

Article

Automatic Verification Flow Shop Scheduling of Electric Energy Meters Based on an Improved Q-Learning Algorithm

Long Peng¹, Jiajie Li¹, Jingming Zhao¹, Sanlei Dang¹, Zhengmin Kong^{2,*}  and Li Ding² 

¹ Meteorology Center of Guangdong Power Grid Co., Ltd., Guangzhou 510600, China; 18825083646@gd.csg.cn (L.P.); lijiajie3@gsbb.gd.csg.cn (J.L.); zhaojingming@gsbb.gd.csg.cn (J.Z.); dangsanlei@gd.csg.cn (S.D.)

² School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China; liding@whu.edu.cn

* Correspondence: zmkong@whu.edu.cn

Abstract: Considering the engineering problem of electric energy meter automatic verification and scheduling, this paper proposes a novel scheduling scheme based on an improved Q-learning algorithm. First, by introducing the state variables and behavior variables, the ranking problem of combinatorial optimization is transformed into a sequential decision problem. Then, a novel reward function is proposed to evaluate the pros and cons of the different strategies. In particular, this paper considers adopting the reinforcement learning algorithm to efficiently solve the problem. In addition, this paper also considers the ratio of exploration and utilization in the reinforcement learning process, and then provides reasonable exploration and utilization through an iterative updating scheme. Meanwhile, a decoupling strategy is introduced to address the restriction of over estimation. Finally, real time data from a provincial electric energy meter automatic verification center are used to verify the effectiveness of the proposed algorithm.

Keywords: reinforcement learning; Q-learning; flow shop scheduling; electric energy meters automatic verification



Citation: Peng, L.; Li, J.; Zhao, J.; Dang, S.; Kong, Z.; Ding, L. Automatic Verification Flow Shop Scheduling of Electric Energy Meters Based on an Improved Q-Learning Algorithm. *Energies* **2022**, *15*, 1626. <https://doi.org/10.3390/en15051626>

Academic Editor: Francesco Bellotti

Received: 6 January 2022

Accepted: 16 February 2022

Published: 22 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A problem of scheduling is to assign scarce resources to different tasks for a certain period of time. It is a decision process and plays an important role in most manufacturing systems, as well as in the field of information processing [1]. An effective dispatching scheme can improve the production level of a company, making resource utilization more reasonable and improving the company's competitiveness in the market. There are two major aspects of the scheduling problem, namely problem modeling and solution algorithm design, where the former focuses on the system model, scheduling rules, and target functions, while the latter mainly considers algorithm complexity, algorithm convergence, and algorithm properties.

Generally speaking, a production scheduling problem can be described as follows. Considering a task that can be decomposed into several processes under some constraints, how to arrange the processing time, processing sequence, and the resources occupied by each process to achieve the best performance, are vital problems [2–5]. According to the external environment, processing characteristics, processing constraints, and minimization objectives, scheduling problems can be classified as a flow shop scheduling problem (FSP) [6] or a job shop scheduling problem (JSP). FSP is a commonly used model in the field of industry, which is also a hot issue in current academic literature [2]. The hybrid flow shop scheduling problem (HFSP) [7,8] and the permutation flow shop scheduling problem (PFSP) are two classic FSPs.

Automatic verification scheduling of the electric energy meters investigated in this paper is formulated as a flow shop scheduling problem. With the promotion of the 'centralized

inspection and unified distribution' management mode of electric energy meters, several provincial electric power companies have established their own metrological verification centers in recent years. In a metrological verification center, electric energy meter automatic verification flow shop scheduling is a common way for meter verification. Therefore, it has been a major concern to study the technology of metering automatic verification flow shop scheduling.

As the scheduling problem is an extremely difficult combinatorial optimization problem, it has been theoretically proved that a permutation flow shop scheduling problem with more than three machines is an NP-hard problem [8], and no optimization algorithm with polynomial time has been found so far. At present, the solving methods of scheduling problems have transitioned from mathematical methods to artificial intelligence methods, which can be divided into three categories: mathematical programming [9], the heuristic method [4], and intelligent optimization [10]. Mathematical programming simplifies the scheduling problem as the optimization of one or more objective functions under equality constraints or inequality constraints, which can be solved using traditional optimization algorithms, such as the branch and bound method and the pull relaxation method. Mathematical programming can theoretically provide the optimal solution to the problem; however, the searching efficiency lies at a low level so it is difficult to solve large-scale scheduling problems in a reasonable amount of time [9]. Different from the optimization algorithm, the heuristic scheduling method is based on intuitive or empirical solutions to specific problems. The NEH algorithm [11] gives higher priority to jobs with longer total processing times. From this priority, we can determine an initial arrangement, and then constructs a scheduling scheme using the insertion method. This is one of the most effective heuristic methods. Apart from NEH, there are also many other heuristic scheduling methods, such as the Palmer heuristic algorithm, Gupta algorithm, Raj algorithm, and so on. However, the heuristic algorithm is less efficient in large-scale scheduling problems. Intelligent optimization schemes adopt the genetic algorithm [12], tabu search algorithm [13], particle swarm optimization algorithm [14], and other optimization algorithms to solve the scheduling problem, which can quickly converge on the local optimal solution.

In the field of flow shop scheduling, accurate time consumption of workpieces on different machines is needed when using conventional approaches like mathematical programming. In fact, these data are usually absent in practice, and the performance of heuristic algorithms is limited due to their fixed structure, although for some specific applications there are usually some specific methods that perform well. In this case, the powerful ability of reinforcement learning to learn and explore the environment can be utilized to handle this problem well. It is more robust and there is no need for a fixed model.

At present, reinforcement learning has not been widely used in production flow shop scheduling [15–24]. Q-learning algorithm is a model-free reinforcement learning algorithm that solves large-scale optimization problems [16]. Lu et al. [17] used the Q-learning algorithm to solve this decision-making problem of energy management in a hierarchical electricity market to adjust flexible loads on the demand side. A reliable adaptive routing algorithm based on heuristic Q-learning algorithm is proposed to effectively predict the reliability of links between vehicles by dynamically adjusting routing paths through interaction with the surrounding environment [18]. In the single machine job scheduling problem, He and other scholars [20] found that agents can select better rules based on the Q-learning algorithm, which proves the effectiveness of reinforcement learning in the field of production flow shop scheduling. Some researchers also applied it to the non-permutation flow shop scheduling problem [25]. Previous studies were mainly based on the combination of reinforcement learning with other algorithms, and did not define the reward function in line with the flow shop scheduling problem. This paper focuses on solving the automatic verification flow shop scheduling problem of electric energy meters using single agent reinforcement learning, and puts forward a reward function suitable for the scheduling

problem considering some reality-based constraints. The contributions of this paper are as follows:

- This paper focuses on solving the automatic verification flow shop scheduling problem of electric energy meters using a single agent reinforcement learning algorithm.
- This paper puts forward a reward function suitable for the scheduling problem considering some reality-based constraints.
- This paper proposes a policy that decouples the selection and evaluation process in standard Q-learning, thus prevents the overoptimistic value estimates.

The following contents are arranged as follows. Section 2 provides the mathematical model of the automatic verification flow shop scheduling problem of electric energy meters, and presents the final scheduling goal, which denotes the maximum processing time. Section 3 provides the automatic verification flow shop scheduling algorithm of electric energy meters based on Q-learning, and puts forward a suitable reward function. In Section 4, a group of examples of a provincial automatic verification center of electric energy meters are simulated and analyzed to verify the effectiveness of the algorithm.

2. Problem Formulation

In order to improve the efficiency of the automatic flow verification for smart meters, a dispatch center will put different batches of the same type of measuring instruments (single-phase meters) or different types of instruments with the same voltage level (single-phase meters and terminals) on one line for verification, according to various factors, including user demand, smart inventory reserves, and verification cycle. Specifically, the scheduling problem can be described as the following optimization problem: assuming that there are n electric energy meters, which need to go through a total of m verification procedures. The verification sequence for different electrical energy meters is identical. Given the verification time of each batch of electric energy meters in different verification procedures, finding an optimal verification sequence so that the scheduling target is completed with high efficiency is required. Generally, several assumptions are needed for this kind of optimization problem:

- (1) One electric energy meter can only participate in one verification procedure at a specific time;
- (2) There is only one machine in each verification procedure, and only one equipment can be verified at a time;
- (3) Once a certain verification procedure for an electric energy meter starts, it cannot be stopped;
- (4) The verification sequence of electric energy meters in different verification procedures is identical;
- (5) It is assumed that the buffer zone between different procedures is infinite.

Under the above assumptions, the mathematical description of the scheduling problem for electric energy meters automatic flow verification can be summarized as follows: assuming that there are n electric energy meters and m verification procedures. All equipment should be verified in the order of procedure 1 to m . All electric energy meters should be ranked as $\pi = [\pi_1, \pi_2, \dots, \pi_n]$. The processing time of electric energy meters i on verification procedure j is denoted as p_{ij} , and $C(\pi_i, j)$ is the verification completion time of the workpiece π_i on verification procedure j , that is:

$$\begin{cases} C(\pi_1, 1) = p_{\pi_1, 1} \\ C(\pi_i, 1) = C(\pi_{i-1}, 1) + p_{\pi_i, 1} \\ C(\pi_i, j) = \max(C(\pi_{i-1}, j), C(\pi_i, j-1)) + p_{\pi_i, j} \end{cases} \quad (1)$$

Under the above constraints, the scheduling objective of this paper is to minimize the maximum processing time, $C_{\max}$:

$$\min C_{\max} = \max(C(\pi_i, m)) \quad (2)$$

where the maximum processing time, $C_{\max}$, refers to the completion time of the last verification procedure of the last meters. The minimization of $C_{\max}$ means that the production cycle of the verification unit is minimized, that is, the utilization rate of the unit reaches its maximum.

3. Scheduling Algorithm Based on Q-Learning

Reinforcement learning is a key technology in artificial intelligence, which has a strong real-time learning performance. The main feature of reinforcement learning is its interaction with the environment. Q-learning represents a classic reinforcement learning algorithm. Reinforcement learning is based on the mathematical model of Markov decision processes (MDPs). An MDP usually consists of an agent, a state space, an action space, a state transition matrix, a reward function, and a discount factor [17].

The agent is denoted as the main body that makes decisions. In the automatic verification and scheduling process of electric energy meters, the machine corresponding to the first verification process is usually regarded as the agent.

State represents the current environment. In the automatic verification and scheduling process of electric energy meters, the state refers to the energy meters that has not been verified, where 0 and 1 are used to indicate whether to start verification. For example, suppose there are nine electric energy meters that need to undergo three verification procedures, the number of the energy metering devices that have been verified at a certain time is 1, 2, 4. And then the current status is marked as 001011111. There are 2^n states in total for n electric energy metering devices. The state space is denoted as $S = \{s_1, s_2, \dots, s_{2^n}\}$.

Action refers to the decision made by the agent. In this problem, the action of the agent is to select a device from the energy metering devices that has never been verified to start the verification. Therefore, in this problem, $A = \{a_1, a_2, \dots, a_n\}$ denotes the selectable action set.

Reward refers to the reward value returned by the environment to the agent after the agent performs an action. In previous works, the reciprocal of the maximum completion time, $C_{\max}$, and the reciprocal of the maximum idle time were often selected as the reward function. This paper proposes a new reward function based on the scheduling problem, called an *sigmoid* – $C_{\max}$ index, as follows:

$$r(\pi) = \frac{e^{-\lambda(C_{\max} - C_{\max}^0)}}{1 + e^{-\lambda(C_{\max} - C_{\max}^0)}} \quad (3)$$

where $C_{\max}$ is the maximum completion time of a certain scheduling sequence, $C_{\max}^0$ is the lower limit of an estimated maximum completion time, and λ is the normalization factor. Compared with the pure reciprocal of the maximum completion time, $C_{\max}$, the new reward function proposed in this paper is more suitable for actual scheduling problems. For example, for a certain scheduling problem, the maximum completion time, $C_{\max}$, obtained by multiple scheduling processes is between 200 and 250. If the reward function is based on the reciprocal of $C_{\max}$, the pros and cons of different scheduling strategies cannot be properly distinguished. On the contrary, based on the *sigmoid* – $C_{\max}$ index proposed in this paper, the pros and cons of different scheduling strategies can be clearly distinguished, which is conducive to the training process of the agent.

The reward is the sum of all rewards from the current moment to the end of a round. In reinforcement learning, a discounted return is often used as the assessment of the current state. The discounted return is defined as:

$$u_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots + \gamma^N r_{t+N}, \quad (4)$$

where $\gamma \in [0, 1]$ is called the discount factor, and at time $t + N$, the round ends.

In reinforcement learning, the state of the agent at time t is denoted as $s_t (s_t \in S)$. According to the state and the agent's internal strategy function, the agent selects an action $a_t (a_t \in A)$ to get the state at the next moment, s_{t+1} , and the instantaneous reward,

r_t . According to this cycle, the agent continuously interacts with the environment, and constantly adjusts its behavior according to the rewards obtained by learning, so as to obtain the largest long-term reward. The specific process is shown in Figure 1.

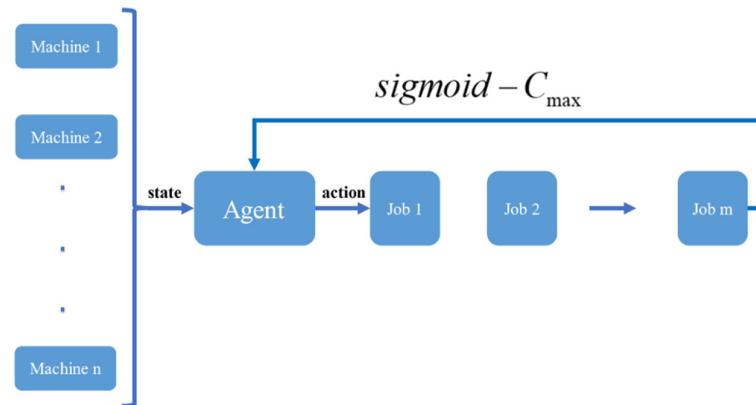


Figure 1. Algorithm flow chart.

In the Q-learning algorithm, the key part is the iteration of the Q-value. Finding the mathematical expectation of u_t , one can obtain the Q-value as:

$$Q_{\pi}(s_t, a_t) = E_{S_{t+1}, S_{t+2}, \dots, S_{t+N}}[u_t], \quad (5)$$

where S_{t+N} is the state at the end of the round at time $t + N$, which is 00000000 in the case of nine electric energy meters undergoing three verification procedures.

The Q value iteration formula of Q-learning is:

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max(Q_{a_{t+1}}(s_{t+1}, a_{t+1}))), \quad (6)$$

where α represents the learning rate and γ is the discount factor. The greater the learning rate is, the greater impacts the return has on the current $Q(s_t, a_t)$. The greater the discount factor, γ , is, the greater the agent's emphasis on future rewards will be. Although $\max(Q_{a_{t+1}}(s_{t+1}, a_{t+1}))$ can make the Q value quickly converge on the possible optimal value, it will also lead to over estimation, which makes the estimation value deviate significantly from the true value. To solve this problem, a decoupling strategy is adopted to decouple the selection of target Q-value action and the calculation of target Q-value. In other words, the largest Q-value of each action is not found directly in the target Q network, but the action corresponding to the largest Q-value is found in the current Q network. This decoupling strategy can be described by:

$$a^{\max}(S_t, r_t) = \operatorname{argmax}_{a_t} Q^c(\phi(S_t), a_t, r_t) \quad (7)$$

Then, chosen action $a^{\max}(S_t, r_t)$ can be used to calculate the target Q value in the target network, which yields:

$$Q(s_t, a_t) = (1 - \alpha)Q^t(s_t, a^{\max}(S_t, r_t)) + \alpha(r_t + \gamma \max(Q^t_{a_{t+1}}(s_{t+1}, a^{\max}(S_{t+1}, r_{t+1})))) \quad (8)$$

In Equations (7) and (8), Q^c and Q^t are the selection and evaluation, respectively, which are decoupled to prevent overoptimistic value estimates. As for the action selection strategy, the agent has two different modes: exploration and utilization. The ε greedy strategy is one of the common strategies for the agent to select its action. The agent calculates the account immediately with probability ε , and selects the action with the greatest value with the probability $1 - \varepsilon$. The smaller the ε is, the more this strategy emphasizes utilization, which means using existing successes to select actions, and the larger the ε is, the more

this strategy emphasizes exploration. In the proposed strategy, the ratio of exploration and utilization is gradually updated by gradually decreasing ϵ .

Based on the above analysis, this paper proposes an automatic verification scheduling algorithm of electric energy meters based on an improved Q-learning (IQL) method. The specific flow of the Algorithm 1 is shown below.

Algorithm 1 Automatic verification scheduling algorithm of electric energy meters based on improved Q-learning

Input: Processing time matrix $\{p_{ij}\}$
Output: $Q(s, a)$
Initialization: $\epsilon = 1, \alpha, \gamma, ep, Q(s, a)$
For each episode:
 Initialize state $s_0 = 111 \dots 1, \pi = [], r_t = 0$
 $\epsilon = (1 - ep)\epsilon$
 While $s_t \neq 000 \dots 0$:
 Current state is s_t
 If $\text{rand} < \epsilon$:
 Randomly choose an optional act a_t ;
 else:
 Choose the act a_t with the maximum value
 Act a_t , and obtain the next state $s_{t+1}, \pi = [\pi, a_t]$
 If $s_{t+1} = 000 \dots 0, r_t = r(\pi)$:
 Calculate $a^{\max} Q(s_t, r_t) = \arg\max_{a_t} (\phi(S_t), a_t, r_t)$
 Update $Q(s, a)$ as follow:
 $Q(s_t, a_t) = (1 - \alpha)Q(s_t, a^{\max}(S_t, r_t)) + \alpha(r_t + \gamma \max_{a_{t+1}} (Q_{a_{t+1}}(s_{t+1}, a^{\max}(S_{t+1}, r_{t+1}))))$
 Update $s_t = s_{t+1}$
 Else:
 Stop criterion and output the result

The time complexity of the proposed algorithm depends on the number of iterations (N) and the state space and action space dimensions. In the flow shop scheduling problem, the state space and action space are fixed, which are the number of workpieces (n) and the number of machines (m), respectively. Therefore, the algorithm complexity is $O(Nnm)$.

4. Experimental Results

In order to verify the effectiveness of the proposed algorithm, this paper uses a case provided by a provincial electric energy meter automatic verification center to carry out the test. Specifically, the case requires the verification of nine different batches of electric energy meters. Each batch has to go through the verification of the five major processes, the appearance, pressure test, multi-function test, sorting test, sealing test, and automatic labeling test. The time consumed in the different verification processes is different for different batches. The corresponding time consumption of the specific verification process is shown in Table 1.

Based on the improved Q-learning algorithm proposed in Section 3, this paper formulates an electric energy meter automatic verification flow shop scheduling strategy, and analyzes the effects of different parameters in the algorithm on the scheduling results. This algorithm is implemented using python 3.8 on an Intel (R) Core (TM) i5-8400 CPU @2.80 Ghz 2.81 GHz (RAM 8.00 GB), and the program run time is 442 milliseconds. Generally speaking, since the completion time in the reinforcement learning process decreases until it converges on the optimal solution as the number of iterations increases, the average maximum completion time in the learning process is usually used as an evaluation index, which can reflect the influence of parameters on the final result well. A small average maximum completion time means a fast convergence speed and good algorithm performance, which

also corresponds to better algorithm parameters. Therefore, this paper adopts the average maximum completion time, $C_{\max}$, as the evaluation index of the algorithm performance.

Table 1. Job verification time of each step.

Job/Step	Step1/min	Step 2/min	Step 3/min	Step 4/min	Step 5/min
Job 1	12	13	20	11	10
Job 2	12	14	16	23	18
Job 3	14	16	22	17	17
Job 4	17	12	16	20	16
Job 5	21	20	15	24	24
Job 6	14	12	23	22	21
Job 7	14	14	20	20	26
Job 8	20	22	18	20	18
Job 9	10	24	19	13	18

Since the number of iterations of the flow shop optimization scheduling algorithm for different parameters to converge on the corresponding optimal solution is different, $C_{\max}$, here, is the average time of the algorithm converging on the global solution under corresponding parameters within 5000 iterations. In order to show the convergence speed of the algorithm under different parameters, this paper also records the number of iterations for the improved algorithm to reach the optimal solution under the same parameters, as shown in Figure 2.

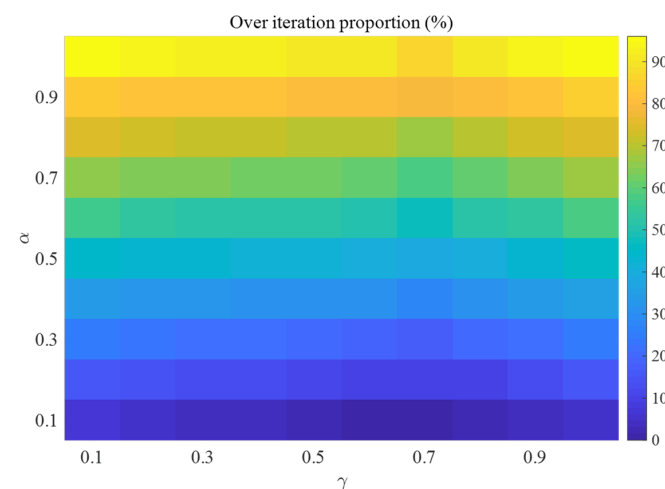


Figure 2. Influence of different parameters on the convergence iteration times of the IQL algorithm.

The least number (1089) of iterations among all parameters was used as the corresponding percentage of zero over iteration value (ZOIV), and 5000 was used as the corresponding percentage of full over iteration value (FOIV). The number of over iterations, N , is defined as the times the algorithm continued to iterate until 5000 when the algorithm reached the optimal solution under the corresponding algorithm parameters. Over iteration proportion (OIP) is defined as:

$$\text{OIP} = 1 - \frac{N}{\text{FOIV} - \text{ZOIV}} \quad (9)$$

It can be seen that the smaller the OIP is, the faster convergence speed will be under the corresponding algorithm parameters. Figure 3 shows the average maximum completion time, $C_{\max}$, before and after the algorithm is improved when one parameter takes the optimal value and another parameter changes. It can be seen from Figure 3 that the average maximum completion time of the improved algorithm is relatively smaller than the original algorithm, which means that the convergence speed of IQL is faster than that

of QL. Figure 3 also shows that algorithm parameter α has a greater impact than γ on the results. In addition, the improved algorithm is more robust to the parameters.

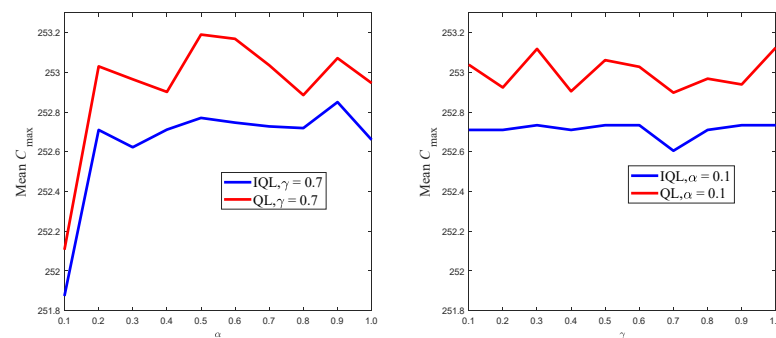


Figure 3. Comparison of impacts on average finishing time of different parameters, before and after the improvement.

According to Figures 2 and 3, the paper finally determined the optimal parameters and selected a learning rate of $\alpha = 0.1$, discount factor of $\gamma = 0.7$, and greedy rate iteration parameter of $ep = 10^{-4}$, which can determine the iterative process of the optimal solution of the calculation example. Figure 4 shows the iterative process of the optimal solution under different algorithm parameters, before and after the algorithm is improved. The orange line is the iterative process of the original QL algorithm under the optimal parameters. The blue line is the iterative process of the IQL algorithm under the optimal parameters. The red line is the iterative process of the IQL algorithm under another set of parameters, from which it can be seen that the algorithm shows a slower convergence speed. The algorithms all converge on the global optimal solution, before and after the improvement, but the improved algorithm shows a faster convergence speed.

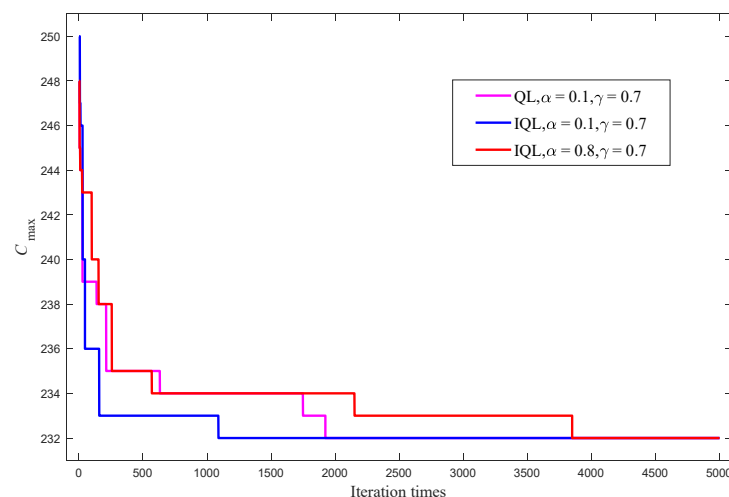


Figure 4. Criterion step of optimal solution under different parameters, before and after the algorithm improvement.

For the above example, IQL and QL are compared using the same number of iterations. The results of 10 independent runs are shown in Table 2.

Table 2. Comparison of IQL and QL.

Time	1	2	3	4	5	6	7	8	9	10
IQL	232	233	232	232	233	232	232	232	232	232
QL	234	233	232	234	234	234	234	234	233	234

It can be seen from Table 2 that the IQL algorithm finds an optimal solution 8 times out of 10 independent runs. On the other hand, it can be seen from Figure 4 that, even if the optimal solution is found at the same time, the speed of the IQL algorithm is faster than that of the QL algorithm. Therefore, the average performance of the IQL algorithm is better than that of the QL algorithm.

The Gantt chart of the optimal solution obtained by the improved algorithm under suitable solution parameters is shown in Figure 5. The optimal solution of this example is (2,7,4,9,5,6,8,3,1), the maximum completion time $C_{\max}$ is 232.

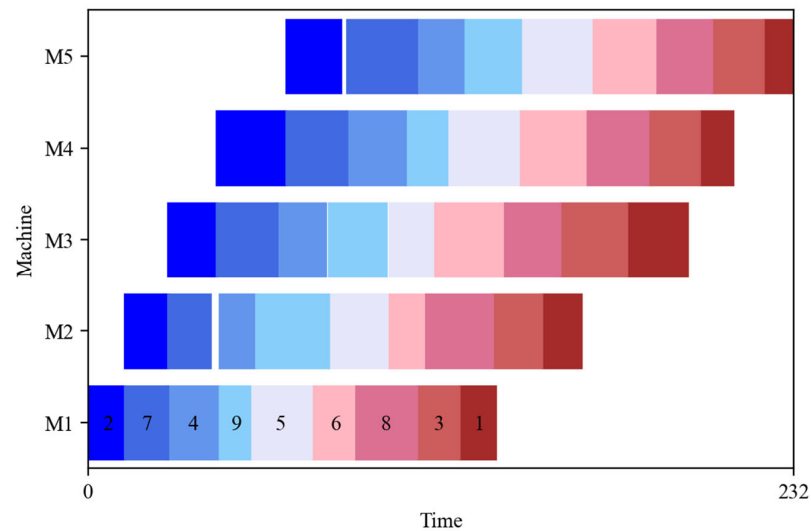


Figure 5. Gantt chart of the optimal solution.

For comparison, the optimal solution of the QL algorithm is shown in the Figure 6.

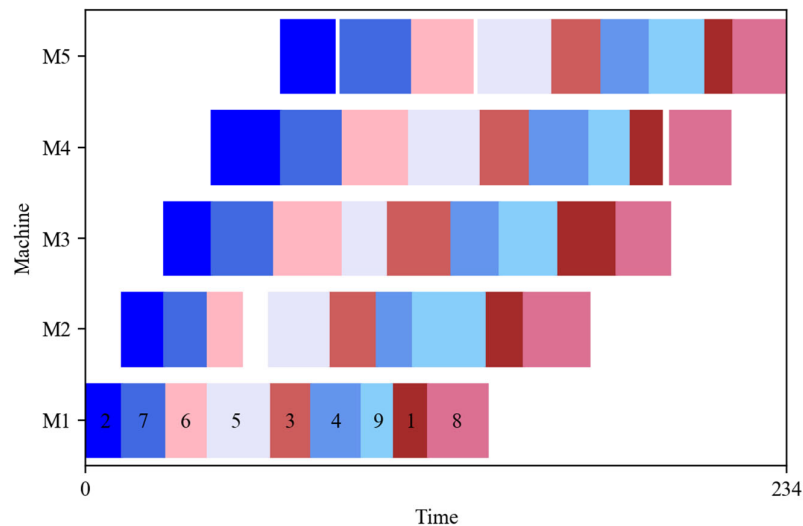


Figure 6. Gantt chart of the optimal solution based on the QL algorithm.

5. Conclusions

In this paper, an improved reinforcement learning strategy based on the Q-learning algorithm is proposed to solve the flow shop the scheduling problem of electric energy meters automatic verification. Based on the traditional Q-learning algorithm, this paper analyzes the state, action, and reward function of the flow shop scheduling problem for electric energy meter automatic verification. A new reward function named the *sigmoid* – $C_{\max}$ index is proposed, and the combination of exploration and utilization by the iterative up-

dating of ε has improved the effectiveness of the strategy. In addition, a decoupling strategy is introduced to solve the problem of overestimation. Finally, the proposed algorithm is verified using the case of automatic flow verification provided by a provincial verification center. Through parameter analyses and reasonable selection, the optimal solution of the verification strategy is obtained, which further demonstrates the convergence, effectiveness, and robustness of the proposed algorithm.

Author Contributions: Conceptualization, L.P. and J.L.; methodology, L.P.; software, J.L.; validation, L.P., J.L. and J.Z.; formal analysis, L.P. and S.D.; data curation, J.Z. and S.D.; writing—original draft preparation, L.P.; writing—review and editing, J.L., L.D. and Z.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by China Southern Power Grid Science and Technology Program 035900KK52200005 (GDKJXM20200741).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Serrano-Ruiz, J.C.; Mula, J.; Poler, R. Smart manufacturing scheduling: A literature review. *J. Manuf. Syst.* **2021**, *61*, 265–287. [\[CrossRef\]](#)
2. Gao, K.; He, Z.M.; Huang, Y.; Duan, P.; Suganthan, P.N. A survey on meta-heuristics for solving disassembly line balancing, planning and scheduling problems in remanufacturing. *Swarm Evol. Comput.* **2020**, *57*, 100719. [\[CrossRef\]](#)
3. Rossit, D.A.; Tohm, E.F.; Frutos, M. The non-permutation flow-shop scheduling problem: A literature review. *Omega* **2018**, *77*, 143–153. [\[CrossRef\]](#)
4. Ruiz, R.; Maroto, C. A comprehensive review and evaluation of permutation flowshop heuristics. *Eur. J. Oper. Res.* **2005**, *165*, 479–494. [\[CrossRef\]](#)
5. Singh, H.; Oberoi, J.S.; Singh, D. Multi-objective permutation and non-permutation flow shop scheduling problems with no-wait: A systematic literature review. *Rairo-Oper. Res.* **2021**, *55*, 27–50. [\[CrossRef\]](#)
6. Li, J.; Sang, H.; Han, Y.; Wang, C.; Gao, K. Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions. *J. Clean. Prod.* **2018**, *181*, 584–598. [\[CrossRef\]](#)
7. Wang, G.; Gao, L.; Li, X.; Li, P.; Tasgetiren, M.F. Energy-efficient distributed permutation flow shop scheduling problem using a multi-objective whale swarm algorithm. *Swarm Evol. Comput.* **2020**, *57*, 100716. [\[CrossRef\]](#)
8. Fernandez-Viagas, V.; Perez-Gonzalez, P.; Framinan, J.M. The distributed permutation flow shop to minimise the total flowtime. *Comput. Ind. Eng.* **2018**, *118*, 464–477. [\[CrossRef\]](#)
9. Mousakhani, M. Sequence-dependent setup time flexible job shop scheduling problem to minimise total tardiness. *Int. J. Prod. Res.* **2013**, *51*, 3476–3487. [\[CrossRef\]](#)
10. Xu, L.Z.; Xie, Q.S.; Yuan, Q.N.; Huang, H.S. An intelligent optimization algorithm for blocking flow-shop scheduling based on differential evolution. *Int. J. Simul. Model.* **2019**, *18*, 678–688. [\[CrossRef\]](#)
11. Maassen, K.; Perez-Gonzalez, P.; Günther, L.C. Relationship between common objective functions, idle time and waiting time in permutation flow shop scheduling. *Comput. Oper. Res.* **2020**, *121*, 104965. [\[CrossRef\]](#)
12. Zou, P.; Rajora, M.; Liang, S.Y. Multimodal Optimization of Permutation Flow-Shop Scheduling Problems Using a Clustering-Genetic-Algorithm-Based Approach. *Appl. Sci.* **2021**, *11*, 3388. [\[CrossRef\]](#)
13. Umam, M.S.; Mustafid, M.; Suryono, S. A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem. *J. King Saud Univ. —Comput. Inf. Sci.* **2021**, *in press*. [\[CrossRef\]](#)
14. Rameshkumar, K.; Rajendran, C. A novel discrete PSO algorithm for solving job shop scheduling problem to minimize makespan. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2018; Volume 310, p. 012143.
15. Brammer, J.; Lutz, B.; Neumann, D. Permutation flow shop scheduling with multiple lines and demand plans using reinforcement learning. *Eur. J. Oper. Res.* **2021**, *299*, 75–86. [\[CrossRef\]](#)
16. Sadeghzadeh, M.; Calvert, D.; Abdullah, H.A. Self-Learning Visual Servoing of Robot Manipulator Using Explanation-Based Fuzzy Neural Networks and Q-Learning. *J. Intell. Robot. Syst.* **2015**, *78*, 83–104. [\[CrossRef\]](#)
17. Lu, R.; Hong, S.H.; Zhang, X. A dynamic pricing demand response algorithm for smart grid: Reinforcement learning approach. *Appl. Energy* **2018**, *220*, 220–230. [\[CrossRef\]](#)
18. Zhang, D.; Zhang, T.; Liu, X. Novel self-adaptive routing service algorithm for application in VANET. *Appl. Intell.* **2018**, *49*, 1866–1879. [\[CrossRef\]](#)

19. César, Y.; Fonseca-Reyna, Y.; Puris, A.; Martinez, Y.; Trujillo, Y. An Improvement of Reinforcement Learning Approach for Permutation of Flow-Shop Scheduling Problems. *RISTI—Rev. Iber. Sist. E Tecnol. Inf.* **2019**, *E18*, 257–270.
20. He, Z.; Wang, K.; Li, H.; Song, H.; Lin, Z.; Gao, K.; Sadollah, A. Improved Q-learning algorithm for solving permutation flow shop scheduling problems. *IET Collab. Intell. Manuf.* **2021**. [[CrossRef](#)]
21. Oztop, H.; Tasgetiren, M.F.; Kandiller, L.; Pan, Q. A Novel General Variable Neighborhood Search through Q-Learning for No-Idle Flowshop Scheduling. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8.
22. Fonseca-Reyna, Y. Q-learning algorithm performance for m-machine, n-jobs flow shop scheduling problems to minimize makespan. *Investig. Oper.* **2017**, *38*, 281–290.
23. Yang, S.; Xu, Z.; Wang, J. Intelligent Decision-Making of Scheduling for Dynamic Permutation Flowshop via Deep Reinforcement Learning. *Sensors* **2021**, *21*, 1019. [[CrossRef](#)] [[PubMed](#)]
24. Pan, Z.; Wang, L.; Wang, J.; Lu, J. Deep Reinforcement Learning Based Optimization Algorithm for Permutation Flow-Shop Scheduling. *IEEE Trans. Emerg. Top. Comput. Intell.* **2021**. [[CrossRef](#)]
25. Xiao, P.; Zhang, C.; Meng, L.; Hong, H.; Dai, W. Non-permutation Flow Shop Scheduling Problem based on Deep Reinforcement Learning. *Jisuanji Jicheng Zhizao Xitong Comput. Integr. Manuf. Syst. CIMS* **2021**, *27*, 193–206.