

## Research Article

# A Business Management Resource-Scheduling Method based on Deep Learning Algorithm

Jing Wang 

Zhengzhou Vocational College of Finance and Taxation, Zhengzhou 450000, China

Correspondence should be addressed to Jing Wang; 12007008@zzcsjr.edu.cn

Received 24 April 2022; Revised 18 May 2022; Accepted 25 May 2022; Published 8 July 2022

Academic Editor: Vijay Kumar

Copyright © 2022 Jing Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As China's economic level and industrial volume continue to develop and expand, the traditional methods of business management and resource scheduling demonstrate significant limitations and misalignments with China's daily economic activities. Traditional industrial and commercial management resource scheduling is highly reliant on manual labor. When confronted with the requirements of large-scale project management, the pure manual mode cannot track the real-time progress of the project flexibly, effectively, and in a timely manner, resulting in the inability to complete the corresponding resource-scheduling work efficiently, which will affect the overall operation progress of the project. Deep neural networks have a high-application potential for solving extremely complex and highly nonlinear optimization problems. Artificial intelligence and deep learning research have advanced at a rapid pace over the past few decades, thanks to the efforts of numerous researchers. Combined with GPU technology, the deep learning framework can provide an extremely complex optimization problem with a practical and feasible optimization scheme and corresponding solution path in a very short amount of time. Therefore, this paper explores the potential application of deep learning technology to industrial and commercial resource-scheduling management. By analyzing the benefits of deep learning technology and the bottleneck issues of existing industrial and commercial resource scheduling, a real-time optimized industrial and commercial resource-scheduling model based on deep learning technology is developed. The model is evaluated using the respective data set. The test results demonstrate that the resource-scheduling model proposed in this paper has strong real-time and high-operation efficiency and can assist engineers in completing the corresponding resource-scheduling tasks.

## 1. Introduction

In recent years, science and technology have made significant strides, resulting in the development of large-scale businesses. Additionally, the traditional vertical economy has evolved into the horizontal economy, and the single-project operation company system has evolved into the multiproject cooperation company system. The traditional project management model under the operation of a single project has struggled to meet the expanding resource-scheduling requirements under the operation of multiple projects concurrently. Experts and scholars engaged in project management research and practice at home and abroad have focused on project integration management, also known as project group management, in light of this context [1–4].

With the development of project management theory, resource allocation management, which integrates strategic management and project management, has been widely used in manufacturing, *t*, communication, and other industries and achieved success. The object of traditional project management is a single project, and more than 90% of the projects are in a multiproject environment. Therefore, the transformation from project management to project group management is an important process in line with the current development trend of the construction industry and improving management efficiency. The theory of project group management has also become a hot issue in the field of strategic management and project management [5]. However, the research and application in the construction industry are still in their infancy and lack research. Therefore,

it provides a new management concept for construction enterprises, which is helpful for construction enterprises to improve their operation mode and improve their management performance.

This paper provides a brief introduction to the research background and significance of cloud computing, summarizes the research status of domestic and foreign scholars and research groups in cloud resource management and task scheduling, and explains it primarily from three perspectives: a heuristic technique for resource scheduling, reinforcement-learning resource scheduling, and deep reinforcement-learning resource scheduling [6–8]. The paper then introduces the principle of the deep reinforcement-learning model and its advantages in solving the perceptual decision-making problem of complex systems, and proposes a deep reinforcement learning-based cloud resource management and scheduling model. The paper then examines the current challenges of cloud computing, including the difficulty of making real-time, online decisions regarding resource management in a complex cloud environment. The conflict between cloud service providers who seek to reduce energy consumption and customers who seek to maximize service quality. This study proposes a cloud resource management and scheduling framework based on deep reinforcement learning to address the challenges of cloud computing by integrating the powerful perception and decision-making abilities of deep reinforcement learning.

## 2. Related Works

*2.1. Overview of Business Administration.* Business administration includes the execution or management of business operations and decision-making, as well as the efficient organization of people and other resources to guide activities toward common goals and objectives. In general, “administration” refers to the larger management function, which includes financial, personnel, and information technology (IT) services [1].

Administration is the bureaucratic or operational performance of daily office activities, which are typically internally focused and reactive rather than proactive. Administrators perform a variety of tasks to assist a business in achieving its goals. “The five aspects of administration” are the administrator’s “functions,” according to Henri Fayol (1841–1925). [3] According to Fayol, the five functions of management are planning, organizing, commanding, coordinating, and controlling. Creating output, which includes all procedures that result in the company’s sold product, is sometimes included as a sixth factor [4].

*2.2. Resource-Scheduling Method.* Almost every endeavor has a finite amount of resources or capabilities. As a result, a project’s success is determined by how well resources are handled. A resource is anything needed to successfully carry out and complete a job. Labor, equipment, supplies, facilities, or whatever else is required to finish a project [9].

Project managers can outline completion dates for activities allocated to their teams using a structured approach

to resource allocation. Customers or a board of directors can then receive reports from them. We will look at what resource scheduling is, why it is important, and how to construct an effective resource-scheduling plan in this post.

Using a structured approach to resource allocation, project managers can specify completion dates for activities assigned to their teams. They can then send reports to clients or a board of directors. In this post, we will discuss what resource scheduling is, why it is important, and how to create an effective resource-scheduling plan.

As we all know, the ultimate objective of scheduling the various types of resources is to arrange system resources in accordance with system demand and resource supply using scientific-technical methods and an acceptable allocation mechanism in order to meet system resource demand while achieving the objectives of minimum resource use and maximum efficiency. The objective of resource scheduling is to maximize resource consumption while optimizing the system. The scientific approach to resource scheduling and formulation of resource allocation are the primary tools for achieving this objective. The premise is to clarify each subproject’s resource demand, resource demand progress, and resource retention, as well as to evaluate the scheduling priority based on the project’s experts and leaders. The focus of resource scheduling is not the resource satisfaction of a project, but rather the system’s overall effect.

Resource scheduling is not only a requirement of project group management but also the central issue that the Project Group Management Office must address (PMO). Frequently, a project group includes multiple projects. There are numerous types of resources that are required for projects. Integrating all projects’ resources is a massive undertaking. All projects must have their needs, types, and demand plans clarified. Therefore, it is necessary to establish a platform for managing resource information to enable PMO managers to master resource information and project information and to formulate reasonable resource implementation plans [10–13].

In the field of engineering construction, resource scheduling has always been a difficult problem, and the research on resource-scheduling methods is also one of the difficulties. The commonly used resource-scheduling methods mainly include the following.

*2.2.1. PERT Network Analysis.* The PERT network analysis approach introduces Program Evaluation and Review Technique (PERT), a program review technology first created by the US Navy in order to plan and control the development of Polaris missiles. Polaris submarine development has been sped up by two years, thanks to PERT technology. PERT is a technique that makes and evaluates plans using network analysis. It can coordinate the many operations of the entire plan, allocate labor, material resources, time, and cash efficiently, and speed up the plan’s completion. PERT is a commonly used approach to modern planning and analysis, and it is an important tool and method of current management [14]. The PERT network analysis approach is based on the network diagram and

optimizes the network diagram's progress to adjust resource usage.

**2.2.2. Mathematical Programming.** Mathematical programming is a branch of operation research that combines early study, quick development, widespread application, and mature methodologies. It is a mathematical strategy for helping people organize scientific data. The extremal problem of objective functions under constraints is studied using mathematical theory and methodologies. It is a key branch of operations research that is employed in military operations, economic analysis, operations and management, and engineering technologies. It establishes a scientific foundation for making rational use of finite human, material, and financial resources. Mathematical programming is a method to solve the optimal answer by building mathematical models and using MATLAB and other programs. At present, in resource-scheduling problems, many algorithms (ant group algorithm, particle group algorithm, etc.) need mathematical modeling [15].

**2.3. Deep Learning-Related Technologies.** Convolution neural network, which is a special type of neural network, is a new network structure invented by Zhang et al. who is a professor at New York University, inspired by the mechanism of the biological nervous system processing visual information. CNN model shines brilliantly in the ILSVRC image recognition competition in 2012, showing its strong ability and development potential in image data processing [16]. CNNs are composed of several feature extraction stages. For each stage, there exist three different functions in it, that is, convolution, pooling, and nonlinear activation function (ReLU), with the convolution layer being the most important since it implements the procedure of information and feature extraction. The specific network structure is shown in Figure 1 below.

Traditional convolutional neural networks frequently use the original image pixels as the network's input without excessive manual preprocessing, and through the convolution layer, the convolution operation is performed. Each convolution layer contains a variety of convolution cores, adds bias, extracts the most fundamental local image features, maps them to multiple feature images, and processes the filtering output results of the convolution core with a nonlinear activation function, such as the ReLU function [17]. The pooling layer might help you to avoid using too many parameters in the network which you have to deal with. By subsampling the feature mapping of the convolution layer using the output result of the activation function as the input, retain the effective feature information, make the feature extraction have translation invariance, and reduce the influence of pixel displacement on the feature extraction. Finding the maximum of a special region of a feature map or finding the average value of a particular feature map is the approach that is heavily used in the implementation of a convolutional neural network. After deep feature extraction of image information by a multilayer

alternating convolution pooling layer, these abstract features are inferred and calculated by the full connection layer to realize the classification task.

In a particular deep learning architecture, there are many different types of layers in it, and each of those different types of layers has its own functions. Here, we will introduce those types of layers briefly.

The basic purpose of the convolutional layer is to extract features from the data with different abstract levels [18]. That is, the convolutional layers located in the low level of the whole network architecture is mainly used to learn the basic feature of an image, such as the line segment, the color, and so on. The convolutional layers located at the middle position of the whole network architecture are mainly used to learn the middle level of the feature of an image, such as the angle, the degree, and so on. The convolutional layers located at the high position of the whole network are mainly used to learn the most abstract features of an image. This type of feature often includes the face, the shape, and so on. As a convolutional layer of a neural network, it usually has the following architecture shown in Figure 2.

The convolutional layer uses the convolution operator executed on a particular image to extract the information contained in the image, and the convolution operator is implemented by the following steps:

First, we must specify the kernel size of a convolution kernel, it is often an odd number, and in most of the proposed CNN models, the kernel size is 3, 5, or 7. Since a larger size kernel often contains more parameters and we can get the same effect with a large kernel with multiple small kernels, we can just use the small kernels in the convolutional network [19].

Once we specify the kernel size of a kernel, we can execute the convolution operator on the image by equation (1):

$$\begin{aligned} \text{Similarity}(i, j) &= (I * K)(i, j) \\ &= \sum_m \sum_n I(m, n)K(i - m, j - n), \end{aligned} \quad (1)$$

where  $I$  stands for an image,  $K$  stands for a particular kernel with two-dimensional. For an image with width  $w$  and height  $h$ , if we choose  $k$  as the kernel size of the kernel  $K$ , after we do the convolution operation, the resulting feature map has the shape which satisfies the following equations (2) and (3) if we use one as the stride  $s$ :

$$F_w = \frac{W - k}{s} + 1, \quad (2)$$

$$F_H = \frac{H - k}{s} + 1. \quad (3)$$

If we need to make the shape of the feature map after convoluting the same with the input images, we can use the padding technology. That is, if we pad the input image with  $p$  pixels in each dimension, the output image for the convolution operator can be calculated by equations (4) and (5):

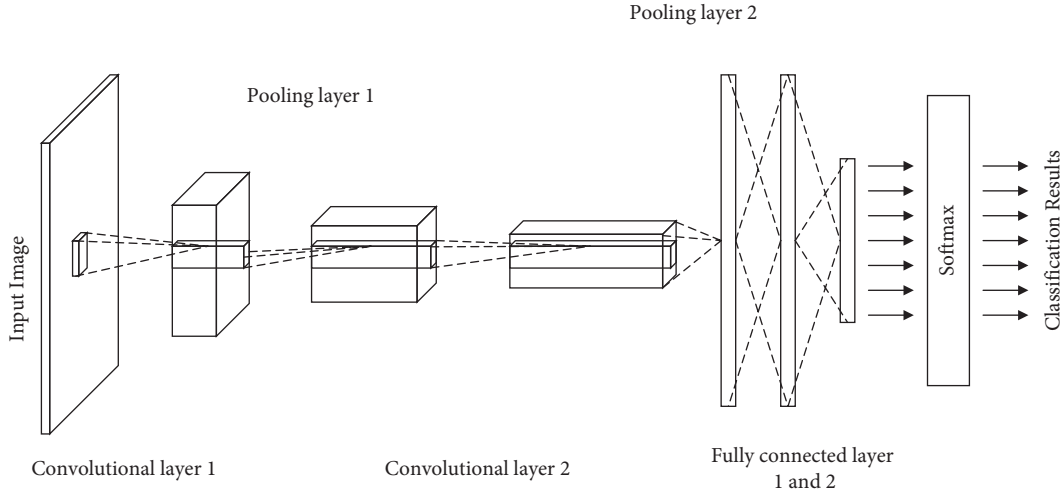


FIGURE 1: Convolutional neural network structure.

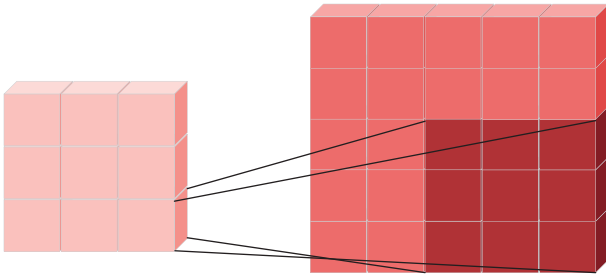


FIGURE 2: The architecture of the convolution layer.

$$F_w = \frac{W - k + 2 * p}{s} + 1, \quad (4)$$

$$F_H = \frac{H - k + 2 * p}{s} + 1. \quad (5)$$

After we complete the operation of convolution, in order to put the nonlinear property to our network, we can apply some activate function elementwise to the output of the convolution operator, and it can be formulated in equation (6):

$$a(i, j) = f(S(i, j)), \quad (6)$$

where  $f$  is the activation function we choose to use in our model, and it can be the following functions (described in equations (7) to (9)):

The ReLU function:

$$f(x) = \begin{cases} x, & \text{if } x \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

The sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (8)$$

The Leaky ReLU function:

$$f(x) = \max(0.1x, x). \quad (9)$$

Each of those can be used in any convolutional neural network.

For the pooling layer, it is often used as an  $n$  down-sampling function in our model. The most commonly used pooling technology is the max-pooling layer, and it just keeps one value for a particular region of the feature map or image, and the kept value is the biggest value of that region. The pooling procedure can be seen in Figure 3:

As can be seen in Figure 4, the fully connected layer is used to implement the function of classifying the input data for the whole neural network. The complete connection layer maps the learned feature representation to the sample's label space. The core operation of full connection is matrix-vector product (showed in equation (10)):

$$y = Wx. \quad (10)$$

The linear transition from one feature space to another is the essence of a comprehensive connection layer [20]. It is believed that each dimension of the source space influences each dimension of the target space, which is a copy of the hidden layer. Regardless of rigor, the target vector can be defined as the weighted sum of the source vector. In CNN, the full link is frequently located in the final few layers, which are used for weighting and summing the properties of the prior design. In the mist, for instance, feature engineering is equivalent to frontend convolution and pooling, whereas the entire backend serves a similar purpose as feature weighting. (Convolution is the deliberate weakening of an entire relationship.) Outside of the immediate area, there is not much of a presence, and the region is directly wiped to zero influence, inspired by the local field of vision; a small amount of coercion is also employed, and the parameters used in different regions are actually identical. Weakening reduces the parameters, saves calculation time, and focuses on local perfection; forcing further parameter reduction).

### 3. Resource-Scheduling Model Based on Deep Learning Technology

This paper transforms the multiresource-scheduling problem into a multiresource task packing problem and constructs a cloud resource-scheduling model based on deep

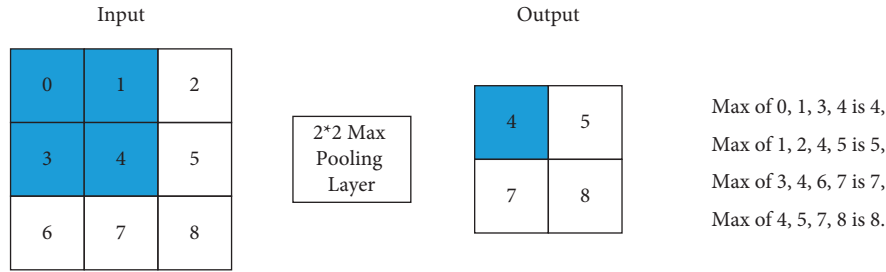


FIGURE 3: The demo of the pooling operation.

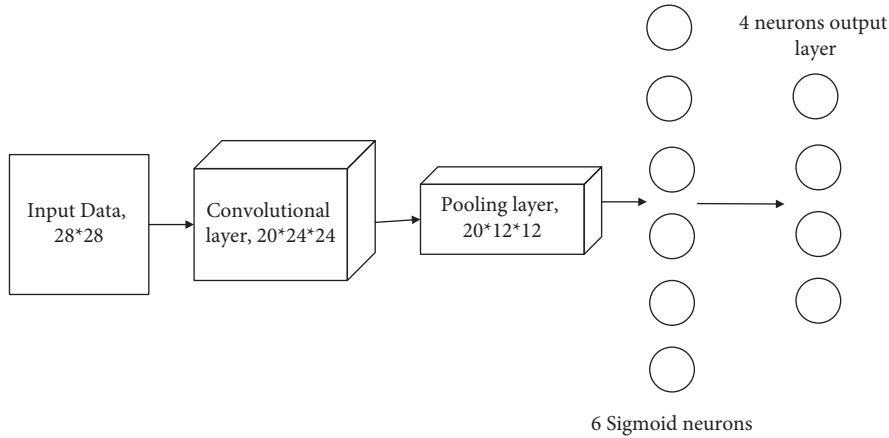


FIGURE 4: The fully connected layer in a neural network.

reinforcement learning using the deep q-network algorithm of the convolutional neural network, combined with the characteristics of cloud resource allocation and job scheduling. The system resources of the model are represented in the form of clusters, and the jobs arrive online in discrete time according to Poisson distribution. The goal is to efficiently and reasonably schedule jobs to the cluster and minimize the average slow down or average completion time of jobs. The main innovations of the proposed model are as follows: (1) the model abstracts the resource state of the cloud system into the form of an “image” and extracts the features of the resource state through the deep convolution neural network. (2) In the process of training the model, the incremental  $\epsilon - gree\ dy$  strategy selection action is adopted to strengthen the early exploration of the optimal scheduling strategy, which is conducive to finding the optimal global solution and ensuring the convergence. (3) Improve its action value evaluation method so that the agent can judge the action value more accurately and effectively, which is conducive to the agent finding the optimal scheduling strategy faster. The experimental results suggest that this learning strategy produces better outcomes and converges faster than a learning strategy based on the traditional strategy gradient method [21].

The system resources are expressed in the form of a cluster with  $d$  resource types (such as CPU and memory). The total amount of cluster CPU resources is  $R_{cpu}$ . The total amount of memory resources is  $R_{mem}$ , and the duration is  $T$ . The job arrives at the cluster job buffer in a discrete-time online manner. At each time step, the intelligent scheduler

selects to schedule one or more waiting jobs, assuming that the job’s resource requirements are known when it reaches the cluster. The resource attribute of each job is expressed as a vector  $r = (r_{i,cpu}, r_{i,mem}, \dots, r_{i,d})$ ,  $r_{i,cpu}$  indicates the number of CPU resources that job  $i$  needs to occupy,  $r_{i,mem}$  reflects the amount of RAM that the job requires, and  $t_i$  is the duration of the job  $i$ . At the same time, the cluster form is used as the collection of resources, ignoring the influence of machine fragments and other factors, but the model contains the basic elements of multiresource scheduling, which can verify the effectiveness of the deep reinforcement-learning method in the field of cloud computing resource scheduling.

**3.1. State-Space.** As depicted in Figure 5, the system’s state-space comprises the current cluster’s configuration of machine resources as well as the resource demands of jobs in the waiting queue. The cluster state denotes the configuration of resources for jobs awaiting service in the subsequent  $t$  time steps ( $t=20$ ). In the cluster state graphic, different occupations are represented by distinct colors. For instance, the red portion of the image indicates that the task requires 1 unit of CPU and 2 units of memory, with a 1-time step duration. The image of the job queue depicts the resource requirements of the jobs awaiting scheduling. For instance, job 1 requires two CPU units and three memory units, and its duration is two time steps. The Poisson distribution predicts that a certain number of job days will arrive at each time step. The job buffer backlog is used to store incoming

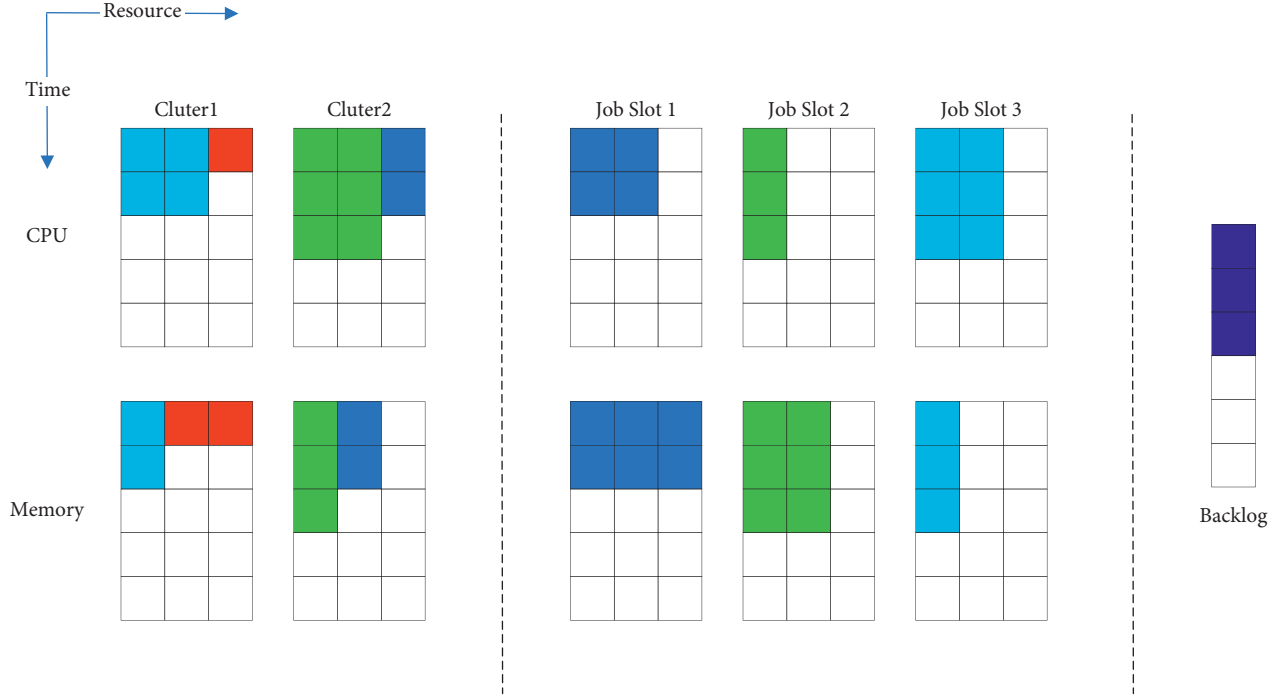


FIGURE 5: State space.

job queues that are not represented in the state space. As input to the model neural network, the system state space will take the form of a binary matrix (with the color unit represented by one and the blank unit represented by zero). Only the properties of  $M$  tasks can therefore be fixed in the state space. The remaining jobs will be overstocked in the scheduling queue's buffer backlog, awaiting transfer into the scheduling queue. This strategy simultaneously reduces the action space, resulting in a more efficient model learning process.

**3.2. Action Space.** In this model, the scheduling queue containing  $M$  jobs is represented as  $\{J_0, J_1, \dots, J_{M-1}\}$ , and  $N$  clusters are represented as  $\{C_0, C_1, \dots, C_{N-1}\}$ . Therefore, the number of action spaces is  $N * M + 1$ , which means  $\{0, 1, 2, \dots, N * M - 1, \emptyset\}$ , action  $j * M + i$  means scheduling job  $J_i$  to cluster  $C_j$ , action  $\emptyset$  means empty action, and no job is scheduled. At each time step, the scheduler will select multiple jobs from the job queue and schedule them to the cluster until the empty action or invalid action is selected. The scheduler will assign necessary resources to scheduled jobs and check the cluster resource health of the system, take out the corresponding number of jobs from the buffer and supplement them to the scheduling queue to ensure that the number of jobs in the scheduling queue remains unchanged, and update the status of the job queue until all jobs waiting for service are scheduled.

**3.3. Return Function.** The return function serves as a guide for the agent as it interacts with the environment and investigates the optimization goals. For various optimization goals, separate return functions must be defined. As

optimization targets in this work, we want to reduce average job slowness and average time to complete a job. The slow down of operation is defined as  $S_i = C_i/T_i (S_i \geq 1)$ , and  $C_i$  represents the completion time of operation  $i$ , and the corresponding return function is designed as follows in equation (11):

$$R = \sum_i^J \frac{-1}{T_i}, \quad (11)$$

where  $J$  represents all jobs waiting for scheduling that have reached the system, and  $T_i$  represents the duration of the job  $i$ .

The return function for minimizing the job completion goal can be designed as follows in equation (12):

$$R = -|\psi|, \quad (12)$$

where  $\psi$  represents the number of uncompleted jobs in the current time step system.

Throughout the training process, 100 job sets with different arrival sequences are used, and each job set contains 100 jobs. In each training round, 20 rounds of exploration were carried out on the same job set. Record the current status information  $S_t$  of all time steps in each round, the selected action  $a$ , the obtained return value  $r$ , and the next state information  $s_{t+1}$ . When the round ends, the cumulative discount return  $v_t$  obtained for each time step of the current round is calculated. In order to boost the agent's early stage of training investigation of the best strategy, the incremental  $\epsilon$ -greedy strategy is adopted to select the action (the initial value of  $\epsilon$  is 0.5, 0.9, and each training round sees a 0.001 increase.), that is, to increase the probability of agent choosing random action in the early stage of

```

(1) Set the capacity of replay memory  $D$  to  $M$ .
(2) Use random weights  $\theta$  to start the action-value function.
(3) Set up the action-value function  $Q^\sim$  that you want to use as a target with random weights  $\theta^\sim = 0$ 
(4) For each iteration:
(5)   For each jobset:
(6)     Run episode  $I = 1, 2, \dots, N$ :
(7)        $\{s_1^i, a_1^i, r_1^i, s_2^i, \dots, s_{L_i}^i, a_{L_i}^i, r_{L_i}^i, s_{L_i+1}^i\} \sim \pi_\theta$ 
(8)       Compute returns:  $v_t^i = \sum_{s=t}^{L_i} \gamma^{s-t} r_s^i$ 
(9)       For  $t = 1$  to  $L$ :
(10)        Compute baseline:  $b_t = \sum_{i=1}^N (v_t^i/N)$ 
(11)        For  $i = 1$  to  $N$ :
(12)           $\Delta r_t^i = v_t^i - b_t^i$ 
(13)          Store transition  $(s_t^i, a_t^i, \Delta r_t^i, s_{t+1}^i)$  in  $D$ 
(14)        End the for loop
(15)      End the for loop
(16)    End the for loop
(17)  Sample random minibatch of transition  $(s_t^i, a_t^i, \Delta r_t^i, s_{t+1}^i)$  from  $D$ 
(18)  Update network parameters  $\theta$ 
(19)   $L_i(\theta_i) = E_{(s,a,r,s') \sim D(M)} [(r + \gamma \max Q(s', a'; \theta_i^\sim) - Q(s, a; \theta))^2]$ 
(20)  Update  $Q^\sim = Q$ 
(21) End for

```

ALGORITHM 1: The pseudocode of the algorithm.

training and increase the exploration of action space. With the progress of training, the model continues to learn knowledge from experience, and the model begins to increase the probability of selecting the action of  $Q$  value in order to make full use of the knowledge that the agent has learned to select the best action to obtain the maximum cumulative discount return value.

The training of the model adopts the minibatch training method, from which  $M = 32$  empirical information is randomly selected, and the parameters that exist in the  $Q$  network will be updated by the algorithm of random gradient descent, in which we use 0.001 as the learning rate, and we adopted the CNN structure in Figure 1. The current  $Q$  network's parameter values are copied to the target  $Q^\sim$  Network every  $C$  training round and the target network parameters are updated once. The detailed pseudocode of the training process is shown in Algorithm 1.

#### 4. Experiment and Results

This section will compare the DQN algorithm used in this paper with the classical heuristic short job first SJF algorithm. Teris\* algorithm and strategy gradient algorithm DeepRM are in optimizing the average completion time and the average slowdown of jobs through experimental analysis. In order to ensure the fairness of the experimental comparison, the selection of our hyperparameters is consistent with that in Reference 4. The following are the outcomes of the experiment:

- (1) Figure 6 depicts the results of the experiment. The DQN algorithm converges faster and is more stable than DeepRM, reducing the average completion time of eventually converged work by 5.2% compared to DeepRM. In the first 100 rounds of training, the

DQN and DeepRM curves are relatively oscillatory and unstable, and the average work completion time is longer than with the Teris\* algorithm and the heuristic SJF method. After 200 rounds of training, the curve becomes more stable, the operation completion time is significantly less than that of the SJF and Teris\* algorithms, and it converges eventually.

- (2) Figure 7 demonstrates that as training progresses, the average total return and the highest total return attained by the agent when the scheduling operation is completed in each training round continue to rise until convergence, and the average return curve approaches and converges to the maximum return curve. In terms of convergence, it is synchronized with the job completion time curve in Figure 7, which demonstrates that the agent's return value continues to increase during the process of continuous learning and optimization of the scheduling strategy toward the goal.

As shown in the figure, the average job slowdown of short jobs scheduled with the Teris\* algorithm is significantly greater than that of long jobs, but the average job slowdown of short jobs scheduled with DQN and DeepRM algorithm is significantly less than that of long jobs. According to the research, the scheduling technique based on the Teris\* algorithm tends to make full use of effective resources in order to deploy more jobs. As a result, short jobs must wait for resource release and cannot be planned in a timely manner when the job load is reasonably substantial. According to the definition of a work slowdown given above, under the same waiting time, the work slowdown of short work is far greater than that of long work. In addition, in the

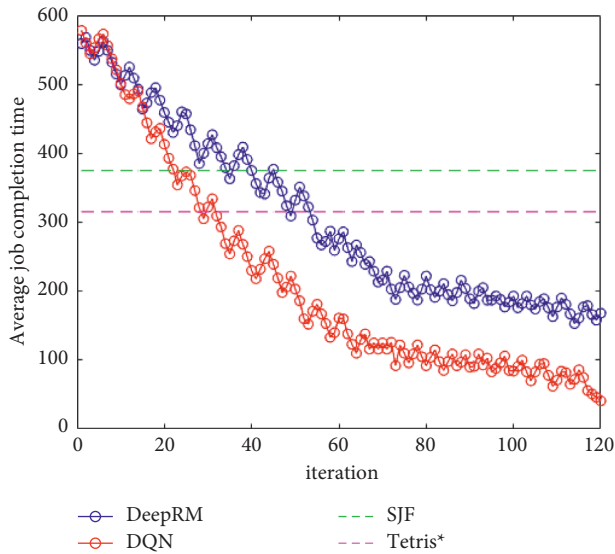


FIGURE 6: The average job completion time.

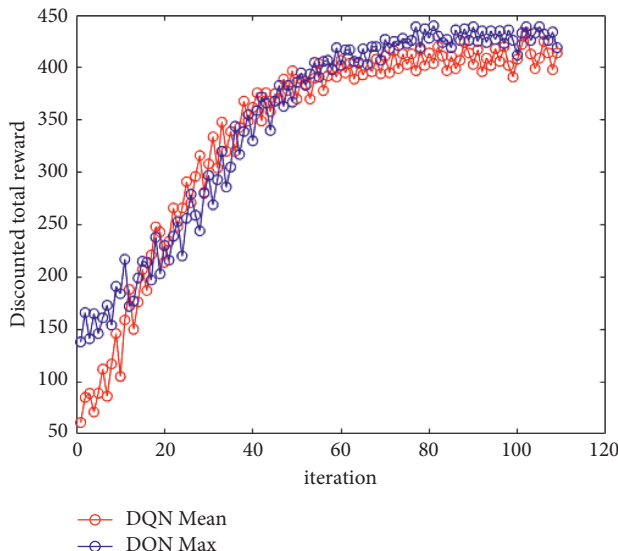


FIGURE 7: Discounted total reward.

experimental homework set, the number of short homework accounts for a large proportion, so the average slowdown of all homework is large. DQN and DeepRM scheduling strategy from experience learning to schedule strategy, if reducing the short job's slow down, it will help to reduce all job's slow down. Therefore, in the scheduling process, the scheduling strategy will be biased to allocate resources to short jobs, reduce the short job's job slowdown, and ultimately reduce the average job loss of all jobs.

## 5. Conclusion

This paper provides a brief introduction to the research background and significance of cloud computing, outlines the current state of cloud resource management and task scheduling research conducted by domestic and

international experts and research groups, and explains it primarily from three perspectives: resource scheduling based on a heuristic method, reinforcement learning-based resource scheduling, and deep reinforcement learning-based resource scheduling. The paper then introduces the principle of the deep reinforcement-learning model and its advantages in solving the perceptual decision-making problem of complex systems, and proposes a deep reinforcement learning-based cloud resource management and scheduling model. The paper then examines the current challenges of cloud computing, focusing primarily on the difficulty of online resource management decision-making in a complex cloud environment and the conflict between cloud service providers seeking to minimize energy consumption and users seeking to maximize service quality. In light of the aforementioned issues, this paper develops a cloud resource management and scheduling framework based on deep reinforcement learning to solve cloud computing's problems by integrating the high perceptual and decision-making abilities of deep reinforcement learning. Experiments demonstrate that the algorithm proposed in this study outperforms any single priority-scheduling rule and has a significantly shorter operation time than a commonly used meta-heuristic algorithm. This study's intelligent scheduling priority rules decision-making system provides technical support for appropriately guiding the scheduling process and enhances the real-time intelligence of production scheduling decisions. Future research will consider conducting additional studies on resource investment in an uncertain environment, such as uncertain resource availability, emergency order insertion, etc.

## Data Availability

The data used to support the findings of this study are available from the author upon request.

## Conflicts of Interest

The author declares that he has no conflict of interest.

## References

- [1] R. Gray and P. Bamford, "Issues in program integration," *International Journal of Project Management*, vol. 17, p. 361~366, 1999.
- [2] S. Pellegrinelli, M. Young, and D. Partington, "Attributes and levels of program management competence: an interpretive study," *International Journal of Project Management*, vol. 23, p. 87~95, 2005.
- [3] A. Pmbok, *Guide to the Project Management; Newtown Square*, Project Management Institute, Pennsylvania, 2000.
- [4] Z. Deng, C. Ma, X. Mao, S. Wang, and Niu, "Historical payoff promotes cooperation in the prisoner's dilemma game," *Chaos, Solitons & Fractals*, vol. 104, pp. 1-5, 2017.
- [5] Bo Gao, Z. Deng, and D. Zhao, "Competing spreading processes and immunization in multiplex networks," *Chaos, Solitons & Fractals*, vol. 93, pp. 175-181, 2016.
- [6] L. Xiao, "Research on the optimization of enrollment data resources based on cloud computing platform," *Review of Computer Engineer Studies*, vol. 2, no. 2, pp. 9-12, 2015.



- [7] Z. Lu and T. Shi, "Modeling and optimization of resource investment problem with activity splitting," *Computer Integrated Manufacturing Systems*, vol. 24, no. 3, pp. 602–611, 2018.
- [8] R. H. Möhring, "Minimizing costs of resource requirements in project networks subject to a fixed completion time," *Operations Research*, vol. 32, no. 1, pp. 89–120, 1984.
- [9] S. B. Rodrigues and D. S. Yamashita, "An exact algorithm for minimizing resource availability costs in project scheduling," *European Journal of Operational Research*, vol. 206, no. 3, pp. 562–568, 2010.
- [10] X. Zhu, R. Ruiz, S. Li, and X. Li, "An effective heuristic for project scheduling with resource availability cost," *European Journal of Operational Research*, vol. 257, no. 3, pp. 746–762, 2017.
- [11] S. Shadrokh and F. Kianfar, "A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty," *European Journal of Operational Research*, vol. 181, no. 1, pp. 86–101, 2007.
- [12] A. A. Najafi, T. A. Niaki, and M. Shahsavar, "A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations," *Computers & Operations Research*, vol. 36, no. 11, pp. 2994–3001, 2009.
- [13] B. Afsha R-Nadja Fi, "Multi-mode resource availability cost problem with recruitment and release dates for resources," *Applied Mathematical Modelling*, vol. 38, no. 21–22, pp. 5347–5355, 2014.
- [14] H. H. Li, Z. G. Chen, Z. H. Zhan, J. Zhang, and K. J. Du, "Renumber coevolutionary multiswarm particle swarm optimization for multi-objective workflow scheduling on cloud computing environment," in *Proceedings of the Companion publication of the 2015 conference on genetic and evolutionary computation*, pp. 1419–1420, ACM, Madrid, Spain, July 2015.
- [15] M. Dorigo, V. Maniezzo, A. Colomi, and P. Li, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.
- [16] H. Zhang, P. Li, Z. Zhou, and X. Yu, "A PSO-Based Hierarchical Resource Scheduling Strategy on Cloud Computing," in *Proceedings of the In International Conference on Trustworthy Computing and Services*, pp. 325–332, Berlin, Heidelberg, 2012 May.
- [17] S. Xue, M. Li, X. Xu, J. Chen, and S. Xue, "An ACO algorithm for task scheduling in the cloud environment," *Journal of Software*, vol. 12, pp. 466–473, 2014.
- [18] Z. Peng, D. Cui, J. Zuo, and Li, "Random task scheduling scheme based on reinforcement learning in cloud computing," *Cluster Computing*, vol. 18, no. 4, pp. 1595–1607, 2015.
- [19] Z. Peng, D. Cui, J. Zuo, and Lin, "Research on cloud computing resources provisioning based on reinforcement learning," *Mathematical Problems in Engineering*, vol. 2015, no. 9, pp. 1–12, Article ID 916418, 2015.
- [20] J. Z. Liu, Y. X. Zhang, Y. Z. Zhou, and Zhang, "Aggressive resource provisioning for ensuring QoS in virtualized environments," *IEEE Transactions on Cloud Computing*, vol. 3, no. 2, pp. 119–131, 2015.
- [21] Z. Peng, D. Cui, Y. Ma, J. Xiong, B. Xu, and W. Lin, "A reinforcement learning-based mixed job scheduler scheme for cloud computing under SLA constraint," in *Proceedings of the 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing*, pp. 142–147, CSCloud, Beijing, June 2016.