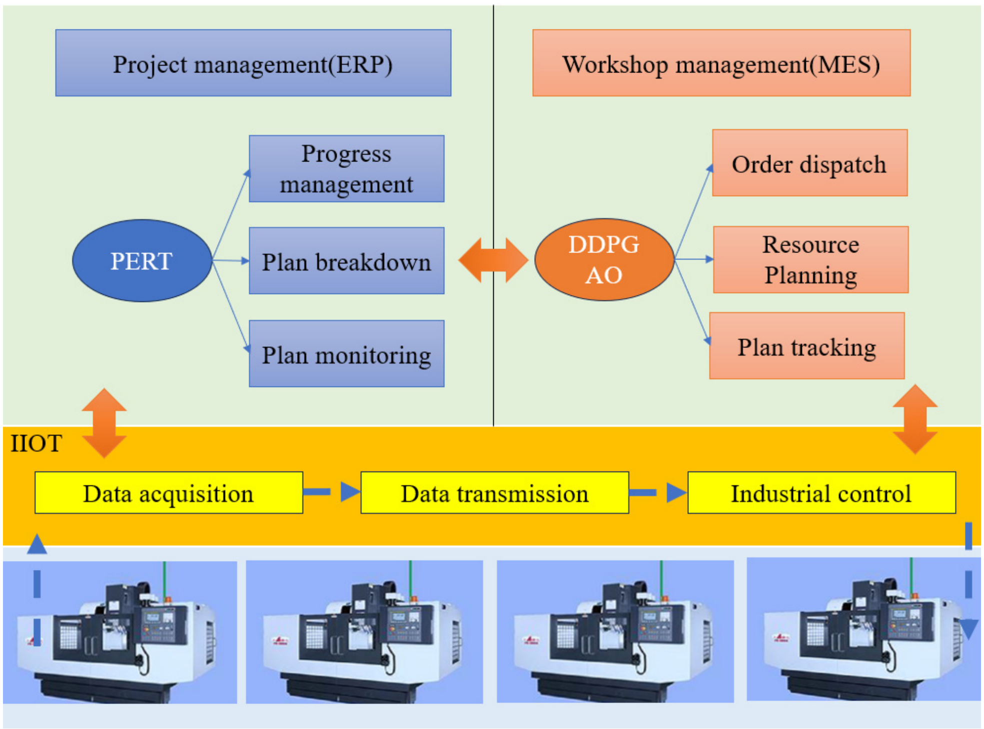

Figure 4. Industrial informatization implementation framework.

Mechanical processing is the most common processing method used in manufacturing enterprises. The main types of mechanical processing equipment used are small mechanical processing equipment, medium mechanical processing equipment, and large mechanical processing equipment. Among them, large-scale mechanical processing equipment is expensive, with a small number of pieces of equipment, and most of the time, it undertakes the processing of key and difficult products. In the actual production process, resource conflicts often occur.

Therefore, this paper focuses on the mechanical processing tasks of large-scale products in manufacturing enterprises, collects resource plans on large-scale mechanical processing equipment through production information systems, and conducts experimental analysis. The collection frequency of the resource plans is 7 days.

Through the experimental environment, the paper extracted the resource plan of a certain device and formed Table 1 using a visual approach. The outcomes of the resource planning matrix are presented in Table 1, which indicates that the paper represents three different items on the equipment in different colors, and the number of tasks for the day is calculated according to the previous formula. Then, the difference between the number of tasks and the equipment capacity is calculated to obtain the carrying capacity. If the total number of tasks is greater than the device's faculty, the image is red. If the number of tasks is less than the device's capacity, the image is green. Large-scale mechanical processing equipment is generally single-piece processing, and the equipment's capacity is based on a 12 h working system with a maximum of two product tasks to be undertaken on the same day.

Table 1. Resource plan matrix data.

Time (Day)	Project 1 Blue	Project 2 Orange	Project 3 Gray	Total Tasks	Equipment Total Tasks	Load Index
1	1	0	0	1	2	Green
2	0	1	0	1	2	Green
3	1	1	1	3	2	Red
4	1	1	1	3	2	Red
5	1	0	0	1	2	Green
6	1	1	1	3	2	Red
7	1	1	1	3	2	Red

This paper adopts visualization processing to contrast the actual performance of the model after reinforcement learning training. The preliminary extracted resource plan visualization graph is illustrated in Figure 5, which indicates the task distribution of three large-scale machining tasks over a period of 7 consecutive days. The green and red columns show that there is no conflict issue with respect to the tasks on Days 1, 2, and 5, while there is a conflict issue with respect to the tasks on Days 3, 4, 6, and 7; these issues need to be automatically adjusted through the model.

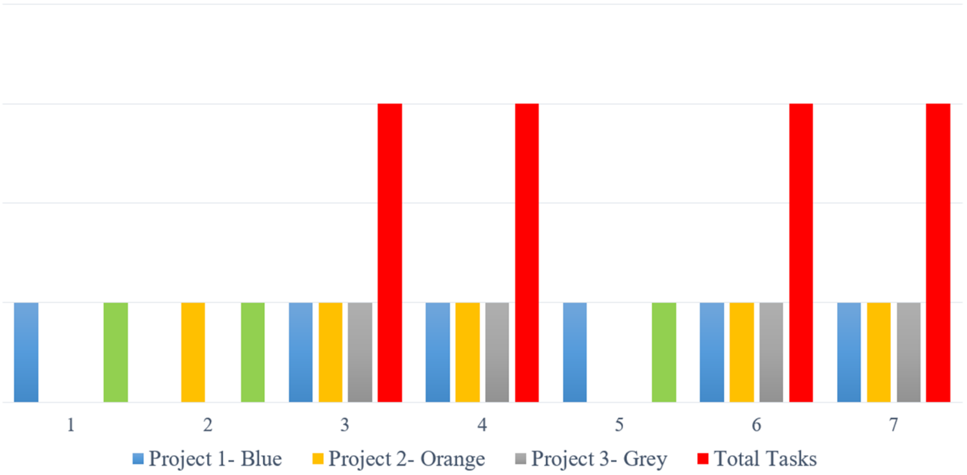

Figure 5. Visual graphics of actual resource planning. Green represents the total number of tasks that have not exceeded the maximum capacity of the device.

4.2. Experiment and Discussion on the Automatic Coordination of Resource Planning

This paper substitutes the data in Table 1 into the new model for calculation and obtains the optimized results of the model. Then, the optimized results are visualized to form Figure 6.

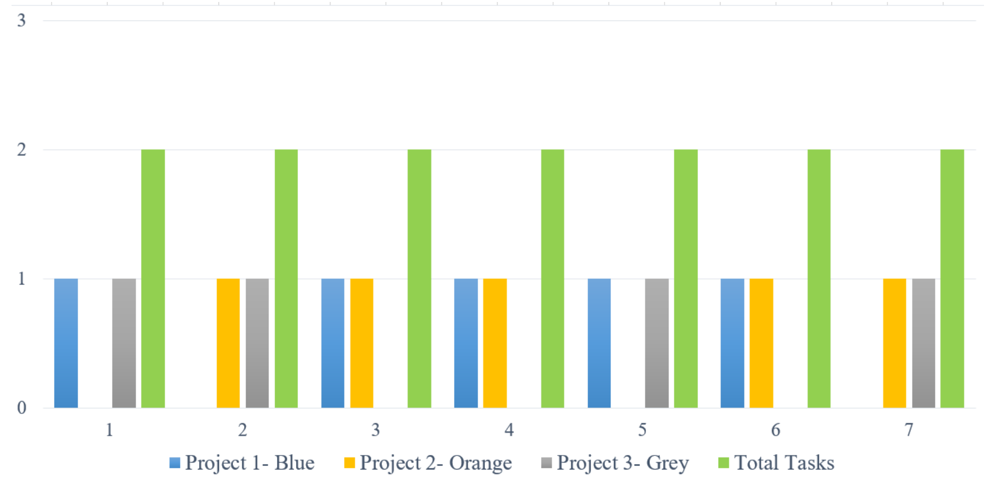

Figure 6. Resource plan after coordination.

Figure 6 shows that the model has successfully coordinated the resource plan through continuous attempts. All tasks do not exceed the maximum capacity of the device. Comparing Figures 5 and 6, this paper finds that the model moved Project 3 from Day 3 to Day 1 and moved Project 3 from Day 4 to Day 2. The main reason for this result is that our reward function stipulates that the fewer parts there are beyond our ability, the greater the total time difference of the task, and the greater the reward. Similarly, Project 3 on Day 6 was moved to Day 5. In subsequent calculations, the paper finds that the model ultimately exhibited low convergence. Therefore, we add a control function for the total time difference in the reward function. The control function of the total time difference calculates the total time difference impact of each project task. This total time difference affects the search for the corresponding project's total time difference. If the impact on the total time difference can be tolerated, there will be rewards; if it cannot be tolerated, there will be punishments. We see this punishment as "hitting a wall". Therefore, the results in Figure 6 are closely related to the setting of the reward function. The feasibility of the model is verified through the experiments in this paper.

In the process of continuous model calculation, this paper discovers another advantage of intelligent algorithms. Managers often have great confidence in their own judgments, with the main goal of solving practical problems. Moreover, intelligent algorithms may achieve leaner results. Therefore, this paper analyzes and processes resource plans with single-point conflicts for experienced managers and intelligent models, forming Figure 7; Figure 7A shows the target resource plan; Figure 7B shows the results of the manager's analysis and processing, and Figure 7C shows the results of the intelligent algorithm's analysis and processing. This paper finds that managers achieve a balance in resource planning by eliminating peaks and filling in low values in management. Managers believe that their experience plays an important role when they do not see the results of intelligent algorithms. The intelligent algorithm yields even better results. This paper requires the optimal sum of the total time difference of project tasks in the reward function. Therefore, the intelligent algorithm obtains the results in Figure 7C. These results not only achieve a balance in resource planning but also improve the total time difference between project tasks compared to Figure 7B. The greater the total time difference, the stronger and more stable the anti-interference ability of the project plan.

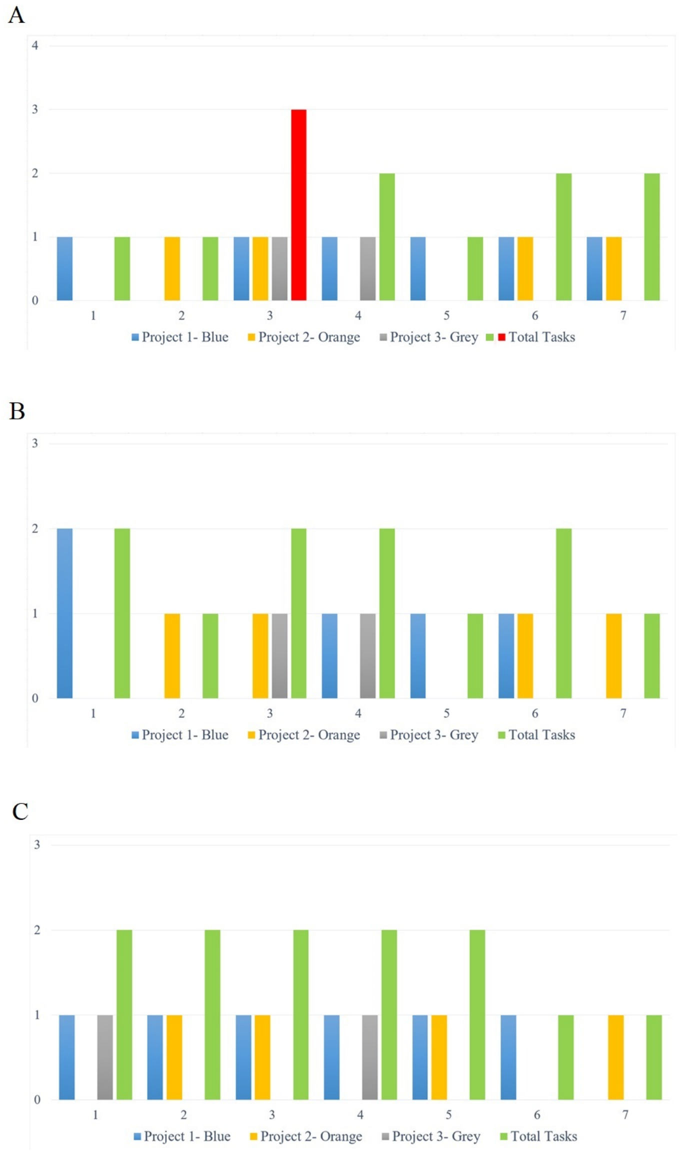

Figure 7. Comparison between the manager processing results and the intelligent algorithm processing results. (**A**) represents conflicting resource plans. (**B**) represents the result of the management personnel's overall planning of (**A**). (**C**) represents the optimization result of the computer algorithm on (**A**).

From the above, it can be seen that managers may be influenced by various factors, such as personal abilities, work environment, and work status, when balancing resources, resulting in inadequate consideration. In other words, most of the decisions made by managers are correct solutions rather than optimal solutions. In order to achieve the goal of an optimal solution, the paper proposes the PERT-RP-DDPGAO algorithm, which allows machines to solve the optimal solution of planning and coordination through self-learning. The PERT-RP-DDPGAO algorithm converts the impact of plan adjustments and the matching of resources and plans by managers into time parameters, resource planning matrices, and reward functions. By using this method, the computer can obtain the optimal solution through rigorous calculations. Of course, computers are also affected by the accuracy of input data, the operating environment, system interfaces, and other factors during calculations, resulting in biased results and risks. Therefore, the output results require manual verification by managers. The results of retesting can only be used for actual production. The computer-aided production scheduling method enables managers to make more unified and scientific decisions, gradually reducing human uncertainty factors. This is also a requirement for the development of modern enterprises.

The final core of the PERT-RP-DDPGAO algorithm lies in the DDPG algorithm section. The DDPG algorithm part is the key environment for implementing the planning and coordination function. This paper aims to further validate the value of the engineering application of DDPG. We conduct comparative experiments between the DDPG algorithm, the DQN algorithm, the greedy search algorithm, and the random search algorithm. This paper first chooses the DQN algorithm because some of the principles of DDPG are the same as those of DQN, but the actor–critic algorithm is added as the basic framework. It is hoped that the superiority of DDPG in continuous control problems can be further verified through new experimental objects. Moreover, we verify whether this algorithm can be used in resource planning and coordination application scenarios, such as whether the response speed is better. Then, heuristic algorithms are considered for comparison to verify the advantages of reinforcement learning in handling such control problems.

This paper compares the convergence and response speed of the four algorithms mentioned above, forming Figures 8 and 9.

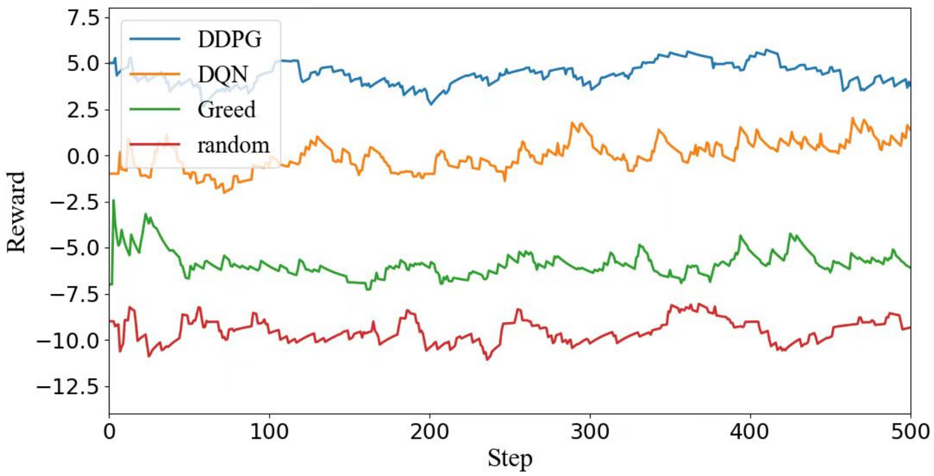

Figure 8. Convergence comparison of four algorithms.

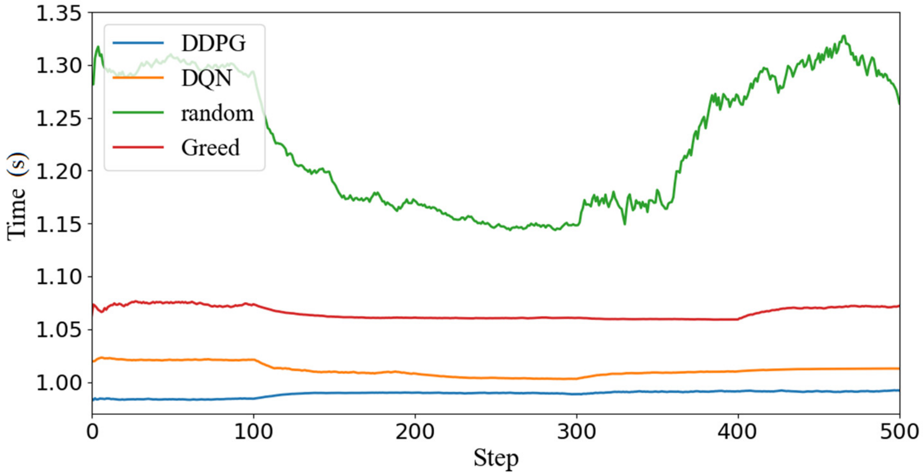

Figure 9. Comparison of the responses of the four algorithms.

The comparison results show that the DDPG algorithm outperforms the other algorithms in terms of convergence and response. The DQN algorithm is superior to heuristic algorithms. First, this paper briefly explains that the DQN algorithm deep Q-network can be used to solve continuous state space problems. The uniqueness of the DQN algorithm lies in the experience replay and the target network. When training the Q-network, experience replay can break the correlation between data and make the data independently distributed, thereby reducing the variance of parameter updates and improving the convergence speed. The use of the target network can alleviate the problem of overestimation to a certain extent and increase the stability of learning. In the application scenario of resource planning, DDPG has more advantages than DQN. In other words, DDPG has its own uniqueness in continuous control problems.

The response time is an important indicator for measuring algorithm efficiency. This paper extracts the average response time values of the DDPG algorithm, the DQN algorithm, the gray search algorithm, and the random search algorithm, shown in Table 2. This table shows that the average response time of the DDPG algorithm is greater than that of the other algorithms. The DDPG algorithm has a response time 3.0% lower than that of the DQN algorithm. The DDPG algorithm has a response time that is 8.4% shorter than that of the gray search algorithm. The DDPG algorithm has a response time reduced by 19.7% compared to that of the random search algorithm.

Table 2. Comparison table of response times for different algorithms.

Algorithm Name	Response Time (s)
DDPG algorithm	0.98
DQN algorithm	1.01
Grey search algorithm	1.07
Random search algorithm	1.22

The deep deterministic policy gradient algorithm is an optimization of the DQN that combines the idea of the deterministic policy gradient algorithm and innovatively adopts a model-free deep reinforcement learning algorithm. The dual neural network architecture is used in the DDPG algorithm architecture. Both the strategy function and value function use a dual neural network model architecture (i.e., an online network and a target network). This dual structure makes the learning process of the algorithm more stable and accelerates the convergence speed. Moreover, the DDPG algorithm introduces an experience replay mechanism; the experience data samples generated by the interaction between the actor and the environment are stored in the experience pool, and batch data samples are extracted for training. This training method is similar to the experience replay mechanism of the DQN. This mechanism can eliminate the correlation and dependency of samples, facilitating algorithmic convergence. This is also the reason why the DDPG algorithm can achieve good results in comparative analysis.

5. Conclusions

This paper proposes a new intelligent scheduling model, PERT-RP-DDPGAO. This model decomposes project plans using PERT technology. After the project plan is decomposed, a resource plan is formed, and the resource plan is trained as an intelligent agent in the DDPG model to achieve automatic coordination of multiple projects and multilevel plans in an enterprise.

The PERT-RP-DDPGAO model adds a feedback environment to traditional PERT techniques, improving the robustness of traditional algorithms. This paper studies resource planning, which has received little attention in scheduling algorithm research. For the first time, the resource plan is transformed into a resource plan matrix through the matrix formula, and the control action of the resource plan is simulated through matrix operation. After the resource plan is matrixed, the DDPG algorithm is used to achieve automatic coordination of the resource plan. After analysis, the results of automatic coordination have practical managerial implications and potential for engineering applications.

Finally, this paper conducts comparative experiments on the DDPG part of the new algorithm with the DQN algorithm, random search algorithm, and greedy search algorithm, analyzing and verifying that the DDPG algorithm is superior to the other algorithms in terms of convergence and response speed. The DDPG algorithm has a response time 3.0% lower than that of the DQN algorithm. The algorithm has a response time that is 8.4% shorter than that of the gray search algorithm. The algorithm has a response time reduced by 19.7% compared to that of the random search algorithm.

The PERT-RP-DDPGAO algorithm considers engineering applications, simplifies the complexity of the production process, and mainly focuses on the core planning parameters of project management. The core input parameters of the algorithm are planned time, time parameters, and resource capability. The planned time and time parameters are derived from the PERT algorithm, while the resource capability is obtained through manual filling. Therefore, the core of the application of the algorithm is for enterprises to have the ability to track product plans. Through practical application, it has been found that enterprises with ERP and MES systems can use this algorithm. However, this algorithm still has the following limitations: Firstly, algorithms require enterprises to have basic information technology capabilities, such as establishing ERP or MES systems that can provide real-time feedback on product progress. Secondly, the algorithm has only been applied to machining devices, and the generalization of the resource planning matrix still needs to be improved. Thirdly, the application of algorithm results also needs to be combined with management processes. The fourth issue is that the algorithm does not involve risk management.

6. Future Work

This study can be used to solve the complex scheduling problem of multilevel planning and management for multiple projects in enterprises. By applying this algorithm, automatic coordination and management of multilevel plans from the project level to the workshop level and then to the equipment level can be achieved. This algorithm is highly efficient in planning and scheduling management, does not have a high dependence on enterprise information construction, and has strong potential for engineering applications.

However, this paper mainly conducts experiments on large-scale machining equipment tasks as the main research object. For other device plans, it is necessary to optimize the resource planning matrix algorithm to further enhance the generalization of the algorithm. In the future, the research team will explore more application scenarios based on this algorithm and consider more influencing factors, such as value and quality.

In the process of conducting more practical scenario research, we will incorporate risk management content. By designing evaluation models, conduct more in-depth research on machine results and manual results.

Author Contributions: Conceptualization, L.W. and H.L.; methodology, validation, L.W. and M.X.; analysis, L.W.; resources, H.L.; writing—original draft preparation, L.W.; writing—review and editing, Y.W.; visualization, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Key Research and Development Program of China (2021YFB1716200), Research Funds for Leading Talents Program (048000514122549) and Research Project on the Development of Intelligent Manufacturing Strategy at the First Research Institute (F-Technology Committee-23250).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Minghao Xia and Yu Wang were employed by the company Capital Aerospace Machinery Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Pereira, J.; Vilà, M. An exact algorithm for the mixed-model level scheduling problem. *Int. J. Prod. Res.* **2015**, *53*, 5809–5825. [CrossRef]
2. Abdul Nazar, K.P.; Madhusudanan Pillai, V. A bit-wise mutation algorithm for mixed-model sequencing in JIT production systems. *Int. J. Prod. Res.* **2015**, *53*, 5931–5947. [CrossRef]
3. Siala, M.; Hebrard, E.; Huguet, M.J. A study of constraint programming heuristics for the car-sequencing problem. *Eng. Appl. Artif. Intell.* **2015**, *38*, 34–44. [CrossRef]
4. Sun, H.; Fan, S. Car sequencing for mixed-model assembly lines with consideration of changeover complexity. *J. Manuf. Syst.* **2018**, *46*, 93–102. [CrossRef]
5. Lopes, T.C.; Sikora, C.G.S.; Michels, A.S.; Magatão, L. An iterative decomposition for asynchronous mixed-model assembly lines: Combining balancing, sequencing, and buffer allocation. *Int. J. Prod. Res.* **2020**, *58*, 615–630. [CrossRef]
6. Rauf, M.; Guan, Z.; Sarfraz, S.; Mumtaz, J.; Shehab, E.; Jahanzaib, M.; Hanif, M. A smart algorithm for multi-criteria optimization of model sequencing problem in assembly lines. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101844.
7. Mosadegh, H.; Fatemi Ghomi, S.M.T.; Süer, G.A. Heuristic approaches for mixed-model sequencing problem with stochastic processing times. *Int. J. Prod. Res.* **2017**, *55*, 2857–2880. [CrossRef]
8. Mosadegh, H.; Ghomi, S.F.; Süer, G.A. Stochastic mixed-model assembly line sequencing problem: Mathematical modeling and Q-learning based simulated annealing hyper-heuristics. *Eur. J. Oper. Res.* **2020**, *282*, 530–544. [CrossRef]
9. Nazar, K.A.; Pillai, V.M. Mixed-model sequencing problem under capacity and machine idle time constraints in JIT production systems. *Comput. Ind. Eng.* **2018**, *118*, 226–236. [CrossRef]
10. Wang, B.; Guan, Z.; Ullah, S.; Xu, X.; He, Z. Simultaneous order scheduling and mixed-model sequencing in assemble-to-order production environment: A multi-objective hybrid artificial bee colony algorithm. *J. Intell. Manuf.* **2017**, *28*, 419–436. [CrossRef]
11. Zhong, Y.G.; Lv, X.X.; Zhan, Y. Sequencing problem for a hull mixed-model assembly line considering manufacturing complexity. *J. Intell. Fuzzy Syst.* **2016**, *30*, 1461–1473.
12. Zhang, B.; Xu, L.; Zhang, J. A multi-objective cellular genetic algorithm for energy-oriented balancing and sequencing problem of mixed-model assembly line. *J. Clean. Prod.* **2020**, *244*, 118845. [CrossRef]
13. Manavizadeh, N.; Rabbani, M.; Radmehr, F. A new multi-objective approach in order to balancing and sequencing U-shaped mixed model assembly line problem: A proposed heuristic algorithm. *Int. J. Adv. Manuf. Technol.* **2015**, *79*, 415–425. [CrossRef]
14. Defersha, F.M.; Mohebalizadehgashti, F. Simultaneous balancing, sequencing, and workstation planning for a mixed model manual assembly line using hybrid genetic algorithm. *Comput. Ind. Eng.* **2018**, *119*, 370–387. [CrossRef]
15. Hong, S.; Han, J.; Choi, J.Y.; Lee, K. Accelerated dynamic programming algorithms for a car resequencing problem in automotive paint shops. *Appl. Math. Model.* **2018**, *64*, 285–297.
16. Leng, J.; Jin, C.; Vogl, A.; Liu, H. Deep reinforcement learning for a color-batching resequencing problem. *J. Manuf. Syst.* **2020**, *56*, 175–187. [CrossRef]
17. Kampker, A.; Kreisköther, K.; Schumacher, M. Mathematical model for proactive resequencing of mixed model assembly lines. *Procedia Manuf.* **2019**, *33*, 438–445. [CrossRef]
18. Taube, F.; Minner, S. Resequencing mixed-model assembly lines with restoration to customer orders. *Omega* **2018**, *78*, 99–111. [CrossRef]
19. Wu, J.; Ding, Y.; Shi, L. Mathematical modeling and heuristic approaches for a multi-stage car sequencing problem. *Comput. Ind. Eng.* **2021**, *152*, 107008. [CrossRef]
20. Lu, P.H.; Wu, M.C.; Tan, H.; Peng, Y.H.; Chen, C.F. A genetic algorithm embedded with a concise chromosome representation for distributed and flexible job-shop scheduling problems. *J. Intell. Manuf.* **2018**, *29*, 19–34. [CrossRef]
21. Arık, O.A. Population-based Tabu search with evolutionary strategies for permutation flow shop scheduling problems under effects of position-dependent learning and linear deterioration. *Soft Comput.* **2021**, *25*, 1501–1518. [CrossRef]
22. Vela, C.R.; Afsar, S.; Palacios, J.J.; Gonzalez-Rodriguez, I.; Puente, J. Evolutionary tabu search for flexible due-date satisfaction in fuzzy job shop scheduling. *Comput. Oper. Res.* **2020**, *119*, 104931. [CrossRef]
23. Harbaoui, H.; Khalfallah, S. Tabu-search optimization approach for no-wait hybrid flow-shop scheduling with dedicated machines. *Procedia Comput. Sci.* **2020**, *176*, 706–712. [CrossRef]
24. Gmira, M.; Gendreau, M.; Lodi, A.; Potvin, J.Y. Tabu search for the time-dependent vehicle routing problem with time windows on a road network. *Eur. J. Oper. Res.* **2021**, *288*, 129–140. [CrossRef]
25. Zarrouk, R.; Bennour, I.E.; Jemai, A. A two-level particle swarm optimization algorithm for the flexible job shop scheduling problem. *Swarm Intell.* **2019**, *13*, 145–168. [CrossRef]
26. Zhao, F.; Qin, S.; Yang, G.; Ma, W.; Zhang, C.; Song, H. A factorial based particle swarm optimization with a population adaptation mechanism for the no-wait flow shop scheduling problem with the makespan objective. *Expert Syst. Appl.* **2019**, *126*, 41–53.
27. Zheng, X.; Zhou, S.; Xu, R.; Chen, H. Energy-efficient scheduling for multi-objective two-stage flow shop using a hybrid ant colony optimisation algorithm. *Int. J. Prod. Res.* **2020**, *58*, 4103–4120. [CrossRef]
28. Chen, T.L.; Cheng, C.Y.; Chou, Y.H. Multi-objective genetic algorithm for energy-efficient hybrid flow shop scheduling with lot streaming. *Ann. Oper. Res.* **2020**, *290*, 813–836. [CrossRef]

29. Liu, G.S.; Zhou, Y.; Yang, H.D. Minimizing energy consumption and tardiness penalty for fuzzy flow shop scheduling with state-dependent setup time. *J. Clean. Prod.* **2017**, *147*, 470–484. [CrossRef]
30. Yu, C.; Semeraro, Q.; Matta, A. A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. *Comput. Oper. Res.* **2018**, *100*, 211–229. [CrossRef]
31. Wang, S.; Wang, X.; Chu, F.; Yu, J. An energy-efficient two-stage hybrid flow shop scheduling problem in a glass production. *Int. J. Prod. Res.* **2020**, *58*, 2283–2314. [CrossRef]
32. Jamrus, T.; Chien, C.F.; Gen, M.; Sethanan, K. Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing. *IEEE Trans. Semicond. Manuf.* **2017**, *31*, 32–41. [CrossRef]
33. Elmi, A.; Topaloglu, S. Cyclic job shop robotic cell scheduling problem: Ant colony optimization. *Comput. Ind. Eng.* **2017**, *111*, 417–432. [CrossRef]
34. Sha, Z.; Xue, F.; Zhu, J. Scheduling strategy of cloud robots based on parallel reinforcement learning. *J. Comput. Appl.* **2019**, *39*, 501.
35. Asghari, A.; Sohrabi, M.K.; Yaghmaee, F. Task scheduling, resource provisioning, and load balancing on scientific workflows using parallel SARSA reinforcement learning agents and genetic algorithm. *J. Supercomput.* **2021**, *77*, 2800–2828. [CrossRef]
36. Luo, S. Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Appl. Soft Comput.* **2020**, *91*, 106208. [CrossRef]
37. Zhang, C.; Song, W.; Cao, Z.; Zhang, J.; Tan, P.S.; Chi, X. Learning to dispatch for job shop scheduling via deep reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1621–1632.
38. Swarup, S.; Shakshuki, E.M.; Yasar, A. Task scheduling in cloud using deep reinforcement learning. *Procedia Comput. Sci.* **2021**, *184*, 42–51. [CrossRef]
39. Silva, M.A.L.; de Souza, S.R.; Souza, M.J.F.; Bazzan, A.L.C. A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems. *Expert Syst. Appl.* **2019**, *131*, 148–171. [CrossRef]
40. Zhao, C.; Wang, S.; Yang, B.; He, Y.; Pang, Z.; Gao, Y. A coupling optimization method of production scheduling and logistics planning for product processing-assembly workshops with multi-level job priority constraints. *Comput. Ind. Eng.* **2024**, *190*, 110014. [CrossRef]
41. Wan, C.; Zheng, H.; Guo, L.; Liu, Y. Hierarchical scheduling for multi-composite tasks in cloud manufacturing. *Int. J. Prod. Res.* **2023**, *61*, 1039–1057. [CrossRef]
42. Manna, A.K.; Bhunia, A.K. Investigation of green production inventory problem with selling price and green level sensitive interval-valued demand via different metaheuristic algorithms. *Soft Comput.* **2022**, *26*, 10409–10421. [CrossRef]
43. Tripathi, K.K.; Jha, K.N. An empirical study on performance measurement factors for construction organizations. *KSCE J. Civ. Eng.* **2018**, *22*, 1052–1066. [CrossRef]
44. Kadri, R.L.; Boctor, F.F. An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *Eur. J. Oper. Res.* **2018**, *265*, 454–462. [CrossRef]
45. Olivieri, H.; Seppänen, O.; Denis Granja, A. Improving workflow and resource usage in construction schedules through location-based management system (LBMS). *Constr. Manag. Econ.* **2018**, *36*, 109–124.
46. Tripathi, K.K.; Jha, K.N. Determining success factors for a construction organization A structural equation modeling approach. *J. Manag. Eng.* **2018**, *34*, 04017050.
47. Habibi, F.; Barzinpour, F.; Sadjadi, S.J. A mathematical model for project scheduling and material ordering problem with sustainability considerations: A case study in Iran. *Comput. Ind. Eng.* **2019**, *128*, 690–710.

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.