

SUPPLY CHAIN SCHEDULING USING DOUBLE DEEP TIME-SERIES DIFFERENTIAL NEURAL NETWORK

Wei Lu^{1,a}, Hua Tan^{*2,b}, Xiaohui Yan³, Cixing Lv⁴

¹School of Electrical Engineering and Intelligentization, Dongguan University of Technology, College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen, China

²Guangdong Linkcom Technology Co., Ltd, Dongguan, China

³School of Electrical Engineering and Intelligentization, Dongguan University of Technology, Dongguan, China

⁴School of Electrical Engineering and Intelligentization, Dongguan University of Technology, Dongguan, China

Abstract—The purpose of supply chain scheduling is to be able to find an optimized plan and strategy so as to optimize the benefits of the entire supply chain. This paper proposes a method for processing tightly coordinated supply chain task scheduling problems based on an improved Double Deep Timing Differential Neural Network (DDTDN) algorithm. The Semi-Markov Decision Process (SMDP) modeling of the state characteristics and action characteristics of the supply chain scheduling problem is realized, so as to transform the task scheduling problem of the tightly coordinated supply chain into a multi-stage decision problem. The deep neural network model can help fit the state value function, and the unique reinforcement learning online evaluation mechanism can realize the selection of the best action strategy combination, and optimize it under the condition of only the stator processing time. Finally, the optimal action strategy group is obtained.

1 Introduction

Supply chain occupies a core position in global trade. Manufacturing is a national basic industry, and its supply chain situation directly represents its position in the international division of labor [1-2]. Compared with traditional supply chain systems, open real-time supply chain systems are quite different in terms of tasks, services, resources, optimization goals, and uncertainties [3-4]. The traditional manufacturing scheduling model [5] has been unable to adapt to the rapid changes in open supply chain tasks. The supply chain task scheduling system used by manufacturing companies is far from the actual production situation, and it is difficult to apply in complex dynamic open situations [6], so there is an urgent need for dynamic and intelligent scheduling methods [7].

In terms of research status at home and abroad, Pan A et al.[8] constructed a "production-scheduled" collaborative process based on multi-agent downstream manufacturers and upstream suppliers; Wang G et al. [9] used ontology-based methods to study the supply and demand of resources and services in the supply chain; Serban R et al. [10] studied the effect of implementing a differential negotiation strategy on the agent of supply and sales tasks in an e-commerce environment; Jihang Z et al. [11] transferred reinforcement learning by reusing the knowledge of the opponent's behavior learned before, and promoted the negotiation between the collaborative tasks of the supply chain; Daniel, J.S.R. et al. [12-14] use genetic algorithms to optimize supply chain scheduling problems; Ma Yuge et al. [15-16] combine particle swarm

optimization with supply chain scheduling problems to quickly obtain convergence results. Among the above algorithms, the particle swarm algorithm is simple to use and has fast convergence speed, but it is easy to be premature and fall into local optimality; while genetic algorithm has strong global search ability, but the search speed is slow; reinforcement learning algorithm is unstable and cannot handle large-scale scheduling problems, and there has not been much breakthrough.

Based on the DQN algorithm [17-18], this paper proposes a method for handling the task scheduling problem of the tightly coordinated supply chain based on the Double Deep Timing Differential Neural Network (DDTDN) algorithm, which transforms the task scheduling problem of a tightly coordinated supply chain into a multi-stage decision-making problem, the given supply chain sub-order processing time is used to optimize the supply chain task scheduling process, and finally the optimal action strategy group is obtained. The advantages of this algorithm include:

1) The action space of the supply chain scheduling problem is a multi-dimensional discrete space, which is not suitable for Q learning using one-dimensional discrete action functions. TD learning is used to replace the Q learning algorithm to solve the problem.

2) The combination of deep neural network and reinforcement learning makes the system capable of handling more complex and larger-scale supply chain task scheduling problems.

^{b*} Corresponding author: 20998353@qq.com

3) The traditional DQN algorithm has an overestimation problem. Using DDTDN to replace the DQN algorithm to solve this problem.

2 SYSTEM IMPLEMENTATION METHOD

The focus of the tightly cooperative supply chain scheduling task on the production side is to reasonably allocate the pending supply chain orders j ($j=1, 2 \dots n$) to the processing sequence composed of m suppliers. The optimization goal of scheduling is to minimize the waiting time of each supplier in order to minimize the total supply chain order completion time. The task scheduling problem of a tightly cooperative supply chain must satisfy the following constraints:

- 1) The overall processing flow of each supply chain order is fixed, but the processing sequence of each supplier's order queue can be changed;
- 2) Each supplier can only process one supply chain sub-order at each time and no interruption is allowed;
- 3) Each supply chain order j has a supply chain sub-order processing time p_{ij} corresponding to supplier i ($i=1,2, \dots,m$), and preparation time and logistics time are included in the processing time.

3 SYSTEM MODELING

Aiming at the task scheduling problem of a tightly coordinated supply chain, this paper proposes a processing method based on Double DTDN to realize the semi-Markov decision-making process for the state and action characteristics of the tightly coordinated supply chain task scheduling problem. The details are shown in Figure 1:

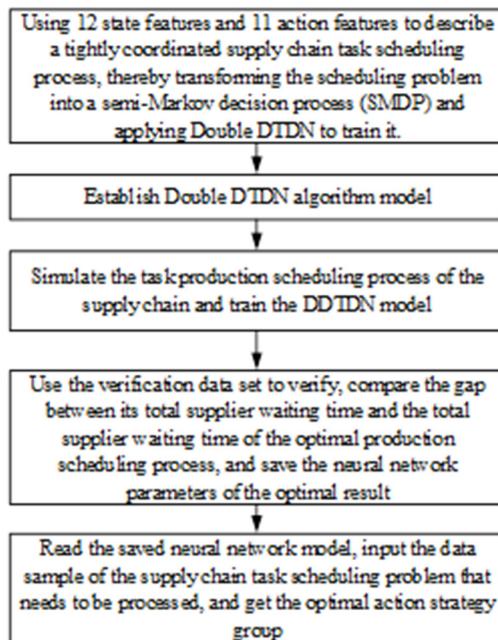


Fig 1. SUPPLY CHAIN SCHEDULING USING Algorithm

3.1 System State Feature Modeling

In this paper, 12 state characteristics are selected to formulate the tasks of the tightly coordinated supply chain and form the basis for judging the busyness of each supplier [19]. Status feature 1 describes the distribution of supply chain order quantities of different suppliers; status feature 2 describes the current workload allocated to each supplier; status feature 3 describes the total workload that each supplier needs to complete from the current status; features 4, 5 describe the longest or shortest processing time of supply chain orders currently in each waiting queue; status feature 6 represents the remaining processing time of the processing job, which in turn characterizes the supplier's free/busy status; status feature 7, 8 represents the longest or shortest remaining processing time for the supplier to wait to complete the supply chain sub-order; status features 9, 10 describe the ratio of the processing time of the supply chain order at a certain supplier to the processing time at the next supplier; state features 11 and 12 describe whether the supplier can adopt Johnson-Bellman rule heuristic behavior.

3.2 System Action Feature Modeling

This paper selects 11 allocation rules as the action characteristics of the tightly coordinated supply chain tasks based on 113 scheduling performance indicators [20]. Among them, actions 1-8 are common allocation rules for scheduling problems, actions 9, 10 are Johnson-Bellman allocation rules, and action 11 is the first-come-first-processing rule, when there is only one supply chain order in the queue, action 11 would be adopted. When the DDTDN algorithm works, it selects the behavior that suits the current supply chain order queue according to the status value input of each supplier, and then selects the supply chain sub-orders for processing.

3.3 Reward Function

The production cycle is closely related to the busyness of the supplier. Define $\sigma_i(t)$ as the supplier's free/busy status indicator function:

$$\sigma_i(t) = \begin{cases} 0 & \text{Supplier } i \text{ is busy at time } t \\ 1 & \text{Supplier } i \text{ is free at time } t \end{cases} \quad (1)$$

The reward function is defined as:

$$r = \sum_{i=1}^m \int_{t_{k-1}}^{t_k} \sigma(t) \quad (2)$$

In the formula, r represents the reward that the system gets when it moves to the state at time t_{k+1} after executing the behavior at the decision time t_k . Obviously, r is equal to the inverse number of the total idle time of the supplier in the interval $[t_k, t_{k+1}]$.

3.4 Double DTDN Modeling

Suppose the objective function value is $V_T(S_{t+1})$, in the DQN algorithm, it is directly calculated from the current neural network, as:

$$V_T(S_{t+1}) = \max V_{eval}(S_{t+1}) \quad (3)$$

The estimated value output by the network model of the DQN algorithm is larger than the true function value, and for different states, the overestimation amplitude will be different, which directly leads to the change of the optimal action strategy selection.

In the DDTDN algorithm, on the basis of the current value function of the subsequent state corresponding to each action calculated by the current neural network, the action corresponding to the largest current value function value is found, namely:

$$a^{\max} = \arg \max V_{eval}(S_{t+1}) \quad (4)$$

Then the simulation environment executes the selected action a^{\max} to obtain the corresponding subsequent state S'_{t+1} , and calculate the corresponding value function in the target network, namely:

$$V_T(S'_{t+1}) = \max V_{target}(S'_{t+1}) \quad (5)$$

$V_T(S'_{t+1})$ is the value of the objective function we require, so that the over-estimation problem caused by the DQN algorithm is well solved.

The neural network in DDTDN is divided into two types: current neural network (eval_net) and target neural network (target_net) (as shown in Figure 2). The system triggers a learning flag every n steps to perform learning and update the current neural network parameters, besides, the target neural network parameters will be updated at the end of each complete supply chain order. The current neural network and the target neural network have the same network structure (shown in Table 3), the loss calculation method is variance calculation, and the optimizer is the RMSPropOptimizer. What DDTDN seeks is an iterative form of harvest function related to the value function of the next state, which means when performing reinforcement learning, you only need to obtain the value function of the current cycle state and the next state, without obtaining a complete cycle.

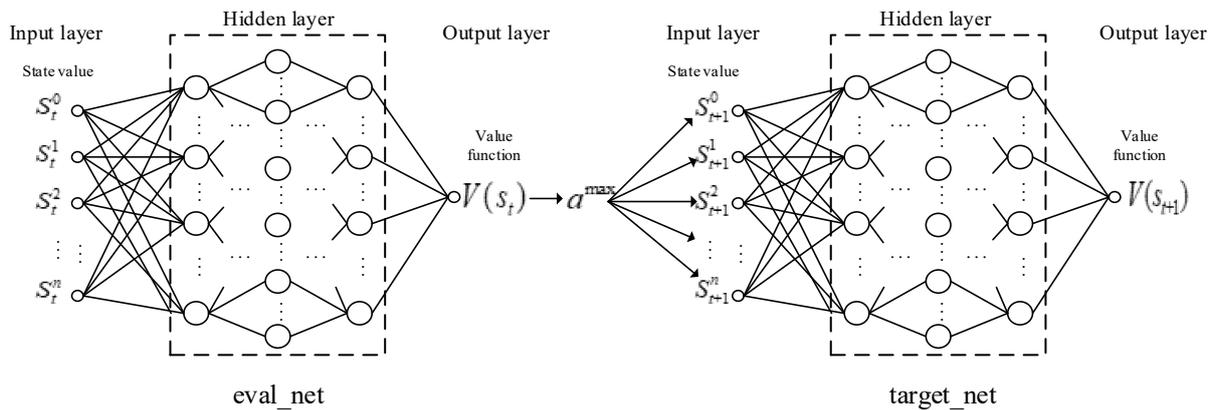


Fig 2. Algorithm implementation steps

Table1. DOUBLE DTDN STRUCTURE

Layers	Type	Input dimension	Output dimension	Activation function
1	Input layer	(None, Feature size)	(None, 200)	ReLU
2	Hidden layer	(None, 200)	(None, 300)	ReLU
3	Hidden layer	(None, 300)	(None, 400)	ReLU
4	Hidden layer	(None, 400)	(None, 300)	ReLU
5	Hidden layer	(None, 300)	(None, 200)	ReLU
6	Hidden layer	(None, 200)	(None, 100)	ReLU
7	Output layer	(None, 100)	(None, 1)	ReLU

3.5 Double DTDN Algorithm Training

The DDTDN algorithm flow is shown in Figure 3. The system mainly contains two layers of loops. The inner loop simulates the sub-order processing process of the supply chain, stores the obtained single-step samples in the memory, and updates the current neural network

parameters when the learning flag is triggered; the outer loop is used to repeatedly execute the inner layer loop, and update the state transition at the end of each episode, replace the target neural network parameters, and output the optimal production cycle corresponding strategy combination when the episode reaches the set value Max_Episode, and record its current neural network parameters.

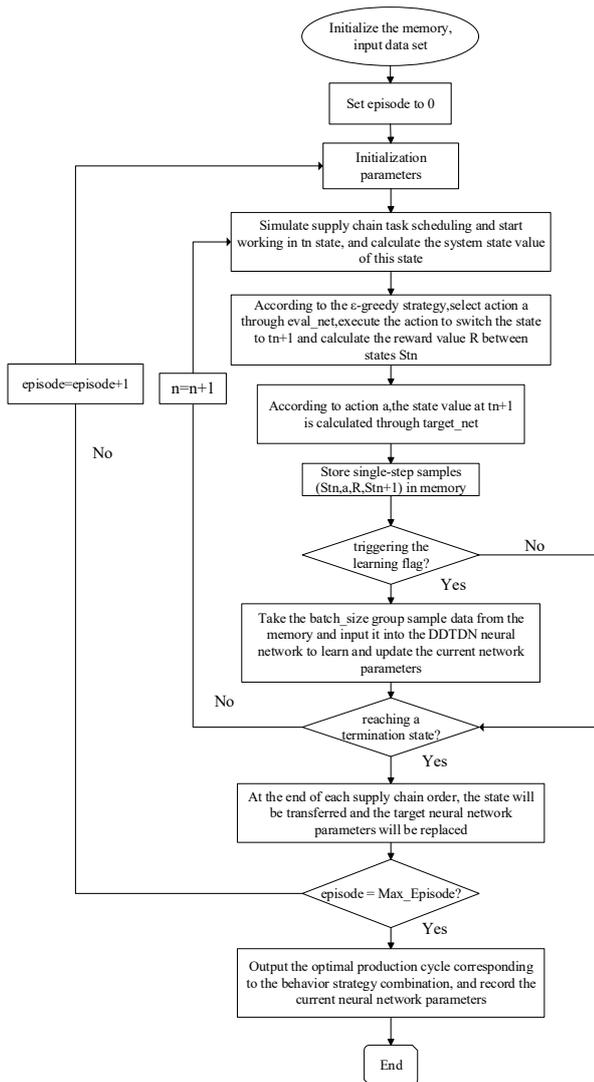


Fig 3. DDTDN algorithm training flowchart

4 EXPERIMENTAL SIMULATION

4.1 Experimental Platform and Data Set

The experiment was carried out under the environment of Intel Core I5-9300HF 2.40GHz, 16G memory, interpreter python3.7, and deep neural network modeling using TensorFlow.

In order to evaluate the effectiveness of the algorithm in the task scheduling of tightly coordinated supply chain, this paper uses various data sets of different scales to verify it.

The data set includes: the number of suppliers m , the number of pending supply chain orders n , and the pending

supply chain sub-order processing time data set. Collect historical supply data of the enterprise, and define the supply chain task for the supply demand of product manufacturing: decompose the supply chain task into multiple (5-20) supply chain orders, and each supply chain order contains multiple sub-orders (5-20), each sub-order is completed by a supplier to complete the construction of the supply chain task data set. In addition, in order to increase the sample size of the data set, the simulation data set is also constructed with reference to the historical supply data of the company to randomly generate the simulation data set, that is, the historical supply data of the company is referred to, and the corresponding processing time curve of the sub-orders of the historical pending supply chain is randomly generated. The sub-order processing time data set.

The experimental data set is generated randomly, and the sub-order processing time of the supply chain obeys a discrete and uniform distribution of 1-100. The size of the test data set includes the number of 4 types of suppliers ($m = 5, 6, 7, 10$) and the number of 5 types of orders ($n = 5, 6, 7, 10, 20$), which are set by m and n , resulting in a total of 6 groups data.

In terms of neural network parameter settings, set the discount factor γ to 0.95; set the maximum in ϵ -greedy strategy $\epsilon = 0.9$, and the initial $\epsilon = 0$, and increase at a rate of 1×10^{-4} after each learning; set the initial learning rate $\alpha = 5 \times 10^{-4}$, and it will be halved when the number of learning times reaches 1×10^4 to prevent the occurrence of over-fitting; set the maximum number of learning $MAX_EPISODE=500$.

4.2 Experimental Result and Analysis

Comparing the results of the execution of this algorithm with the results of several other common scheduling rules, the results are shown in Table 2. The evaluation criteria of the algorithm in the table are based on the total idle time of the supplier in the entire scheduling process, in seconds (s), the shorter the idle time is, the better the algorithm scheduling is.

The comparison rules used in the experiment are common allocation rules in scheduling problems. The experimental results show that compared with these common scheduling rules, this algorithm can achieve better results in the above-mentioned supply chain scheduling tasks of different scales. Effectively reduce the waiting time of the supplier's production queue, improve the efficiency of task scheduling in a tightly coordinated supply chain, and increase productivity.

Table2. COMPARISON OF EXPERIMENTAL RESULTS

	LPT	LPT/TWK	LWKR	MWKR	Johnson1	Johnson2	DDTDN
5_5.xlsx	1802	1752	1502	2052	1777	1567	1167
6_6.xlsx	2299	2317	2173	1927	2029	1981	1645
7_7.xlsx	4006	3341	3593	3299	3164	3852	2879
10_10.xlsx	8497	7727	7745	7637	8387	9037	7445

20_5.xlsx	3344	2529	2662	2679	2529	3159	2154
20_10.xlsx	11011	9711	9171	10351	9601	10491	8891

5 Conclusion

This paper proposes a double deep timing differential neural network (DDTDN) to optimize the task scheduling process of a tightly coordinated supply chain under the condition of only a given processing time for sub-orders of the pending supply chain, and finally get the optimal action strategy group.

The advantages of the proposed algorithm include:

It has learning ability and strong real-time performance. When the neural network training is mature, the past experience patterns are stored in the network parameters, and scheduling decisions can be made in real time.

The development potential is great, and there is a lot of room for development at the moment when the theory of deep neural networks continues to advance and the computing capabilities of computers continue to improve.

Acknowledgment

The authors would like to thank the School of Electrical Engineering and Intelligentization Dongguan University of Technology, and the School of Mechanical and Electrical Engineering of Shenzhen University.

The work was supported by National key R&D program of China (Grant. 2018YFB1004004) and Introduced innovative scientific research team project of Dongguan City (No. 2018607204010)

References

1. Dmitry Ivanov , Dolgui, Alexandre . A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0[J]. International Journal of Production Research.2014, 152(SI) : 131 — 143.
2. Petersen, KJ; Handfield, RB; Ragatz, GL. Supplier integration into new product development: coordinating product, process and supply chain design [J]. Journal of Operations Management.2014, 23(3-4):371 — 388.
3. Xu, Li Da. Information architecture for supply chain quality management [J]. International Journal of Production Research .2011, 49(1):183 — 198.
4. Linton, Jonathan D.; Klassen, Robert; Jayaraman, Vaidyanathan . Sustainable supply chains: An introduction [J]. Journal of Operations Management.2007, 25(6):1075 — 1082.
5. Li Jian-xiang; Tang Li-xin. A Review of Production Planning and Scheduling in Iron and Steel Supply Chain [J]. Control Engineering China, 2010,17(2): 123-6.

6. Seuring, Stefan. A review of modeling approaches for sustainable supply chain management [J]. Decision Support Systems.2013 , 54(4):1513 — 1520.
7. Qian X L, Tang L X, Liu W X. Dynamic scheduling: A survey of research methods[J]. Control and Decision, 2001,16(2): 141-145.
8. A Pan, TM Choi. An agent-based negotiation model on price and delivery date in a fashion supply chain[J]. Annals of Operations Research, 2013, 2:1 - 29.
9. G Wang, TN Wong, X Wang. An ontology based approach to organize multi-agent assisted supply chain negotiations[J]. Computers & Industrial Engineering, 2013, 65(01): 2 - 15.
10. S Radu, AM Florea. Bargain strategies for agent automated negotiation in an e-business environment[J]. Scalable Computing: Practice and Experience, 2014, 15(02): 121 - 129.
11. HB Ammar, S Chen, K Tuyls, G Weiss. Automated transfer for reinforcement learning tasks [J]. Künstliche Intelligenz, 2014, 28(01): 7 - 14.
12. Daniel, J.S.R.; Rajendran, C. A simulation-based genetic algorithm for inventory optimization in a serial supply chain [J]. International Transactions in Operational Research, 2005, 22(01): 101 - 27.
13. Jiang, Dali; Li, Haitao; Yang, Tinghong. Genetic algorithm for inventory positioning problem with general acyclic supply chain networks [J]. EUROPEAN JOURNAL OF INDUSTRIAL ENGINEERING, 2016, 10(03): 367 - 384.
14. Zhao Hanping; Jiang Jiadong; Feng Yuncheng. Coordinating supply chain of Stackelberg game model based on evolutionary game with GA-RL [J]. Systems Engineering-Theory & Practice , 2010, 30(04): 667 - 672.
15. Ma Yuge; Liu Yong. Synergy optimizing of material purchase in JIT supply chain using particle swarm optimization algorithm [J]SCComputer Engineering and Applications,2016, 54(02): 259 - 65.
16. Ma Bin; Wu Zezhong. Research on Supply Chain Network Equilibrium Model Based on Improved Particle Swarm Optimization [J]. Operations Research and Management Science,2020, 29(02): 122 - 128.
17. MNIH V, KAVUKCUOGLU K, SILVER D, et al.Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540):529-533.
18. MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing Atari with Deep Reinforcement Learning[J].Computer Science, 2013.
19. ZHANG Zhicong, ZHENG Li. Manufacturing system scheduling based on reinforcement learning[M]. Beijing: Science Press, 2016.

20. ISKANDER P W. A Survey of Scheduling Rules[J]. *Operations Research*, 1977, 25(1):45-61. Covington, P., Adams, J., & Sargin, E. (2016, September). Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 191-198).