Multi-agent reinforcement learning-based dynamic task assignment for vehicles in urban transportation physical internet

Wei Qin, Yanning Sun, Zilong Zhuang, Zhiyao Lu and Yaoming Zhou
School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
Corresponding author: wqin@sjtu.edu.cn

Abstract: *The transportation task assignment for vehicles plays an important role in city logistics of the physical internet, which is the key to cost reduction and efficiency improvement. The development of information technology and the emergence of "sharing economy" create a more convenient logistics mode, but also bring a greater challenge to efficient operation of urban transportation physical internet. On the one hand, considering the complex and dynamic environment of urban transportation, an efficient method for assigning transportation tasks to idle vehicles is desired. On the other hand, to meet the users' expectations on immediate response of vehicle, the task assignment problem with dynamic arrival remains to be resolved. In this paper, we proposed a dynamic task assignment method for vehicles in urban transportation physical internet based on the multi-agent reinforcement learning. The transportation task assignment problem is transformed into a stochastic game process from vehicles' perspective, and then an extended actor-critic algorithm is proposed to obtain the optimal strategy. Based on the proposed method, vehicles can independently make decisions in real time, thus eliminating a lot of communication cost. Compared with the methods based on FCFS (first come first service) rule and classic contract net, the results show that the proposed method can obtain higher acceptance rate and average return in the service cycle.*

Keywords: *urban transportation physical internet, transportation task assignment, multi-agent reinforcement learning, actor-critic algorithm.*

1 Introduction

With the increasingly fierce market competition and the advancement of information technology, the existing city logistics modes are developing towards an energy-saving, efficient and shareable manner. In particular, the novel mode combining city logistics with physical internet, so-called hyperconnected city logistics (Ballot et al., 2014; Kubek and Więcek, 2019;), makes traffic management system to operate more effectively by big data analysis and machine intelligence algorithms (Zhong et al., 2017; Kaffash et al., 2020). In this kind of traffic management system, assigning transportation tasks to vehicles is one of the most important services. Rapidity and rationality are the guarantee for the satisfaction of both users and drivers. However, the sharp increase in transportation demands and vehicle quantities have brought great challenges to existing task assignment methods.

Traditional modeling methods are usually based on simplified constraints and steady-state assumptions, such as mathematical programming (Russell, 2017), graph theory (Xia et al., 2019) and Markov model (Hasan and Ukkusuri, 2017), which are difficult to handle complex and dynamic task assignment problem for vehicles. The rule-based task assignment method can better ensure the real-time decision-making, but the acceptance rate of task assignment and the average return of the system should be further improved. With the storage of vehicles operation data, it is theoretically possible to obtain a decision scheme through existing data learning (Morin et al. 2020). Multi agent can use distributed structure to describe complex and dynamic urban transportation system, so as to reduce the complexity of the system. Reinforcement learning interacts with environment through trial and error, which is suitable for decision-making problems of the complex dynamic system with large uncertainty and difficult to be solved by traditional methods (Haydari and Yilmaz, 2020). Therefore, the task assignment problem is described as a multi-person multi-stage stochastic game process under cooperative conditions in this paper. A reward-driven decision evaluation method is adopted and the multi-agent reinforcement learning algorithm serves as a solution framework for the problem.

The main works and contributions of this paper include: 1) For task assignment problem, to meet the requirement of immediate response to transportation tasks of users, a stochastic-game-based event-driven task assignment model is developed. It models nodes in transportation network as agents, vehicles at node as agents' resources. Dynamic transportation tasks will trigger the corresponding nodes to make decisions. 2) An extended actor-critic (AC) algorithm is proposed to solve the developed task assignment model and obtain the optimal strategy. This algorithm consists of several actor networks and one centralized critic network. In training process, agents update parameters of actor and critic networks based on experiences of interacting with environment and state value generated by critic network, and achieve ideal synergy. In testing process, agents are able to provide online decision only based on their state. 3) Simulation and comparison experiments was carried out in Didichuxing's open source data (DiDi, 2020), which shows that the proposed model and algorithm for dynamic task assignment of vehicles can significantly improve the acceptance rate of task assignment and the average return of the system. This study can also provide a reference for practical applications.

The rest of paper is organized as follows. Section 2 gives the literature review on the related works. Then in Section 3, we proposed the networked description of urban transportation and developed an event driven task assignment model based on stochastic game. The extended actor-critic algorithm was put forward for model solution in Section 4. Simulation experiments and results analysis are given in Section 5. Finally, the conclusions are summarized in Section 6.

2 Related works

Task assignment problem has always been a hot topic in the fields of enterprise staffing (Bouajaja and Dridi, 2015), factory machine scheduling (Liu et al., 2019), satellite resource scheduling (Gabrel and Vanderpooten, 2002) and transportation (Lin et al., 2001; Srivastava et al., 2008; Glaschenko et al. 2009; Seow et al., 2009; Zhen et al., 2019; Zhang et al., 2018). Transportation task assignment is to reasonably arrange the correspondence between vehicles and tasks, and to propose an immediate task assignment scheme. This problem involves multiple dynamic tasks and limited resources, which is a typical combinatorial optimization problem and also an NP-hard problem. It requires online response to randomly arrived demand, and the information at the time of decision-making is incomplete, including only the current and historical resources and demand information. These features make it difficult to be effectively solved as the general assignment (Chekuri and Khanna, 2005) or knapsack problems (Kleywegt and Papastavrou, 1998). The current literature mainly employs mathematical programming, graph theory, simulation or multi-agent models to solve it.

When the target problem only contains a small-scale task or a single type of resource, the mathematical programming model can be established to obtain the exact solution. Considering the individual and collaborative factors involved, Chen et al. (2009) established a multi-objective optimization model to solve the matching problem between employees and tasks. Some researchers also employed heuristic algorithms to solve complex problems with many constraints, which greatly reduce the computation time and memory consumption. Deng et al. (2016) proposed an accurate algorithm and an approximate algorithm for the matching of staffs and tasks in the crowdsourcing platform, in which the accurate algorithm is difficult to run because of excessive memory consumption, but the response time of the approximate algorithm is less than millisecond. Abstracted the task allocation problem of unmanned aerial vehicles (UAV) as a collaborative multi-task allocation problem, Jia et al. (2018) developed the mathematical model with kinematic, resource and time constraints, and used the improved genetic algorithm to get the solution of the problem.

The structure of the system can be described intuitively by the node, link and weight in the graph theory model. Gabrel and Vanderpooten (2002) established a graph theory model for the problem of satellite and observation task matching. Further, the shortest path algorithm is used to obtain the task planning scheme to achieve the maximum benefit. Kachroo and Sastry (2016) proposed a travel time function based on traffic density, and established a mathematical programming model to solve the user balance and route allocation schemes by using the node traffic balance in the directed graph with consideration of the intersection time delay.

When the dynamic characteristics cannot be fully expressed by mathematical equations, simulation models can be employed to model the problem. Lin et al. (2001) simulated the freight transportation system in the production logistics by using the combination of the first come first serve rule and the nearest vehicle first rule. Theoretically, the more perfect the actual situation is, the more detailed and accurate the simulation model is, and the more credible the simulation results are. Jorge et al. (2014) confirmed that the mathematical model can get the optimal results, but it needs longer computation time than the simulation model. As for some problems with random and uncertain events, the simulation models can better reflect the effectiveness of the algorithm. However, the modeling and maintenance costs of the simulation models are higher, so it is not suitable for complex systems.

With the advantages of solving large-scale problems, multi-agent systems for task assignment problem have been widely concerned (Srivastava et al., 2008; Seow et al., 2009; Glaschenko et al., 2009; Hao et al., 2013; Lan, 2018). This method essentially enables information sharing between agents through direct or indirect communication to achieve decision sharing. Moreover, some studies have applied multi-agent-based reinforcement learning methods to transportation industry and have achieved good results. A distributed multi-agent deep reinforcement learning method was adopted to solve the problem of controlling traffic signals in a complex urban transportation network, and good results were achieved in terms of optimality and robustness (Chu et al., 2019). Lin et al. (2018) proposed two algorithms based on multi-agent reinforcement learning framework to generate a decision-making scheme for large-scale fleet management of a travel platform. The algorithms can capture supply and demand changes in high-dimensional spaces and formulate corresponding balancing strategies. It is verified in practice that multi-agent systems can significantly improve the utilization of transportation resources. These studies in the context of transportation show that the idea of employing multi-agent reinforcement learning to solve transportation task assignment is feasible.

In short, researches of task assignment problem in many fields are gradually increasing and deepening, and have achieved good results in practical applications. However, there are still some problems such as lack of consideration of random and uncertain factors in practice, and the resulting decision scheme has low flexibility and lag. Especially for the complex and dynamic environment of urban transportation, many algorithms cannot be directly applied. Therefore, in order to improve and solve the above problems, this study proposed a multi-agent reinforcement learning algorithm to solve the problem of transportation task assignment.

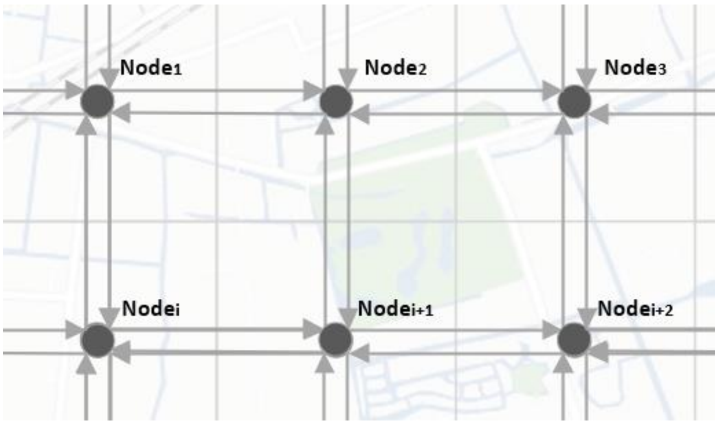

Figure 1: Networked description for urban transportation

3 Event-driven task assignment model based on stochastic game theory

In this section, the networked description of urban transportation is proposed, and an event driven task assignment model based on stochastic game is developed.

3.1 Networked Description of Urban Transportation

Based on the idea of graph theory, the complex urban transportation system is abstracted as a complex network $G = (N, E)$ composed of nodes and edges (see Figure 1). $N = \{Node_1, Node_2, \ldots, Node_n\}$ is the set of nodes in the complex network, which represent various areas of urban roads. $E = \{Edge_{12}, Edge_{21}, \ldots, Edge_{ij}\}$ is the set of edges in the complex network. There are two edges $Edge_{ij}$ and $Edge_{ji}$ connected between any two adjacent nodes $Node_i$ and $Node_j$. In our opinion, any known urban transportation system can be described by G.

Vehicles and tasks in the transportation network are denoted by V and T, where $V = \{Vehicle_1, Vehicle_2, \ldots\}$ is the set of vehicles, and $T = \{Task_1, Task_2, \ldots\}$ is the set of tasks. We defined c_{it} as the total vehicle resource at $Node_i$ at time t, $l_{i,t}$ as the total transport task for $Node_i$ at time t. The service period is usually divided into days or months, which is expressed as P. In order to describe the dynamic changes in the environment and resources, time is discretized, and the service period of the vehicle between any two adjacent nodes is taken as the time interval Δt.

Before developing task assignment model, we made the following assumptions based on the networked description for urban transportation:

1) Modeling objects are moments and places where demand is greater than supply. Based on analysis of real scenarios, when demand is less than supply or supply and demand are balanced, as long as any demand arrives, timely response can ensure that the global benefit is maximized. In that case, no task assignment and evaluation are required.

2) Each period in service cycle is the assignment period of the transportation task, $P = [P^{start}, P^{end}]$ where P^{start} is the start time of the round of assignment, P^{end} is the end time of the round of assignment, $\Delta t = P^{end} - P^{start}$ is the time interval.

3) Vehicle resources of node are updated before start time P^{start}, which includes: the remaining vehicles of the node in the previous period, the vehicles that arrived from other nodes in the previous period, and the vehicles that completed the transportation task to reach the destination node.

4) In the same assignment period, except for assignment decisions, the number of vehicles at a node will not increase or decrease due to external factors. The number of vehicles at a node is the maximum number of tasks that the node can accept during this period.

5) Transport tasks are represented as $task = \{No^{task}, t^{e}, t^{w}, t^{d}, node^{dep}, node^{dest}, v, m\}$, where No^{task} is the number of tasks; t^{a}, $t^{w}$and t^{d} denote the task assignment, waiting and delivery time, respectively; $node^{dep}$ and $node^{dep}$ are places of departure and destination, respectively; v is task value and m is the order in which tasks arrive.

6) For tasks that are not accepted during the task assignment period, if there is a waiting time $t^{w} \neq 0$, the assignment request can be re-initiated in multiple assignment periods of $t^{a} + t^{d}$, and it has higher priority in new assignment period, which means $m^{old} < m^{new}$.

3.2 Stochastic game and model development

Stochastic game. Multi-agent reinforcement learning has the characteristics of multi-stage of the Markov decision process, and also has the characteristics of multi-participant of matrix games, so it is usually expressed by a stochastic game that combines the two. Stochastic game is a type of dynamic game with state probability transition, which is performed by one or more participants. It can be defined as:

$$SG = (n, S, A_i, P, R_i) \tag{1}$$

where n is the number of agents; S is the state set of the environment; A_i refers to action set that agent i can choose; P represents the state transition probability; R_i is the agent's return function. In this process, multiple agents make a choice of actions, and the next state and reward of the environment is determined by the joint actions of multiple agents (see Figure 2).

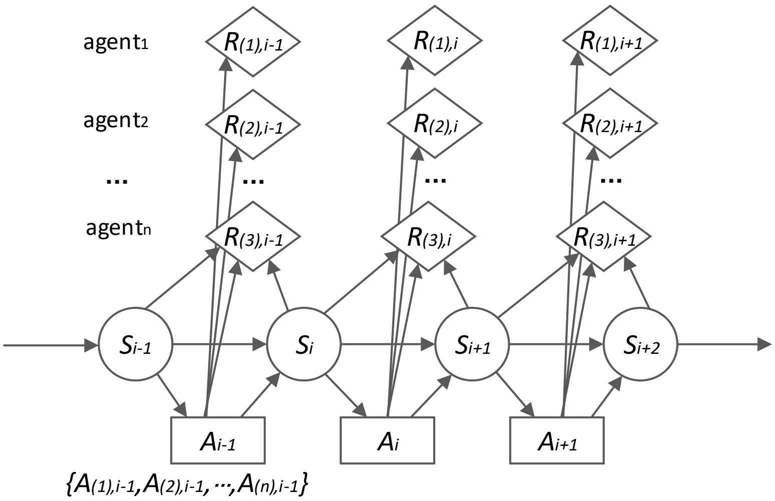

Figure 2: Stochastic game

Stochastic games are aimed at solving the Nash equilibrium, but under normal conditions, the transfer function and return function are unknown. In reinforcement learning, the agent learns the equilibrium strategy through interaction with the environment, and uses the rationality and convergence to evaluate algorithm performance (Bowling and Veloso, 2002).

Agent. Each node in the transportation network is considered as an agent. Without considering factors such as driver's historical order acceptance rate and preferences, the vehicles are no difference in the same or similar locations. Therefore, each node has two states: demand vehicles or supply vehicles, which also denotes agent states. Vehicles are the resource owned by nodes, that is, attributes of agents. In practice, there is a one-to-one assignment relationship between the transportation task and the vehicle. If each vehicle is assigned to a transportation task, most of the other joint actions are invalid. Compared with considering vehicles as agents in the literature (Gupta et al. 2017), our agent setting method can greatly reduce the number of agents, and further reduce the environment's joint action space and calculations.

State. When task arrives, the task's destination and the estimated value can be observed by the node. The resource of other nodes has little influence on the decision of the vehicles in this node, thus only the remaining vehicle resources of this node are considered. The environmental information observed by each agent can be defined as the resource remaining, task information and time information of the node where the vehicle is located:

$$s_i = \{c_i^{remain}, s_i^{task}, s_t^{time}\} \tag{2}$$

where c_i^{remain} is the remaining resources of current node and s_t^{time} is the current assignment time. The task that arrives can be expressed as,

$$task_i = \{i, t^e, t^w, t^d, node^{dep}, node^{dest}, v_i, m\} \tag{3}$$

where the state of the task can be represented as s_t^{task} = $\{node^{dest}, v_i\}$.

Action. For any task k that arrives at node i, its departure node and its neighboring nodes can choose whether to accept the task,

$$a_{i,k} = \{0,1\} \tag{4}$$

where $a_{i,k} = 0$ denotes that the task is rejected and $a_{i,k} = 1$ denotes that the task is accepted.

Reward. The rewards obtained from the interactive feedback between nodes and the environment are determined by the node state and actions simultaneously. When the $task_k = \{k, t^e, t^w, t^d, node^{dep}, node^{dest}, v_i, m\}$ arrives at t, the reward received by the node is defined as,

$$r_{i,k} = \begin{cases} 0, \text{ when node } i \text{ rejects task } k \\ \alpha \dfrac{v_i}{\sum a_{i,k}} + \beta c_j^{remain}, \text{ when node } i \text{ accepts task } k \end{cases} \tag{5}$$

where v_i is task value and $\Sigma a_{i,k}$ is the number of nodes that choose to accept the task. When more nodes choose to accept the task, the nodes can get less rewards. c_j^{remain} is the remaining resources of the current node. When there are more remaining resources j, the greater the reward that the node can get, the more inclined it is to accept the task. α and β are normalized coefficients for task value and resource consumption, which is to eliminate the difference in feature vector values of different dimensions.

State probability transition. The vehicle resource distribution and node location information during the service period are known, but the specific information of the next arrival task is unknown. And the environment condition will refresh between periods, so that the vehicle resource distribution changes on each node and the state transition probability function is unknown.

End time. For the entire assignment process, task assignment is terminated when the service cycle ends. In a certain assignment period, when the vehicle resources of each node in the transportation network run out, the next assignment period is started.

$$\begin{cases} t^{end} = T \\ t^{curr} = t^{curr} + \Delta t, if \sum_{node_i \in Node} c_{i,t^{curr}} = 0 \end{cases} \tag{6}$$

where t^{curr} refers to the current time of the environment and $\Sigma_{node_i \in Node} c_{i,t^{curr}}$ is the total number of vehicle resources in the transportation network during the t^{curr} period.

4 Extended actor-critic algorithm for model solution

The AC algorithm (Konda and Tsitsiklis, 2000; Bhatnagar et al., 2008; Babaeizadeh et al., 2016) is the basic framework we adopt, which combines value function-based and policy gradient-based methods, improves the limit of the state space dimension in the value function-based method, and solves the randomness of the environment that causes the estimated policy gradient to have a large variance in multiple samplings. The framework consists of two networks, one is the actor network $\pi(s, a, \theta)$, which is used to optimize agent strategies; the other is the critic network $\hat{q}(s, a, \omega)$, which is used to estimate the value function. Parameters of the neural network are θ and ω, respectively. Based on critic's evaluation for the action taken, actor will adjust its strategy, and critic will update the value function based on experience and rewards.

4.1 Network Structure

In the extended AC framework, we establish different actor networks for different agents, which can maintain its own network parameters. In actual situations, there is a difference in the probability distribution of tasks arriving at different locations in the city. For example, the tasks at the center of the city have a short distance and a short time, and tasks at the edge of the city may take longer and be more valuable higher. If a network is simply described by shared parameters, the differences between nodes cannot be reflected, which may cause problems such as the difficulty in convergence of results. Therefore, we proposed a centralized training and distributed execution structure. During the training process, each agent learns strategies from observations and actions of its own environment. A centralized critic network uses the observation status of each node as input, and updates the rewards obtained by the actor's action feedback based on the environment. In this process, centralized training can make the strategies of each agent achieve tacit coordination, while decentralized execution can extract the local strategies of each agent from the global strategy, thereby achieving the purpose of task assignment.

Figure 3 shows the distributed network structure used in this study. There are two parts, multiple networks for executing strategies and a centralized value function network. Strategy network and value function network in the figure are both multi-layer feedforward neural networks with three layers.

4.2 Network Training

Actor's policy gradient is calculated by,

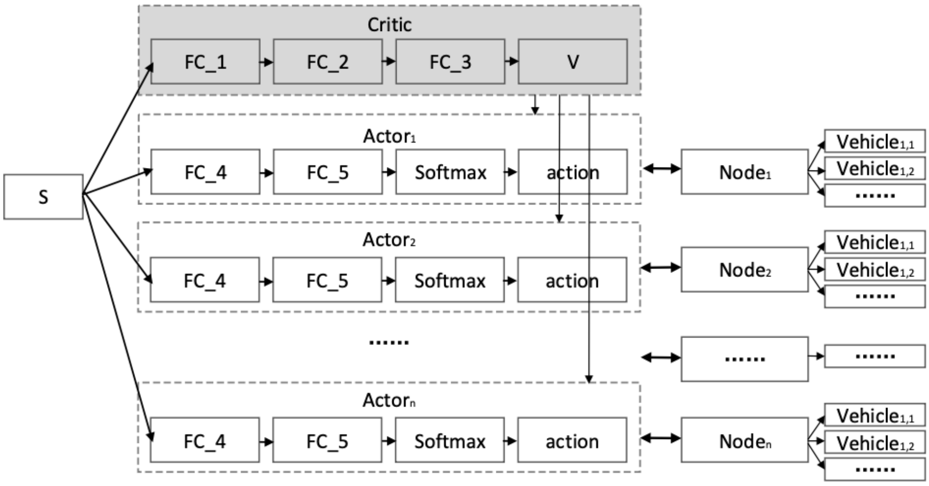

Figure 3: Extended AC decision model framework

Table 1: Extended AC algorithm

```
input: environment, number of iterations $N$, period $T$, number of nodes $n$, state
space dimension, action space dimension, step size $\alpha$, $\beta$, attenuation factor $\gamma$,
exploration rate $\varepsilon$, critic network structure and actor network structure
output: actor network parameters $\theta_1$, $\theta_2$, ..., $\theta_n$, critic network parameters $\omega$
-----------------------------------------------------------------------------------
Initialize network parameters
for $i$ from 1 to $N$ do
    Initialize the environment and get the initial state $s_0$
    for $t$ from 1 to $T$ do
        j = 0
        while there are tasks and vehicle resources left at target node
            for $k$ from 1 to $n$ do
                use $s_t$ in the network as input, output action $a_{t,k}$
                perform actions to get feedback $r_t$ and next state $s_{t+1}$
            end for
            calculate dominance function and target critic network value function
            $j = j + 1$
        for $m$ from $j$ to 1 do
            Actor network parameter update
            Critic network parameter update
        end for
    end for
end for
```

$$\nabla J(\theta) = E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s_t,a_t)V(s_t,\omega)] = E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s_t,a_t)A(s_t,t,\omega)] \tag{7}$$

Advantage function *A* is used as the evaluation point of critic network, which can be defined as the difference between the action value function and the state value function, and replaced by its unbiased estimate.

$$A(s,t) = r + \gamma V(s_{t+1}) - V(s) \tag{8}$$

Critic network loss is the squared loss of actual state value and estimated state value, and its parameters are updated using time difference (TD).

$$\begin{aligned} &\min \sum (V_\pi (s_{t+1}, \omega') - V(s_t, \omega))^2 \\ &V_\pi (s_{t+1}, \omega') = \sum \pi(s_t, a_t)(r_{t+1} + \gamma V(s_{t+1}, \omega')) \end{aligned} \tag{9}$$

Whenever a new task arrives, vehicles at the same node will accept the same decision, that is, intelligent node will give a unified decision of vehicles at that node, and select one of the vehicles to complete the real matching action. This method can reduce the task contradictions between matching decisions. In addition, when multiple nodes are involved in task matching, there may still be conflicts between the actions given by the nodes. In order to meet the constraint of the task's uniqueness, the state value generated by the centralized evaluation network is used as the basis for the final action selection for task coordination between nodes. The pseudocode is shown in Table 1.

5 Experiment

5.1 Data and Simulation Environment

Didichuxing's open source data (DiDi, 2020) is used to verify the effectiveness of the proposed method. Some data samples are shown in Table 2. By analyzing and visualizing the data, it can be seen that the attributes of the task have different characteristics in different periods, such as 7: 30-7: 40 and 19: 50-20: 00, as shown in Figures 4, 5, respectively. The tasks submitted in the two periods are divided according to the places of departure and destination. The number of tasks contained in each place can be seen from the figure. The place of departure is more scattered, and the place of destination is relatively concentrated for the period 7: 30-7: 40, while the period 19: 50-20: 00 is the opposite. The results of this analysis are also consistent with actual life experiences.

Table 2: Data samples

Orders number	mjiwdgk	f78cfb7e	5c33acbf	…
Start billing time	1501581031	1477963587	1477965143	…
End billing time	1501582195	1477965143	1477959461	…
Longitude of departure position	104.11225	104.05412	104.07139	…
Latitude of departure position	30.66703	30.67206	30.71631	…
Longitude of destination position	104.07403	104.06614	104.05733	…
Latitude of destination position	30.686300	30.709336	30.699250	…

The proposed method uses a distributed network structure with high complexity. In order to reduce training costs and time, we only considered a part of nodes in the urban transportation network. In this experiment, five nodes were selected as the modeling objects. The total number of tasks and the total number of vehicle resources for these selected regional nodes are shown in Figure 6. The average order acceptance rate of the nodes is about 82.866%, which means the demand for vehicle resources exceeds the supply for a long time.

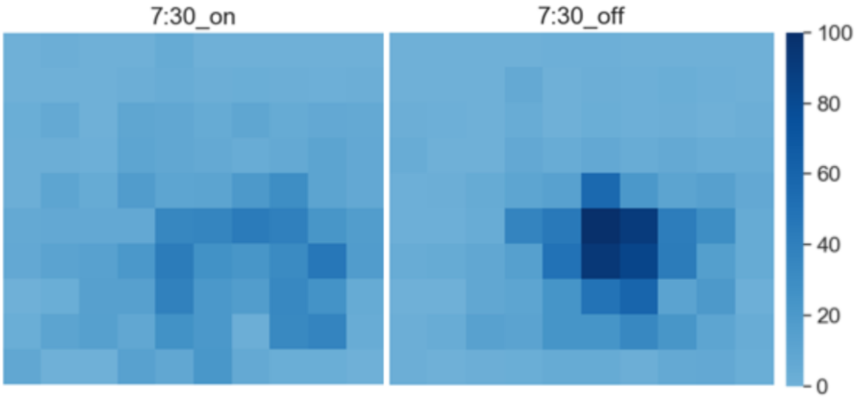

Figure 4: Distribution of orders' departure and arrival in 7:30-7:40

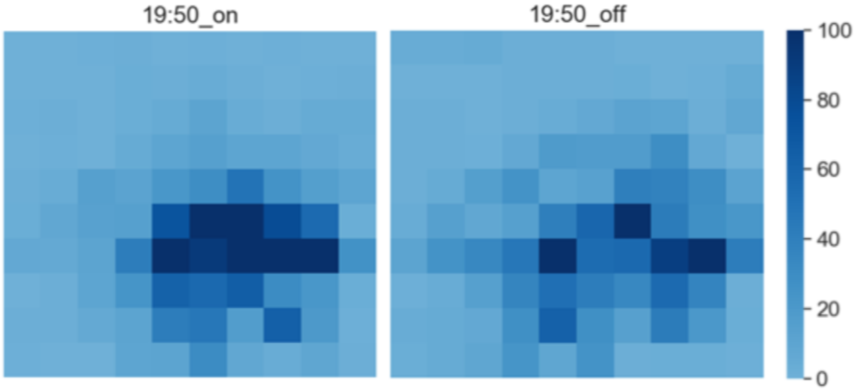

Figure 5: Distribution of orders' departure and arrival in 19:50-20:00

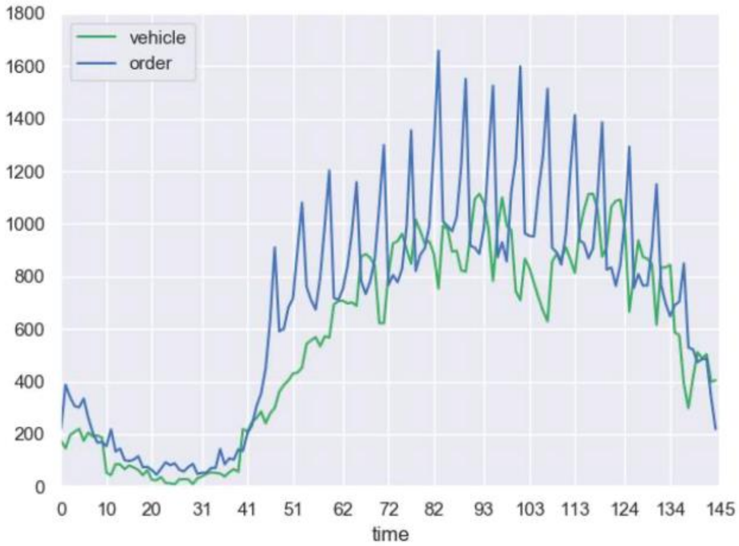

Figure 6: Number of orders and vehicles in local areas

5.2 Result Analysis

Task acceptance rate and profit rate are used to evaluate the performance of the method. Task acceptance rate is the ratio of the number of tasks accepted to the total number of tasks, and profit rate is the ratio of the total value of accepted tasks to the total task value.

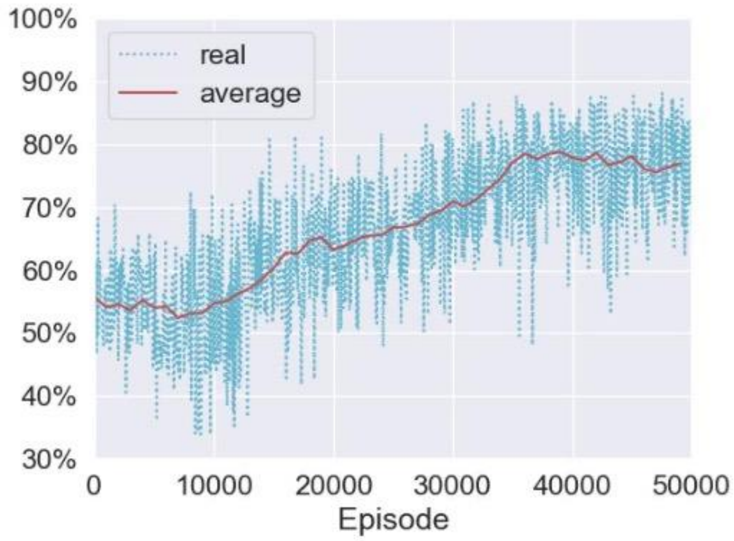

Figure 7: Task acceptance rate change with training episodes

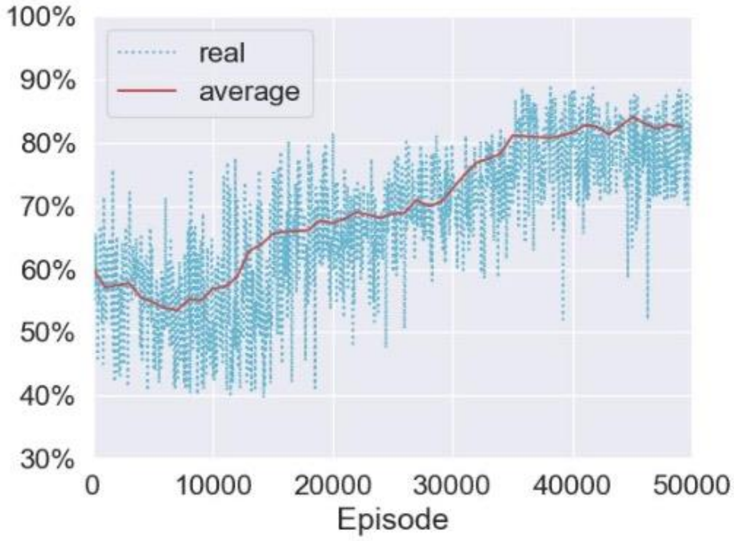

Figure 8: Profit rate change with training episodes

Figures 7 and 8 show the changes in the task acceptance rate and profit rate of the proposed method with training rounds, where blue is the true value and red is the average. The trends in the two figures are basically consistent. At the beginning of training, the agent belongs to the tentative exploration stage, and the task acceptance rate and profit rate have both declined slightly, but then gradually increased. It can be seen that the task acceptance rate and profit rate gradually stabilized after training to about 35,000 rounds, when the model gradually converges.

Figure 9 shows the number of remaining vehicles in each round during the training process. A large amount of vehicle resources was idle during the initial training period. By learning the acceptance and rejection strategy for transportation tasks, the wasted vehicle resources of each modeling node are reduced gradually.

First-come-first-served (FCFS) task assignment method and contract network are used for comparing with the proposed method. FCFS method means that if a transport task arrives and the task departure node still has idle vehicles remaining, the task is assigned to it. If there is no idle vehicle remaining at the node, the transportation task is assigned to the neighboring node with idle vehicles. The contract net algorithm (CNA) is appropriately modified to fit the context of this paper (Hu et al., 2019).

Table 3 compares the experimental results of different algorithms with the proposed method. As can be seen from the data in this table, the task acceptance rate of each algorithm is similar, but our method performs better in profit rate, which means that it is effective and can make the vehicle obtain greater benefits.

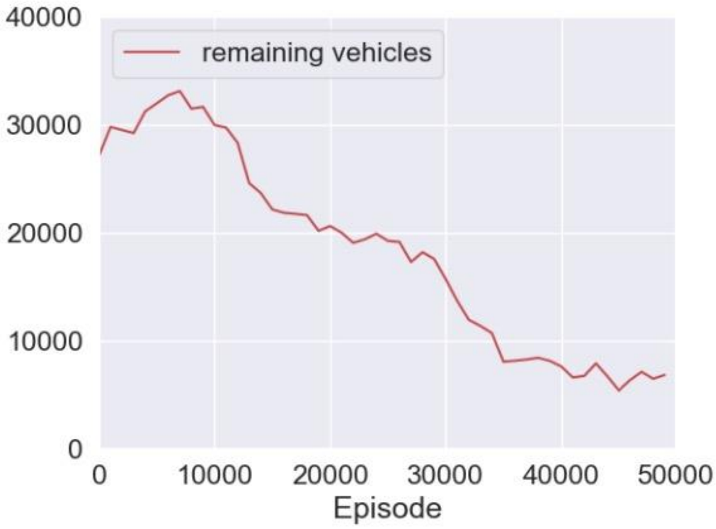

Figure 9: Number of remaining vehicles change with training episodes

Table 3: Comparison of task acceptance rate and profit rate of different algorithms

Algorithms	task acceptance rate	profit rate
FCFS	89.373%	87.068%
CNA	89.565%	88.203%
Proposed method	89.555%	89.159%

6 Conclusion

A reasonable and efficient task assignment method is the direct means to improve the revenue. This paper proposed a dynamic task assignment method for vehicles in urban transportation based on multi-agent reinforcement learning. Aiming at the problem of unreasonable task assignment due to greedy choice, an event-driven random game model was developed to describe the task assignment problem of vehicles. An extended actor-critic (AC) algorithm is proposed for model solution. The distributed network structure is used to construct a learning framework with the positions of various nodes as the decision-making subject in the urban transportation network. By comparing with the mainstream task assignment methods, our method can make vehicle operators achieve higher revenues while ensuring immediate response to transportation tasks.

Since the adopted framework involves the parallel computation of multiple neural networks and takes a long time for training and parameter optimization, the proposed method still has some shortcomings. In the subsequent research, the framework structure or mapping relationship can be further optimized to reduce its complexity and thus have more practical value.

Acknowledgment

The authors would like to acknowledge financial supports of the National Natural Science Foundation of China (No. 51775348), and the National Key Research and Development Program of China (No. 2019YFB1704401).

References

- Babaeizadeh, M., Frosio, I., Tyree, S., Clemons, J., Kautz, J. (2016). Reinforcement learning through asynchronous advantage actor-critic on a gpu. arXiv preprint arXiv:1611.06256.
- Ballot, E., Montreuil, B., Meller, R. (2014): The physical internet.
- Bhatnagar, S., Ghavamzadeh, M., Lee, M., Sutton, R. S. (2008). Incremental natural actor-critic algorithms. Advances in neural information processing systems, 105-112.
- Bouajaja, S., Dridi, N. (2015). Research on the optimal parameters of ACO algorithm for a human resource allocation problem. 2015 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 60-65, doi: 10.1109/SOLI.2015.7367412.
- Bowling, M., Veloso, M. (2002). Multiagent learning using a variable learning rate. Artificial Intelligence, v136, no2, 215-250.
- Chekuri, C., Khanna, S. (2005). A polynomial time approximation scheme for the multiple knapsack problem. SIAM Journal on Computing, v35, no3, 713-728.
- Chen, X., Fan, Z. P., Li, Y. H. (2009). Matching Problem of Employee and Task Based on Individual and Cooperative Factors. Industrial Engineering and Management, v14, no2, 120-124. (in Chinese).
- Chu, T., Wang, J., Codecà, L., Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. IEEE Transactions on Intelligent Transportation Systems, v21, no3, 1086-1095.
- Deng, D., Shahabi, C., Demiryurek, U., Zhu, L. (2016). Task selection in spatial crowdsourcing from worker’s perspective. GeoInformatica, vol20, no3, 529-568.
- DiDi. (2020). GAIA open dataset. https://gaia.didichuxing.com.
- Gabrel, V., Vanderpooten, D. (2002). Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite. European Journal of Operational Research, v139, no3, 533-542.
- Glaschenko, A., Ivaschenko, A., Rzevski, G., Skobelev, P. (2009). Multi-agent real time scheduling system for taxi companies. 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), 29-36.
- Gupta, J. K., Egorov, M., Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. International Conference on Autonomous Agents and Multiagent Systems Springer, Cham, v10642, 66-83.
- Hao, H., Jiang, W., Li, Y., Yuan, Z. (2013). Research on agile satellite dynamic mission planning based on multi-agent. Journal of National University of Defense Technology, v35, no1, 53-59. (in Chinese).
- Hasan, S., Ukkusuri, S. V. (2017). Reconstructing activity location sequences from incomplete check-in data: a semi-Markov continuous-time Bayesian network model. IEEE Transactions on Intelligent Transportation Systems, v19, no3, 687-698.
- Haydari, A., Yilmaz, Y. (2020). Deep reinforcement learning for intelligent transportation systems: a survey. IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2020.3008612.

- Hu, Y., Li, C., Zhang, K., Fu, Y. (2019). Task allocation based on modified contract net protocol under generalized cluster. Journal of Computational Methods in Sciences and Engineering, v19, no4, 969-988.
- Jia, Z., Yu, J., Ai, X., Xu, X., Yang, D. (2018). Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm. Aerospace Science and Technology, v76, 112-125.
- Jorge, D., Correia, G., H., A., Barnhart, C. (2014). Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. IEEE Transactions on Intelligent Transportation Systems, v15, no4, 1667-1675.
- Kachroo, P., Sastry, S. (2016). Traffic assignment using a density-based travel-time function for intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems, v17, no5, 1438-1447.
- Kaffash, S., Nguyen, A. T., Zhu, J. (2020). Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis. International Journal of Production Economics, v231, no107868, 1-15.
- Kleywegt, A. J., Papastavrou, J. D. (1998). The dynamic and stochastic knapsack problem. Operations research, v46, no1, 17-35.
- Konda, V., R., Tsitsiklis, J., N. (2000). Actor-critic algorithms. Advances in neural information processing systems.
- Kubek, D., Więcek, P. (2019). An integrated multi-layer decision-making framework in the physical internet concept for the city logistics. Transportation Research Procedia, v39, 221-230.
- Lan, C. (2018). Research on multi-task rapid scheduling technology for satellite networks. M. S. thesis, Xidian University, China. (in Chinese).
- Lin, J., T., Wang, F., K., Yen, P., Y. (2001). Simulation analysis of dispatching rules for an automated interbay material handling system in wafer fab. International Journal of Production Research, v39, no6, 1221-1238.
- Liu, J., L., Wang, L., C., Chu, P., C. (2019). Development of a cloud-based advanced planning and scheduling system for automotive parts manufacturing industry. Procedia Manufacturing, v38, 1532-1539.
- Lin, K., Zhao, R., Xu, Z., Zhou, J. (2018). Efficient large-scale fleet management via multi-agent deep reinforcement learning. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1774-1783.
- Morin, M., Gaudreault, J., Brotherton, E., Paradis, F., Rolland, A., Wery, J., Laviolette, F. (2020). Machine learning-based models of sawmills for better wood allocation planning. International Journal of Production Economics, v222, no107508, 1-10.
- Russell, R. A. (2017). Mathematical programming heuristics for the production routing problem. International Journal of Production Economics, v193, 40-49.
- Seow, K. T., Dang, N. H., Lee, D. H. (2009). A collaborative multiagent taxi-dispatch system. IEEE Transactions on Automation science and engineering, v7, no3, 607-616.
- Srivastava, S. C., Choudhary, A. K., Kumar, S., Tiwari, M. K. (2008). Development of an intelligent agent-based AGV controller for a flexible manufacturing system. The International Journal of Advanced Manufacturing Technology, v36, no7-8, 780.
- Xia, F., Wang, J., Kong, X., Zhang, D., Wang, Z. (2019). Ranking station importance with human mobility patterns using subway network datasets. IEEE Transactions on Intelligent Transportation Systems, v21, no7, 2840-2852.

- Zhang, Y. H., Gong, Y. J., Chen, W. N., Gu, T. L., Yuan, H. Q., Zhang, J. (2018). A dual-colony ant algorithm for the receiving and shipping door assignments in cross-docks. IEEE Transactions on Intelligent Transportation Systems, v20, no7, 2523-2539.
- Zhen, L., Yu, S., Wang, S., Sun, Z. (2019). Scheduling quay cranes and yard trucks for unloading operations in container ports. Annals of Operations Research, v273, no1, 455-478.
- Zhong, R. Y., Xu, C., Chen, C., Huang, G. Q. (2017). Big data analytics for physical internet-based intelligent manufacturing shop floors. International journal of production research, v55, no9, 2610-2621.