Sensors & Transducers, Vol. 242, Issue 3, March 2020, pp. 6-11

Sensors & Transducers

Published by IFSA Publishing, S. L., 2020
http://www.sensorsportal.com

An Experimental Model of Deep Learning Logistics Distribution based on Internet of Things

[1,] Jixiang LU and [2 *]Xiao HAN

[1] School of Mechanical Engineering , Northwestern Polytechnical University, Xi'an 710072, China
[2] Hamburg University of Technology, Hamburg, 21107, Germany
[1] Tel.: +86 13319288957 (CHN)
E-mail: lujixiang@nwpu.edu.cn

Received: 25 February 2019 /Accepted: 25 March 2019 /Published: 31 March 2020

Abstract: Intelligent logistics is not only an important industry in the world, but also an application and research hotspot of artificial intelligence. It is based on the Internet of things (IoT). Improving the user experience is the key of intelligent logistics, which needs to be optimized according to the change of user needs. The design of IoT based intelligent scheduling is one of the key technologies to improve user experience. First, we use the deep learning network to learn the urban environment, and guide the logistics company to wait for the vehicle to contact the receiving and delivering users in the shortest time. Different from the existing scheduling designs which are based on reinforcement learning, the high-dimensional logistics environment is considered in this paper during establishment of reinforcement learning scheduling model. These factors have a great impact on improving the efficiency of delivery and receiving as well as the benefit of couriers.

Keywords: Internet of things, optimization design, intelligent logistics, intelligent scheduling, deep learning, reinforcement learning

1. Introduction

With the continuous development of modern intelligent transportation, the transportation vehicles are becoming more networked and intelligent. Herein, it is based on the Internet of Things (IoT). Intelligent logistics is the most important industry in intelligent transportation, which is also the application and research focus of artificial intelligence. The intelligent logistics system obtains the information of transportation goods effectively through RFID, sensor network and other technologies of IoT. At the same time, it obtains the information of vehicles through the Internet of vehicles.it collects the location information of vehicles through GPS, and conducts intelligent scheduling with GIS system.

The proliferation of online e-commerce, has popularized IoT and GIS-based logistics deployment system country all over China which are practiced by the application of new logistics companies such as JD.com, SF Express and Alibaba. In 2018, 50.5 billion packages were delivered, and the whole industry's revenue was nearly 600 billion yuan. In the future, logistics should not only provide more efficient, accurate and agile services, but also realize the unmanned operation, intelligent operation and intelligent decision-making for the entire logistics system. Current logistics companies (such as jd.com) have built a full link intelligent logistics system for forecasting, inventory, warehousing, transportation and distribution. However, how to further improve the user experience of the service, improve the company's efficiency and save costs requires continuous optimization of the existing system.

To improve the user experience is the key of intelligent logistics, which needs the constant optimization based on the variety of users' demands. The design of intelligent scheduling is one of the key technologies to improve user experience. The system of logistics companies applies intelligent path planning by collecting valid data of delivery vehicles (or couriers) and the consignee/consignor scheduling. And the whole efficiency can be improved if we consider how couriers deliver the goods based on their behaviors and paths. Specifically, by applying the integration of couriers' experience and big data, we can clearly see the optimal path from the all options. Therefore, the service efficiency of delivery vehicles and consignee/consigner shall be optimized.

In order to provide better user experience, logistics companies need to quickly match the consignees and delivery vehicles based on the vehicles' routes and then carry out effective scheduling. However, delivery vehicles are not always directly matched with consignees and consigners. For example, when the consignee is unable to receive the goods in time for some reasons, the delivery vehicles need to wait for a long time, and we call the vehicles as waiting vehicles (couriers). In the practical urban scenario, the logistics companies will suffer some latency in collecting the consignee's data, since the valid data can only be obtained when the goods is delivered and signed by the consignee. Therefore, in the real world, the couriers must call for collection many times according to their own experience and wait for a long time to reach the users, especially when they are far away from the users. In addition to the response time of the consignee/consigner, the delivery demand and weather in different periods are also the factors leading the courier to be late for delivery, which is also the reason why the consignee/consigner needs to spend more time in waiting. Such a phenomenon directly reduces the user experience of the service and affects the revenue of enterprises and couriers. Therefore, designing effective vehicle scheduling strategy can significantly improve the user experience of receiving and mailing goods from logistics companies, and increase the efficiency and revenues.

Deep Q network (DQN) was proposed by Google Deepmind in 2013 [1]. This deep reinforcement learning (DRL) scheme combines reinforcement learning with deep neural network, and has made remarkable achievements in many applications due to its superior performance over reinforcement learning. In order to solve the above problems, this paper investigates the intelligent route scheduling method of delivery vehicles based on big data from logistics company scheduling service, and then develops an effective online dynamic framework based on deep reinforcement learning to improve service efficiency and user experience[2-4]. Our proposed scheme has potential to reduce the waiting time of consignee/consignor, improve the revenue of express delivery as well as the user experience. Our contributions are summarized as follows:

- First, we use deep-Q-network to learn the urban environment and guide waiting vehicles of logistics companies to reach the customers in the shortest time。
- Different from the existing scheduling schemes, which are based on reinforcement learning, we consider the high-dimensional logistics environment when building the scheduling model with reinforcement learning. These factors have a great impact on the improvement of delivery efficiency and revenues of couriers.
- Our proposed dynamic model is based on real-time big data and historical data, which can be used to schedule the paths of the delivery vehicles of logistics company during the model training period. At the same time, the model can be applied to real dynamic urban environment.

2. Vehicle Scheduling Model and Reinforcement Learning

In this section we first explain the relationship between the scheduling model of the vehicles of logistics companies and reinforcement learning, and then present the scheduling model based on reinforcement learning.

2.1. System Model

The goal of intelligent path scheduling is to minimize the waiting time and search time of users and couriers meanwhile maximizing the revenue of couriers. First, we divide the city into many small areas. Specifically, we assume that the shape of the city is a rectangular grid which can be evenly divided into many cells, where each cell represents an area in the city, as shown in Fig. 1.

The scheduling problem of a vehicle of a logistics company is described as follows. The vehicle searches for a consignee/consigner in a certain cell. At the beginning, there is no historical data regarding the customer, since such data is only available when the customer requests service. Therefore, the delivery vehicle communicates with the logistics company service center, informing its status. And service center directs the online guides the vehicle to an adjacent cell. Next, the vehicle searches consignee/consigner in the new cell continuously reports his status to the service center when it is empty and occupied. If the status is occupied, the guidance of the service center is finished, and the next request is to start after the vehicle has dropped off the packages for the customers and reported the revenue. If the status is empty, the request and response from the vehicle and service center will continue until the vehicle detects the consignee/consigner. It is worth mentioning that there is competitions among vehicles which are from different when processing these requests.

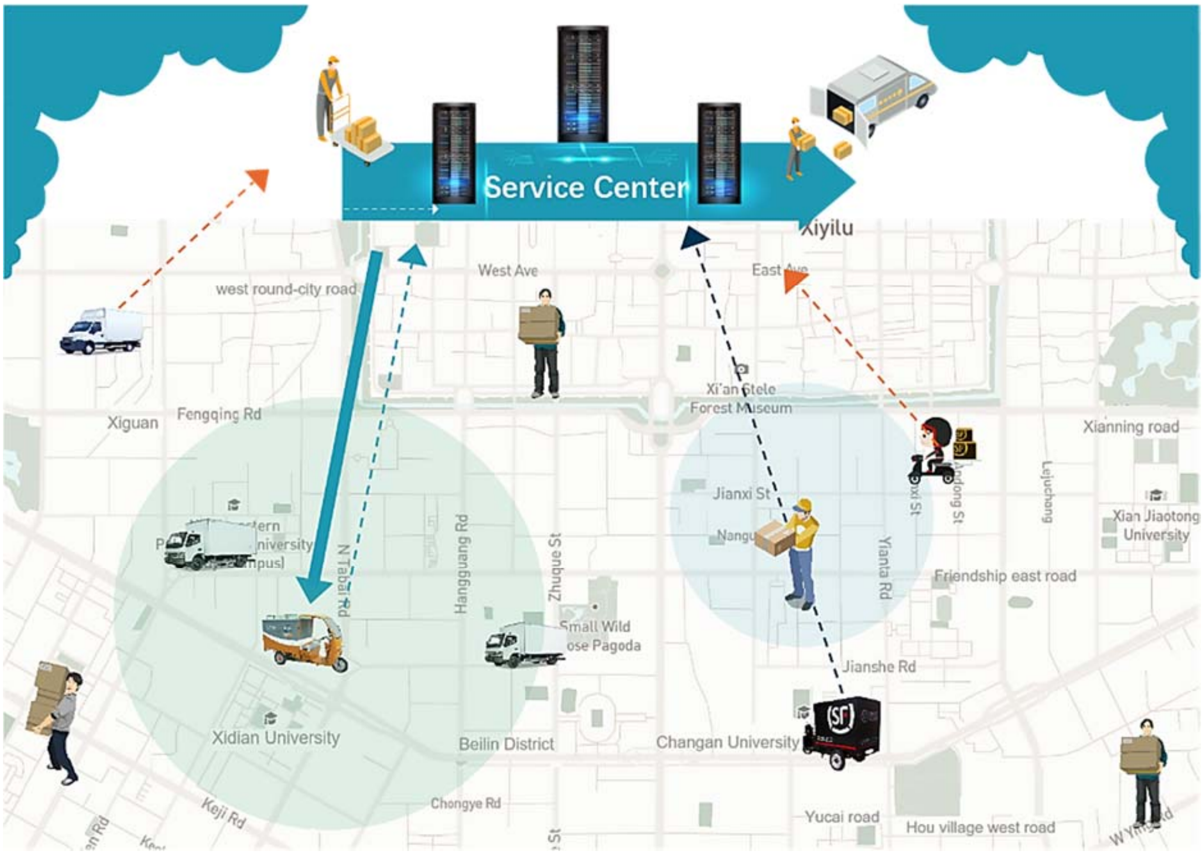

Fig. 1. Scheduling Model of Delivery Vehicles.

2.2. Basic Work

Reinforcement learning has been widely used to find optimal strategies and achieve optimal control [5-8]. In the reinforcement learning framework, the agent observes the environment, tries the action $a\in A$, and gets the reward $r\in R$, and then generates a strategy that maps the state $s\in S$ to the action a. According to the strategy, the environment will move to the next state with probability $p\in P$, and then repeat the process step by step until the optimal control is reached. Therefore, quaternion (S,A,R,P) can be used to represent reinforcement learning models. In the deterministic model which is similar to that of this paper, P is often omitted in system Settings.

2.3. Introduction of the Model

The vehicle scheduling model can be presented as a Markov decision process (MDP), but it is noted that in this paper, the transition probability is unknown. Therefore, in this case the idea of model-free reinforcement learning is more suitable. Typically, for model-free reinforcement learning, the goal of the agent (the service center) is to minimize the search time of the vehicle and maximize the revenue of the courier. In MDP, the components of model-free reinforcement learning method for the vehicle scheduling are described as follows:

1) State S: The information of the vehicle in a certain time slot.

Instead of relying on historical data, our dynamic model can be applied to real-time environments. It considers the vehicle scheduling problem in different time and spatial domain. On the one hand, the differences among the users' demands of mailing and delivering during the weekdays and weekends and at different times of the same day should be taken into consideration. Above is the application scenario of the urban environment that needs to be considered before scheduling. At the beginning of each step of the reinforcement learning, the agent observes the urban environment and collects some relevant parameters as the input state. On the other hand, each vehicle is not a separate entity, the distribution of other vehicles in the surrounding areas will also have a significant impact on the guiding policies, because there is competition among them. Based on the above factors, the parameters of the state of the reinforcement learning model are defined as follows:

• X_m: The horizontal coordinate of the vehicle, $m\in(1,2,\dots M)$.

• Y_n: The vertical coordinate of the vehicle, $n\in(1,2,\dots N)$.

• T_k: One day of the week, which is classified as weekday and weekend, where $k\in(1,2)$.

• P_l : Different time of a day, such as morning rush hour, evening rush hour, etc., $l\in(1,2,\ldots L)$.

• $C=\{C_s,C_n,C_e,C_w\}$: The number of vehicles from other logistics companies in the east, west, north and south, which indicates the competition among the directed vehicle and other vehicles.

Therefore, each input state at k-th step is $S(k)=\{X_m,Y_n,T_k,P_l,C\}$, which represents the set involving all parameters.

2) Action A: For each state, the agent will direct the vehicle to take one of the following actions to reach the next state.

The logistics company service center has four options $A=\{a_s,a_n,a_e,a_w\}$.In each state the service center will direct the logistics company to navigate the vehicles, which means the service center can direct the vehicles to one of the four adjacent cells which are in the south, north, east and west, respectively.

3) Reward R: Immediate reward after each action is taken by the vehicle

Only if the vehicle contact a consignee/consignor will it get reward, which is the profit of the order. Otherwise the reward is zero. Considering the competition of the vehicles from other logistics companies nearby, there is a probability that a vehicle will fail to receive an order under the guidance. When the vehicle successfully contacts a consignee/consignor with guidance, we define the reward r as the profit of the order r_p multiplied by the success rate of the order. Therefore, at the k-th step the immediate reward $r(k)$ can be expressed as:

$$r(k)=\begin{cases}\rho r_p, & \text{if the vehicle received the service request}\\ 0, & \text{otherwise}\end{cases}, \tag{1}$$

The couriers generated profits after contacting with consignee/consignor i. The order profit obtained by a courier can be expressed as:

$$r_p(i)=e(i)-c(i), \tag{2}$$

where $e(i)$ and $c(i)$ are the revenue and cost from contacting with consignee/consignor i, respectively. In order to achieve an effective scheduling, we define revenue, success rate, and vehicle consumption based on the rules considered by real-world logistics companies. According to the weight of the goods and the delivery distance, each order has different revenue.

Besides, the defined cost includes the transportation cost of the goods themselves as well as the fuel consumption and labor cost of the vehicle in the initial pickup phase of logistics.

Another factor needed to be considered during contacting with a consignee/consignor is that the competition among the vehicles from different logistics companies. The vehicles of other logistics companies may contact with the consignee/consignor earlier, resulting in a failure of delivery for the directed vehicle. Specifically, to quantify this situation, we need to consider the number of vehicles of other logistics companies competing with the directed vehicle. The success rate of contacting with a consignee/consignor ρ is defined as follows:

$$\rho=\frac{P_s}{C_0\cdot p_k+\sum_{n=1}^{N}C_n}, \tag{3}$$

where P_s is the number of consignees/consignors at the location of the guided vehicle, P_k is the probability that the vehicle searches for the consignee/consignor in the current cell, C_0 is the number of the vehicles at the same location, and C_n is the number of the equivalent vehicles of other logistics companies involved in the cells of which the distance to the consignees/consignors is n.

3. Demand and Vehicle Scheduling

We introduce the deep reinforcement learning algorithm to realize the best vehicle scheduling of logistics companies.

3.1. Reinforcement Learning

According to [9], the policy in reinforcement learning is $\pi=\{a_1, a_2, \ldots, a_K\}$, where $a_{n\in[1,K]}\in A$, and K is the total step for contacting with a consignee/consignor. The agent execute $a\in A$ in the k-th step, and then the agent jumps to another state S_{k+1}, obtaining the reward r_k . The goal of the agent is to maximize the expected cumulative reward R_k, which is defined as

$$R_k=\sum_{t=k+1}r_t \tag{4}$$

The learning policy is to maximize the expected cumulative reward, so the strategy can be expressed as

$$\pi^*=\arg\max_{\pi\in A}R_k^\pi, \tag{5}$$

where $R_k^\pi=\sum_{t=k}^{K}\gamma^{t-k}r_k$, and γ is the discount factor which is used to balance the immediate reward and future reward. In Q-learning algorithm, the expected cumulative reward starts with state S_k and follows the action a_k, which is called state-action value $Q(s_k,a_k)$ or Q value, which is given by

$$\begin{aligned}Q(s_k,a_k)&=\mathbb{E}\left[r_k+\gamma r_{k+1}+\gamma^2 r_{k+2}+\ldots\mid s_k,a_k\right]\\&=\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r_{k+t}\mid s_k,a_k\right]\end{aligned} \tag{6}$$

Let $Q^*(s_k,a_k)$ denote the optimal Q value, which is presented by optimal Bellman formula

$$Q^*\left(s_k, a_k\right) = \mathbb{E}\left[r_k + \gamma \max_{a_k'} Q^*\left(s_k', a_k'\right) \mid s_k, a_k\right], \tag{7}$$

where s'_k and a'_k represent the state and action in the next step, respectively. Thus, the correlation of the current state and next state has been established, and the optimal Q value can be calculated iteratively. In Q-leaning, the iterative function is defined as

$$\begin{aligned} Q\left(s_k, a_k\right) &= Q\left(s_k, a_k\right) + , \\ \alpha\Big[r_k &+ \gamma \max_{a_k'} Q\left(s_k', a_k'\right) - Q\left(s_k, a_k\right)\Big] \end{aligned} \tag{8}$$

where α is the learning rate, γ is the discount rate. Finally, the Q value will be converged to the optimal value, i.e., $Q_t \to Q^*$ when $k \to \infty$, and the optimal policy π^* is obtained through Q table, i.e.,

$$\pi^* = \arg\max_{a_k} Q^*(s_k, a_k).$$

However, there are two challenges for searching the Q table for the optimal strategy: one is that it is difficult to store the Q table in memory when the state is high dimensional. Another reason is that the learning process is very slow, because only one state value can be updated in one step. Therefore, in order to achieve our needed dynamic vehicle scheduling model with strong immediate and high dimensional state, conventional Q-learning is not feasible for this problem. In order to solve the above two problems and improve resource utilization, we further introduce deep Q-network.

3.2. Deep Q-Network

The main idea of deep Q-network (DQN) is to approximate the function of state-action value by using function approximation technique. The neural network is a good approximator for this nonlinear function. Different from general Q-learning, DQN uses a deep neural network to replace the traditional Q table, so that the state-action value can be expressed as a function by the neural network, which means that the process of updating the Q table is the process of training the deep neural network. In the process of updating the neural network, we need to consider the Q value at current step and the next step. Therefore, we use two independent networks to represent two Q values, respectively. Thus, we define two types of Q-value neural networks as [6]

$$\begin{aligned} Q\left(s_k, a_k\right) &= Q\left(s_k, a_k; w\right) \\ Q\left(s_k', a_k'\right) &= \hat{Q}\left(s_k', a_k'; w^-\right), \end{aligned} \tag{9}$$

where w is the parameter evaluating the Q-network, w^{-} is the parameter of the targeted Q-network. Due to the in-homogeneity between the high-dimensional state space and the low-dimensional action space, these two neural networks only use the state as the input, and the output is the Q value of each possible action.

In order to obtain the optimal policy based on Q-Q*network, we use supervised learning to training the Q-network. We consider the targeted Q value $r_j + \gamma \max_{a'} \hat{Q}(s_{j+1}, a'; w^-)$ as the label. The loss function of the neural network is expressed as

$$L(w) = \mathbb{E}\left[r_k + \gamma \max_{x'} \hat{Q}\left(s_{k+1}, a'; w^-\right) - Q\left(s_k, a_k; w\right)\right]^2 \tag{10}$$

3.3. Overall Architecture

During the training, we use gradient descent regarding w to minimize the loss function in i-th iteration. Considering the neural network as an approximation for replacing the Q table in reinforcement learning may lead to non-convergence of the network. To avoid this problem, Deep-Mind's paper [6] introduces two methods: 1) experience replay and 2) fixed target networks. Therefore, the derivative of the loss function can be expressed as

$$\begin{aligned} &\nabla_w L(w) = \mathbb{E}_{s_j, a_j, r_j, s_{j+1} \in \mathbb{M}} \\ &\left[\left(r_j + \gamma \max_{a'} \hat{Q}\left(s_{j+1}, a'; w^-\right) - Q\left(s_j, a_j; w\right)\right) \nabla_w Q\left(s_j, a_j; w\right)\right], \end{aligned} \tag{11}$$

where M is the empirical reset memory, which can store the passing sequence, and then let the agent randomly select any passing sequence for the subsequent training process. Experience reset memory can break the relationship between training data. At each fixed step, the parameter of the target network ω^- is replaced by the parameter of the evaluation network ω, which is also called fixed target network.

The entire process of scheduling an logistics company vehicle to a consignee/consignor can be represented as follows. After collecting real-time data, we use deep Q-network, experience reset, and fixed Q target to learn and interact with the urban environment at a specific time. Then an individual vehicles can be scheduled to reduce the search time. In the state S_k, according to the guidance from service center, the vehicle can randomly take an action a_k to get into a new cell with Ò-greedy policy. Meanwhile, the vehicle gets a reward r_k after executing a_k and jumps to the next state S_{k+1}. After this, the vehicle will report passing sequence (S_k, a_k, r_k, S_{k+1}), which will be stored in memory by the service center. If the consignee/consignor is contacted with the directed vehicle which is in the same cell, the phase is finished after reporting the immediate revenue, and a new guidance process starts. Otherwise the service center will continue to direct the vehicle to receive the service request. Besides, when enough transition samples are collected, the service center will randomly select the batched passing sequence (s_i, a_i, r_i, s_{i+1}) to train the neural networks. For the training process, the service center utilizes the gradient descent method to minimize the difference between the target Q value and the Q assessment (MSE), and let $\hat{Q} = Q$ for every L steps. Afterwards, when the vehicle sends its current location information to the service center and asks for guidance, the service center will provide guidance based on the learning results of DQN.

3. Conclusion

This paper have studied the optimal route scheduling model for the vehicles of logistics companies based on deep reinforcement learning, This model could guide the delivery vehicles through the interaction between the service center of the logistics company and the urban environment. We took time, space, and competition between vehicles from different logistics companies into account. These models and methods can improve the user experience of vehicle allocation service. The next step of our work is to build the research foundation for unmanned distribution.

References

[1]. D. Zhang, X. Han and C. Deng, Review on the research and practice of deep learning and reinforcement learning in smart grids, *CSEE Journal of Power and Energy Systems*, Vol. 4, No. 3, September 2018, pp. 362-370.

[2]. A. Zappone, M. Di Renzo and M. Debbah, Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both ?, *IEEE Transactions on Communications*, Vol. 67, No. 10, October 2019, pp. 7331-7376.

[3]. R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, *Machine Learnin*g, 8, 1992, pp. 229-256.

[4]. Z. M. Fadlullah et al., State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems, *IEEE Communications Surveys & Tutorials*, Vol. 19, No. 4, 2017, pp. 2432-2455.

[5]. O. L. Georgeon, J. B. Marshall and S. Gay, Interactional Motivation in artificial systems: Between extrinsic and intrinsic motivation, *in Proceedings of IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL'2012)*, San Diego, CA, 2012, pp. 1-2.

[6]. I. Goodfellow, Y. Bengio, A. Courville, F. Bach, Deep Learning, *MIT Press*, Cambridge, USA, 2016.

[7]. Y. Yu, J. Chen, S. Lin and Y. Wang, A Dynamic QoS-Aware Logistics Service Composition Algorithm Based on Social Network, *IEEE Transactions on Emerging Topics in Computing*, Vol. 2, No. 4, December 2014, pp. 399-410.

[8]. P. Jaillet, J. Qi, M. Sim, Routing optimization under uncertainty, *Oper. Res.*, Vol. 64, No. 1, 2016, pp. 186-200.

[9]. J. J. Q. Yu, W. Yu and J. Gu, Online Vehicle Routing With Neural Combinatorial Optimization and Deep Reinforcement Learning, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 10, October 2019, pp. 3806-3817.