Deep Reinforcement Learning Approach for Capacitated Supply Chain Optimization under Demand Uncertainty

1st Zedong Peng
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
zdpeng@zju.edu.cn

2nd Yi Zhang
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
drzhangyi@zju.edu.cn

3rd Yiping Feng
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
ypfeng@zju.edu.cn

4th Tuchao Zhang
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
ztchao1996@zju.edu.cn

5th Zhengguang Wu
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
nashwzhg@zju.edu.cn

6th Hongye Su
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
hysu69@zju.edu.cn

***Abstract*—With the global trade competition becoming further intensified, Supply Chain Management (SCM) technology has become critical to maintain competitive advantages for enterprises. However, the economic integration and increased market uncertainty have brought great challenges to SCM. In this paper, two Deep Reinforcement Learning (DRL) based methods are proposed to solve multi-period capacitated supply chain optimization problem under demand uncertainty. The capacity constraints are satisfied from both modelling perspective and DRL algorithm perspective. Both continuous action space and discrete action space are considered. The performance of the methods is analyzed through the simulation of three different cases. Compared to the baseline of (r, Q) policy, the proposed methods show promising results for the supply chain optimization problem.**

***Index Terms*—supply chain optimization, deep reinforcement learning, demand uncertainty, vanilla policy gradient**

I. Introduction

According to China Federation of Logistics and Purchasing(CFLP), China's domestic social logistics cost amounted to almost 14.8% of GDP in 2018. Compared to the proportion of about 10% in developed countries, there is still room to enhance the efficiency of supply chain for Chinese enterprises. Supply Chain Management(SCM) is one of the key technologies to help enterprises to gain a competitive advantage in the marketplace [1]. Typically, SCM includes all processes that transform raw materials into final products. When the structure of the supply chain is determined, the optimization problem looks for the most efficient way of the production, shipment, and distribution of the products and attempts to fulfil the demand over the planning horizon. There are several challenges in supply chain optimization [2].

Firstly, there are a lot of uncertainties in supply chain network, such as the demand fluctuation and supply disruption, equipment failure and etc. All these uncertainties increase the difficulty in decision making. More robust strategies should be designed to lower the risk brought by these uncertainties.

Secondly, the global economic integration has increased the complexity of the structure of supply chain network. Decision makers need to consider more interactions between plants and retailers at the same time to achieve the optimal operation.

Traditional solutions for supply chain optimization problem mainly include rule-based methods and operation research(OR)-based methods. (r, Q) policy and (s, S) policy are the two basic rule-based methods [3]. In an (r, Q) inventory system, the inventory of the item is reviewed continuously. A fixed quantity Q will be ordered when the inventory position drops to the reorder point r. These policies are easy to understand and implement, so they have been widely used in practice. As for OR-based methods, forecasting techniques are generally first adopted to estimate the distribution of future demand; then, optimization methods, such as robust optimization [4], stochastic programming [5] and chance-constrained optimization [6], are apply to find the optimal decision.

In recent years, motivated by the great success of Deep Reinforcement Learning, several DRL-based methods have been proposed to solve resource allocation problems and scheduling problems, such as resource management in cluster [7], device placement optimization [8], power allocation in satellites [9], railway lines scheduling [10], cellular network traffic scheduling [11], dynamic flowshop scheduling [12] and etc. These works demonstrate the potential of DRL in

This work is supported by Intelligent Manufacturing Integrated Standardization and New Model application Project (No.2018-ZJ-003) and the Science Fund for Creative Research Groups of NSFC (Grant No.61621002).

modeling complex systems and making decisions in uncertain environment.

Reinforcement learning was first applied to supply chain optimization and inventory management by Giannoccaro and Pontrandolfo in 2002 [13]. Recently, DRL-based methods have also been proposed to solve supply chain optimization problems. Oroojlooyjadid et al. proposed an algorithm based on deep Q-networks for Beer Game without the knowledge of the demand probability distribution [14]. Fuji et al. proposed a multi-agent reinforcement learning technique to optimize supply chain performance using DNN-weight evolution [15]. Li et al. proposed novel sophisticated multi-agent reinforcement learning approach for resource balancing problem in complex logistics network [16].

However, most of the above literature focuses on discrete action space and rarely take capacity constraints into consideration. In this paper, we focus on multi-period supply chain optimization problems with capacity constraints under demand uncertainty. We proposed two DRL-based methods to solve the supply chain optimization problem with capacity constraints. A general method for both continuous and discrete action space and a method for discrete action space. The rest of this paper is organized as follows: Section 2 describes the formulation of the supply chain optimization problem. Section 3 presents the deep RL-based method to solve the problem. Section 4 presents the results of proposed methods and finally Section 5 outlines the conclusion of the paper.

II. Problem Statement

In this paper, we focus one the multi-period supply chain optimization problem. The supply chain is structured as Fig.1 and consists of plants, plant warehouses, retailers and consumers. The horizon of the optimization problem is divided into a set of periods with the same length. At the beginning of each period, the decision maker reviews the amount of on-hand inventory or backorders of the plant warehouse and each retailer, then decide the amount of products to produce and delivered to each retailers. Due to the capacity constraints, there is an upper limit on the amount of products the plant can produce and each retailer can store. The products produced by plants will be handed over to plant warehouse at the end of each period all together. The supply of raw material is assumed to be adequate and stable. The demand from consumers are stochastic and possibly seasonal and the decision maker satisfies the demand to the fullest extent by the on-hand inventory. In addition, the overstocked products or unsatisfied demands are carried over to the next period, which means that the decision made in a period will affect the inventory levels in the future periods. In the case of excess inventory, a storage cost is incurred per unit of overstocked products. Otherwise, in the case of deficient inventory, a penalty cost is incurred per unit of unsatisfied demand.

The objective is to maximize the total profit considering the revenue from sold products, production cost, storage cost, penalty cost, transportation cost incurred in all the periods. The demands across the periods are independent, though not necessarily identically distributed. This problem encapsulates the dilemma of matching supply with volatile demand in the presence of capacity constraints. The supply chain optimization problem can be formulated as follows.

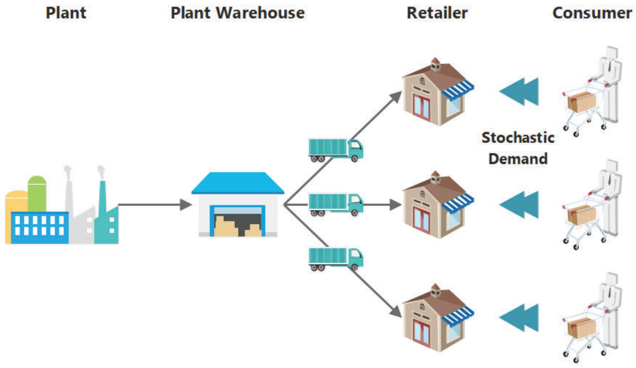


Fig. 1: Structure of Supply Chain

$$max \sum_{t=1}^{T} \left[\begin{array}{l} \nu_1 \sum_{j=1}^{K} Dem_{j,t} - \nu_2 p_t - \nu_5 \sum_{j=1}^{K} \left\lceil \frac{d_{j,t}}{\gamma} \right\rceil \\ -\nu_3 \sum_{j=1}^{K} \max\{inv_{j,t}, 0\} \\ +\nu_4 \sum_{j=1}^{K} \min\{inv_{j,t}, 0\} \end{array} \right] \tag{1}$$

subject to

$$0 \leq p_t \leq P_{\max}, \forall t \in \{1, ..., T\} \tag{2}$$

$$\sum_{j=1}^{K} d_{j,t} \leq inv_{j=0,t}, \forall t \in \{1, ..., T\} \tag{3}$$

$$inv_{j=0,t} + p_t \leq C_{j=0,t}, \forall t \in \{1, ..., T\} \tag{4}$$

$$inv_{j,t} + d_{j,t} \leq C_{j,t}, \forall j \in \{1, ..., K\}, \forall t \in \{1, ..., T\} \tag{5}$$

$$inv_{j=0,t+1} = inv_{j=0,t} + p_t - \sum_{j=1}^{K} d_{j,t}, \forall t \in \{1, ..., T\} \tag{6}$$

$$inv_{j,t+1} = inv_{j,t} + d_{j,t} - Dem_{j,t}, \forall j \in \{1, ..., K\}, \forall t \in \{1, ..., T\} \tag{7}$$

where $\nu_1, \nu_2, \nu_3, \nu_4, \nu_5$ are the corresponding coefficients of revenue from sold products, production cost, storage cost, penalty cost and transportation cost. p_t denotes the production target for the plant in period t and $d_{j,t}$ denotes the amount of products delivered to retailer j. $inv_{j,t}$ denotes the inventory level of plant warehouse and retailers, where $j = 0$ denotes plant warehouse and $j = 1, ..., K$ denotes retailers. The value of $inv_{j,t}$ becomes negative when backorder occurs with insufficient inventory.

There is a production capacity limit for the plant as constraint (2) and the total delivered amount should be no more than the current inventory level of plant warehouse as constraint (3). Constraints (4) and (5) represent the storage capacity constraints of the plant warehouse and retailers. In the plant warehouse, the sum of current inventory and newly produced products should not exceed the capacity limit. For retailers, the sum of current inventory level and newly delivered products should not exceed the capacity limit. Constraints (6) and (7) represent the material balance for plant warehouse and retailers. When the demand $Dem_{j,t}$ is deterministic and known in advance, the mathematical program can give a optimal solution. However, when the demand is stochastic and unpredictable, even robust optimization methods and stochastic programming methods can not find a reliable solution in practice. Generally, model-free DRL based algorithms do not use transition probability distribution. Hence they are able to make decisions without the knowledge of demand probability distribution.

III. Deep Reinforcement Learning Setup

In this section, two deep reinforcement learning based methods are proposed by to solve the above problem. Generally, the environment in DRL is modelled as Markov Decision Process(MDP). The difference between the supply chain optimization problem and standard environments, such as LunarLander and CarRacing in Gym [17], lies in the constraints (2)-(5), which is more similar to constrained Markov decision process(CMDP) and stochastic action set Markov decision process (SAS-MDP). The action space of supply chain optimization problems is variable due to the inventory capacity constraints. For each state $s \in S$, a set $A(s)$ of possible action is available and $A(s_1)$ and $A(s_2)$ might be different. For example, considering a plant warehouse with the capacity of 100, the action space is $[0, 50]$ when the state is 50 and the action space becomes $[0, 30]$ when the state increase to 70.

As we know, policy-based and value-based deep reinforcement learning approximates decision policies using neural network and value-based reinforcement learning approximate the Q function using neural network. Generally, If the output dimension of both two kinds of neural networks is set equal to the dimension of action spaces, the structure of the neural network would change when state transition occurs. This problem can be solved from the viewpoints of both the modelling of MDP and the DRL algorithm. In this section, two approaches have been proposed to solve this problem. Compared to value-based DRL, policy-based DRL has three main advantages :better convergence properties, effectiveness in high-dimensional or continuous action spaces, ability to learn stochastic policies. Since the production target p_t and delivered amount $d_{j,t}$ might be discrete or continuous variables for different industries, we choose policy-based DRL, more precisely Vanilla Policy Gradient algorithm, to solve the supply chain optimization problem.

A. From modelling perspective: mapping infeasible actions to feasible actions

In this approach, the structure of neural network will not alternated. When sampling trajectories, the action chosen by agents will be further mapped into feasible action space. For example, when the action space is $[0, 100]$ and feasible action space is $[0, 30]$, the subset of $(30, 70]$ will be mapped to 30.

Action space. According to the formulation in section 2, the action a_t is defined as a vector including the production target of the plant and the delivered amount to each retailer in period t.

$$a_t = \{p_t, d_{j,t} | j \in \{1, ..., K\}\} \tag{8}$$

The value of p_t ranges from 0 to P_{max} as constraint (2). To satisfy constraints (4) and (5), the value of p_t and $d_{j,t}$ are clipped between 0 and $C_{j,t} - inv_{j,t}$. When constraint (3) is violated, we will scale down the value of the action in an appropriate proportion. The detailed method for continuous action space is presented as equation (9). For discrete action space, the action vector is first scaled according to equation (9) and then the scaled action will round down to the nearest feasible action.

$$d_{j,t} \leftarrow d_{j,t} \cdot \frac{inv_{j=0,t}}{\sum\limits_{j=1}^{K} d_{j,t}} \tag{9}$$

State space. The state s_t is defined as a vector including the inventory level of the plant warehouse and each retailer.

$$s_t = \{inv_{0,t}, inv_{j,t}, Dem_{j,t-1}, Dem_{j,t-2} | j \in \{1, ..., K\}\} \tag{10}$$

The fluctuation of demand may be regular or seasonal, so we include the last demands to the state space and therefore allow the agent to have limited knowledge of the demand history.

State transition. The state transition is deterministic and the transition function is implemented according to the material balance constraints (6) and (7).

Reward function. The one step reward function is accomplished through equation (11), which is derived from the objective function (1).

$$\begin{aligned} r_t =& \nu_1 \sum_{j=1}^{K} Dem_{j,t} - \nu_2 p_t - \nu_3 \sum_{j=1}^{K} \max\{inv_{j,t}, 0\} \\ & - \nu_5 \sum_{j=1}^{K} \left\lceil \frac{d_{j,t}}{\gamma} \right\rceil + \nu_4 \sum_{j=1}^{K} \min\{inv_{j,t}, 0\} \end{aligned} \tag{11}$$

Thus, the capacitated supply chain optimization problem is modeled as an Markov Decision Process. The DRL architecture of method A is shown in Fig.2. Besides the mapping method as equation (9), designing negative reward for infeasible actions might be another choice. The negative reward of infeasible actions will lower the probability of being chosen in policy-based algorithms. However, designing proper negative reward for infeasible actions is pretty tricky and the reward function needs to be redesigned when the parameters of

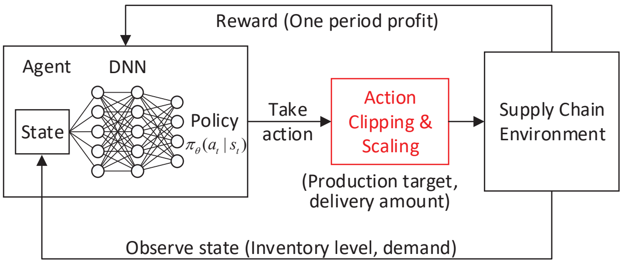


Fig. 2: DRL architecture of Method A

the mathematical model change. Therefore, we don't adopt negative reward method in this paper. The Vanilla Policy Gradient algorithm we applied to solve the Markov Decision Process is shown in Algorithm 1.

Algorithm 1 Vanilla Policy Gradient Algorithm

```
**Initialization**: Input: Initial policy parameters $\theta_0$, initial value function parameter $\phi_0$
**for** k=0,1,2,... **do**
    Collect set of trajectories $D_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
    Compute rewards-to-go
        $\hat{R}_t = \left(\sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})\right)$
    Update value function network using target $\hat{R}_t$ by regression on mean-squared error:
        $\phi_{k+1} = \arg\min_{\phi} \frac{1}{|D_k| T} \sum_{\tau \in D_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t\right)^2$
    Compute advantage estimates $\hat{A}_t$ based on the current value function $V_{\phi_k}$.
        $\hat{A}_t(s_t, a_t) = \left(\sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})\right) - V_{\phi_k}(s_t)$
    Estimate policy gradient as
        $\hat{g}_k = \frac{1}{|D_k|} \sum_{\tau \in D_k} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta_k} \hat{A}_t$
    Compute policy gradient update using standard gradient ascent
        $\theta_{k+1} = \theta_k + \alpha_k \hat{g}_k$
**end for**
```

B. From DRL algorithm perspective: New output activation function

Besides the above modelling method, the capacity constraints can also be satisfied from DRL algorithm perspective in the case of discrete action space.

First, we model the supply chain optimization problem without considering the capacity constraints. The action space is defined the same as equation (8). The value of p_t ranges from 0 to P_{max} and the value of $d_{j,t}$ ranges from 0 to $C_{j,t}$. There are no clipping or scaling process as equation (9). The state space, state transition and reward function are the same as the equation (6) (7) (10) and (11).

In method A, standard Vanilla Policy Gradient algorithm is adopted and the output of the neural network is directly used as policy $\pi(a|s)$ to sample trajectories. To help the algorithm satisfy capacity constraint, we make small modifications to algorithm 1. Motivated by the ApprOpt proposed by Bhatia et al [18], the output of the neural network is further processed before used to sample trajectories.

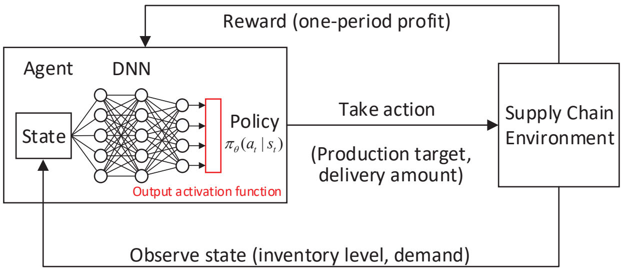

Reward (one-period profit)
Agent
DNN
State
Policy
$\pi_\theta(a_t\,|\,s_t)$
Output activation function
Take action
(Production target,
delivery amount)
Supply Chain
Environment
Observe state (inventory level, demand)

Fig. 3: DRL architecture of Method B

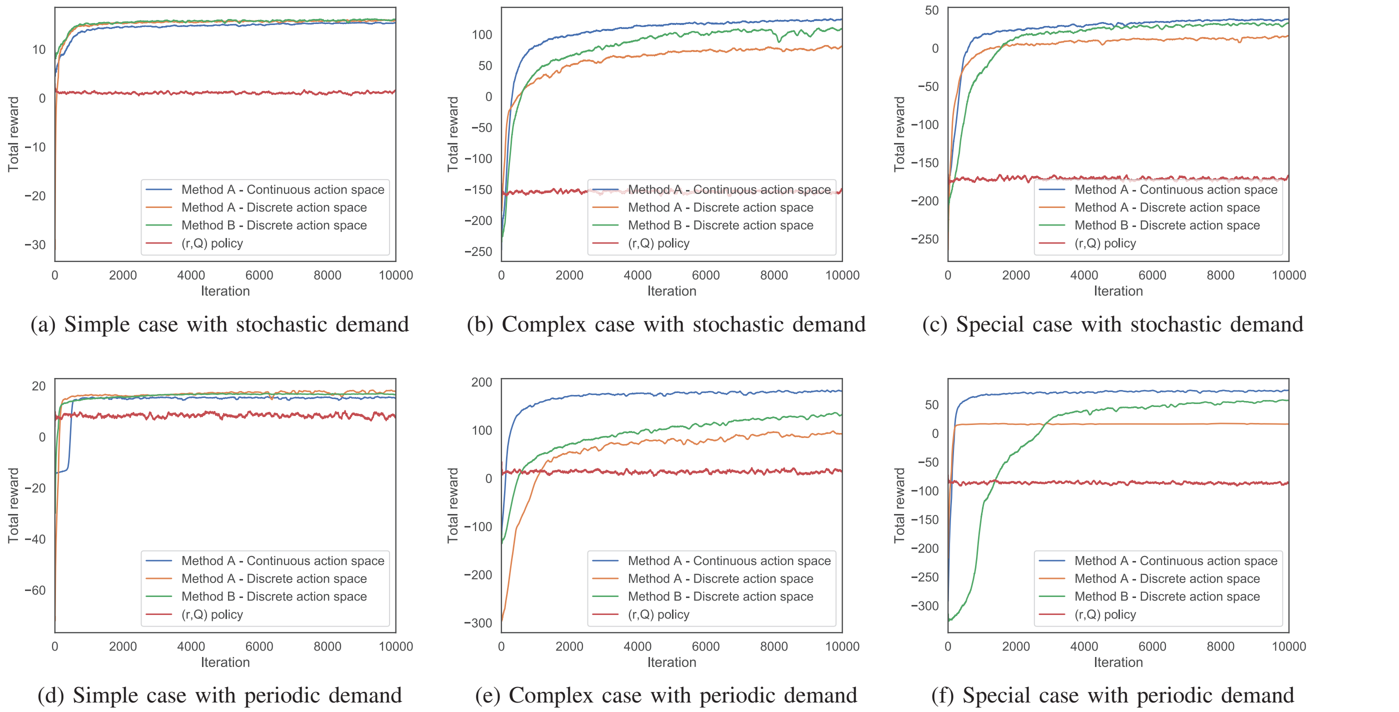
A new output activation function is added behind the output of original neural network and it is defined as follows. Firstly, An indicator function $\eta(a_i|s)$ was designed to represent the feasibility of action a_i in current state s as equation (12). The criteria to judge the feasibility of actions is according to constraints (3) (4) and (5). Therefore, the indicator function $\eta(a_i|s)$ have different values to different state s.

$$\eta(a_i|s) = \begin{cases} 1 & \text{if action } a_i \text{ is allowed in state } s \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Then the probability of choosing infeasible actions is set to zero and the new policy is normalized to guarantee the probability sum up to 1 as equation (13).

$$\pi_\theta(a_i|s) = \frac{\pi_\theta(a_i|s) * \eta(a_i|s)}{\sum\limits_i \pi_\theta(a_i|s) * \eta(a_i|s)} \tag{13}$$

The aim of output activation function (13) is to reduce the probability of choosing infeasible action to zero. With this modification, only feasible action would be chosen by the DRL agent. The DRL architecture of method B is shown in Fig.3. The difference between method A and method B is that method A directly projects infeasible actions to feasible action space, while method B project the output of neural network to feasible action space.



(a) Simple case with stochastic demand

(b) Complex case with stochastic demand

(c) Special case with stochastic demand

(d) Simple case with periodic demand

(e) Complex case with periodic demand

(f) Special case with periodic demand

Fig. 4: Comparison on the two proposed methods and (r, Q) policy

IV. Case Study

To assess the performance of the two proposed methods, we conduct experiments on the capacitated supply chain optimization problem with different settings, including a simple case, a complex case and a special case. In the simple case, there are only one plant, one plant warehouse and one retailer in the supply chain network. Only sales revenue storage cost and penalty cost are considered. In the complex case, there are one plant, one plant warehouse and three retailers in the supply chain network. Sales revenue, production cost, storage cost, transportation cost and penalty cost are considered. The special case has the same supply chain structure as the complex case. The difference is that the second, third retailers has no storage cost and the third retailer has no transportation cost. This special case is designed to test if DRL agent could learn to make decision in relatively imbalanced environment. The length of each episode is set to 25 steps. Two kinds of demand are designed, including stochastic demand and periodic demand. The periodic demand has both seasonal fluctuations and stochastic fluctuations, while stochastic demand has only stochastic fluctuations.

The (r, Q) policy is adopted as the baseline for the proposed methods. When the inventory reaches level r, a replenishment order for Q units is placed. The detailed implementation can be referred to the work of Kammer [19]. The policy network is parameterized by 4-layer, 6-layer, 10-layer, 15-layer, 20-layer multi layer perception (MLP) with node size of 64 in each layer. All the DRL methods are trained for 10000 episodes. We use $AdamOptimizer$ with a learning rate of 10^{-4}. The batch size is fixed to 16.

The results of the simple case, complex case and special case with different demand are shown in Fig.4. From all the results of these three cases, the DRL agent outperforms the (r, Q) policy. In the simple case, Method A with continuous action space, Method A with discrete action space and Method B with discrete action space present comparable performance after 10000 iterations. In the complex case and the special case, since the continuous action space has a broader feasible region, the Method A with continuous action space obtains higher reward compared to the others. From Fig.4b, Fig.4c, Fig.4e and Fig.4f, we can see that the Method A with continuous action space has a faster convergence speed and generally converges to the optimal policy in 2000 iterations.

V. Conclusion

In this paper, two DRL-based methods have been proposed to solve multi-period capacitated supply chain optimization problem under demand uncertainty. In the first method, action clipping and scaling rules are designed to satisfy the capacity constraints, which applies to both continuous action space and discrete action space. In the second method, a new output activation function are proposed to enforce actions in the feasible region. The results from the case study demonstrate that both the two methods always converge to a better policy than the (r, Q) policy in different setting of demand uncertainty and supply chain network.

References

[1] J. T. Mentzer, W. DeWitt, J. S. Keebler, S. Min, N. W. Nix, C. D. Smith, and Z. G. Zacharia, "Defining supply chain management," *Journal of Business logistics*, vol. 22, no. 2, pp. 1–25, 2001.

[2] D. J. Garcia and F. You, "Supply chain design and optimization: Challenges and opportunities," *Computers & Chemical Engineering*, vol. 81, pp. 153–170, 2015.

[3] H. Galliher, P. M. Morse, and M. Simond, "Dynamics of two classes of continuous-review inventory systems," *Operations Research*, vol. 7, no. 3, pp. 362–384, 1959.

[4] D. Bertsimas and A. Thiele, "A robust optimization approach to supply chain management," in *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 2004, pp. 86–100.

[5] F. Xie and Y. Huang, "A multistage stochastic programming model for a multi-period strategic expansion of biofuel supply chain under evolving uncertainties," *Transportation Research Part E: Logistics and Transportation Review*, vol. 111, pp. 130–148, 2018.

[6] B. Vahdani, R. Tavakkoli-Moghaddam, F. Jolai, and A. Baboli, "Reliable design of a closed loop supply chain network under uncertainty: An interval fuzzy possibilistic chance-constrained model," *Engineering Optimization*, vol. 45, no. 6, pp. 745–765, 2013.

[7] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. ACM, 2016, pp. 50–56.

[8] A. Mirhoseini, H. Pham, Q. V. Le, B. Steiner, R. Larsen, Y. Zhou, N. Kumar, M. Norouzi, S. Bengio, and J. Dean, "Device placement optimization with reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2430–2439.

[9] J. J. G. Luis, M. Guerster, I. del Portillo, E. Crawley, and B. Cameron, "Deep reinforcement learning architecture for continuous power allocation in high throughput satellites," *arXiv preprint arXiv:1906.00571*, 2019.

[10] H. Khadilkar, "A scalable reinforcement learning algorithm for scheduling railway lines," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–11, 2018.

[11] S. Chinchali, P. Hu, T. Chu, M. Sharma, M. Bansal, R. Misra, M. Pavone, and S. Katti, "Cellular network traffic scheduling with deep reinforcement learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] J. Wang, S. Qu, J. Wang, J. O. Leckie, and R. Xu, "Real-time decision support with reinforcement learning for dynamic flowshop scheduling," in *Smart SysTech 2017; European Conference on Smart Objects, Systems and Technologies*. VDE, 2017, pp. 1–9.

[13] I. Giannoccaro and P. Pontrandolfo, "Inventory management in supply chains: a reinforcement learning approach," *International Journal of Production Economics*, vol. 78, no. 2, pp. 153–161, 2002.

[14] A. Oroojlooyjadid, M. Nazari, L. Snyder, and M. Takáč, "A deep q-network for the beer game: A reinforcement learning algorithm to solve inventory optimization problems," *arXiv preprint arXiv:1708.05924*, 2017.

[15] T. Fuji, K. Ito, K. Matsumoto, and K. Yano, "Deep multi-agent reinforcement learning using dnn-weight evolution to optimize supply chain performance," in *Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018.

[16] X. Li, J. Zhang, J. Bian, Y. Tong, and T.-Y. Liu, "A cooperative multi-agent reinforcement learning framework for resource balancing in complex logistics network," *arXiv preprint arXiv:1903.00714*, 2019.

[17] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[18] A. Bhatia, P. Varakantham, and A. Kumar, "Resource constrained deep reinforcement learning," *arXiv preprint arXiv:1812.00600*, 2018.

[19] L. Kemmer, H. v. Kleist, D. d. Rochebouet, N. Tziortziotis, and J. Read, "Reinforcement learning for supply chain optimization," in *European Workshop on Reinforcement Learning 14*, 2018, pp. 1–9.