Hindawi
Scientific Programming
Volume 2022, Article ID 8750580, 12 pages
https://doi.org/10.1155/2022/8750580

Research Article

Research on Fresh Product Logistics Transportation Scheduling Based on Deep Reinforcement Learning

Hongshen Yu

Changchun University of Finance and Economics, Changchun 130177, Jilin, China

Correspondence should be addressed to Hongshen Yu; yuhongs301@ccufe.edu.cn

Received 24 November 2021; Revised 16 December 2021; Accepted 23 December 2021; Published 14 February 2022

Academic Editor: Baiyuan Ding

With the improvement of the economic level, people's quality of life continues to improve, the demand for fresh food is increasing, and the logistics of fresh products is also developing rapidly. We effectively balance the relationship between transportation costs and service levels in fresh product logistics and transportation businesses, improve the transportation capacity and efficiency of logistics transportation businesses, and improve the resource utilization of businesses. It is important for the development of the logistics transportation scheduling industry for fresh products. *Significance*. Based on this, this paper proposes a DNQ algorithm based on pointer network, which solves the single fresh product distribution service center-regional efficient logistics scheduling problem, and a feasible logistics transportation scheduling scheme can be obtained through simulation experiments. The simulation results show that the algorithm is superior to other common intelligent algorithms in terms of accuracy and stability, which proves that the algorithm is effective and feasible (the research results cannot be directly shown in the abstract and need to be supplemented) At the same time, it further explored the DNQ algorithm to improve the correction network, which can solve the complex problem of multiple fresh product distribution service centers-regional efficient logistics scheduling. It is a successful attempt to improve the solution algorithm. Complex logistics and transportation scheduling problems provide ideas and have good guidance and reference significance.

1. Introduction

With the further improvement of residents' living standards, the scale of online and offline demand in the fresh product market has gradually expanded. How to achieve accurate delivery of customer needs has put forward higher requirements for the efficient transportation of fresh product logistics. The logistics and transportation of fresh products need to be comprehensively considered from the aspects of precooling and refrigeration technology, fresh product logistics links, etc.; for the huge fresh market demand, the continuous improvement of the logistics distribution optimization problem of fresh products becomes the problem especially important. According to the latest research, the transportation cost of fresh product logistics is 40%–60% higher than the logistics of ordinary goods, and the cost incurred is on the rise. How to plan vehicle scheduling scientifically and reasonably plays a very important role in reducing distribution costs and total logistics costs, as well as providing better services to consumers. For the optimization of the logistics and distribution vehicle scheduling of agricultural products and fresh products, it is necessary to apply modern information technology to carry out real-time and accurate positioning of agricultural products cold-chain transportation vehicles, driving data collection, customer information collection and analysis, agricultural product transportation volume analysis, and distribution route planning to achieve distribution. Intelligent and efficient vehicle scheduling can meet consumer demand at the lowest cost and guarantee the quality of agricultural products. This article summarizes the life scenarios of residents, the distribution network of fresh products is scattered, the production and sales are separated, the distribution path is too long, the transportation capacity is wasted, the vehicle loading rate is too low, and the transportation efficiency is low. Research on the efficient logistics transportation

scheduling problem of the distribution center has certain guidance and reference significance for solving the complex logistics transportation scheduling problem of fresh products.

2. Related Work

The research content of fresh product logistics and transportation mainly focuses on the development status and countermeasures of fresh products, safety management, route optimization, precooling and cold storage technology, and optimization of fresh product logistics links. A brief introduction to the research contents of fresh product logistics and transportation in recent years is now given.

In terms of qualitative research, Macheka et al. [1] pointed out that the core technology and core equipment of fresh product logistics are backward, the cold chain supply chain is severely disconnected, and smooth operations cannot be formed. Fresh product logistics lacks systematic laws and regulations and authoritative logistics standards and other internal development contradictions. Musavi and Bozorgi-Amiri [2] put forward that my country's fresh food logistics is facing problems such as reprocessing and packaging and last-mile delivery. Neves-Moreira et al. [3] used the fuzzy analytic hierarchy process to analyze the weights of IS indicators and summarized three indicators that have the greatest impact on the development of agricultural and fresh products logistics. Rahimi et al. [4] found out the restrictive factors restricting the development of my country's agricultural and fresh products logistics from the five perspectives of policy, personnel, hardware, software, and management. Lei [5] analyzed the development conditions of my country's fresh agricultural products and fresh product logistics from economic factors and social factors. Samuel Mercier [6] summarized and analyzed the current situation of the food cold chain in Canada and believed that the main problems of the cold chain include seasonal outbreaks of food waste and food poisoning, cold chain transportation in summer, and products produced in severe cold areas, freezing damage, and other issues. The research of Xin et al. [7–9] focused on the technical improvement of the fresh-keeping link in the logistics of fresh food products. New Zealand scholar James K. Carson [10] focused on how to optimize the preservation process to extend the life cycle of products on the shelf. In the case study, he introduced how to improve the carton packaging and stacking methods of kiwifruit to extend the life of the product. Marlies de Keizer [11] believes that the different spoilage rates of different types of products will have an impact on the effect of the fresh agricultural product logistics network. He believes that the product delivery cycle and the different spoilage rates of products should be considered as factors in the logistics network model.

In terms of quantitative research, the research on the logistics of fresh agricultural products and fresh products has a wide range of research. Sun Zhidan and others used cost-saving methods and human ant colony algorithms to solve the path optimization problem of fresh agricultural products and fresh products logistics distribution [12–14]. Ge Changfei et al. [15] optimized the existing logistics and transportation model of fresh food products from the perspective of product quality and order quantity. Huang Chunhui's research [16] focuses on the emergency complementation of intersupplier inventory in the food cold chain and the optimization of the entire fresh product logistics and transportation network. Zhang Wenfeng [17] established an optimization model for the layout of fresh product logistics outlets with the goal of outlet operating costs and construction costs and used a particle swarm algorithm to solve the model. Xiunian Zhan [18] established a value-based decision model. In order to improve the efficiency of the cold chain of fresh products, Soto-Silva et al. [19] designed three models to improve the efficiency of the cold chain of fresh products: fresh product purchasing model, fresh product storage model, and the combination of purchasing, transportation, and warehousing. The model was finally validated with a Chilean dried apple processing plant. Ghezavati et al. [20–22] established an optimization model for the supply chain distribution network of fresh agricultural products from the place of production to the customer with the maximum benefit as the objective function and verified the effectiveness of the model. Marco Bortolini and others created a three-objective distribution planning model to optimize the logistics distribution path of fresh products with the minimum cost as the goal [22, 23].In solving the scheduling problem model, Zhang Jianfeng and others used a deep recurrent neural network model embedded with a pointer network to solve the job shop scheduling problem.

3. Related Theories

In reinforcement learning, modeling strategy $\pi \vee (a \vee s)$ and value function $V^{\pi}(s)$ and $Q^{\pi}(s, a)$ are generally required. Early reinforcement learning algorithms mainly focused on the discrete and limited problems of states and actions, and tables can be used to record these probabilities. But in many practical problems, the number of states and actions of some tasks is very large. In order to effectively solve these problems, in order to be able to design a stronger strategy function, so that the agent can deal with complex environments, learn better strategies, and have better generalization capabilities, the pointer network model is used.

3.1. Pointer Network Model. Compared with the traditional local search algorithm, the pointer network model has two obvious advantages. First, compared with the traditional local search algorithm, the pointer network model responds faster when entering new data. In the traditional heuristic algorithm, when a new case is entered, it needs to be recalculated without any experience. Second, the output of the pointer network model is related to the length of the input but has nothing to do with the dictionary. The versatility of the model is greatly improved. The same model is used in the same type of problem, saving the trouble of training the model in each case.

Pointer Networks (Ptr-Nets) are composed of an encoder (Encoder), a decoder (Decoder), and a shallow neural network. The encoder is a two-way long and short-term memory network, and the decoder is a single Xiang's long and short-term memory network [24]. The structure of the pointer network is shown in Figure 1. [24].

Sequence to sequence is a conditional sequence generation problem. Given a sequence $x_{1:s}$, generate another sequence $y_{1:T}$. The length S of the input sequence and the length of the output sequence can be different. The goal of the sequence-to-sequence model is to estimate the conditional probability as

$$p_\theta(y_{1:T} \vee x_{1:S}) = \prod_{t=1}^{T}\left(y\middle|t \vee y_{1:(t-1),x_{1:S}}\right), \tag{1}$$

where is an element in the set V. Given a set of training data, the maximum likelihood estimation can be used to train the model parameters.

$$\max_\theta \sum_{n=1}^{N} \log p_\theta\left(y_{1:T_n} \vee x_{1:S_n}\right). \tag{2}$$

After the training is completed, the model can generate the most likely target sequence based on an input sequence X,

$$\hat{y} = \arg\max_y p_\theta(y \vee x). \tag{3}$$

The specific generation process can be completed by the greedy method or beam search.

Similar to the general sequence generation model, conditional probability can be implemented using various neural networks. The most direct way to achieve sequence to sequence is to use two recurrent neural networks to encode and decode, respectively, which is also known as the Encoder-Decoder model.

3.1.1. Encoder. First, use a recurrent neural network Ren. To encode the input sequence $x\_(1{:}s)$ to obtain a fixed-dimensional vector u, u is generally the hidden state of the encoding recurrent neural network at the last moment.

$$\begin{aligned} h_t^e &= f_{enc}\left(h_{t-1}^e, e_{x_{t-1}}, \theta_{enc}\right), \forall t \in [1{:}S], \\ u &= h_S^e. \end{aligned} \tag{4}$$

Among them, $f_{\text{enc}}(\cdot)$ is the coded recurrent neural network, its parameter is θ_{enc}, and e_x is the vector of x.

3.1.2. Decoder. When generating the target sequence, another recurrent neural network R_{dec} is used to decode. In the t-th step of the decoding process, the generated prefix sequence is $y_{1:t-1}$. Let $\overset{t}{h}$ denote the hidden state in the network R_{dec}; $o_t \in (0,1)^{V\vee}$ is the posterior probability of all words in the vocabulary; then,

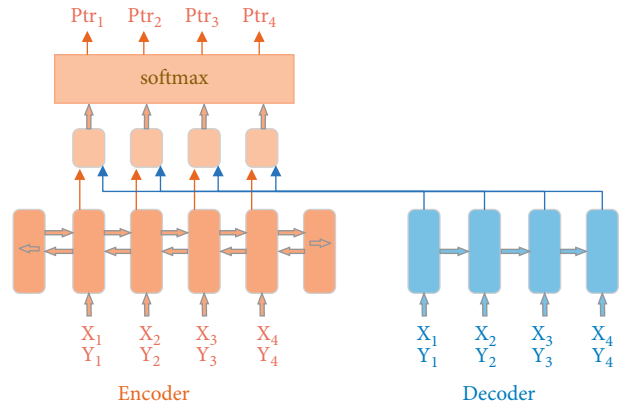

Figure 1: Schematic diagram of pointer network structure.

$$\begin{aligned} h_0^d &= u, \\ h_t^d &= f_{\text{dec}}\left(h_{t-1}^d, e_{y_{t-1}}, \theta_{\text{dec}}\right), \\ o_t &= g\left(h_t^d, \theta_0\right), \end{aligned} \tag{5}$$

where $f_{\text{enc}}(\cdot)$ is the Decoder recurrent neural network, $g(\cdot)$ is the last layer of the feedforward neural network with the softmax function, θ_{dec} and θ_0 are the parameters of the neural network, e_y is the vector of y, and y_θ is a special symbol, such as "$."

Although the sequence-to-sequence model based on the recurrent neural network is relatively easy to implement in theory, there are some shortcomings: (1) the length of the input sequence is difficult to determine, so the capacity of the encoding vector cannot be determined, and the information of the input sequence cannot be saved; (2) length dependence is an unavoidable problem of cyclic neural networks. If the input sequence is too long, it is easy to lose information.

In order to obtain more abundant input sequence information, the attention mechanism can be used to select useful information from the input sequence in each step. The hidden layers of Encoder and Decoder are represented by $(e_1, \ldots, e_s)$ and $(d_1, \ldots, d_{t(T)})$; then,

$$\begin{aligned} h_0^d &= u, \\ h_t^d &= f_{\text{dec}}\left(h_{t-1}^d, e_{y_{t-1}}, \theta_{\text{dec}}\right), \\ o_t &= g\left(h_t^d, \theta_0\right). \end{aligned} \tag{6}$$

In the t-th step of the decoding process, the model takes the node e_j of the hidden layer of input j and the current state d_t as simple input through the network, and the output obtained is the corresponding input j at this time, and all the inputs are calculated. After that, the normalized processing is performed by softmax, and then the processing result is weighted and summed with the node e_j, which is the probability distribution of the next input.

The attention mechanism can be used alone, but more often as a component in a neural network. The attention mechanism is mainly used for information screening, selecting relevant information from the input information. The attention mechanism can be divided into two steps: one is to calculate the attention distribution α, and the other is to calculate the weighted average of the input information according to α [25].

The input of the pointer network [26] is a vector sequence $X = x_1, \ldots, x_{n\_}$ n of length n, and the output is a subscript sequence $c_{1:m} = c_1, c_2, \ldots, c_m, c_i \in [1, n], \forall i$. Unlike general sequence-to-sequence tasks, the output sequence here is the subscript (index) of the input sequence.

The calculation formula of the conditional probability $p(c_{1:m} \vee x_{1:n})$ is

$$\begin{aligned} p\left(c_{1:m} \vee x_{1:n}\right) &= \prod_{i=1}^{m} p\left(c_i \vee c_{1:m}, x_{1:n}\right) \\ &\approx \prod_{i=1}^{m} p\left(c_i \vee c_1, c_2, \cdots, c_m\right). \end{aligned} \tag{7}$$

In formula (7), the conditional probability $p(c_i \vee x_{c_1}, \ldots, x_{c_{i-1}}, x_{c_{1:n}})$ is calculated using the attention distribution. For $x_{c_1}, \ldots, x_{c_{i-1}}, x_{c_{1:n}}$, if a recurrent neural network is used to encode the vector h_i, then

$$p\left(c_{1:m} \vee x_{1:n}\right) = \operatorname{softmax}\left(s_{i,j}\right). \tag{8}$$

In formula (8) $s_{i,j}$ is the unnormalized attention distribution in each input vector when decoding at the i-th step:

$$s_{i,j} = v^T \tanh\left(W x_j + U h_i\right), \quad \forall j \in [1, n]. \tag{9}$$

In formula (14), v, W, and U represent learnable parameters.

3.2. Deep Reinforcement Learning Technology. Agents can perceive their environment through sensors and act on anything in the environment with the help of actuators. Reinforcement learning can obtain the optimal strategy for sequential decision-making by maximizing the cumulative reward value that the agent obtains from the environment, which is more suitable for exploration of effective strategies to solve problems, with strong decision-making ability but lack of perception ability.

Deep learning is another important research field of machine learning, and it has made considerable progress in recent years. DL mainly combines low-level features of things through multilayer neural networks and nonlinear transformations to form abstract and easily distinguishable high-level feature representations to realize effective perception and expression of things. Although deep learning has a strong perception ability, its decision-making ability is insufficient.

Deep reinforcement learning integrates deep learning and reinforcement learning. It can give full play to their respective advantages and integrate perception and decision-making capabilities. Specifically, deep learning methods are used to obtain abstractions of large-scale input data. Representation is used as the environmental observation value for reinforcement learning, thereby obtaining the optimization of the problem-solving strategy. The basic process of deep reinforcement learning is shown in Figure 2.

The basic iterative process of deep reinforcement learning is as follows.

In the first step, the agent obtains the observations about the environment state by interacting with the environment and uses the deep learning method to realize the perception of the environment based on the obtained environment state information and determine the characteristics of the environment state. The second step is to use reinforcement learning methods to map the current state features to corresponding actions through a certain strategy.

In the third step, the environment gives feedback to the action taken by the agent and forms the next state, returning to the first step.

Deep Q-Learning (DQN, Deep Q Network) is a typical deep reinforcement learning algorithm to solve the problem that the Q-Learning algorithm cannot be applied when the state space is too high or the action space is continuous.

The basic idea of DQN is to use a neural network instead of the action value function, use a neural network to receive state-action pair as input, and output the corresponding Q-value. DQN algorithm converts the original Q-Learning problem into the corresponding deep learning problem; namely, the problem of updating the Q-table is transformed into a problem of approximating the Q function using neural network fitting, as shown in Figure 3.

DQN introduces an experience playback mechanism and dual network mechanism in the training process.

The experience playback mechanism mainly relies on the storage unit of the experience pool. At each time step, the trajectory transfer sample $e_t = (s_t, a_t, r_t, s_{t+1})$ obtained from the interaction between the agent and the environment is stored in the experience pool $P = (e_1, \ldots, e_t)$, as shown in Figure 4. When training the DQN network, each time a small batch of trajectory transfer samples is randomly taken from P, this approach effectively reduces the correlation between training samples.

In addition to using a neural network to approximate the current state-action value function, DQN also uses another network alone to generate the target Q-value. Specifically, $f(s, a \vee \theta)$ represents the output of the current Q-value network; $f(s, a \vee \theta')$ represents the output of the target Q-value network; usually, $Y = r + \gamma \max_{a'} f'(s, a \vee \theta')$ approximately represents the goal of value function optimization, namely, the target value Q_{target}. During each iteration, the parameters θ of the current Q-value network will be updated. After several iterations, the parameters θ of the current Q-value network will be used to replace the target Q-value network parameters θ', so that the target will be within a period of time. The Q-value remains unchanged, thereby reducing the correlation between the current Q-value and the target Q-value, as shown in Figure 5.

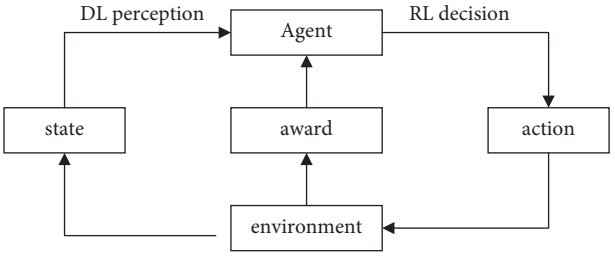

Figure 2: The basic process of deep reinforcement learning.

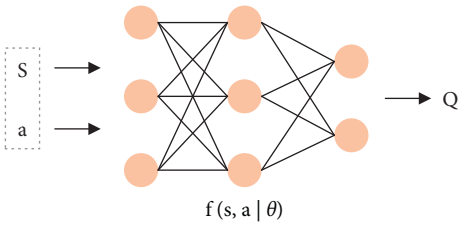

Figure 3: Schematic diagram of neural network fitting Q function.

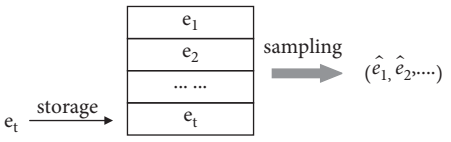

Figure 4: Schematic diagram of experience playback mechanism in DQN.

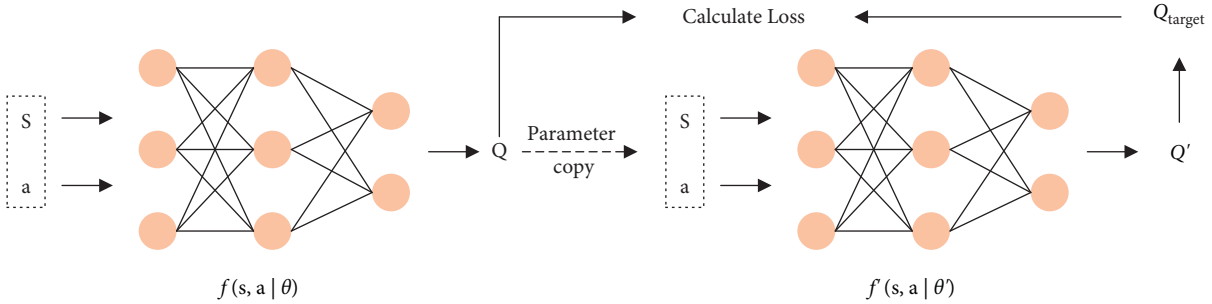

Figure 5: Schematic diagram of DQN dual network.

3.3. DQN Algorithm. In Q-Learning, let $Q(s,a)$ directly estimate the optimal state value function $^{Q}(s,a)$. The estimation method of the Q function is

$$Q(s,a) \longleftarrow Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q\left(s',a'\right) - Q(s,a)\right). \tag{10}$$

When the Q-Learning algorithm is used in practical problems, the problem of dimensional explosion often occurs; that is, the state space or action space is often very large, and Q-table storage cannot be used, because the hardware cost of storage and the time cost of querying are very high. In the continuous reading state and action space, the dimensions are even greater, and obviously, storage is impossible. So we need a function $Q_{\varnothing}(s,a)$ fitted value function $Q^{\pi}(s,a)$.

$$Q_{\varnothing}(s,a) \approx Q^{\pi}(s,a). \tag{11}$$

The function $Q_{\varnothing}(s,a)$ is usually a function with a parameter of $\varnothing$, usually a neural network, and the output is a real number. If the action is a finite discrete m action "$a_1, \ldots, a_m$," we can make the Q network output a dimensional vector, where each dimension is represented by $Q_{\varnothing}(s,a_i)$, and the corresponding value function $Q(s,a_i)$ approximate value.

$$Q_{\varnothing}(s) = \begin{matrix} Q_{\varnothing}(s,a_1) \\ \vdots \\ Q_{\varnothing}(s,a_m) \end{matrix} \approx \begin{matrix} Q^{\pi}(s,a_1) \\ \vdots \\ Q^{\pi}(s,a_m) \end{matrix}. \tag{12}$$

It is necessary to learn a function $Q_{\varnothing}(s,a)$ containing the parameter $\varnothing$ to approximate the value function $Q^{\pi}(s,a)$. By fitting the mapping relationship between Q-value and s-a, the limited Q-value is replaced. To obtain good reinforcement learning results, a better fitting function is needed. In order to obtain the required fitting function, the pointer network introduced in Section 3.1 will be used. Driven by this kind of thinking, the DeepMind team proposed the Deep Q-Learning Network (Deep Q-Learning Network, DQN). According to the knowledge of the Q-Learning algorithm in the previous section, the core of each iteration of the algorithm is to seek

$$Q(s,a) = E_{s',a'}\left[r + \max_{a'} Q(s',a')\right]. \tag{13}$$

Introduce a deep neural network with parameter $\varnothing$ to fit $Q_{\varnothing}(s,a) \approx {}^{Q}(s,a)$, and train the network by gradient descent at each iteration, and the sample mean square error is caused by the loss function instead:

$$\begin{aligned} L_i(\varnothing_i) &= E_{s,a}\left[\left(y_i - Q_{\varnothing_i}(s,a)\right)^2\right], \\ y_i &= E_{s',a'}\left[r + \max_{a'} Q_{\varnothing_{i-1}}(s',a')\right]. \end{aligned} \tag{14}$$

The gradient of the loss function on $\varnothing$ is

$$\begin{aligned} \nabla_{\varnothing_i} L_i(\varnothing_i) &= E_{s,a,s'} \\ &\cdot \left[\left(r + \max_{a'} Q_{\varnothing_{i-1}}(s',a') - Q_{\varnothing_i}(s,a)\right)\nabla_{\varnothing_i} Q_{\varnothing_i}(s,a)\right]. \end{aligned} \tag{15}$$

In order to make the training of deep neural networks more efficient and convergent, the deep Q network algorithm also introduces an experience playback mechanism and a target network mechanism [27].

4. Optimization Algorithm for Logistics Transportation of Fresh Products Based on Deep Reinforcement Learning

4.1. Algorithm Structure Design

4.1.1. Design of DNQ Algorithm Structure Based on Pointer Network. For tasks that require the perception of high-dimensional raw input and output decision-making control at the same time, deep reinforcement learning algorithms have made huge and substantial progress. Because the convolutional neural network has natural advantages in image processing, the deep Q network has a level of competence with humans when it solves the complex problems related to image processing but is close to the real environment. However, the problems faced by deep reinforcement learning often have a strong time dependence. If there is a delay in the rewards in the environment, the agent needs a long number of steps to optimize the strategy. Faced with such problems, the deep Q network is not good. Performance and recurrent neural networks are suitable for handling problems related to time series [28]. Based on this, this chapter uses the pointer network model introduced in Section 3.1 to replace the convolutional neural network in the traditional deep Q network model. This modified deep Q network is called the deep Q network based on the pointer network (Deep Q Network) (DQN-PN).

In the structure diagram of the pointer network model shown in Figure 1, the encoder of the model is a two-way long and short-term memory network, and the decoder is a one-way long and short-term memory network. Assuming that a single fresh product logistics service center has 4 customer points in the deterministic logistics transportation scheduling problem, the location information of the 4 customer points is input into the encoder, and the output of the encoder each time focuses on the customer point information and other 4 points. Concerning the information of a customer point, such output is called the node of the customer point. We input the information of the customer point where the agent is at this time into the decoder and input the node containing the previous path information and the 4 customer points obtained before into a shallow neural network, and 4 outputs can be obtained. The output is normalized by softmax, and what is obtained is the location information of the next client point that the agent will visit. The size of the output dictionary is the number of client points input to the encoder.

4.1.2. Design of DNQ Algorithm Structure Based on Improved Pointer Network. When using the value function and strategy function of the pointer network model-fitting algorithm proposed in the previous section, it is found that for the efficient logistics transportation scheduling problem of multiple fresh product service centers, after the distribution service center is added, the number of transportation vehicles will increase or decrease at the same time. The input becomes very complicated; therefore, the encoder input becomes inefficient, which affects the performance of the algorithm. In order to improve the efficiency of input, this section improves the pointer network model, simplifies its structure, and makes it suitable for solving the efficient logistics transportation scheduling problem of multiple fresh product distribution service centers.

The RNN encoder adds additional complexity to the encoder, but it is not actually necessary, and the method can be made more versatile by omitting it. The encoder RNN is necessary only when the input transfers order information, and when there is no meaningful order in the input set, it is not necessary to use it for combinatorial optimization problems in the encoder. Therefore, in the model established by et al., the encoder RNN is simply omitted, and the embedded input is used directly instead of the hidden state of the RNN. With this modification, much of the computational complexity disappears without reducing the efficiency of the model.

As shown in Figure 6, the model consists of two main components. The first is a set of embeddings, which maps the input to a D-dimensional vector space. There may be multiple embeddings corresponding to different elements of the input, but they are shared between the inputs. The

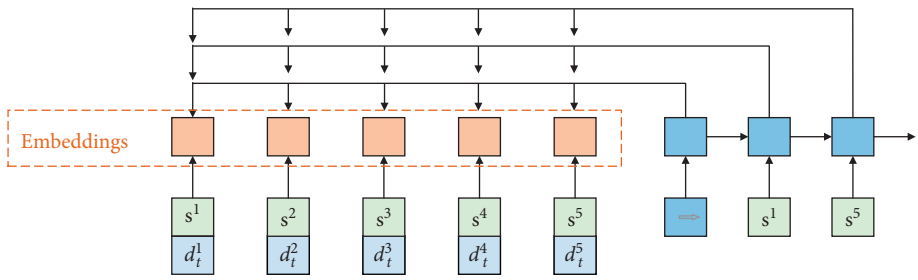

Figure 6: Improved pointer network model.

second component of the model is the decoder, which points to the input of each decoding step. Use RNN to simulate the decoder network, and provide static elements as input to the decoder network, and dynamic elements can also be the input of the decoder.

In the model in Figure 6, the embedding layer on the left maps the input to a high-dimensional vector space. On the right, the RNN decoder stores the information of the decoded sequence. Then, the hidden state and embedded input of the RNN use the attention mechanism to generate a probability distribution on the next input.

4.2. Algorithm Steps. Through the introduction of the pointer network and the deep Q network in the previous chapter, the specific algorithm steps of the deep Q network based on the pointer network are as follows: d_t^1

4.3. Algorithm Environment Design. From the steps of Algorithm 1 in the previous section, it can be seen that it essentially uses the reinforcement learning of the deep learning fitting value function. Therefore, the process of the deep Q network algorithm based on the pointer network is also the continuous interaction between the agent and the environment. If we want to use this algorithm, we need to design a single fresh product efficient logistics and transportation scheduling problem, including the state space S design, the action space A design, and the action reward r design. The discount rate γ, the learning rate α, and the parameter update interval C are relatively simple inputs, and their values will be given in the experimental part, and no design will be done in this section.

4.3.1. State Space S Design. One of the inputs of the algorithm is the state space S, so the state space S is designed first. According to the algorithm structure design ideas in Section 4.1, the fresh product logistics transportation scheduling problem model is determined. At time t, the state s_t of the environment at this time is represented by a set of vectors $(p, q_t, j_1, j_2, \ldots, j_N)$. The state of the environment is constantly changing over time. Although time is a continuous variable, according to the Markov property of reinforcement learning, the state change trajectory of the environment can be divided into countless discrete states in a certain short period of time. Here, the unit is separated by the length of time the delivery vehicle moves from one customer point to another customer point. Assuming there are T states, in the fresh product logistics transportation scheduling problem, $T = N + 2m$, m is the number of times to return to the distribution center. All states form a state set, that is, the state space S, $S = \{s_t: 0, \ldots, T\}$.

4.3.2. Action Space A Design. Another input of the algorithm is the action space A. The output of the deep Q network algorithm based on the pointer network is the mapping relationship function between state and action. Therefore, the design of the action space is indispensable for environmental design. In the logistics and transportation scheduling problem of fresh products, in each step of the iteration, the agent chooses the next customer point to be visited which is the action. The agent can only choose one action to execute. In addition, the time discretization of the action space is consistent with the state space of the environment. In the problem to be solved, the transportation vehicle needs to depart from the distribution service center and eventually return to the distribution service center, so the default " $a_0 = a_T = p_0$, that is, the first action and the last action select the distribution center. At other times, it has never chosen one of the customer points visited. If the remaining load capacity of the vehicle cannot meet the needs of the customer point, the strategy is to return the vehicle to the distribution service center to unload the goods and then continue to provide services.

4.3.3. Action Reward r Design. Action reward is the key to the algorithm because it determines the learning direction and efficiency of the deep Q network algorithm of the pointer network. In each step of the iterative process, the environment will give a reward value according to the current state and the action chosen to be executed, and the evaluation and improvement of the strategy will be carried out according to the reward r. In the fresh product logistics transportation scheduling problem, the goal is to make the total visit distance the smallest, and the reward for each step can be represented by the distance between two customer points.

5. Simulation Experiment and Result Analysis

5.1. Experimental Environment and Parameter Settings. All the experimental algorithms in this article are implemented on the Linux system based on the Tensorflow

platform using Python. The computer's CPU is Intel Core i7 8700, the CPU frequency is 3.2 GHz, the graphics card is NVIDIA GeForce GTX 1080Ti, and the RAM is 16 GB.

The parameter settings are shown in Table 1.

5.2. Simulation Experiment Results and Analysis. Simulation Experiment 1: solve a single fresh product distribution service center-regional efficient logistics scheduling problem.

Since there is no public cvRPSD calculation example set, in order to verify the solution effect of Deep Q Network based on pointer network (DQN-PN), in this section, the experimental data will be obtained from the benchmark example of the deterministic CVRP example set (http://www.branchandcut.org//data/), and then through the improvement of the deterministic CVRP example set, the experiments needed in this section will be obtained. CVRPSD calculation example set is as follows. Compared with the CVRP calculation example, other information is not changed except for the needs of the customer. Assume that the random demand of the customer point in this experiment obeys a discrete distribution, and the expected value of the random demand is equal to the demand of the corresponding customer point in the CVRP calculation example. The random distribution function of the demand of customer point i is shown in Table 2. The random distribution function of the demand of other customer points obeys the same formula as that of customer point i, and the difference lies in the different distribution probabilities. Table 2 is only used to show the form of the data and does not have the data details of the examples. The space is limited, and the data details of each customer point are no longer listed. At the same time, since no result set of using deep reinforcement learning to solve such problems was found, the results of the frequently used simulated annealing algorithm (simulated annealing, SA) were used as a reference.

In Table 3, BKS represents the best-known value of the calculation example; BS represents the best value obtained in the algorithm experiment; AS is the average value obtained in the algorithm experiment; CT represents the average time consumed in the algorithm experiment and the unit of time consumption for seconds.

Combining Table 3 and Figure 7 shows that for different types and scales of calculation examples, the deep Q network algorithm based on the pointer network is effective and stable for solving the single fresh product distribution service center-regional efficient logistics scheduling problem. There is a higher solution accuracy. When the solution time is not much different, most of the results are better than the solution results of the simulated annealing algorithm. This shows that the method designed in the article can effectively solve the problem of single fresh product distribution service center-regional efficient logistics scheduling. The main reason is that the pointer network model network with better approximation performance is adopted.

Simulation Experiment 2: solve the problem of multiple fresh product distribution service centers-regional efficient logistics scheduling.

In order to verify the effectiveness of the improvement based on the improved DNQ algorithm of pointer network (I-DNQ-NP) in solving multiple fresh product distribution service centers-regional efficient logistics scheduling problems, this section designs and the previous one contrast experiment of section algorithm. Since there is no public MDCVRPSDTW example set, and it is not meaningful to directly use the result data of simulation experiment 1, this section changes the examples Q1 and Q2 in the related literature [30] to change the definite requirements of each customer point used after being a random demand variable.

Calculation example Q1 is to solve the logistics scheduling problem of multiple fresh product distribution service centers with 3 distribution service centers and 15 customer points. The specific information of the distribution service center is shown in Table 4, and the specific information of the customer points is shown in Table 5.

Example Q2 is the logistics scheduling problem of multiple fresh product distribution service centers with 4 distribution service centers and 25 customer points. The specific information of the distribution service center is shown in Table 6, and the specific information of customer points is shown in Table 7.

In Tables 4–7, the X and Y coordinates represent the coordinates of the customer point and the coordinate position of the freight yard; K represents the number of vehicles available in the distribution center; Q represents the carrying capacity of each vehicle in the distribution center; I represents the basic salary of the driver for each vehicle startup in the distribution service center; H represents the hourly salary of the driver of the distribution center; E is the expected value of the customer's random demand; ET is the time when the service can be started; LT is the time when the customer can be serviced.

The comparison of the experimental results is shown in Table 8. In the table, TO in BS represents the timeout time of the optimal solution, DIS in BS represents the path length of the optimal solution, TO in AS represents the average timeout time of the solution obtained by solving the calculation example 20 times, and DIS in AS represents the average path length of the solution obtained by solving the calculation example 20 times, F is the driver's salary, CT is the calculation time, and the unit of time is seconds.

Through simulation test two, it can be known that under the conditions of the above experimental parameters, both DNQ-NP and I-DNQ-NP can obtain feasible solutions that do not exceed the carrying capacity of the transport vehicle and meet the customer's point time. Analyzing the data in Table 8 shows that under the same parameter conditions, I-DNQ-NP is equivalent to DNQ-NP in terms of solution quality and accuracy (no difference between the optimal value and the average value). However, it has been drastically reduced. The experimental results prove that the modification of the pointer network is effective. The RNN encoder does add additional complexity to the encoder, simply omitting the encoder RNN, and directly using embedded input instead of RNN hidden state. Through this modification, a lot of computational complexity disappears, which greatly improves the efficiency of the model.

Input: State space S, action space A, discount rate γ, learning rate α, parameter update interval C
(1) Initialize experience pool D, the capacity is N_D;
(2) Randomly initialize the parameters of the Q network $\varnothing$;
(3) Randomly initialize the parameters of the target Q network $\widehat{\varnothing} = \varnothing$;
(4) repeat
(5) Initialize the initial state s;
(6) repeat
(7) In state S, choose action $a = \pi^{\varepsilon}$
(8) Perform action a, observe the environment, get an instant reward r and a new state s';
(9) Put s, a, r, s' into D;
(10) Sample ss,aa,rr,ss' from D;
(11) $y = \begin{cases} rr, ss' \text{Terminal state,} \\ rr + \gamma \max\limits_{a'} Q_{\varnothing}(ss', a') \text{other.} \end{cases}$;
(12) Use $(y - Q_{\varnothing}(ss, aa))^2$ as the loss function to train the Q network;
(13) $s \longleftarrow s'$;
(14) Every C steps, $\widehat{\varnothing} = \varnothing$;
(15) Until s is the termination state;
(16) Until $\forall s, a$, $Q_{\varnothing}(s, a)$ converges;
output : Q network $Q_{\varnothing}(s, a)$

Algorithm 1: Deep Q network based on pointer network

Table 1: Parameter setting [29].

Parameter	Parameter description	Value
Enc-net	Encoder parameters	$(N+2) \times 256 + (N+2) \times 256^2$
Dec-net	Decoder parameters	$(N+2) \times 256 + (N+2) \times 256^2$
γ	Discount rate	1.0
α	Learning rate	0.001
β	Learning rate	0.002
ε	Parameters of ε − greedy method	[0.1, 1.0]
C	Parameter update interval	0
N_D	Capacity of experience pool D	5000

Table 2: The random distribution function table of the demand of customer i.

Demand	ζ_1	ζ_2	⋯	ζ_k
Probability distributions	$p_{i,1}$	$p_{i,2}$	⋯	$p_{i,k}$

Table 3: Comparison of result sets of CVRPSD calculation examples.

Examples	BKS	SA		DNQ-PN		
		BS	CT	BS	AS	CT
E-n22-k4	375	411.57	104.1	406	409	84.6
P-n23-k8	529	619.53	21.1	580	642	17.2
A-n32-k5	784	853.6	199.8	822	868	138.4
A-n33-k4	661	704.2	178.2	695	730	196.5
E-n33-k4	835	850.27	371.7	813	855	239.2
A-n34-k5	778	826.87	236.4	788	823	240.7
A-n36-k5	799	858.71	276.1	815	823	235.2
A-n39-k5	822	869.18	257.6	754	809	243.6
P-n40-k5	458	472.5	367.2	430	495	338.1
A-n44-k6	937	1025.48	281.4	1019	1136	253.0
A-n45-k7	1146	1264.99	216.0	1213	1285	248.9
P-n50-k10	696	760.94	150.6	732	764	157.6
E-n51-k5	521	552.26	586.0	594	621	445.8
A-n55-k9	1073	1179.11	265.5	1149	1269	248.5
A-n60-k9	1354	1529.82	393.7	1477	1504	354.8

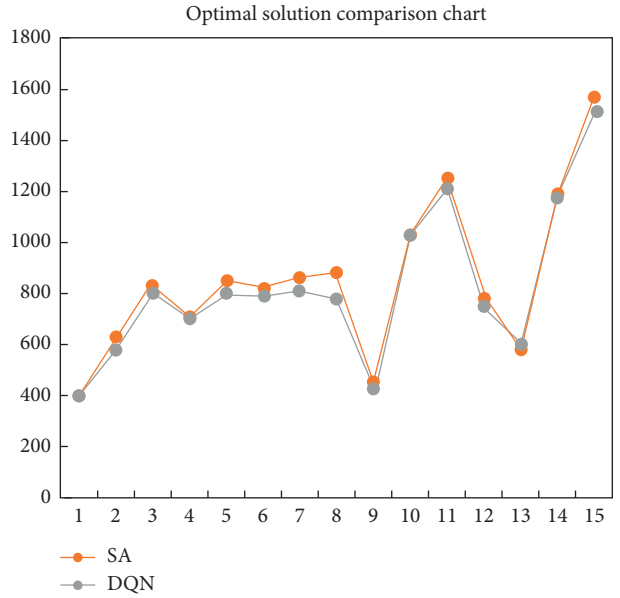

Figure 7: Comparison of optimal solutions.

Table 4: Calculation example Q1 distribution service center information table.

Distribution Service center	X coordinate	Y coordinate	K	Q	I	H
1	60	50	2	100	1000	10
2	10	80	2	100	1200	12
3	100	80	2	100	1500	15

Table 5: Example Q1 customer point information table.

customer	X coordinate	Y coordinate	$E(j)$	ET	LT
1	9	115	10	140	510
2	80	86	30	100	260
3	90	112	10	65	146
4	40	112	10	410	640
5	72	65	10	15	67
6	60	99	20	265	710
7	25	66	20	50	300
8	10	60	20	10	70
9	38	70	10	334	705
10	35	66	10	100	350
11	35	89	10	260	450
12	25	105	20	152	321
13	52	75	30	30	135
14	65	85	10	167	620
15	20	100	40	384	529

Table 6: Calculation example Q2 distribution service center information table.

Distribution Service center	X coordinate	Y coordinate	K	Q	I	H
1	30	80	3	150	1200	12
2	22	20	3	150	1200	12
3	80	90	3	150	1500	15
4	75	40	3	150	1800	18

Table 7: Example Q2 customer point information table.

Customer	X coordinate	Y coordinate	$E(j)$	ET	LT
1	65	40	10	129	450
2	100	86	30	100	170
3	32	66	10	65	146
4	85	90	10	10	70
5	75	65	10	115	367
6	90	69	20	150	210
7	80	46	20	170	225
8	88	98	20	50	95
9	45	70	10	234	605
10	80	66	10	200	450
11	65	29	10	448	505
12	25	85	20	0	100
13	22	75	30	30	92
14	52	85	10	367	467
15	50	80	40	184	429
16	20	95	40	155	548
17	58	65	20	99	148
18	35	95	20	179	254
19	45	90	10	258	345
20	30	50	10	10	73
21	39	42	20	310	532
22	28	36	20	12	433
23	18	41	10	102	397
24	25	20	10	0	60
25	5	14	40	250	300

Table 8: Comparison of experimental results.

			Q1	Q2
DNQ-NP	BS	TO	0	0
		DIS	4450	6267
		F	10055	16632
	AS	TO	0	0
		DIS	4521	6748.5
		F	10127	17684
	CT		194	383
I-DNQ-NP	BS	TO	0	0
		DIS	4461	6194
		F	10169	15568
	AS	TO	0	0
		DIS	4487	6446.8
		F	10207	17006
	CT		161	328

6. Conclusion

The language expression of the conclusion analysis part is more colloquial. It is recommended to clarify the logical relationship and make a rigorous expression. In today's highly developed material civilization, the daily fresh product needs of residents' lives are more personalized, and at the same time, the effectiveness of customer needs is gradually becoming higher. This further brings challenges to the logistics and transportation scheduling of the fresh product market (how to achieve it). The precise delivery of customer needs puts forward higher requirements for the efficient transportation of fresh product logistics. Based on this, this paper proposes a DNQ algorithm based on a pointer network to solve the single fresh product distribution service center-regional efficient logistics scheduling problem, and a feasible logistics transportation scheduling plan can be obtained through simulation experiments. The simulation results show that the algorithm is superior to other common intelligent algorithms in terms of accuracy and stability, which proves that the algorithm is effective and feasible. Based on this, we further explored the improved DNQ algorithm of the correction network, which can solve multiple fresh product distribution service centers-regional efficient logistics scheduling problems. The improved algorithm reflects its efficient model solving efficiency on complex problems, and through simulation, the analysis can further obtain multiple fresh product distribution service centers-regional efficient logistics scheduling schemes. It is a successful attempt to improve the solution algorithm. It is of great significance for further research on solving the complex logistics transportation scheduling problems of fresh products. Through this research, we can see that the DNQ algorithm of the pointer network has an important position and practical value in solving the actual problem of fresh product logistics transportation scheduling, but the transportation scheduling model in this article is a very abstract model. When the follow-up is further combined with the classic algorithm research, it needs to be further improved in practical applications.

Data Availability

The dataset can be accessed upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Research and Practice of Applied Undergraduate Talent Training Mode from the perspective of AACSB International Certification No. zd20054.

References

[1] L. Macheka, E. Spelt, J. G. A. J. van der Vorst, and P. A. Luning, "Exploration of logistics and quality control activities in view of context characteristics and postharvest losses in fresh produce chains: a case study for tomatoes," *Food Control*, vol. 77, pp. 221–234, 2017.

[2] M. Musavi and A. Bozorgi-Amiri, "A multi-objective sustainable hub location-scheduling problem for perishable food supply chain," *Computers & Industrial Engineering*, vol. 113, pp. 766–778, 2017.

[3] F. Neves-Moreira, D. Pereira da Silva, L. Guimarães, P. Amorim, and B. Almada-Lobo, "The time window assignment vehicle routing problem with product dependent deliveries," *Transportation Research Part E: Logistics and Transportation Review*, vol. 116, pp. 163–183, 2018.

[4] M. Rahimi, A. Baboli, and Y. Rekik, "Multi-objective inventory routing problem: a stochastic model to consider profit, service level and green criteria," *Transportation*

Research Part E: Logistics and Transportation Review, vol. 101, pp. 59–83, 2017.

[5] N. Lei, "Intelligent logistics scheduling model and algorithm based on Internet of Things technology," *Alexandria Engineering Journal*, vol. 61, no. 1, pp. 893–903, 2022.

[6] qingX. Xin, Z. Fu, Z. Zhu, and X. Zhang, "Improved preservation process for table grapes cleaner production in cold chain," *Journal of Cleaner Production*, vol. 221, pp. 1171–1179, 2019.

[7] H. Zhao, S. Liu, C. Tian, G. Yan, and D. Wang, "An overview of current status of cold chain in China," *International Journal of Refrigeration*, vol. 88, no. 88, pp. 483–495, 2018.

[8] X. Zhang and G. Li, "Study on the time-space optimization for cold-chain logistics of fresh agricultural products," in *Proceedings of the 2010 International Conference on Future Information Technology and Management Engineering*, no. 1, pp. 331–333, Changzhou, China, October 2010.

[9] J. K. Carson and A. R. East, "The cold chain in New Zealand-a review," *International Journal of Refrigeration*, vol. 87, no. 87, pp. 185–192, 2018.

[10] M. de Keizer, R. Akkerman, M. Grunow, J. M. Bloemhof, R. Haijema, and J. G. A. V. D. Vorst, "Logistics network design for perishable products with heterogeneous quality decay," *European Journal of Operational Research*, vol. 262, pp. 535–549, 2017.

[11] M. Grazia Speranza, "Trends in transportation and logistics," *European Journal of Operational Research*, vol. 264, no. 3, pp. 830–836, 2018.

[12] H. Shi, L. Sun, Y. Teng, and X. Hu, "An online intelligent vehicle routing and scheduling approach for B2C e-commerce urban logistics distribution," *Procedia Computer Science*, vol. 159, pp. 2533–2542, 2019.

[13] L. Liu, H. Wang, and S. Xing, "Optimization of distribution planning for agricultural products in logistics based on degree of maturity," *Computers and Electronics in Agriculture*, vol. 160, pp. 1–7, 2019.

[14] Z. Lu, Z. Zhuang, Z. Huang, and W. Qin, "A framework of multi-agent based intelligent production logistics system," *Procedia CIRP*, vol. 83, pp. 557–562, 2019.

[15] J. Cheng, B. Yang, M. Gen, Y. J. Jang, and C.-J. Liang, "Machine learning based evolutionary algorithms and optimization for transportation and logistics," *Computers & Industrial Engineering*, vol. 143, Article ID 106372, 2020.

[16] A. V. Barenji, W. M. Wang, Z. Li, and D. A. Guerra-Zubiaga, "Intelligent E-commerce logistics platform using hybrid agent based approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 126, pp. 15–31, 2019.

[17] C. Qi and L. Hu, "Optimization of vehicle routing problem for emergency cold chain logistics based on minimum loss," *Physical Communication*, vol. 40, Article ID 101085, 2020.

[18] X. Zhang, J. S. L. Lam, Shipping mode choice in cold chain from a value-based management perspective," *Transportation Research Part E: Logistics and Transportation Review*, vol. 110, pp. 147–167, 2018.

[19] W. E. Soto-Silva, M. C. González-Araya, M. A. Oliva-Fernández, and L. M. Plà-Aragonés, "Optimizing fresh food logistics for processing: application for a large Chilean apple supply chain," *Computers and Electronics in Agriculture*, vol. 136, pp. 42–57, 2017.

[20] V. R. Ghezavati, S. Hooshyar, and R. Tavakkoli Moghaddam, "A Benders' decomposition algorithm for optimizing distribution of perishable products considering postharvest biological behavior in agri-food supply chain:a case study of tomato," *CEJOR*, vol. 25, pp. 29–54, 2017.

[21] J. Chai, "Study on route optimization of cold chain logistics of fresh food," *Carpathian Journal of Food.Science and Technology*, vol. 8, no. 2, pp. 113–121, 2016.

[22] M. Bortolini, M. Faccio, E. Ferrari, M. Gamberi, and F. Pilati, "Fresh food sustainable distribution: cost, delivery time and carbon footprint three-objective optimization," *Journal of Food Engineering*, vol. 174, no. 1, pp. 56–67, 2016.

[23] S. Y. Wang, Y. Shi, F. Tao, and H. Wen, "Optimization of vehicle routing problem with time windows for cold chain logistics based on carbon tax," *Sustainability*, vol. 9, no. 694, pp. 1–23, 2017.

[24] X. Dai, M. Chen, and Y. Zhou, "Optimal logistics transportation and route planning based on fpga processor real-time system and machine learning," *Microprocessors and Microsystems*, vol. 80, Article ID 103621, 2021.

[25] O. Stopka, K. Jeřábek, and M. Stopková, "Using the operations research methods to address distribution tasks at a city logistics scale," *Transportation Research Procedia*, vol. 44, pp. 348–355, 2020.

[26] O. Vinyals, M. Fortunato, and N. Jaitly, *Pointer Networks*, Computer Science, 2015, https://arxiv.org/abs/1506.03134.

[27] S. R. Cardoso, A. P. F. D. Barbosa-Póvoa, and S. Relvas, "Design and planning of supply chains with integration of reverse logistics activities under demand uncertainty," *European Journal of Operational Research*, vol. 226, no. 3, pp. 436–451, 2013.

[28] Y. Wang, J. Zhang, X. Guan, M. Xu, Z. Wang, and H. Wang, "Collaborative multiple centers fresh logistics distribution network optimization with resource sharing and temperature control constraints," *Expert Systems with Applications*, vol. 165, Article ID 113838, 2021.

[29] B. Bai, K. Zhao, and X. Li, "Application research of nano-storage materials in cold chain logistics of e-commerce fresh agricultural products," *Results in Physics*, vol. 13, Article ID 102049, 2019.

[30] H. M. Stellingwerf, L. H. C. Groeneveld, G. Laporte, A. Kanellopoulos, J. M. Bloemhof, and B. Behdani, "The quality-driven vehicle routing problem: model and application to a case of cooperative logistics," *International Journal of Production Economics*, vol. 231, Article ID 107849, 2021.