drones

MDPI

Article

Deep Reinforcement Learning for Truck-Drone Delivery Problem

Zhiliang Bi [1], Xiwang Guo [1,*], Jiacun Wang [2], Shujin Qin[3] and Guanjun Liu [4]

[1] School of Information and Control, Liaoning Petrochemical University, Fushun 113001, China; bizhiliang@stu.lnpu.edu.cn
[2] School of Computer Science and Software Engineering, Monmouth University, West Long Branch, NJ 07764, USA; jwang@monmouth.edu
[3] School of Economics and Management, Shangqiu Normal University, Shangqiu 476000, China; qinshujin@sqnu.edu.cn
[4] School of Electronic and Information Engineering, Tongji University, Shanghai 201804, China; liuguanjun@tongji.edu.cn
* Correspondence: xguo@monmouth.edu

Abstract: Utilizing drones for delivery is an effective approach to enhancing delivery efficiency and lowering expenses. However, to overcome the delivery range and payload capacity limitations of drones, the combination of trucks and drones is gaining more attention. By using trucks as a flight platform for drones and supporting their take-off and landing, the delivery range and capacity can be greatly extended. This research focused on mixed truck-drone delivery and utilized reinforcement learning and real road networks to address its optimal scheduling issue. Furthermore, the state and behavior of the vehicle were optimized to reduce meaningless behavior, especially the optimization of truck travel trajectory and customer service time. Finally, a comparison with other reinforcement learning algorithms with behavioral constraints demonstrated the reasonableness of the problem and the advantages of the algorithm.

Keywords: reinforcement learning; drone; path planning; road network

Citation: Bi, Z.; Guo, X.; Wang, J.; Qin, S.; Liu, G. Deep Reinforcement Learning for Truck-Drone Delivery Problem. *Drones* **2023**, *7*, 445. https://doi.org/10.3390/drones7070445

Academic Editor: Emmanouel T. Michailidis

Received: 17 May 2023
Revised: 22 June 2023
Accepted: 28 June 2023
Published: 6 July 2023

1. Introduction

An unmanned aerial vehicle (UAV) is an automatic or remotely controlled flying vehicle that can be classified by its wing type, power source, and other features. With the rapid progress of science and technology, the use of UAVs has become increasingly extensive [1,2]. The applications of UAVs have expanded beyond national defense and security to include industries such as agriculture, transportation, and photography. With the rapid growth of e-commerce and logistics worldwide, time and manpower costs have become key factors hindering further expansion. Efficient and fast transportation is needed to address this problem, hence the proposal and study of drone-truck joint delivery [3]. Currently, many large companies, including Amazon and Alibaba, are exploring the use of drones for fast delivery. With the advancement of drone payload and positioning technology, some prototypes and experimental studies have received further research [4,5]. In this article, UAV and drone both refer to unmanned aerial vehicles and will not be distinguished further in the following text.

The practical significance of truck-drone delivery lies in its potential to expand the distribution mode, which can help establish new logistics methods. Currently, logistics distribution faces challenges such as high labor costs, low efficiency, and poor transportation safety. The combination of truck and drone can solve these problems effectively. On the one hand, it can reduce labor costs and improve efficiency. Drones can reach remote areas, such as mountains and islands, that are difficult for trucks to access, thereby reducing delivery time and cost. On the other hand, truck-drone delivery can address the rising cost of human resources and the aging population in developed countries. Drones can reduce the workload and risks of delivery personnel, ensuring their health and safety.

Moreover, truck-drone delivery can improve distribution safety and reduce the impact of human factors on distribution. Overall, truck-drone delivery is an important direction for the future of logistics distribution. It is expected to become a representative of new logistics methods, promote the transformation and upgrading of the logistics industry, and contribute to economic development and social progress.

Reinforcement learning is a popular direction in the field of AI in recent years [6–8]. Based on behaviorism psychology, it influences the decision-making of intelligent agents by providing feedback rewards for their behavior, in order to achieve the final goal. Although various algorithms and mechanisms have been developed to study the carrying capacity and time efficiency of truck-drone transportation, few have employed reinforcement learning technology to simulate and systematically study the application of truck-drone in the field of transportation logistics. In this study, reinforcement learning was used to solve a truck-drone combined logistics delivery problem, and the effectiveness of the results was enhanced by introducing different reinforcement learning algorithms for comparison. The driving state of the delivery problem was divided into different categories, and reinforcement learning algorithms and mathematical models were used to solve the problem. An attempt was made to build a reinforcement learning environment by introducing truck routes and drone delivery times based on real maps and models, and to solve the problem from the perspective of reinforcement learning.

The integration of reinforcement learning with the environment provides a more effective means to simulate vehicle movement trajectories, road network conditions, and the dynamic effects of decision-making in real road network environments. In the context of driving trajectories and customer distribution, reinforcement learning enables direct learning through simulation environments, whereas traditional heuristic algorithms often rely on encoding or complex mathematical mappings for calculations. This approach allows for a more intuitive and adaptable learning process, as the algorithm learns and adapts to different scenarios without relying on explicit rules or extensive pre-defined models. By leveraging the power of reinforcement learning, the algorithm can effectively navigate and optimize vehicle movements in complex road networks, leading to improved performance and more efficient decision-making.

Based on an analysis of the delivery problem under real road network conditions, this study attempted to establish a unique reinforcement learning environment. This was carried out by introducing truck routes and drone delivery times, among other attributes, based on real maps and models, and solving the problem from the perspective of reinforcement learning. The environment included the direction selection of vehicles at each intersection and the drone delivery behavior at the parking point. The truck served as the mobile launch and recovery site for the drone, achieving optimal transportation routes from the perspective of a single target problem. The goal of solving the delivery problem was to minimize customer waiting time, which was modeled as a single-target integer programming problem. Reinforcement learning algorithms were then used to optimize route selection by randomly generating customers in the fixed road network environment. A new Gym-based general environment was created to support reinforcement learning.

The specific contributions of this article are as follows: (1) The use of deep reinforcement learning to study the truck-drone problem, which expands the research ideas for solving this problem; (2) for deep reinforcement learning, a general simulation environment for the truck-drone delivery problem was established for the first time using the Gym library, providing a basic comparative case for future related research; (3) this research focused on enhancing the learning efficiency of the agent in the context of truck-drone delivery. By reducing the action space, the algorithm can focus on selecting more meaningful and efficient actions, improving overall learning efficiency. The aim was to address the truck-drone delivery problem more effectively and provide faster and more efficient delivery solutions for customers.

The rest of the paper is organized as follows. Section 2 introduces the relevant research and technical approaches in the field of truck-drone delivery in recent years. Section 3

elaborates on the mathematical model of the problem and the simulation environment. Section 4 introduces the reinforcement algorithm and element definition. Section 5 compares the performance of the algorithm. Finally, Section 6 summarizes this paper and proposes prospects for future research.

2. Related Work

The distribution problem has always been a major concern in academic and industrial circles, particularly as the logistics industry becomes increasingly developed and important. Currently, logistics distribution within a single city typically relies on trucks for large-scale transportation to specific distribution sites, after which couriers transfer the goods through small vehicles or customers pick them up at home. However, this traditional logistics distribution mode has difficulty meeting the needs of efficient, convenient, and safe modern logistics. Furthermore, as the cost of manpower and time continues to rise, academia and industry have begun exploring the use of drones for distribution to solve the last kilometer distribution difficulties. Many solutions have been proposed in the literature, such as optimizing distribution routes, utilizing unmanned vehicles, and updating distribution methods to minimize last-mile logistics expenditure. As a new logistics distribution tool, drones offer the advantages of small size, high flexibility, fast speed, wide distribution range, and low cost, and are considered to be an important development direction for logistics distribution in the future. Consequently, academia and industry are actively exploring the research and development of drone delivery technology to achieve intelligent and automated logistics distribution [9,10].

In recent years, with the development of technology and cost reduction, drone technology has rapidly progressed and the feasibility of applying drones to the logistics industry has been extensively researched. Numerous optimization models have been developed for various purposes, such as surveillance [11,12], disaster relief activities [13], and package delivery [14]. The drone delivery problem model is a modified version of the vehicle routing problem. Drones possess higher speed, maneuverability, and fast positioning capabilities compared to ground vehicles, making them more predictable in route trajectories and faster in delivery capabilities. However, drones also have certain limitations, such as the problem of flight radius affected by energy, restrictions on carrying capacity, and potential dangers caused by malfunctions, which are currently facing challenges for large-scale applications of drones. Therefore, researchers have proposed a truck-drone delivery model that uses trucks as mobile platforms for launching drones and carriers for large-scale transportation of goods. This model coordinates the fast transportation capability of drones with the loading capacity of trucks to achieve fast, low-cost, and convenient characteristics [15–17].

Similarly, the combined truck-drone delivery model has different constraints and requirements to similar tasks in traditional trucking. In particular, the truck-drone distribution model requires more consideration of drone flight distance, drone route planning, drone safety, and other parameters. Compared to traditional human delivery, the truck-drone delivery problem requires more consideration of the combined location of the parking or drone launch point and the customer, rather than the distance between the two. Therefore, the truck-drone joint distribution mode requires a more intelligent and automated scheme to meet the real-time scheduling, path planning, and flight safety requirements in the distribution process. In short, as a new logistics distribution mode, the truck-drone joint distribution mode requires in-depth research and exploration in many aspects to achieve efficient, safe, and sustainable distribution.

John Gunnar Carlsson et al. [18] determined a logistics system in which drones serve as service providers, traveling back and forth on moving trucks. By combining Euclidean plane theory with simulated real road network data, they determined the superiority of this method in terms of efficiency and related formulas. Some researchers used a mixed integer linear programming (MILP) model for transport route optimization planning in multimodal systems combining truck and drone operations and proposed an effective

truck-drone routing algorithm (TDRA). Based on various problem cases, they confirmed that the truck-drone scheme is better than the pure truck delivery scheme.

The study conducted by Mohammad Moshref-Javadi et al. focused on the utilization of drones to address the Traveling Salesman Problem (TSP) [19]. Some research focused on a scenario where both a drone and a truck were utilized [20]. The authors assumed that the truck could park at the customer's location and deploy the drone for multiple deliveries while in a stationary state, effectively extending the Traveling Repair Problem (TRP). To address this problem, a mixed taboo search simulated annealing algorithm was employed, and extensive experiments and comparisons were conducted .

Pedro L. Gonzalez-R et al. studied the "last mile" problem in urban logistics distribution and took into account the energy constraints of drones, as well as the truck-drone delivery problem when drones require battery replacements or charging. They utilized a greedy heuristic algorithm for optimization [21] and proposed a mathematical model to address the energy limitations of drones, which expanded the use of drones in urban logistics. Furthermore, researchers investigated a new truck-drone collaborative delivery system in response to the COVID-19 pandemic. This system eliminates contact transmission and reduces the risk of disease spread. The researchers implemented an encoder-decoder framework combined with reinforcement learning to solve the routing problem without the need for manual heuristic design, resulting in improved generality.

A comprehensive study addressing the joint allocation problem of drones and vehicles is presented in reference [22]. Unlike general research on the combination of two transportation tools, this study introduced a third tool, consisting of drones and vans for delivery, while trucks were responsible for transporting goods and related equipment to designated stations. The study aimed to reduce the total delivery cost, with the fixed cost of each drone parking lot corresponding to the cost of using a drone station. Additionally, the study proposed a discrete optimization model for the problem, in addition to using a two-stage heuristic algorithm. The test results evaluating the applicability and efficiency of the algorithm demonstrated that combining traditional delivery methods with drones and drone stations can significantly reduce costs and increase profit margins.

Most of the articles mentioned above employed heuristic algorithms, such as traditional genetic algorithms, to optimize the truck-drone joint delivery model [23,24]. However, this paper innovatively adopted reinforcement learning and the Gym environment library to simulate and optimize the truck-drone joint distribution model based on basic road network information. Compared to traditional algorithms, the proposed method can better handle real-time scheduling and path planning requirements in the actual situation and can be carried out multiple times in the simulation environment to obtain more accurate and reliable results. Specifically, the reinforcement learning algorithm used in this paper can constantly adjust the strategy during practical operations to achieve optimal delivery results. Furthermore, the Gym environment library provides an open simulation environment based on reinforcement learning, which facilitates simulation experiments and parameter adjustments.

Currently, routing for pure truck delivery with waiting time as the indicator is referred to as the Minimum Latency Problem (MLP) or the TRP [25,26]. They are customer-centric, with the goal of minimizing customer waiting time. As businesses and logistics organizations increasingly focus on customer satisfaction and faster delivery lead times, and with the special advantages of drones in speedy delivery, this study aimed to minimize customer waiting time. By adopting a truck-drone joint delivery model, customer waiting time can be effectively reduced and delivery efficiency and customer satisfaction can be improved [27,28].

3. Problem Statement and Mathematical Model

In this section, we will introduce the delivery model and road network situation of the truck-drone joint delivery model considered in this article and explain the differences to traditional truck delivery systems. At the same time, we will also explain in detail the

assumptions made by the model and provide a preliminary explanation of the generation and setting of the environment.

3.1. Problem Description

In the truck-drone delivery model, a hybrid delivery system consists of a truck and a set of drones is used. The truck acts as a cargo loading platform and a drone launch platform, leveraging its high load capacity and characteristics as a freight transportation vehicle. As a drone launch platform, the truck provides energy, maintenance, command, and drone positioning services. As a cargo loading platform, it compensates for the limited transport capacity of drones and reduces transportation costs compared to pure drone transport. The drones are launched from the truck at specific locations such as warehouses or parking lots, allowing the truck to effectively expand the service radius of the drones. The objective of this study was to minimize customer waiting time by determining the truck's traveling path in a stochastic environment.

Figure 1 clearly illustrates the research problem addressed in this article. The figure depicts a delivery environment composed of elements such as parking spots, distribution centers, and road networks. The truck initially departs from the distribution center and travels along city roads. Upon reaching a designated parking spot, the vehicle stops, and the drone is launched for delivery based on customer location information. Due to the characteristics of the drone, we can consider the straight-line distance between the parking spot and the customer as the drone's flight trajectory. Additionally, considering the limited flight radius of the drone, we set a radius around the truck as a constraint to confine its flight behavior and prevent the drone from going beyond the control range. After completing the delivery tasks within the designated area, the truck carries the drone and proceeds to make action selections at the available turning points, determining the direction of travel until the tasks are completed.

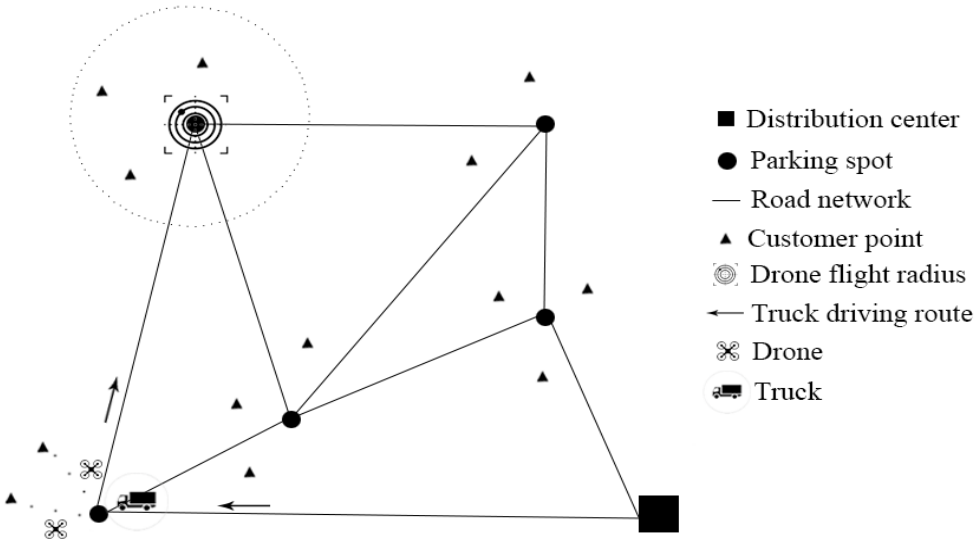


Figure 1. Truck-drone distribution model.

Figure 2 shows the road network information of a Chinese city obtained using the SUMO software and OpenStreetMap satellite image. This study evaluated population density and set up customer distribution areas with several residential points as distribution centers using one truck and five drones in this road network environment. This simulation scheme is practically significant and useful for reinforcing the decision-making skills of reinforcement learning agents in exploration. By defining parking points, this study determined the flight strategy of the drones while minimizing other influencing factors during the flight process of the drones.

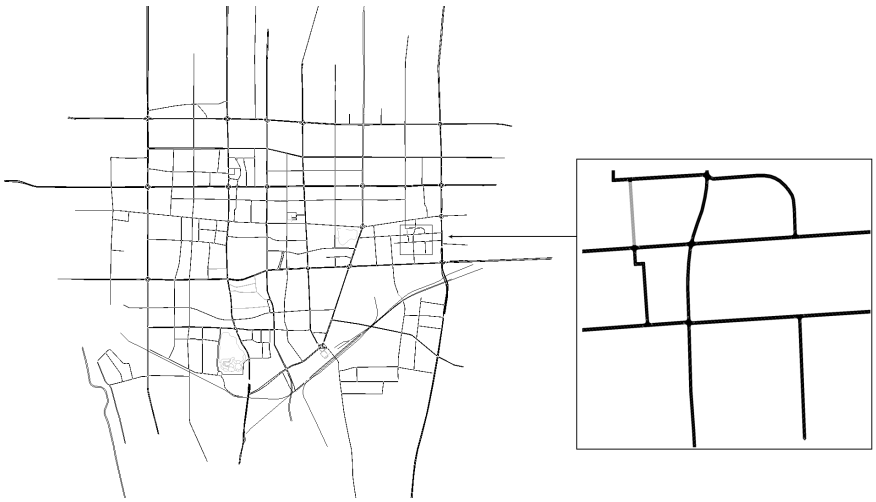


Figure 2. Road network information map.

This study compressed the action space by combining the characteristics of vehicle transportation and compressing the selection time and early learning costs. This avoids falling into the "dimension disaster" and better integrates the algorithm characteristics. The Deep Q-Network(DQN) algorithm used in this study uses the memory pool to help the intelligent agent estimate the number of customers and the time it takes, to some extent.

On the basis of the constraints and objective function, the model developed in this study was based on the following assumptions: the velocities of both drones and trucks were assumed to be constant and the impact of weather factors, such as wind speed and temperature, on drones was not considered. Additionally, any failures of vehicles or drones were not taken into account. Building upon these assumptions, this research addressed the truck-drone delivery problem through simulation-based optimization. By formulating a mathematical model and transforming it into a reinforcement learning environment, we aimed to optimize vehicle routes within the simulated environment to minimize customer waiting time.

3.2. Notations

The model proposed in this paper was based on existing mathematical models reported in various studies, such as [29,30], and considered additional parameters related to truck traveling speed and other aspects of the TSP. However, based on the algorithm and simulation scenarios presented in this paper, we have made extensions and innovations to certain formulas and definitions, enabling the model to better adapt to a reinforcement learning environment and making it more suitable for applications involving a fixed-point launch of UAVs and customer distribution.

3.2.1. Mathematical Notations

1. P The number of parking points;
2. C Number of customers;
3. U Number of vehicle-mounted drones;
4. U_d^{max} Maximum flight distance of drone (flight radius);
5. V_s Truck speed;
6. U_s Drone flight speed;
7. $d_{i,j}^V$ The distance of the truck from parking point i to parking point j, where $i, j \in P$;
8. $d_{i,k}^U$ The distance of drone from parking point i to customer k, where $i \in P, k \in C$;
9. $t_{i,j}^T = d_{i,j}^V / V_s$,the travel time of the truck from parking point i to parking point j, where $i, j \in P$;
10. $t_{i,k}^D = 2 * d_{i,k}^U / U_s$,the flight time of the drone from parking point i to customer k, where $i \in P, k \in C$;

11. U_a If drone a in the truck is used, it is 1; otherwise, it is 0, where $a \in U$;
12. P_i^T Total time spent at parking point i, where $i \in P$;
13. C_k^{max} Maximum waiting time limit for an individual customer k, where $k \in C$;
14. $C_{i,k}$ Waiting time for an individual customer k receiving delivery service at parking spot i, where $k \in C$, $i \in P$;
15. C_k^T Total waiting time for customers, where $k \in C$.

3.2.2. Decision Variables

$$P_i = \begin{cases} 1, & \text{Parking spot } i \text{ is visited} \\ 0, & \text{otherwise.} \end{cases}$$

3.2.3. Objective Function

$$min\, C_k^T \qquad C_k^T = \sum_{i \in P} P_i * (\sum_{i \in P} \sum_{j \in P} t_{i,j}^T + \sum_{i \in P} \sum_{k \in C} t_{i,k}^D).$$

3.2.4. Constraints

(1) The customers to be serviced should be within the flight radius of the drone taking off from the parking spot.

$$d_{i,k}^U \leq U_d^{max};$$

(2) A drone can only be occupied by one customer for the duration of its service.

$$max \sum_{a \in U} U_a \leq U;$$

(3) A parking spot has only one truck docking.

$$P_i \leq 1;$$

(4) It must be delivered within the customer's waiting time limit, or it will be punished for exceeding the time limit.

$$C_k^{max} - C_{i,k} \begin{cases} > 0, & \text{award} \\ < 0, & \text{punishment.} \end{cases}$$

The objective of the objective function is to minimize the total waiting time for all customers. The drone will deliver packages as much as possible within the customer service time limit and try to visit each customer. Constraint (1) ensures that the drone will not cause adverse effects due to exceeding the control distance or flight radius. Constraint (2) ensures that the drone only serves one corresponding customer at the same time to avoid safety issues. Constraint (3) is used to ensure that only one truck is parked at each parking spot to avoid efficiency losses caused by arbitrary parking. Constraint (4) is used to specify the service time and corresponds to the reward in reinforcement learning.

Due to the utilization of reinforcement learning in a simulated Gym environment for testing purposes in this study, it is important to note that the obtained results may not necessarily represent the optimal solution but rather an approximation of it. This is attributed to the considerable computational resources required to solve the problem within this simulation environment. The research was constrained by hardware limitations and computational power, thus preventing the attainment of an optimal solution, particularly concerning the trajectory problem for UAVs. In this study, UAVs were employed solely as delivery vehicles, simplifying the flight trajectory to a direct Euclidean distance between two points. Nonetheless, as evidenced in Tables 1–3, the solving capability of reinforcement learning progressively improves and converges towards the optimal solution by increasing

the number of test steps. The simulation environment offers various advantages, including the ability to conduct generic algorithm testing, generate random customer scenarios, and make informed decisions regarding truck speed and driving trajectory based on real-world conditions.

Table 1. Action constraint contrast.

Action Constraints		Withput Action Constraint	
Turning Point	**Non-Turning Point**	**Turning Point**	**Non-Turning Point**
N-N,S,W,E	S-S	N-N,S,W,E	S-S,N
N-N,S,W,E	N-N	N-N,S,W,E	N-N,S
N-N,S,W,E	W-W	N-N,S,W,E	W-W,E
N-N,S,W,E	E-E	N-N,S,W,E	E-E,W

Table 2. Experimental results of DQN algorithm.

No	Number of Test Rounds	Number of Steps per Game	Memory Pool	Attenuation Coefficient	Run Time (S)	Maximum Reward
1	200	600	1500	0.8	321.3	3102
2	200	600	500	0.999	305.7	2768
3	200	600	1500	0.999	310.3	2533
4	100	800	800	0.999	199.9	5257
5	100	800	2000	0.8	200.8	5214
6	200	800	2000	0.999	416	5476
7	100	1200	2000	0.9999	306.3	11,108
8	100	1200	8000	0.9999	290.6	12,795
9	100	1200	16,000	0.9999	280.2	13,686
10	100	2000	16,000	0.9999	498.8	28,790

Table 3. Comparison of DQN with A3C algorithm.

Test Rounds	Steps per Game	Maximum Reward		Running Time	
		DQN	**A3C**	**DQN**	**A3C**
100	600	2325	1735	135.6 s	133.6 s
200	600	3806	3525	284.9 s	237.3 s
400	600	2678	2565	608.8 s	455.0 s
100	1200	14,152	8246	277.2 s	245.4
200	1200	14,442	9237	617.0 s	459.4 s
100	2000	28,790	22,364	498.8 s	409.6 s

4. Reinforcement Learning for the Truck-Drone Delivery Problem

In this section, we will explain the characteristics and application of reinforcement learning algorithms that have been utilized to solve the problem. We will also introduce the origin of the road network information and discuss the issues regarding the environment setup. This study employed deep reinforcement learning technology, which involves introducing neural networks into basic reinforcement learning and expanding and improving them according to the problem characteristics. The action space and decision-making quantity are compressed so that the agent can make decisions based on specific states. Moreover, a universal environment was constructed using the third-party library Gym, which allows for different deep reinforcement learning algorithms to be universally tested, improving code utilization and enabling more detailed comparisons between various models and algorithms.

4.1. Reinforcement Learning and Deep Q-Network

Reinforcement learning is a machine learning paradigm that takes inspiration from behaviorist psychology. It focuses on enabling intelligent agents to learn through exploration and interaction with their environment to obtain rewards. The ultimate goal is to discover the optimal behavior pattern that maximizes rewards or achieves objectives within the environment. The most common and fundamental model of reinforcement learning is derived from Markov decision processes. Early deep reinforcement learning was initially used to perform comparative experiments in simple games and achieved excellent results [31]. By introducing neural networks, deep reinforcement learning can leverage their technical characteristics to perform various functions, such as parallel testing, memory pooling, adversarial generation, and policy evaluation. In this study, we utilized the characteristics of deep reinforcement learning and the high generality offered by the Gym environment. Through testing multiple algorithms, we implemented a solution to the truck-drone delivery problem based on deep reinforcement learning.

In the early experiments, the study gave the agent full decision-making power over the entire route and only provided certain basic rules based on the driving model. This environment with fewer action constraints produced some initial solutions but caused a lot of wasted time during the exploration and learning phase, with the truck often making U-turns and traveling back and forth to find the correct path. Based on this situation, the study made certain improvements to the action space by specifying that, after one decision, if there were no available turning nodes, the action would be forcibly changed to continue driving until a turning node was encountered for the agent to make the next decision. This method effectively reduced the action and state space, reduced the exploration and learning time, and greatly improved the feasibility of road network simulation testing.

Reward objective function:

$$max(\sum_{i=1}^{P} C_k^{max} - \sum_{k \in C} \sum_{i=2}^{P} C_{i,k} - \sum_{i=1}^{P} C_i^T).$$

This study primarily adopted the DQN algorithm in deep reinforcement learning [32]. DQN is a value-based reinforcement learning algorithm derived from the Q-Learning algorithm, chosen for its advantageous features in simulation environment testing. Firstly, as a value-based approach, DQN is well-suited to addressing the objective function and constraints of this study. Secondly, DQN has undergone extensive discussions and experimentation by researchers, demonstrating a remarkable performance in routing and logistics domains. Additionally, DQN introduces a memory pool mechanism that, when combined with neural networks, enables optimized recording of actions, such as turning at intersections, surpassing other algorithms such as A3C. Furthermore, the powerful processing capability of neural networks helps alleviate the "dimension disaster" and enhances generalization ability. In the following section, we will provide a brief overview of the underlying principles of the DQN algorithm.

A Q-Learning algorithm is value based. The Q value, $Q(S_t, A_t)$, determines how good an action A_t, taken at state S_t, is. When the agent takes some action $A_t(A_t \in A)$, the environment E will give feedback to the new state reached by the action and reward r obtained by the action. Therefore, the main idea of this algorithm is to obtain the corresponding Q table through the state S and behavior A to store Q values. Research and selection are made according to the Q values obtained.

At the same time, both DQN and Q-Learning solve the optimal sequence in the Markov decision process through the Bellman equation, which is consistent with the form of our research object. The state value function $V_\pi(s)$ is used to determine the value of the current state, and the value of each state is not only determined by the value of the current state itself but also by the reachable states that follow. Therefore, the expected cumulative reward

of the requested state can guide the state value of the current state. The Bellman equation for the state value function is as follows:

$$V_\pi(s) = E_\pi[r_{t+1} + \gamma V(S_{t+1})|S_t = S]. \quad (1)$$

In Formula (1), γ is the attenuation coefficient. When the attenuation coefficient approaches 1, it means that the corresponding agent can see the value of the future state more clearly. Pay more attention to the cumulative value of subsequent states.

Although Q-Learning can quickly solve problems with relatively small state and action spaces [33], once the problem becomes complex or has a complex state or action space, it becomes difficult to construct the Q-table, especially for the value matrix disassembled from the input object, where too many parts or operations will lead to an exponential increase in time cost and data, resulting in the "dimensional disaster".

As illustrated in Algorithm 1, DQN is a combination of Q-Learning and neural network algorithms. When confronted with the "dimensional disaster" arising from large state and action spaces, neural networks are employed to replace the Q table and circumvent memory limitations.

As shown in Figure 3, the basic idea is to utilize the memory of a neural network and process a large amount of data using a deep convolutional neural network approximation function. The DQN algorithm then learns from old or processed data using the experience replay mechanism, using the estimated Q value to approach the target Q value.

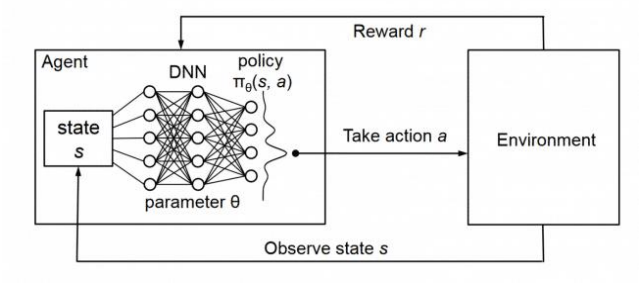


Figure 3. Agent and environment interaction diagram with DQN.

The pseudo-code for DQN training is listed below:

Algorithm 1 DQN training.

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
Initialize target action-value function $\hat{Q}$ with weights $\theta^- = \theta$
for episode 1,M do Initialize sequence $S_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$
  for t=1,T **do**
    With probability ε, select a random action a_t
    Otherwise, select $a_t = argmax_a Q(\phi(s_t), a; \theta)$
    Execute action a_t in the emulator and observe the reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
      Store experience $(\phi_t, a_t, r_t, \phi_{t+1})$ in D
      Sample random minibatch of experiences $(\phi_t, a_t, r_t, \phi_{t+1})$ from D
        Set y_j=

$$\begin{cases} r_j & \text{if episode terminates at step} \\ r_j + \gamma max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^-) & \text{otherwise} \end{cases}$$

      Perform a gradient descent step on $(y_i - Q(\phi_j, a; \theta))^2$ for the weights θ
      Every C steps reset $\hat{Q} = Q$
  end for
end for

The DQN method for solving the problem of a large state space is value function approximation (function approximation). The principle is to represent the function instead of the Q-table. This function can be linear or nonlinear.

$$\hat{v}(s,w) \approx v_\pi(s) \quad \text{or} \quad \hat{q}(s,a,w) \approx q_\pi(s,a),$$

Where w is the weight used to extract the feature values of the input state using a neural network or regression algorithm, and the output is calculated using TD. Then, the function is trained and converged to the point where the estimated value and the true value are close enough.

For the exploration of trial-and-error behavior mentioned above, the solution is greedy exploration. Every time the agent selects an action, it randomly selects an action with a certain probability ϵ and, in other cases, it selects the action with the maximum Q value among the currently available actions.

4.2. Case Description

The test case used in this article was based on a road network file generated from satellite images. Using the information from the road network file, the article converts it into a Gym environment for testing the algorithms.

Considering the real-world situation of the "last mile" problem, the research test environment is a city district in a certain city in China, which is also the residence of some of the researchers. This helps to set the distribution area and density of customers in the environment. At the same time, considering that many countries and regions have strict regulations on the entry of large trucks into urban areas, this article takes into account some real situations and considers the test vehicle as a small to medium-sized truck with a load of five drones and approximately 80 packages. It is assumed that the weight of the packages is within the load range of the drones.

The following is the setting of the case and environment in this article.

4.2.1. Environment

The use of the Gym library for building the reinforcement learning environment in this paper is beneficial in several ways. Firstly, it provides a high degree of versatility, allowing the environment to be used with most numerical calculation libraries. This flexibility makes it easier to integrate the environment with different machine learning algorithms and frameworks. Secondly, Gym provides a shared interface, allowing for the easy comparison and experimentation of different algorithms. This can help to identify the strengths and weaknesses of different approaches and guide further research [34].

The environment itself is designed to be realistic, incorporating road network information, parking points, and customer generation based on normal distribution. This makes the simulation more representative of real-world scenarios and can help to avoid overfitting or ineffective learning caused by unrealistic or uniform data. Overall, the combination of a realistic environment and the flexibility provided by the Gym library creates a strong foundation for tackling logistics and distribution problems using reinforcement learning techniques.

In this study, we made certain simplifications and assumptions regarding the flight aspects of UAVs. Due to hardware and practical limitations, we did not take into account factors such as airspace restrictions, wind speed, and flight control systems that can affect the UAV's flight path. Therefore, during the simulation and testing process, we assumed that the UAV would fly along the shortest distance between two points after takeoff. These simplifications were made for the sake of practical feasibility and computational complexity. In real-world environments, UAV flights involve numerous factors, such as airspace regulations, wind speed, and the operational status of flight control systems. Precise calculation and planning of UAV routes would require considering these factors, which are beyond the scope and feasibility of this study.

4.2.2. State

For different application problems, the definition of state also varies. Based on the characteristics of the truck-drone delivery mode and the road network information, most current research discretizes the continuous time when the truck moves in the environment and defines the discretized time slots as states. However, the definition of time slots varies depending on different scenarios, and is limited by environmental factors such as speed, stops, and distance. For example, some traditional heuristic algorithms treat the truck's movement between nodes as the basic state, while most reinforcement learning-based research defines a fixed time period as the state. At the same time, the composition of states is also an important distinguishing point. In this study, the entire map was represented as a two-dimensional plane using Gym and corresponding mathematical libraries, and Euclidean geometry was used to represent the coordinates of the truck, drone, and customers, which were set as part of the state.

In this study, the state S in reinforcement learning consisted of several elements, including coordinates x and y, time t, score s, and driving direction d. The updates of x and y coordinates are determined by built-in parameters such as velocity and driving direction. The time t is updated with each algorithm step. The score s is determined by the formula $C_k^{max} - C_{i,k}$, C_k^{max} represents the maximum waiting time for customers, while $C_{i,k}$, represents the current waiting time for a specific customer.

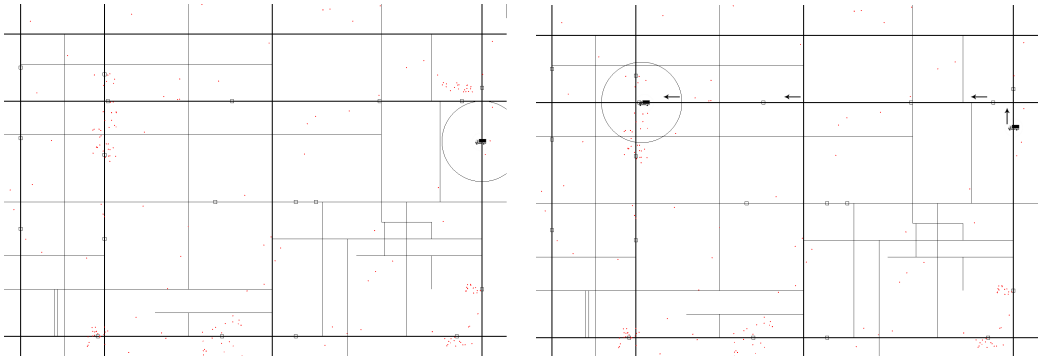
$$S = ((x, y), s, d, t).$$

The coordinates x and y provide spatial information about the agent's position in the environment. They help determine the agent's position relative to other entities such as the truck, drone, or customers. The time t captures the temporal aspect of the environment and allows the agent to track the progress and sequence of events during the delivery process. The score s represents the performance or utility of the agent's actions. It reflects the difference between expected and actual outcomes, guiding the agent to make more optimized decisions.

The driving direction d indicates the expected direction of the agent's movement. It plays a crucial role in determining the next state and subsequent actions that the agent will take. By combining these elements into the state representation, the agent can effectively perceive and interpret the environment, making wise decisions based on the current spatial and temporal context.

Speed is the main parameter of the truck-UAV distribution problem. In this paper, the truck speed was set as a fixed value under certain road conditions, without considering the influence of traffic lights or traffic accidents. At the same time, in order to get closer to the distribution model in the real business environment, this study set a maximum waiting time for the customer and used the time limit as the intermediate point to train the agent with two modes of reward and punishment. Figure 4 shows the state changes in the simulated environment in this article.

The environment depicted in Figure 4 is consistent with Figure 5, allowing us to determine the locations of parking spots and approximate customer distribution through a comparison between the two figures. In Figure 4a, the truck has just departed and has not yet passed any parking spots or launched the drone. As the system operates and the state progresses to Figure 4b, we can clearly observe that the truck has passed four rectangular boxes representing the parking spots and served a significant number of customers, as evidenced by the reduction in pixel points representing customers along the route in Figure 4b. Additionally, arrows have been roughly annotated to indicate the initial direction of travel for the truck and its subsequent trajectory. The state in Figure 4a can be represented as $S = ((1425,\ 650), 0, N, 20)$, while the state in Figure 4b can be represented as $S = ((320,\ 760), 920, W, 280)$.



(**a**) (**b**)

Figure 4. States. (**a**) The state after 20 timesteps of training; (**b**) The state after 20 timesteps of training.

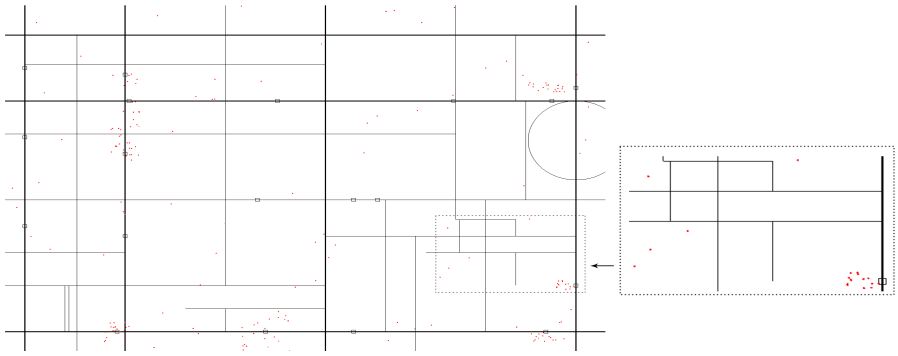


Figure 5. Simulated environment.

Overall, the composition of the state S with coordinates x and y, time t, score s, and driving direction d enables the agent to capture relevant information about its position, temporal progress, performance, and expected movements. This comprehensive state representation aids the agent's decision-making process and enhances its ability to navigate effectively and interact with the delivery environment.

4.2.3. Action

The ability to select and plan actions is a critical metric for evaluating reinforcement learning algorithms. An agent transitions from one state to the next through its behavior. In many reinforcement learning studies on driving, each individual time slice has a distinct control action or even multiple actions combined. However, this paper's primary research objective was the truck-drone distribution mode. Therefore, certain constraints were implemented on the movement space based on the original driving trajectory model to avoid problems caused by a large number of movements. The truck's driving path is constrained to an individual lane to avoid complications in the environment. Additionally, the vehicle's action choice is unrestricted at intersections to enable the agent to make better decisions in specific states. This reduction in motion space also promotes the consistency of state transitions, improving the DQN algorithm's memory pool advantage for decision-making in a specific state.

4.2.4. Reward

In the model establishment phase, the study added drone elements based on the early truck delivery research and considered some relevant parameters based on some truck delivery models and UAV logistics research ideas. The simulation system was optimized as much as possible and the number of parameters with smaller impact coefficients was minimized. In addition to treating all parking points and turning intersections in the entire traffic network as one type of action to adapt to deep reinforcement learning, this study

also used early random exploration strategies for testing to compare the advantages of reinforcement learning techniques.

Regarding the objective function, through some social surveys and related literature research, the study set the objective value to minimize the waiting time of all customers within a certain time limit; specifically, each customer has a fixed time limit. When the truck and drones complete the delivery within the time limit, the customer feedback will be positive and the reward will be the difference between the time limit and the delivery time. When the truck and drones fail to complete the delivery within the time limit, the feedback from each customer will be negative and the punishment will be determined by how much the delivery time exceeds the time limit. This setting is not only more realistic than the traditional approach of minimizing the total waiting time of customers but also facilitates the setting of rewards in reinforcement learning and accelerates the learning efficiency of the agent.

5. Experiment and Results

In this section, we tested the constructed Gym environment. First, we studied the learning efficiency and results with and without action constraints. Secondly, we conducted multiple tests on the model to obtain the impact of different parameters on the results. Finally, we evaluated the effectiveness of the environment and algorithm used, comparing them with constrained random exploration and the A3C algorithm to evaluate the algorithm's ability. All experiments were conducted using a unified environment, with a hardware environment of CPU AMD R7-5800H-3.2GHz, RAM of 16 GB DDR4 3200MHz, and a graphics card of RTX3060 8 GB. The software environment was Python 3.7.12 and Gym 0.21.0.

As shown in Figure 5, we proposed a Gym environment for testing delivery algorithms. Based on a third-tier city in China with a residential population of approximately 1.06 million, we designed a specific case with 19 parking points, seven customer gathering points, 200 customers, one truck, and five drones. These customers were randomly distributed in a square area centered on seven gathering points, following a normal distribution. As the flight radius of the drones was relatively large, the delivery areas centered on the 19 parking points overlapped to some extent. In this case, the truck started delivery from a fixed location.

Using the Gym environment, we were able to simulate a real logistics delivery scenario, test different delivery algorithms, and compare their performance through simulations. In this environment, we were able to observe the running status and effects of the algorithm in real-time and adjust and optimize parameters accordingly. Through such simulation experiments, we were able to gain a better understanding of the performance and optimization direction of the algorithm, providing better support for logistics delivery in real scenarios.

The following is the simulated environment generated for this study:

After multiple rounds of testing using the algorithm on the proposed model and environment, the results were as shown in Table 1. When using the DQN algorithm, expanding the memory pool and approaching a decay factor of 1, to some extent, increases the efficiency of learning by increasing the probability of a greedy strategy. Meanwhile, in terms of the environment, increasing the total number of steps or providing more time for delivery can also affect the results.

In the road network environment set up for this case, In the road network environment set up for this case, as demonstrated in Table 2 and Figure 6, the use of action constraints can increase the efficiency of the agent's learning. as demonstrated by tests 1–3, increasing the memory pool effectively improves the results and saves time. Furthermore, comparing tests 7–10 with the same number of test rounds, it can be observed that increasing the maximum number of steps per test allows the agent to better search for and serve customers, enabling it to find and serve more customers within the service time limit. In situations where the agent is limited by a finite number of steps and cannot explore the entire global

map, it will still strive to find the optimal solution within its feasible range. The comparison of tests 4–6 indicates that, even with a limited number of steps, the algorithm demonstrates the ability to approximate the optimal solution within a certain activity range.

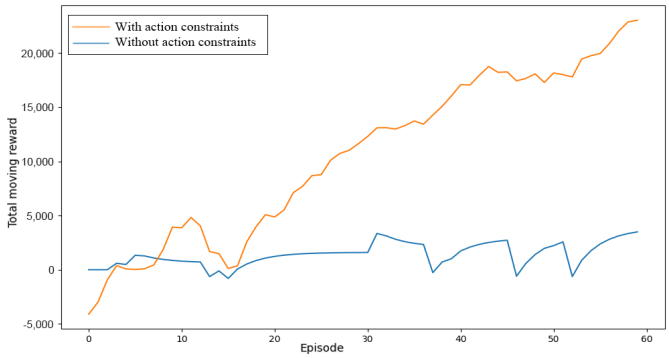
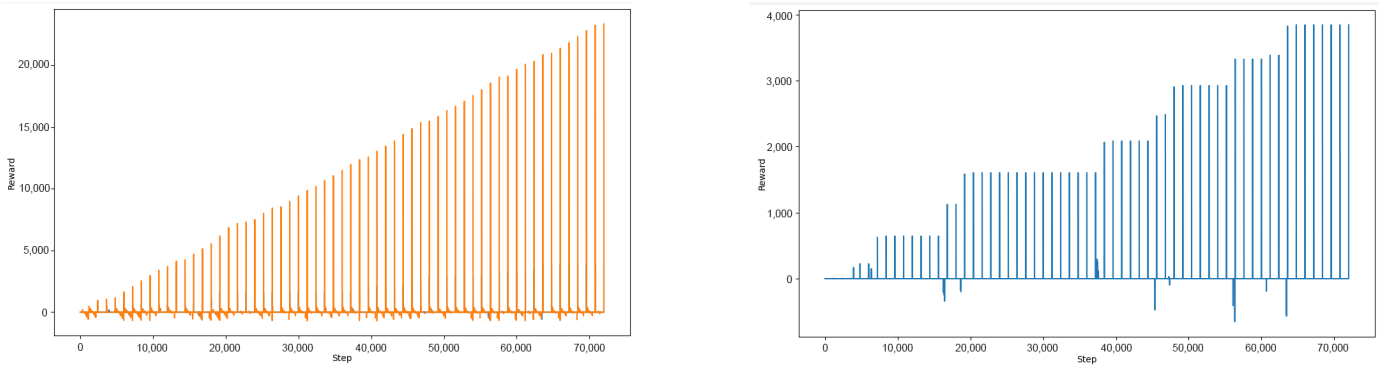


Figure 6. Effect of action constraints.

As can be seen from the comparison in Figure 7, without action constraints, the algorithm accumulates rewards much more slowly and the estimation of future value is not stable enough. This is because the agent spends too much time exploring and backtracking and cannot effectively explore other locations.



(**a**) (**b**)

Figure 7. Action constraint comparison. (**a**) Reword with action constraints; (**b**) Reword without action constraints.

From analyzing the single-round reward curve and the total reward curve, it is evident that the environment with action constraints displays a faster and more stable upward trend. In the last 20 rounds, it even achieves five times the total gain of the environment without action constraints. The single-round reward curve also confirms this, with the environment with action constraints showing a smoother increase while the environment without action constraints often enters a plateau period. Therefore, action constraints play a crucial role in the agent's learning process and can aid the agent in learning faster and achieving better results.

After demonstrating the superiority of the DQN algorithm over other algorithms, we tested its scalability. To address the problem of difficult convergence in random environments, we conducted 1200 games of 2000 steps each and 100 games of 3000 steps each using the DQN algorithm. We found that, although the DQN algorithm cannot guarantee long-term stable convergence, it can obtain stable and good results within a certain period and gradually improve its solving ability as the number of steps increases. Finally, we compared the DQN and A3C algorithms in Table 2 and demonstrated the effectiveness of the DQN algorithm in our research environment.

Overall, the experimental results of this case show that using the DQN algorithm for intelligent vehicle-drone delivery has significant advantages. Further improvements in algorithm learning efficiency and performance can be achieved through parameter adjustments and the use of action constraints and other techniques. These experimental results have significant reference value for the development and optimization of future intelligent vehicle-drone delivery systems [35].

6. Conclusions and Future Research

In this article, we created a simulation environment that is based on a truck-drone combination and uses real road network information to consider delivery tasks performed by a truck-drone combination in a fixed-point environment. We created a truck-drone delivery model with the objective of minimizing the total customer waiting time based on the original truck delivery model. We used the DQN algorithm to learn the optimal delivery schedule. Action constraints were proposed based on the problem characteristics to reduce action space, improve DQN learning efficiency, and reduce time cost.

Our comparative experiments yielded important conclusions regarding the DQN algorithm and its results. We found that an environment with action constraints can effectively improve the algorithm's learning efficiency and solution ability compared to an environment without action constraints. In an environment that minimizes total customer waiting time, we found that the DQN algorithm is better than the A3C algorithm. Additionally, when facing a large state space, amplifying the memory pool and decay coefficient in the DQN algorithm parameters helps the agent better estimate future rewards. Our results also showed that serving more customers does not necessarily mean getting more rewards when minimizing total customer waiting time. Therefore, further research on customer location, vehicle driving trajectory, and service methods under the truck-drone delivery mode will be valuable [36–41].

In terms of the environment, the Gym environment enables reinforcement learning to more universally test the environment, making it easier to compare and adjust different algorithms. The construction of the Gym environment also facilitates the combination of algorithms and mathematical models, as well as the introduction of dynamic environments. It is believed that more and more researchers will construct more universal and valuable reinforcement learning environments in the future [42].

With the continuous development of e-commerce and the logistics industry, the truck-drone delivery model will become increasingly valuable for research. Based on the current research situation, we will explore several possible research directions for the truck-drone transportation model in the future. Firstly, we can consider more complex and general environments by introducing a professional road network simulator, such as SUMO, based on the Gym environment. This will greatly enhance the feasibility and realism of this research direction and simplify the difficulties in environment construction for certain research aspects. Secondly, introducing multi-agent reinforcement learning algorithms for multiple trucks and drones delivery could be a breakthrough, especially when considering the coordinated delivery of multiple drones, which is beneficial for the practical implementation of the truck-drone delivery model. Finally, combining the truck-drone delivery model with reinforcement learning can leverage the advantages of reinforcement learning, particularly multi-agent reinforcement learning, to further investigate the multiple-agent delivery models mentioned earlier. Specifically, using multi-agent algorithms such as Multi-Agent Deep Deterministic Policy Gradient (MADDPG) and Nash Q-Learning can help us delve deeper into the research direction of cooperative game theory in the delivery problem.

Author Contributions: Formal analysis, J.W.; Data curation S.Q. and G.L.; Writing—original draft, Z.B. and X.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The funders had no role in the design of the study.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned aerial vehicle
DQN	Deep Q-Network
MILP	Mixed integer linear programming
TDRA	Truck-drone routing algorithm
MLP	Minimum latency problem
TRP	Traveling repair problem
TSP	Traveling Salesman Problem

References

1. Hu, B.; Wang, J. Deep learning based hand gesture recognition and UAV flight controls. *Int. J. Autom. Comput.* **2020**, *17*, 17–29. [CrossRef]
2. Zhou, P.; Liu, G.; Wang, J.; Wang, Q.; Zhang, K.; Zhou, Z. Lightweight unmanned aerial vehicle video object detection based on spatial-temporal correlation. *Int. J. Commun. Syst.* **2022**, *35*, e5334. [CrossRef]
3. Wang, K.; Yuan, B.; Zhao, M.; Lu, Y. Cooperative route planning for the drone and truck in delivery services: A bi-objective optimisation approach. *J. Oper. Res. Soc.* **2020**, *71*, 1657–1674. [CrossRef]
4. Schermer, D.; Moeini, M.; Wendt, O. A matheuristic for the vehicle routing problem with drones and its variants. *Transp. Res. Part C Emerg. Technol.* **2019**, *106*, 166–204. [CrossRef]
5. Boysen, N.; Briskorn, D.; Fedtke, S.; Schwerdfeger, S. Drone delivery from trucks: Drone scheduling for given truck routes. *Networks* **2018**, *72*, 506–527. [CrossRef]
6. Gu, J.; Wang, J.; Guo, X.; Liu, G.; Qin, S.; Bi, Z. A metaverse-based teaching building evacuation training system with deep reinforcement learning. *IEEE Trans. Syst. Man, Cybern. Syst.* **2023**, *53*, 2209–2219. [CrossRef]
7. Shi, H.; Liu, G.; Zhang, K.; Zhou, Z.; Wang, J. AMARL Sim2real transfer: Merging physical reality with digital virtuality in metaverse. *IEEE Trans. Syst. Man, Cybern. Syst.* **2022**, *53*, 2107–2117. [CrossRef]
8. Guo, X.; Bi, Z.; Wang, J.; Qin, S.; Liu, S.; Qi, L. Reinforcement Learning for Disassembly System Optimization Problems: A Survey. *Int. J. Netw. Dyn. Intell.* **2023**, *2*, 1–14. [CrossRef]
9. Arishi, A.; Krishnan, K.; Arishi, M. Machine learning approach for truck-drones based last-mile delivery in the era of industry 4.0. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105439. [CrossRef]
10. Liu, Z.; Li, X.; Khojandi, A. The flying sidekick traveling salesman problem with stochastic travel time: A reinforcement learning approach. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *164*, 102816. [CrossRef]
11. Zaheer, Z.; Usmani, A.; Khan, E.; Qadeer, M.A. Aerial surveillance system using UAV. In Proceedings of the 2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN), Hyderabad, India, 21–23 July 2016; pp. 1–7.
12. Gohari, A.; Ahmad, A.B.; Rahim, R.B.A.; Supa'at, A.; Abd Razak, S.; Gismalla, M.S.M. Involvement of surveillance drones in smart cities: A systematic review. *IEEE Access* **2022**, *10*, 56611–56628. [CrossRef]
13. Tanzi, T.J.; Chandra, M.; Isnard, J.; Camara, D.; Sebastien, O.; Harivelo, F. Towards" drone-borne" disaster management: Future application scenarios. In *Proceedings of the XXIII ISPRS Congress, Commission VIII (Volume III-8)*; Copernicus GmbH: Gottingen, Germany, 2016; Volume 3, pp. 181–189.
14. Chiang, W.C.; Li, Y.; Shang, J.; Urban, T.L. Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization. *Appl. Energy* **2019**, *242*, 1164–1175. [CrossRef]
15. Shahmoradi, J.; Talebi, E.; Roghanchi, P.; Hassanalian, M. A comprehensive review of applications of drone technology in the mining industry. *Drones* **2020**, *4*, 34. [CrossRef]
16. Lee, T.; Mckeever, S.; Courtney, J. Flying free: A research overview of deep learning in drone navigation autonomy. *Drones* **2021**, *5*, 52. [CrossRef]
17. Zhou, Z.; Liu, G.; Tang, Y. Multi-Agent Reinforcement Learning: Methods, Applications, Visionary Prospects, and Challenges. *arXiv* **2023**, arXiv:2305.10091.
18. Carlsson, J.G.; Song, S. Coordinated logistics with a truck and a drone. *Manag. Sci.* **2018**, *64*, 4052–4069. [CrossRef]
19. Moshref-Javadi, M.; Winkenbach, M. Applications and Research avenues for drone-based models in logistics: A classification and review. *Expert Syst. Appl.* **2021**, *177*, 114854. [CrossRef]
20. Moshref-Javadi, M.; Hemmati, A.; Winkenbach, M. A truck and drones model for last-mile delivery: A mathematical model and heuristic approach. *Appl. Math. Model.* **2020**, *80*, 290–318. [CrossRef]
21. Gonzalez-R, P.L.; Canca, D.; Andrade-Pineda, J.L.; Calle, M.; Leon-Blanco, J.M. Truck-drone team logistics: A heuristic approach to multi-drop route planning. *Transp. Res. Part C Emerg. Technol.* **2020**, *114*, 657–680. [CrossRef]

22. Wang, C.; Lan, H.; Saldanha-da Gama, F.; Chen, Y. On optimizing a multi-mode last-mile parcel delivery system with vans, truck and drone. *Electronics* **2021**, *10*, 2510. [CrossRef]
23. Wu, G.; Mao, N.; Luo, Q.; Xu, B.; Shi, J.; Suganthan, P.N. Collaborative truck-drone routing for contactless parcel delivery during the epidemic. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25077–25091. [CrossRef]
24. Baek, D.; Chen, Y.; Chang, N.; Macii, E.; Poncino, M. Energy-efficient coordinated electric truck-drone hybrid delivery service planning. In Proceedings of the 2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), Torino, Italy, 17–19 November 2020; pp. 1–6.
25. Moeini, M.; Salewski, H. A genetic algorithm for solving the truck-drone-ATV routing problem. In *Optimization of Complex Systems: Theory, Models, Algorithms and Applications, Proceedings of the WCGO 2019, Metz, France, 8–10 July 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1023–1032.
26. Zhao, L.; Bi, X.; Li, G.; Dong, Z.; Xiao, N.; Zhao, A. Robust traveling salesman problem with multiple drones: Parcel delivery under uncertain navigation environments. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *168*, 102967. [CrossRef]
27. Moshref-Javadi, M.; Hemmati, A.; Winkenbach, M. A comparative analysis of synchronized truck-and-drone delivery models. *Comput. Ind. Eng.* **2021**, *162*, 107648. [CrossRef]
28. Jiménez López, J.; Mulero-Pázmány, M. Drones for conservation in protected areas: Present and future. *Drones* **2019**, *3*, 10. [CrossRef]
29. Poikonen, S.; Golden, B.; Wasil, E.A. A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS J. Comput.* **2019**, *31*, 335–346. [CrossRef]
30. Tang, Z.; Hoeve, W.J.v.; Shaw, P. A study on the traveling salesman problem with a drone. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Proceedings of the 16th International Conference, CPAIOR 2019, Thessaloniki, Greece, 4–7 June 2019*; Proceedings 16; Springer: Berlin/Heidelberg, Germany, 2019; pp. 557–564.
31. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
32. Fan, J.; Wang, Z.; Xie, Y.; Yang, Z. A theoretical analysis of deep Q-learning. In Proceedings of the Learning for Dynamics and Control, PMLR, Berkeley, CA, USA, 10–11 June 2020; pp. 486–489.
33. Jang, B.; Kim, M.; Harerimana, G.; Kim, J.W. Q-learning algorithms: A comprehensive classification and applications. *IEEE Access* **2019**, *7*, 133653–133667. [CrossRef]
34. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. Openai gym. *arXiv* **2016**, arXiv:1606.01540.
35. Das, D.N.; Sewani, R.; Wang, J.; Tiwari, M.K. Synchronized truck and drone routing in package delivery logistics. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 5772–5782. [CrossRef]
36. Liu, Y.; Liu, Z.; Shi, J.; Wu, G.; Pedrycz, W. Two-echelon routing problem for parcel delivery by cooperated truck and drone. *IEEE Trans. Syst. Man, Cybern. Syst.* **2020**, *51*, 7450–7465. [CrossRef]
37. Guo, X.; Zhou, M.; Liu, S.; Qi, L. Multiresource-constrained selective disassembly with maximal profit and minimal energy consumption. *IEEE Trans. Autom. Sci. Eng.* **2020**, *18*, 804–816. [CrossRef]
38. Guo, X.; Zhou, M.; Liu, S.; Qi, L. Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints. *IEEE Trans. Cybern.* **2019**, *50*, 3307–3317. [CrossRef] [PubMed]
39. Zhang, G.; Zhu, N.; Ma, S.; Xia, J. Humanitarian relief network assessment using collaborative truck-and-drone system. *Transp. Res. Part E Logist. Transp. Rev.* **2021**, *152*, 102417. [CrossRef]
40. Baldisseri, A.; Siragusa, C.; Seghezzi, A.; Mangiaracina, R.; Tumino, A. Truck-based drone delivery system: An economic and environmental assessment. *Transp. Res. Part D Transp. Environ.* **2022**, *107*, 103296. [CrossRef]
41. Guo, X.; Liu, S.; Zhou, M.; Tian, G. Dual-objective program and scatter search for the optimization of disassembly sequences subject to multiresource constraints. *IEEE Trans. Autom. Sci. Eng.* **2017**, *15*, 1091–1103. [CrossRef]
42. Zhou, Z.; Liu, G.; Zhou, M. A Robust Mean-Field Actor-Critic Reinforcement Learning Against Adversarial Perturbations on Agent States. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–12. [CrossRef]