

Dynamic collaborative truck-drone delivery with en-route synchronization and random requests

Haipeng Cui ^{a,b}, Keyu Li ^b, Shuai Jia ^{c,d,*}, Qiang Meng ^e

^a State Key Laboratory of Intelligent Geotechnics and Tunnelling (Shenzhen University), Shenzhen, 518060, China

^b College of Civil and Transportation Engineering, Shenzhen University, Shenzhen 518060, China

^c Thrust of Intelligent Transportation, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511400, China

^d Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China

^e Department of Civil and Environmental Engineering, National University of Singapore, Singapore 117576, Singapore

ARTICLE INFO

Keywords:

Dynamic truck-drone delivery
Flexible launch and retrieval
Random requests
Reinforcement learning
Mixed integer programming

ABSTRACT

Coordinated truck and drone delivery is gaining popularity in logistics as it can greatly reduce operation costs. However, existing studies on related operations management problems typically ignore the following important features: (i) the random appearance of requests, which require operators to dynamically respond to the requests; and (ii) the decisions of optimal launch and retrieval locations for trucks and drones instead of fixed to customer locations, which can significantly impact the overall time costs. To tackle these challenges, this study investigates the dynamic collaborative truck-drone routing problem with randomly arriving requests and synchronization on routes. We model the problem as a Markov Decision Process (MDP) and solve the MDP via a reinforcement learning (RL) approach. The proposed RL approach determines: (i) whether each request should be serviced upon arrival, (ii) which truck or drone should be assigned for the request, and (iii) the optimal en-route take-off and landing positions for paired trucks and drones. We further employ a framework of decentralized learning and centralized dispatching in RL to increase performance. Numerical experiments are conducted to assess the proposed solution approach on instances generated based on both the Solomon dataset and real-world operational data of a logistics operator in Singapore over several benchmark algorithms under various battery endurance levels of drones and distinct transportation scenarios including node-based dynamic collaborative truck-drone routing problem, dynamic non-collaborative truck and drone routing problem, and dynamic vehicle routing problem. The results show that our RL solution outperforms the benchmark algorithm in total profit by an average of 28.03 %, and our en-route takeoff and landing scenario outperforms the benchmark scenarios in total profit by an average of 8.43 % in multi-day instances. Additionally, compared to the traditional node-based landing scenario, employing our en-route takeoff and landing strategy can save 0.9 h/(drone*day) of waiting time on average.

* Corresponding author.

E-mail addresses: cuihp@szu.edu.cn (H. Cui), 2210474184@email.szu.edu.cn (K. Li), shuaijia@hkust-gz.edu.cn (S. Jia), ceemq@nus.edu.sg (Q. Meng).

1. Introduction

The cooperation of trucks and drones in delivery garners increasing attention due to its distinct advantages over traditional truck delivery. Truck delivery suffers from traffic congestion, environmental pollution, and high operational costs. Some studies (Wang et al., 2020, Wang et al., 2024) have made significant contributions to the research on green transportation. Meanwhile, the collaborative utilization of trucks and drones provides opportunities to overcome these limitations, due to the ability of drones to bypass traffic congestion and lower energy consumption. It greatly increases the flexibility in handling randomly arriving requests, which is beneficial in multiple scenarios such as express delivery and food delivery. This study focuses on the problem of dynamic truck-drone coordination that arises from urban parcel delivery.

In most studies on collaborative truck-drone delivery, the information of requests including delivery locations and time windows is known to the operator in advance (Harbison, 2020, Cointreau et al., 2021), and the take-off and landing positions of drones are fixed at customer sites (Wang et al., 2019). On one hand, the logistics industry is witnessing a surging demand for dynamic logistics solutions currently. For instance, demands for dynamic logistics solutions specially tailored for same-day delivery items such as pharmaceuticals and fresh flowers are growing (Ramos et al., 2023). The demand for efficient and tactful deployment in humanitarian logistics, like relief aid distribution in calamity-stricken areas, underscores the need for dynamic logistics operations (Lu et al., 2023). On the other hand, traditional fixed-location launch and retrieval tend to incur substantial waiting times, resulting in an escalation of operational costs. Planck's proprietary technology equips drones with full autonomy for ground vehicle operations. Such a drone, outfitted with the AVEM system from Planck's suite, can effortlessly initiate flight from a mobile land vehicle without requiring the latter to cease motion. Simultaneously, the drones return to the ground vehicle and execute a precision landing on the bed of the ground vehicle (Li et al., 2022). Therefore, it is challenging to formulate a reasonable dynamic collaborative truck-drone routing problem with comprehensive consideration of key issues such as dynamic requests, path optimization, collaborative scheduling, flexible launch and retrieval, and drone restrictions.

To address these challenges, we investigate a dynamic collaborative truck-drone routing problem with en-route synchronization (DCRP-eS), in which paired trucks and drones respond to dynamically arriving requests. In addition, we consider a flexible situation that the drones can be launched and retrieved at arbitrary positions along the routes, instead of operating the drones at fixed positions. A logistics operator manages a fleet of trucks and drones inside this framework to fulfill delivery requests that arrive at random times. The operator also manages a depot that serves as a hub for unserved packages, idle vehicles, and drones. Each request originates from a depot and is associated with a delivery destination, a time window, and a weight. The operator needs to dynamically respond to customer specific service time windows for delivery operations, aiming to fulfill each accepted request within the designated time window. Deliveries may arrive beyond the expected latest delivery time, but a delay penalty applies.

We formulate the DCRP-eS as a Markov decision process (MDP), and we propose a reinforcement learning (RL) approach to solve the MDP. Specifically, the RL serves the purpose of determining whether a request should be serviced, and which truck or drone should be assigned for the request, while a proposed physical geometric model formulates the appropriate take-off and landing positions. We employ a framework of decentralized learning and centralized dispatching to increase the performance in RL. We train the learning model in a multi-agent training framework. Meanwhile, we formulate the static collaborative truck-drone routing problem with en-route synchronization (CRP-eS) as a mixed integer program (MIP) model for comparison by considering the trucks and drones synchronize at discrete points along the route with full knowledge of the request information. The main contributions of this study are as follows:

- We consider a novel DCRP-eS associated with urban parcel delivery. The unique aspects of this problem involve en-route synchronization between trucks and drones under real-time responses to randomly arriving requests, which is more practical cost-efficient. Additionally, we take into account the varying battery endurance levels of drones. To the best of our knowledge, no literature to date investigated the truck-drone collaborative delivery problem with en-route launch and retrieval in randomly arriving requests.
- We develop an MDP model to capture the dynamic interactions between trucks, drones, and the uncertain environment. This includes aspects such as the changing positions of the vehicles, and the requests randomly arriving. The MDP model is solved using a tailored RL method. Employing RL enables our system to iteratively improve its policy based on trial, eventually converging to one that maximizes the expected rewards represented by the objective function of MDP. Additionally, we employ a physical geometric model to choose the optimal en-route take-off and landing positions in the proposed RL method.
- We conduct extensive numerical experiments on instances generated from both the Solomon benchmark dataset and the real-world operational data of a logistics operator in Singapore. We assess the performance of the proposed RL approach over various benchmark methods, including: (i) a static CRP-eS problem solved by an MIP model that discretizes the points on the routes to achieve flexible launch and retrieval; (ii) a Large Neighborhood Search Algorithm (LNS) for the CRP-eS; (iii) a first-come first-served decision rule (FCFS). And various scenarios used for comparison: (i) a node-based delivery collaborative truck-drone routing approach (DCRP) that requires drones to be launched or retrieved only at customer nodes or the depot; (ii) a dynamic truck and drone routing problem (DTDRP) that truck and drone deliver separately; and (iii) a dynamic vehicle routing approach (DVRP) that schedules only trucks to serve requests. The results demonstrate the superiority of the DCRP-eS and proposed RL approach in terms of efficiency, economy, and flexibility.

The subsequent sections of this paper are structured as follows. Section 2 reviews related literature. In Section 2.3, the problem is defined. The MDP model is presented in Section 4. Section 5 formulates the MIP model. Section 6 elaborates on the RL method. Section

7 presents the computational experiments and the results, along with a discussion. Section 9 concludes and highlights potential avenues for future research.

2. Literature review

In this section, we critically review existing works on three related areas: vehicle routing problems with drone, truck-drone routing problem with flexible synchronization, and vehicle routing problem with random requests. Table 1 summarizes the literature that is most relevant to our work.

2.1. Vehicle routing problem with drone

Murray et al. (2015) is the pioneer in introducing the combination of truck and drone in delivery. They proposed the Flying Sidekick Traveling Salesman Problem (FSTSP), which involves using a combination of a single truck and a single drone for parcel delivery. Agatz et al. (2018) introduced the Traveling Salesman Problem with Drone (TSP-D), which resembles to the FSTSP. However, TSP-D allows the truck to wait at the launch node for the returning drone. These literature (Ha et al., 2018, Es Yurek et al., 2018, Jeong et al., 2019) also explored variants of the TSP-D and addressed them by heuristic algorithms. Furthermore, certain studies (Boccia et al., 2021, Vásquez et al., 2021, Cavani et al., 2021, Dell'Amico et al., 2022) concentrated on investigating exact methods for addressing the FSTSP and TSP-D.

Wang et al. (2017) introduced the Vehicle Routing Problem with Drone (VRP-D), encompassing multiple trucks and drones. Sacramento et al. (2019) also proposed a VRP-D delivery model and developed a heuristic algorithm for this model. In addition, besides the VRP-D with one drone on each truck investigated in these studies (Kuo et al., 2022, Gu et al., 2022, Lei et al., 2022, Luo et al., 2023, Meng et al., 2023), there is also literature (Schermer et al., 2019b, Kitjacharoenchai et al., 2020, Gao et al., 2023) exploring the VRP-D with multiple drones on one truck. However, most existing studies primarily focus on trucks only launching and retrieving drones at customer nodes or pre-determined nodes, inevitably resulting in deadhead waiting times.

The Truck and Trailer Routing Problem (TTRP) shares similarities with the TSP-D and VRP-D as these problems entail routing for two distinct vehicle types. In the TTRP, integration occurs between the truck and the trailer, whereas in the TSP-D and VRP-D, integration occurs between the truck and the drone. These studies (Villegas et al., 2013, Rothenbächer et al., 2018, Accorsi et al., 2020, Cui et al., 2022) explored research on TTRP. It is important to note that while the drone operates autonomously from the truck, the trailer in the TTRP must be transported by connecting truck.

2.2. Truck-drone routing problem with flexible synchronization

Flexible synchronization between trucks and drones can be divided into two categories: flexible paired and flexible launch and retrieval.

Wang et al. (2019) initially introduced the variant of VRP-D, where a drone can land on another truck after takeoff. And they designed an MIP model and devised a branch-and-price algorithm. In the literature of Masmoudi et al. (2022), a drone can return to a different truck than the one it started to swap its depleted battery and/or pick up additional packages. An Adaptive Multi-Start Simulated Annealing (AMS-SA) metaheuristic algorithm is employed to solve this problem. Kim et al. (2023) also relaxed the pair constraints to determine more flexible delivery plans. Long et al. (2024) proposed a Dynamic Truck-UAV (DTU) for urban emergency

Table 1
Summary of related studies.

Reference	Problem	Objective	T	D	TW	Delay	Dynamic	Flexible	Method
Marinelli et al. (2018)	TSP-D	Min cost	1	1			✓		MIP/heuristic
Carlsson et al. (2018)	TSP-D	Min time	1	1			✓		theoretical analysis/heuristic
Schermer et al. (2019a)	VRPDERO	Min time	m	1			✓		MILP/heuristic
Li et al. (2022)	TDRP-SA	Min cost	m	m	✓		✓		MIP/heuristic
Salama et al. (2022)	CTDRSP-FL	Min time	m	m			✓		MILP/heuristic
Masone et al. (2022)	MVDRP-EL	Min time	1	1			✓		MILP/heuristic
Thomas et al. (2023)	CTDRSP-ELR	Min time	1	m			✓		MILP/heuristic
Lan et al. (2024)	TSPTWD-IP	Min time	1	1	✓		✓		MIP/heuristic
Chung et al. (2024)	DTCO	Min time	1	1			✓		heuristic
Dayarian et al. (2020)	VRPDR	Max served orders	1	1	✓		✓		heuristic
Chen et al. (2022)	SDDPPVD	Max served orders	m	—	✓		✓		DQL
Han et al. (2023)	VRPDWTDD	Min cost	m	1	✓		✓		MIP/heuristic
Ramos et al. (2023)	DPDSVRP-LT	Min costs	m	—	✓		✓		MIP/CPLEX
Gu et al. (2023)	D-TDRP-SDOP	Max profit	1	1	✓		✓		MDP/heuristic
Yin et al. (2023)	VRP-D	Min cost	m	1	✓		✓		Branch-and-price-and-cut algorithm
Pina-Pardo et al. (2024)	DTSP-DR	Max served orders	1	m			✓		MDP/online policy
Our study	DCRP-eS	Max profit	m	1	✓	✓	✓	✓	MDP/DQN

T: number of trucks; D: number of drones on a truck; “-”: trucks and drones deliver separately; TW: restricted to time window; Delay: whether allowed to arrive later than the time window of the customer; Dynamic: whether the requests arrive randomly; Flexible: whether drones can take off and land on flexible positions instead customer node.

response in which drones can take off and land on different trucks. [Jiang et al. \(2024\)](#) explored a multi-visit flexible-docking vehicle routing problem in rural areas. The system allows drones to make multiple stops per trip, dock with the same or different trucks, and conduct simultaneous pickup and delivery operations in collaboration with the truck fleet.

[Marinelli et al. \(2018\)](#) considered that a truck can launch and retrieve a drone not only at a customer node but also along a route in TSP-D. In their research, trucks are positioned at suitable locations on the route to wait for drones. [Carlsson et al. \(2018\)](#) considered a drone picks up a package from the truck, which continues moving on its route. After delivering the package, the drone returns to the truck for the next request. [Schermer et al. \(2019a\)](#) proposed a Vehicle Routing Problem with Drones and En-Route Operations (VRPDERO). In contrast to VRP-D, the drones in VRPDERO can be launched and retrieved at both vertices and discrete points along arcs. While take-off and land on discrete points can enhance drone utilization rates and expedite task completion, it still leads to waiting times for trucks. [Li et al. \(2022\)](#) defined the Truck and Drone Routing Problem with Synchronized Arcs (TDRP-SA). The TDRP-SA considered synchronization on arcs, customer classification, direct delivery, multiple trucks, and multiple drones carried by each truck. To address the TDRP-SA problem, a mixed-integer nonlinear programming model was formulated. The researchers proposed a mathematical analysis approach to locate the mobile take-off and landing positions and estimate the arrival time of the drone at the customer location. [Salama et al. \(2022\)](#) considered three key decisions: (i) assigning each customer location to vehicles, (ii) routing trucks and drones, and (iii) scheduling drones taking off and landing. The authors established flexible takeoff and landing points at a set of known feasible truck stop locations, and then proceeded with the selection of suitable locations for drone takeoff and landing. [Masone et al. \(2022\)](#) concentrated on addressing the Multi Visit Drone Routing Problem with Edge Launches (MVDRP-EL). This problem assumes a heterogeneous set of packages, drones capable of carrying multiple packages simultaneously and launching and retrieving along edges, a flexible set of launch/retrieval sites, and a customizable energy depletion function defined by the user. They devised a lower bound formulation based on the Covering Salesman Problem (CSP) and devised a solution approach that incorporates both discretized and continuous methods. [Thomas et al. \(2023\)](#) introduced the single truck multi-drone routing and scheduling problem with en route operations, and proposed a Relax-and-Fix with Re-couple-Refine-and-Optimize (RF-RRO) heuristic approach. Their problem avoids collision during drone takeoff and landing but doesn't determine the specific takeoff and landing point en-route, replacing with an appropriate time. [Lan et al. \(2024\)](#) explored the utilization of Intermediate Points (IPs) as take-off and landing positions for drones, positioned along the arcs connecting customer nodes. The study delved into determining the circumstances wherein leveraging IPs proves advantageous, employing both theoretical analysis and empirical investigations. [Chung et al. \(2024\)](#) proposed a method of pseudo node insertion to achieve synchronization between trucks and drones. Pseudo nodes are generated at the intersection where a drone and a truck can meet simultaneously.

Existing flexible launch and retrieval approaches primarily emphasize the identification of favorable locations along the routes for drones, or positions that trucks and drones can meet simultaneously. The latter is the same as our purpose, but the greater challenge of our problem is to find en-route takeoff and landing points in the dynamic routing process that perfectly synchronizes the truck and drone.

2.3. Dynamic vehicle routing problem

Dynamic routing problems typically involve uncertainties in requests or travel times. The approaches to solving DVRPs typically involve heuristics ([Moghaddam et al., 2012](#), [AbdAllah et al., 2017](#), [Panadero et al., 2020](#), [Fritta et al., 2022](#), [Wang et al., 2022](#)), dynamic programming ([Ulmer et al., 2019](#), [Dastpak et al., 2023](#)), exact algorithm ([Liu et al., 2023](#)), or reinforcement learning methods ([Pan et al., 2023](#), [Chen et al., 2023](#)). We mainly focus on the dynamics of requests, and [Zhang et al. \(2023\)](#) provided a complete overview of dynamic vehicle routing with random requests. Furthermore, [Jia et al. \(2022\)](#) addressed the dynamic problem of containers by establishing a Markov Decision Process (MDP) model to capture the dynamic interaction between the container operator and the uncertain environment. A novel approach integrating reinforcement learning and integer programming was proposed to solve the MDP model. However, the design of the interaction between trailers and tractors in that context does not apply to the collaborative relationship between trucks and drones discussed in this paper. Our agents consist of paired trucks and drones, both actively involved in the delivery process, distinguishing agents from [Jia et al. \(2022\)](#) exclusively employing tractors.

Focusing on the problem most closely related to our research, we examine the truck-drone route planning problem with random requests. [Dayarian et al. \(2020\)](#) investigated the utilization of drones to resupply trucks in SSD problems. [Chen et al. \(2022\)](#) considered employing both trucks and drones for delivery in SSD solving by a deep Q-learning approach, but trucks and drones deliver separately. [Han et al. \(2023\)](#) designed optimization algorithms to address the challenges posed by strict time constraints and ever-changing customer requests. [Ramos et al. \(2023\)](#) introduced truck and drone parallel transportation in the field of medicines delivery, which also requires facing sudden and urgent requests. [Gu et al. \(2023\)](#) explored the Dynamic Truck-Drone Routing Problem with Scheduled Deliveries and On-demand Pickups (D-TDRP-SDOP) within an on-demand logistics framework. They addressed the D-TDRP-SDOP by modeling an MDP. The proposed solution strategy involves an Offline Enhanced Construction Algorithm (OECA) and a segment-based heuristic. [Yin et al. \(2023\)](#) investigated the vehicle routing problem with drones under uncertain demands and truck travel times. They developed an enhanced branch-and-price-and-cut algorithm to solve the problem. [Pina-Pardo et al. \(2024\)](#) addressed the dynamic vehicle routing problem with drone resupply for same-day delivery. They utilized an MDP to optimize real-time order management and routing, improving fill rates and ensuring fast customer service.

We address a novel dynamic collaborative truck-drone routing problem with en-route synchronization. Our problem diverges from previous studies by focusing on determining real-time take-off and landing positions with the requests arriving randomly. This situation mirrors the real-life unpredictability of request arrival. We integrate real-time considerations for takeoff and landing procedures to minimize waiting time and operational costs, thus enhancing overall profit and effectiveness. This nuanced consideration enables

the development of strategies that exhibit adaptability in dynamic environments. To the best of our knowledge, there is no existing research that studied the truck-drone collaborative delivery problem with flexible launch and retrieval in dynamically arriving requests.

3. Problem description

We consider a logistics operator with control over a set of homogeneous trucks $U = \{1, \dots, n\}$ and a set of homogeneous drones $D = \{1, \dots, n\}$ to fulfill requests that arrive randomly within a specified area during the planning horizon $[0, \mathcal{T}]$. The logistics operator also manages a depot located in the area, which is utilized for parking idle trucks and drones and also for storing multiple items. To facilitate modeling, we divide the planning horizon into T successive time intervals. We set $T = \{0, 1, \dots, \mathcal{T}\}$ as the set of feasible decision points. This ensures that trucks and drones can only be dispatched at the start of each time interval. Furthermore, we set up a network $G(N, A)$ to describe a certain area, N which is the node set that represents customer locations and the depot, and A is the arc set that represents routes connecting the locations of N .

Each request $c \in C$ originates from the depot and is associated with a delivery destination $d(c)$, a time window $[e_c, l_c]$ and a weight w_c . If a truck/drone arrives earlier than e_c , it is required to wait at the customer node, while arriving later than l_c is considered to add a delayed penalty. A request $c \in C$ can be served by a drone $d \in D$ if the weight w_c is below the drone capacity; otherwise, it is required to be served by a truck $u \in U$.

Each truck $u \in U$ is required to carry a drone $d \in D$ and the corresponding items for delivery from the depot. Both trucks and drones travel at constant speeds, denoted by v^U and v^D , respectively. Either a truck $u \in U$ or a drone $d \in D$ sequentially delivers the items to each customer location following a dynamic route. Please note that if the first request yields a higher profit through drone delivery, the drone $d \in D$ can also take off directly from the truck $u \in U$ at the depot. After completing the delivery, truck $u \in U$ reloads the respective drone $d \in D$ and returns to the depot. During the delivery process, each drone $d \in D$ is assumed to serve one customer request at once. Our key emphasis lies in achieving synchronized arrivals of both trucks and drones at en-route take-off and landing positions. The takeoff and landing time of drones is relatively short, and its influence on the formulation is negligible, requiring only the addition of constants in the constraints. For simplicity, we omit the takeoff and landing time.

Given the particularity of our research problem, the decision-making process in DCRP-eS involves formulating optimized routes for both trucks and drones and ensuring that these routes are dynamically constructed to accommodate randomly arriving requests. Additionally, the process entails identifying suitable take-off and landing positions to facilitate the en-route synchronized arrival of trucks and drones. The overarching objective is to maximize the anticipated aggregate profit accrued over the decision horizon.

Fig. 1 provides an example of a solution with three requests served by one truck and one drone with each node indicating a random request. At time t_1 , the paired truck and drone receive request 1, and the truck is dispatched to service. After serving request 1, the paired truck and drone receive request 2 at time t_2 , and truck delivery is selected again. At time t_3 , the paired truck and drone receive request 3 during its journey to request 2, and the drone is launched from the truck to serve request 3. At time t_4 , the drone completes request 3, and determines its destination to join the truck using the en-route synchronization approach by tracing the optimal retrieval position along the ongoing route of the truck. At time t_5 , the truck and the drone arrive simultaneously at the designated retrieval position. At time t_6 , after finishing the servicing at node 2, the truck returns to the depot with the drone.

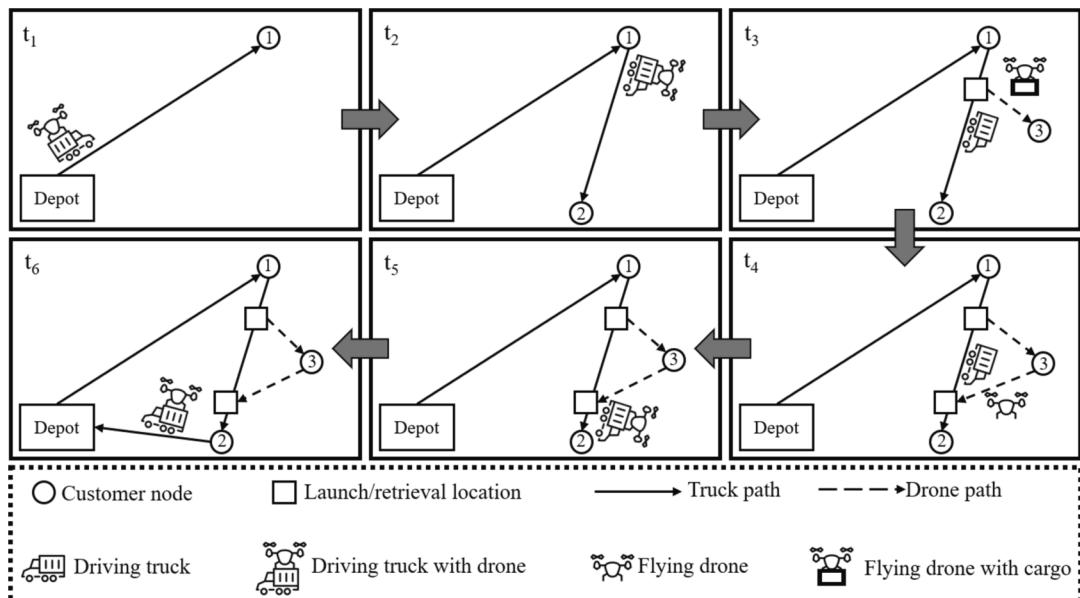


Fig. 1. An illustrative example of a solution for DCRP-eS.

4. Markov decision process model

We formulate the DCRP-eS as an MDP model to address the uncertainties that arise from the random arrival of requests. The constituents of this MDP are elucidated as follows.

4.1. States

The planning horizon encompasses a collection of decision epochs $\varepsilon = \{0, 1, \dots, E\}$. Each decision epoch $e \in \varepsilon$ aligns with a specific time point.

At each epoch, the system state is described by equation (1). $t \in T$ indicates the present time and C is the set of customer requests that arrived but remain unassigned to either a truck or a drone. Each customer request $c \in C$ is associated with a destination $d(c)$, a weight w_c , and a time window $[e_c, l_c]$ for service.

$$s = (t, C, s^U, s^D). \quad (1)$$

The vector $S^U = (s_i^U)_{i \in U}$ defines the states of trucks. For each truck i , $s_i^U = (m_i^{(1)}, m_i^{(2)}, \dots, m_i^{(6)})$. Each element within this tuple is elucidated in Table 2.

The vector $S^D = (s_i^D)_{i \in D}$ defines the states of drones. For each drone i , $s_i^D = (n_i^{(1)}, n_i^{(2)}, n_i^{(3)}, n_i^{(4)}, n_i^{(5)})$. Each element of this tuple is explicated in Table 3.

4.2. Actions

4.2.1. Action space

In state s , each pair of trucks and drones (referred to as paired agents) is assigned a collection of actions represented by $A(s)$. Action $a \in A(s)$ is defined as a vector of six binary elements, namely, $a = (a_1, a_2, \dots, a_6)$. If $a_1 = 1$, the paired agent remains its present states (either proceeds with on-going request, or waits for further assignments) with no additional status assigned, i.e., idle. $a_2 = 1$ implies the arriving request is assigned to the drone, and the truck is set to idle. $a_3 = 1$ indicates that the incoming request is assigned to the truck, and the drone remains idle. $a_4 = 1$ describes that the drone lands on the truck, and the truck is set to idle. $a_5 = 1$ represents that the truck returns to the depot, and the drone is idle. Namely, a paired agent assigned with a_5 the truck can directly return to depot with the drone already retrieved or collect the drone via another action a_4 when travelling back to depot. $a_6 = 1$ represents that the truck dwells at the customer node waiting for other requests, and the drone is idle.

For each paired agent i , if $m_i^{(1)} \neq 0$, then $a_3 = 0$; $n_i^{(1)} \neq 0$, then $a_2 = 0$. That is, a truck i or drone i cannot be assigned any status until its present status has been completed. If $w_c > W_d$, then $a_2 = 0$. That is, drones cannot be assigned overweight requests.

4.2.2. Action constraints related to endurance

Due to the limited endurance of the drone, we need to ensure that only requests in the nearby vicinity or within a feasible range are accommodated. Therefore, the truck should not serve requests that are too far away when the drone is in operation, to ensure that the drone has sufficient energy to land on the truck. Specifically, based on the actions defined in Section 4.2.1, we impose the following two constraints on infeasible actions for the truck and drone:

- (i) If $s_dis + maxr_dis > e^D$ or $2f_dis > e^D$, then $a_2 = 0$. s_dis represents the flight distance of the drone serving the request; $maxr_dis$ denotes the maximum distance that the drone needs to cover to land on the truck after completing the service, that is, the distance from the service node to the current destination of the truck; and e^D defines the endurance of the drone. That means a

Table 2

Notations and definitions in truck states.

Notation	Definition
$m_i^{(1)}$	The truck type of present status, $m_i^{(1)} \in \{0, 1, 2\}$. 0, 1, and 2 respectively correspond to idle, return, and service.
$m_i^{(2)}$	The status information of the truck, $m_i^{(2)} \in \{0\} \cup N_0 \cup C$. If $m_i^{(1)} = 0$ (idle), then $m_i^{(2)} = 0$; if $m_i^{(1)} = 1$ (return), then $m_i^{(2)} \in N_0$ is the depot; if $m_i^{(1)} = 2$ (serve), then $m_i^{(2)} \in C$ is the request to be served in this status.
$m_i^{(3)}$	The final location of the present status, $m_i^{(3)} \in N$. If $m_i^{(1)} = 0$ (idle), then $m_i^{(3)}$ is the present location of truck i . If $m_i^{(1)} = 1$ (return), then $m_i^{(3)}$ is the location of the depot. If $m_i^{(1)} = 2$ (serve), then $m_i^{(3)}$ is the destination of the request served by truck.
$m_i^{(4)}$	The completion time of the present status, $m_i^{(4)} \in T$. If $m_i^{(1)} = 0$ (idle), then $m_i^{(4)}$ equals present time t . If $m_i^{(1)} = 1$ (return), then $m_i^{(4)}$ equals the time to arrive at the depot. If $m_i^{(1)} = 2$ (serve), then $m_i^{(4)}$ equal the time that the truck will arrive at the location.
$m_i^{(5)}$	The final location of previous status, $m_i^{(5)} \in N$.
$m_i^{(6)}$	The time of previous status, $m_i^{(6)} \in T$. $m_i^{(5)}$ and $m_i^{(6)}$ are employed to deduce the positional information of truck when the drone completes its assigned request, together with $m_i^{(3)}$ and $m_i^{(4)}$.

“idle” refers to the state where the truck is stationary, and available to accept new requests.

Table 3

Notations and definitions in drone states.

Notation	Definition
$n_i^{(1)}$	The drone type of present status, $n_i^{(1)} \in \{0, 1, 2\}$. 0, 1, and 2 respectively correspond to idle, retrieve, and service.
$n_i^{(2)}$	The status information of the drone, $n_i^{(2)} \in \{0\} \cup L \cup C$. If $n_i^{(1)} = 0$ (idle), then $n_i^{(2)} = 0$; if $n_i^{(1)} = 1$ (retrieve), then $n_i^{(2)} \in L$ is the mobile retrieval locations where the drone i should be retrieved to the truck i ; if $n_i^{(1)} = 2$ (serve), then $n_i^{(2)} \in C$ is the request to be served in this status.
$n_i^{(3)}$	The final location of the present status, $n_i^{(3)} \in N$. If $n_i^{(1)} = 0$ (idle), then $n_i^{(3)}$ is the present location of drone i . If $n_i^{(1)} = 1$ (retrieval), then $n_i^{(3)}$ is the retrieval location. If $n_i^{(1)} = 2$ (serve), then $n_i^{(3)}$ is the destination of the request served by drone.
$n_i^{(4)}$	The completion time of the present status, $n_i^{(4)} \in T$. If $n_i^{(1)} = 0$ (idle), then $n_i^{(4)}$ equals the present time t . If $n_i^{(1)} = 1$ (retrieve), then $n_i^{(4)}$ equals the time to arrive at the location. If $n_i^{(1)} = 2$ (serve), then $n_i^{(4)}$ equals the time that the truck will arrive at the location.
$n_i^{(5)}$	The distance that the drone has flown of the present status. $n_i^{(5)}$ is employed to limit the battery endurance (abbreviated as endurance for simplicity) of drones.

"idle" refers to the state where the drone is stationary, and available to accept new requests.

drone i cannot be assigned the request where the sum of the fly distance to the customer and the maximum retrieval distance exceeds the endurance of the drone, or where the take-off distance surpasses half of the endurance of the drone.

- (ii) If $n_i^{(1)} \neq 0$, and $n_i^{(5)} + \text{maxr_dis}' > e^D$, then $a_3 = 0$. $\text{maxr_dis}'$ denotes the distance from the drone service node to the new request node. That is, if the drone in this paired agent i is in operation, then the truck can only serve new customer requests within the remaining range of the drone to ensure a retrieval position for the drone. The $2f_dis > e^D$ of previous condition is to ensure an adequate service range for the truck, thus avoiding excessive undermining of its performance.

4.3. Transition function

The system will transition from state s to the subsequent state s' when the logistics operator a acts. This transition is determined by the present state s , the chosen action a , and the occurrences in the upcoming epoch. The occurrences take place at an epoch include the arrival of new requests, status completions of trucks and drones, or the absence of any event.

To fabricate the subsequent system state s' , the time of the next epoch t' is recorded. Additionally, the request set C and the vectors s^U and s^D are updated based on the chosen action a and the observed environment. Further details regarding the transition from state s to subsequent state s' can be found in [Appendix A](#).

4.4. Reward function

After taking an action from any state s ($a \in A(s)$), the logistics operator attains a reward $R(s, a)$. This reward is calculated by subtracting the travel cost incurred by the truck or drone from the revenue gained by serving requests. Specifically, if an action a assigns request c to paired agent i at state s , the operator collects revenue R_c . The travel cost incurred by serving request c is denoted as $C_u(L_{u(l),d(c)}/v_u)$ or $C_d(L_{d(l),d(c)}/v_d)$, Where $d(l)$ is the present location of the drone, $u(l) = m_i^{(4)}$ is the present location of the truck at state s .

In addition, for each request c in the set C , a binary indicator $w_c(s, a)$ takes the value of 1 if action a assigns request c to a paired agent at state s , and 0 otherwise. Let $\eta(s, a)$ represents the cumulative cost of trucks and drones to complete the assigned status and the delay penalty of the customer. Equation (2) formulates the reward function.

$$R(s, a) = \sum_{c \in C} R_c w_c(s, a) - \eta(s, a) \quad (2)$$

4.5. Objective function

Consider π as a policy that consists of a series of decision guidelines $(X_0^\pi, X_1^\pi, \dots, X_E^\pi)$, where X_E^π represents a function that associates a given state s at the epoch e to an action $a \in A(s)$. The set of all policies is denoted by Π , and the system state at epoch e is represented by s_e .

The objective function, denoted as $F(\pi^*)$, seeks an optimal policy π^* that maximizes the expected total reward from the initial state s_0 throughout the planning horizon. Here, π^* represents the optimal policy, essentially the sequence of best actions to take under various states. More specifically, the mathematical expression of the objective function is:

$$F(\pi^*) = \max_{\pi \in \Pi} E[\sum_{e \in \xi} R(s_e, X_e^\pi(s_e)) | s_0] \quad (3)$$

5. Mixed integer programming model

MIP and MDP have various capabilities and may be used for distinct decision-making problems. MIP effectively optimizes resource allocation by statically streamlining under the complete information of the requests. It aims to identify the most efficient integer

solutions within certain constraints, efficiently handling the intricacies involved in scheduling scenarios. While MDP operates resource scheduling through real-time dynamic tracking. It determines the likelihood of prospective states for each current state and action, along with their respective rewards. MDP enables adaptive decision-making based on real-time observations, facilitating the optimization of dynamic processes.

We propose an MIP model for addressing the static CRP-eS problem. The MIP model entails strategically setting discrete points along the routes to ensure the approximate concurrent arrival of both trucks and drones. This model is introduced for comparative analysis with DCRP-eS, where the dynamics of request arrivals require real-time route adjustments. Table 4 shows the notations and definitions of the CRP-eS model.

The MIP model for the problem can be formulated as follows:

$$\text{Max} \sum_{i \in C} z_i r_i - f_1 - f_2 - f_3 \quad (4)$$

$$f_1 = \sum_{u \in U} \sum_{i \in N_L} \sum_{j \in N_R} \alpha^U d_{ij}^U x_{iju} \quad (5)$$

$$f_2 = \sum_{d \in D} \sum_{i \in N_L} \sum_{j \in N_R} \sum_{k \in C} \sum_{h \in H_j} \alpha^D d_{jkh} (\delta_{-ijkhd} + \delta_{+ijkhd}) \quad (6)$$

$$f_3 = \sum_{i \in C} \alpha^C m_i \quad (7)$$

Table 4

Notations and definitions in MIP.

Notation	Definition
Sets	
N	Set of customer nodes and depot nodes
N_L	Set of customer nodes and begin depot node
N_R	Set of customer nodes and return depot node
C	Set of customer nodes
H	Set of all discrete points, where $H = \{H_{ij} \forall i, j \in N\}$, and H_{ij} denotes the set of discrete points along the route (i, j)
U	Set of homogeneous trucks
D	Set of homogeneous drones
Parameters	
t_{ij}^U	Travel time from node i to node j by truck
t_{ij}^D	Travel time from node i to node j by drone
d_{ij}	Distance from node i to node j
d_{jkh}	Distance from discrete point h on route (i, j) to node k
v^U	Truck velocity
v^D	Drone velocity
s_c	Truck service time at customer node c
e^D	Battery endurance of drone
α^U	The transportation cost per unit of time for the truck
α^D	The transportation cost per unit of time for the drone
α^C	The delay penalty per unit of time for the customer
q^D	The maximum loading weight of the drone
Q_c	The weight of the requests required by the customer node c
$[e_c, l_c]$	Lower/upper bound of time window of customer node c
r_c	The profit obtained from completing the request in customer node c
M	A large number
Decision variables	
x_{iju}	1 if arc (i, j) is traversed by truck u ; 0 otherwise
y_{ikjd}	1 if drone d takes off around the node i , serves customer node k , and lands around node j ; 0 otherwise
z_i	1 if customer node i is served
p_{iju}	1 if node i is visited before node j by truck u ; 0 otherwise
δ_{-ijkhd}	1 if drone d takes off at discrete point h on route (i, j) to serve node k ; 0 otherwise
δ_{+ijkhd}	1 if drone d lands at a discrete point h on route (i, j) to serve node k ; 0 otherwise
o_{iu}	The order of node i in the route of truck u
m_i	The delay time at customer node i
τ_{+iu}	The time of truck u arriving at node i
τ_{+id}	The time of drone d arriving at node i
τ_{-iu}	The time of truck u leaving from node i
ψ_{-ikju}	The time of truck u launches drone on route (i, k) to node j
ψ_{+ikju}	The time of truck u retrieves drone on route (i, k) from the node j

$$\text{s.t. } \sum_{u \in U} \sum_{i \in N_L} x_{iju} + \sum_{d \in D} \sum_{i \in C} \sum_{k \in C} y_{ijkl} \leq 1 \quad \forall j \in C \quad (8)$$

$$\sum_{i \in N_L} x_{iju} = \sum_{k \in N_R} x_{jku} \leq 1 \quad \forall j \in \{C : j \neq i, j \neq k\}, \forall u \in U \quad (9)$$

$$z_j \geq \sum_{u \in U} \sum_{i \in N_L} x_{iju} \quad j \in C \quad (10)$$

$$z_j \geq \sum_{d \in D} \sum_{i \in C} \sum_{k \in C} y_{ijkl} \quad j \in C \quad (11)$$

$$z_j \leq \sum_{u \in U} \sum_{i \in N_L} x_{iju} + \sum_{d \in D} \sum_{i \in C} \sum_{k \in C} y_{ijkl} \quad (12)$$

$$m_j \geq \sum_{i \in N_L} x_{iju} (\tau_{+ju} - l_j) \quad j \in C, u \in U \quad (13)$$

$$m_j \geq \sum_{i \in C} \sum_{k \in C} y_{ijkl} (\tau_{+jd} - l_j) \quad j \in C, d \in D \quad (14)$$

$$o_{iu} - o_{ju} \geq 1 - (c + 2)p_{iju} \quad \forall i, j \in \{C : j \neq i\}, \forall u \in U \quad (15)$$

$$o_{iu} - o_{ju} \leq -1 + (c + 2)(1 - p_{iju}) \quad \forall i, j \in \{C : j \neq i\}, \forall u \in U \quad (16)$$

$$p_{iju} + p_{jiu} = 1 \quad \forall i, j \in \{C : j \neq i\}, \forall u \in U \quad (17)$$

$$o_{iu} - o_{ju} + 1 \leq (c + 2)(1 - x_{iju}) \quad \forall i \in N_L, \forall j \in \{N_R : j \neq i\}, \forall u \in U \quad (18)$$

$$\sum_{j \in C} x_{0ju} = \sum_{i \in C} x_{i0u} \quad \forall u \in U \quad (19)$$

$$\tau_{+ju} \geq \tau_{-iu} + t_{ij}^U - M(1 - x_{iju}) \quad \forall i \in N_L, \forall j \in \{N_R : i \neq j\}, \forall u \in U \quad (20)$$

$$\tau_{-ju} \geq \tau_{+ju} + s_j - M(1 - \sum_{i \in N_L} x_{iju}) \quad \forall j \in \{C : i \neq j\}, \forall u \in U \quad (21)$$

$$\psi_{-kbju} \geq \psi_{+kbmu} - M(5 - y_{ijkl} - y_{bmnd} - p_{ibu} - \sum_{h \in H_{kb}} \delta_{-kbmh} - \sum_{h \in H_{kb}} \delta_{+kbjh}) \quad \forall i, j, k, b, m, n \in C, \\ \forall u \in U, \forall d \in \{D : d = u\} \quad (22)$$

$$\sum_{a \in C} \sum_{i \in C} \sum_{d \in D} \sum_{h \in H_{ai}} \delta_{-ajhd} \leq \sum_{a \in C} \sum_{i \in C} \sum_{d \in D} y_{ajid} \quad \forall j \in C \quad (23)$$

$$\sum_{a \in C} \sum_{i \in C} \sum_{d \in D} \sum_{h \in H_{ai}} \delta_{+ajhd} \leq \sum_{a \in C} \sum_{i \in C} \sum_{d \in D} y_{ajid} \quad \forall j \in C \quad (24)$$

$$\sum_{i \in C} \sum_{k \in C} y_{ijkl} \leq 1 \quad \forall j = \{C : i \neq j, k \neq j\}, \forall d \in D \quad (25)$$

$$\sum_{i \in C} \sum_{k \in C} y_{ijkl} Q_j \leq q^D \quad \forall j \in C, \forall d \in D \quad (26)$$

$$\sum_{h \in H_{ai}} \delta_{-ajhu} d_{ajih} + \sum_{h \in H_{ik}} \delta_{-ikjh} d_{ikjh} + \sum_{h \in H_{ik}} \delta_{+ikjh} d_{ikjh} + \sum_{h \in H_{kb}} \delta_{+kbjh} d_{kbjh} \leq e^D + M(3 - y_{ijkl} - x_{iku} - x_{kbu}) \quad \forall a \in N_L, \forall b \in N_R, \forall i, j, k \in C, \forall u \\ \in U, \forall d \in \{D : d = u\} \quad (27)$$

$$\tau_{+jd} \geq \psi_{-ajd} + d_{ajh} \delta_{-ajhd} / \psi^D - M(1 - \delta_{-ajhd}) \quad \forall a \in N_L, \forall b \in N_R, \forall i, j, k \in C \\ \forall h \in H_{ai}, \forall u \in U, \forall d \in \{D : d = u\} \quad (28)$$

$$\psi_{+kbju} \geq \tau_{+jd} + d_{kbjh} \delta_{+kbjh} / \psi^D - M(1 - \delta_{+kbjh}) \quad \forall a \in N_L, \forall b \in N_R, \forall i, j, k \in C \\ \forall h \in H_{kb}, \forall u \in U, \forall d \in \{D : d = u\} \quad (29)$$

$$\sum_{h \in H_{ai}} \delta_{-ajhu} + \sum_{h \in H_{ik}} \delta_{-ikhu} \leq 1 + M(3 - y_{ijkl} - x_{aiu} - x_{iku}) \forall a \in N_L, \forall i, j, k \in C, \\ \forall u \in U, \forall d \in \{D : d = u\} \quad (30)$$

$$\sum_{h \in H_{ai}} \delta_{-ajhu} + \sum_{h \in H_{ik}} \delta_{-ikhu} \geq 1 - M(3 - y_{ijkl} - x_{aiu} - x_{iku}) \forall a \in N_L, \forall i, j, k \in C, \\ \forall u \in U, \forall d \in \{D : d = u\} \quad (31)$$

$$\sum_{h \in H_{ik}} \delta_{+ikjh} + \sum_{h \in H_{kb}} \delta_{+kbjh} \leq 1 + M(3 - y_{ijkl} - x_{iku} - x_{kbu}) \forall i, j, k \in C, \forall b \in N_R, \\ \forall u \in U, \forall d \in \{D : d = u\} \quad (32)$$

$$\sum_{h \in H_{ik}} \delta_{+ikjh} + \sum_{h \in H_{kb}} \delta_{+kbjh} \geq 1 - M(3 - y_{ijkl} - x_{iku} - x_{kbu}) \forall i, j, k \in C, \forall b \in N_R, \\ \forall u \in U, \forall d \in \{D : d = u\} \quad (33)$$

$$3y_{ijkl} \leq \sum_{a \in N_L} x_{aiu} + x_{iku} + \sum_{b \in N_R} x_{kbu} \forall i, j, k \in C, \forall u \in U, \forall d \in \{D : d = u\} \quad (34)$$

$$\tau_{+iu} - d_{aih} \delta_{-ajhu} / v^U \leq \psi_{-ajhu} + M(1 - \delta_{-ajhu}) \forall a \in N_L, \forall i, j \in C, \forall h \in H_{ai}, \\ \forall u \in U, \forall d \in \{D : d = u\} \quad (35)$$

$$\tau_{+ku} - d_{ikh} \delta_{+ikjh} / v^U \leq \psi_{+ikjh} + M(1 - \delta_{+ikjh}) \forall i, j \in C, \forall k \in N_R, \forall h \in H_{ik}, \\ \forall u \in U, \forall d \in \{D : d = u\} \quad (36)$$

$$x_{iju}, y_{ijkl}, z_i, p_{iju}, \delta_{+ikjh}, \delta_{-ikjh} \in \{0, 1\} \forall u \in U, \forall d \in D, \forall i, j, k \in N \quad (37)$$

$$0 \leq o_{iu} \leq c + 1 \forall i \in N \quad (38)$$

The objective function (4) maximizes the profit gained, which is calculated as the total revenue subtracted from the total cost. The cost consists of the travel costs of trucks and drones, as formulated in equations (5) and (6), respectively. Equation (7) illustrates the delay penalty incurred when failing to reach the customer node within the designated time window.

Constraints (8)~(14) pertain to the delivery interactions between trucks, drones, and customers. Constraints (8) imply that each customer node receives service from either a drone or a truck at most once. Constraints (9) guarantee the flow conservation, i.e., if a truck/drone enters a customer node, it must leave the same node. Constraints (10)~(12) restrict whether a customer request is serviced. Constraints (13) and (14) are utilized for computing the delay time of service at the customer node.

Constraints (15)~(21) are related to truck transportation. Constraints (15)~(17) are applied to preserve the sequence of truck deliveries at the customer nodes. Constraints (18) are introduced to eliminate sub-routes within the route of the trucks. Constraints (19) ensure that all utilized trucks must return to the depot. Constraints (20) ensure the continuity of the timeline. Constraints (21) restrict the trucks staying at the customer node until deliveries are completed. It is worth noting that, compared to the traditional VRPs that only have o_{iu} to describe truck routes, the CRP introduces p_{iju} to capture the order of non-adjacent nodes in the route of trucks, considering the coordination between the trucks and drones.

Constraints (22)~(33) define the constraints related to drone transportation. Constraints (22) ensure that a drone can only be

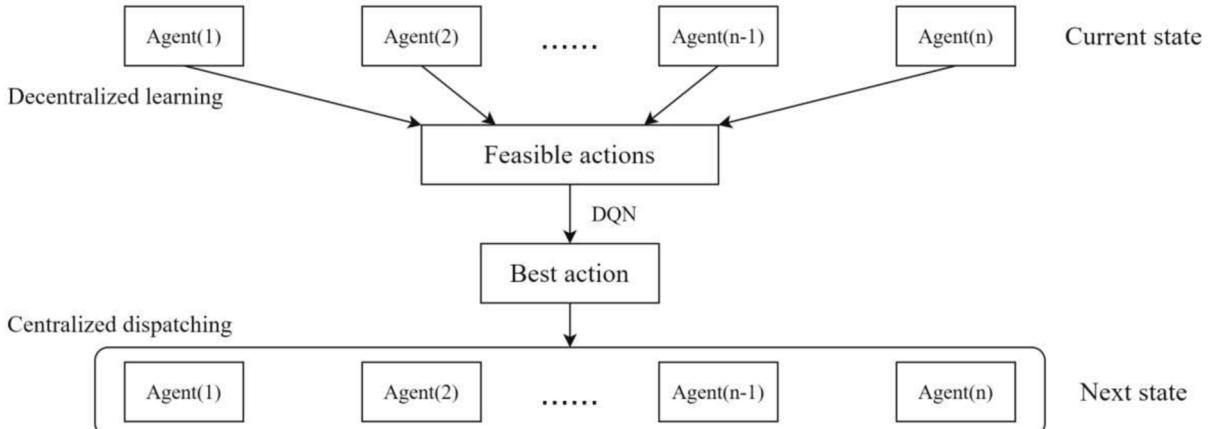


Fig. 2. Framework of the proposed RL method.

assigned to a new request after its retrieval. Constraints (23) and (24) restrict the existence of mobile take-off and landing positions when the drones are serving. Constraints (25) limit the drones to serving only one request during each flight. Constraints (26) impose a capacity limit on drones. Constraints (27) impose an endurance limit on the drones. Constraints (28) and (29) maintain the continuity of the timeline of drones. Constraints (28) ensure that the drone flies from launch position h to node j before reaching node j , and Constraints (29) ensure that a drone flies from node j to the retrieval position h before reaching the retrieval position h . Constraints (30)~(33) ensure that the take-off and landing positions on the routes are unique and correspond to each serve action.

Constraints (34)~(36) ensure the spatial and temporal coordination between the trucks and the drones. Constraints (34) restrict the drones taking off and landing on the same truck. Constraints (35) and (36) ensure that the trucks reach the mobile take-off and landing positions when launching and retrieving the drones. Constraints (37) and (38) specify the variable domains.

6. Reinforcement learning

We propose an RL method to obtain an optimal policy of the MDP model. The proposed RL algorithm applies a multi-agent learning framework adapted from Jia et al. (2022), where each paired truck and drone is treated as an individual agent that learns the value of taking actions in specific states through interactions with the environment. Compared to Equation (1), the single MDP model is a brief summary that concentrates on the states and actions of each truck-drone pair. The single MDP model is used as the foundation for the decentralized offline training of the agents. We provide a centralized dispatching technique that allocates action to agents at each epoch, attempting to maximize the total estimated state-action value while avoiding agent disputes, to ensure coordination among the agents. Fig. 2 illustrates the framework of the RL method that decentralized learning and centralized dispatching. The pseudo-code and hyperparameters of the proposed RL algorithm are provided in Appendix B.

6.1. Flexible launch and retrieval

To achieve real-time coordination between trucks and drones, and inspired by the synchronous time dependence proposed by Dohn et al. (2011), we develop a planar geometric-based physical model for Flexible Launch and Retrieval (FLR). The FLR model tracks the geospatial positions of the trucks and drones during operation and utilizes these positions to generate optimal take-off and landing positions based on solving encounter problems in basic physics. Namely, the position where the truck and drone will simultaneously arrive is considered as the optimal retrieval position. The FLR model ensures the synchronization between truck and drone for retrieval and prevents deadhead waiting.

Fig. 3 demonstrates the flexible retrieval of a drone. In Fig. 3, the drone is at position 1 with coordinates (x_1, y_1) . The truck is at position 2 with coordinates (x_2, y_2) , which can be obtained through present state (destination $m_i^{(3)}$ and completion time $m_i^{(4)}$), and previous state (origin $m_i^{(5)}$ and completion time $m_i^{(6)}$). Additionally, the destination customer location $m_i^{(3)}$ of the truck is position 3 denoted as (x_3, y_3) . The speeds of the drone and the truck are denoted as v_d and v_u , respectively. Given these parameters, we aim to obtain the coordinates of the retrieval location (position 0), denoted as (x_0, y_0) , where the truck and the drone can arrive simultaneously along the future route of the truck.

The retrieval location should satisfy two conditions: (i) both the truck and the drone take the same amount of time Δt to reach the retrieval location; and (ii) the retrieval location lands on the future driving route of the truck, rather than a location that has already been visited. The above conditions are formulated as follows:

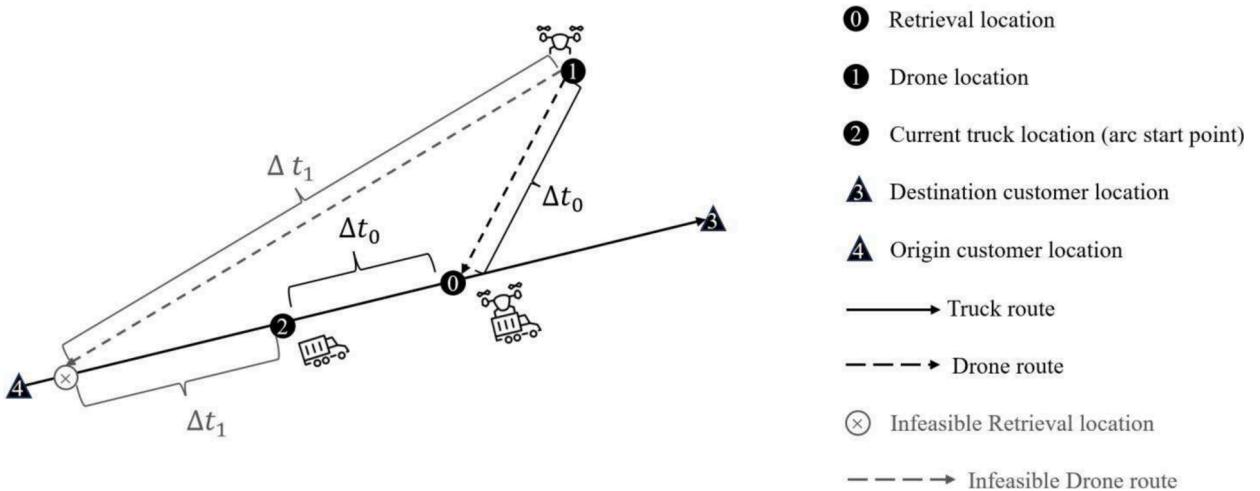


Fig. 3. An example of flexible drone retrieval.

$$\Delta t = \frac{\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}}{v_d} = \frac{\sqrt{(x_0 - x_2)^2 + (y_0 - y_2)^2}}{v_u} \quad (39)$$

$$\frac{y_2 - y_1}{x_0 - x_1} = \frac{x_2 - x_1}{x_0 - x_1} = \frac{m_i^{(4)} - t}{\Delta t} \quad (40)$$

The geometric method in equation (39) solves the retrieval locations. However, due to its quadratic form, the coordinates solved by this equation yield two solutions, Δt_0 and Δt_1 , as shown in Fig. 3. These two solutions correspond to two retrieval points: one at a location where the truck has already passed and another at a point where the truck will pass in the future. Hence, We incorporate time considerations into Equation (40), where $m_i^{(4)} - t$ represents the time needed for the truck to continue to the next customer node, and Δt represents the time it takes for the truck to reach the retrieval position. This ratio of the time coincides with the ratio of the x-y coordinates of the distance from the truck to the next customer node and the retrieval point, ensuring $\Delta t = \Delta t_0$ and the retrieval occurs only along the future route of the truck. Furthermore, if the computed retrieval point lies beyond the designated route, it can be inferred that the optimal position for drone retrieval would be at the customer node.

6.2. Decentralized learning

We adopt a decentralized learning approach to train the paired agents. A single model defines the environment for each agent, where each epoch initiates based on the arrival of a new request or by status completion of paired agents. Thus, the state of $i \in U$ agent in the single model is characterized by equation (41). Let C_t be the set of requests that arrive on time t .

$$s = (t, C_t, s_i^U, s_i^D). \quad (41)$$

If $C_t \neq \emptyset$, the state s is initiated by the arrivals of requests in C_t . Otherwise, the state s is initiated by the completion of a status by a paired agent i . In a given state s , each action a in the agent i action space $A_i(s)$ is characterized as $a = (a_1, a_2, \dots, a_6)$. If $a_1 = 1$, no additional status is assigned to the paired agent i . $a_2 = 1$ describes the additional request assigned to the drone i . $a_3 = 1$ depicts the additional request assigned to truck i . $a_4 = 1$ describes that drone i lands on truck i . $a_5 = 1$ describes that truck i returns to the depot. $a_6 = 1$ represents that truck i stops at the customer node and waits for other requests.

The proposed deep Q-learning approach utilizes a state-action value, denoted as $Q_i(s, a)$, to assess the effectiveness of the paired agent i performing action a from state s . This state-action value, also known as the Q-value, equals the sum of the immediate reward and the expected future rewards obtained by performing action a from state s . This recursive relationship can be expressed using Bellman's equation (42), where $R_i(s, a)$ denote the reward of the paired agent i performing action a from state s , s' represents the next state, and $\gamma \in [0, 1]$ is a discount factor.

$$Q_i(s, a) = R_i(s, a) + \gamma E[\max_{a' \in A_i(s')} Q_i(s', a')|s, a] \quad (42)$$

The optimal policy of the agent is to select the action corresponding to the highest Q-value at each epoch when the Q-values are available. However, due to the large number of state-action pairs, it is practically impossible to enumerate all Q values. We apply artificial neural networks to approximatively calculate the Q-values in order to escape the dimensionality curse. Deep Q-networks (DQNs) are the usual name for these neural networks.

The algorithm iteratively updates the parameters θ_i that define the DQN throughout the learning process. The usual gradient descent technique is utilized to minimize a loss function in this update. The squared difference between the estimated Q-values and the desired Q-values serves to calculate our loss function. We propose to increase the precision of DQN in estimating the true Q-values by minimizing the loss function.

The Q-value approximated by the DQN, denoted as $Q_i(s, a; \theta_i)$, is used in our approach, where θ_i represents the parameters of the DQN. As the true value $Q_i(s, a)$ is unavailable, we utilize an estimate of $Q_i(s, a)$ as the training target $\hat{Q}_i(s, a)$. The value of $\hat{Q}_i(s, a)$ is iteratively updated using the temporal-difference method (Sutton, 1988). This method allows us to update the estimated Q-values based on the difference between the current estimation and a future estimation, incorporating the reward received and the Q-value of the subsequent state.

$$\hat{Q}_i(s, a) = \begin{cases} R_i(s, a), & \text{if } s' \text{ is the terminal state;} \\ R_i(s, a) + \gamma \max_{a' \in A_i(s')} Q_i(s', a; \theta_i), & \text{otherwise} \end{cases} \quad (43)$$

The temporal-difference method is commonly employed in deep Q-learning as it offers a simple and efficient approach for updating the Q-values based on the subsequent state s' . This method is widely favored by researchers in the field.

In Section 6.3, we propose a centralized dispatching method for assuring the feasibility of actions taken by the truck-drone system at each time interval. However, achieving convergence in DQN implementation may demand numerous update iterations. A balance needs to be struck between computational budget and solution effectiveness. In our practical application, we perform a maximum of 2000 iterations if convergence has been achieved.

We integrate two proven extensions to enhance the standard Q-learning process: experience replay (Lin, 1992) and target DQN (Van Hasselt, 2016). Experience replay stabilizes the learning process by enhancing data utilization through the reuse of experiences across multiple updates and by eliminating correlations in previous experiences of agents. The goal of the target DQN is to stabilize the

convergence of the learning process by maintaining a fixed training target for multiple steps. Additionally, to ensure convergence of the learning process, it is crucial to thoroughly evaluate all actions over time. We apply the ϵ -greedy policy to find an equilibrium between exploration and exploitation in action selection.

6.3. Centralized dispatching

At each epoch, centralized dispatching is executed to synchronize the actions of paired trucks and drones, ensuring that each request is served by at most one vehicle (truck or drone). Maximizing the total estimated Q-value produced by the actions apportioned to each paired agent is the objective.

We allocate resources for the paired trucks and drones by solving an asymmetric assignment problem at each epoch. The objective is to apportion actions to paired trucks or drones G to maximize the total estimated Q-value. To formulate the assignment problem, we construct a bipartite graph with two sets of nodes: G representing the groups including truck and drone and P representing the possible actions for the groups. The nodes in P correspond to varied actions that can be taken by the groups. These actions include service actions, return actions, retrieval actions, and idle action. Each arc in the bipartite graph connects a node $i \in G$ to a node $p \in P$, representing the possibility of assigning the action $a(p)$ to paired agent i . The weight of each arc w_{ip} is given by the following equation, where the feasibility of an action is determined by the rules described in Section 4.2.

$$w_{ip} = \begin{cases} Q_i(s, a(p); \theta_i), & \text{if } p \in P \text{ and } a(p) \text{ is feasible for group } i; \\ +\infty, & \text{otherwise} \end{cases} \quad (44)$$

For each decision epoch $e \in \mathcal{E}$, let $P(e) = \cup_{i \in G} \{p(e, i)\}$. For each $i \in G$, if the current decision epoch is a new request arrival, let $P(i) = (a_1, a_2, a_3)$; else $P(i) = (a_1, a_4, a_5, a_6)$. The asymmetric assignment problem assigns actions to paired agents, with the objective of maximizing the total weight. It entails solving the following integer program:

$$\max \sum_{i \in G} \sum_{p \in P} w_{ip} x_{ip} \quad (45)$$

$$\text{s.t. } \sum_{p \in P} x_{ip} = 1, \quad i \in G \quad (46)$$

$$\sum_{i \in G} x_{ip} \leq 1, \quad p \in P(c); e \in \mathcal{E} \quad (47)$$

$$x_{ip} \in \{0, 1\}, \quad i \in G; p \in P \quad (48)$$

In the integer program presented above, if action $a(p)$ is assigned to paired agent i the value of decision variable x_{ip} is 1, otherwise, it is 0. Constraints (46) guarantee that each paired agent performs an action at present epoch. Constraints (47) guarantee that each request is allocated to at most one paired agent. Constraints (48) specify the variable domains.

7. Numerical experiments

To evaluate the effectiveness of the RL method for the DCRP-eS, we conduct experiments using problem instances generated from two dissimilar datasets, and compare with various benchmark methods.

The off-line learning and computational experiments were conducted on a computer with a 13th Gen Intel (R) Core (TM) i7-13700 processor, 32.0 GB RAM, and Nvidia GeForce RTX 4060Ti. Gurobi 10.0.1 was used as the standard solver to solve the linear and integer programming models involved in our solution method.

7.1. Experimental setup

Due to the high complexity of the MIP model, achieving results for extensive instances within a reasonable timeframe was unattainable. We design small-scale instances to compare the results of the MIP and RL methods. The large-scale instances are used to demonstrate the stability and efficiency of our MDP model and RL algorithm.

Small-scale instances: We randomly select 6 out of 50 requests from the Solomon dataset (Solomon, 1987), and two paired trucks and drones, and one depot are considered. The mean travel speed of a truck is 40 km/h, and the mean travel cost is \$0.79/km (Jia et al., 2022). We set the speed of the drone to 80 km/h and the cost to \$0.07/km based on the dynamic differences between drones and trucks (Carlsson et al., 2018). The endurance distance of the drone is 70 km (Jiang et al., 2024).

Large-scale instances: We select 25 locations from the dataset provided by Jia et al. (2022) as the destinations for our requests, with the first facility serving as the depot. The operator holds 10 trucks and 10 drones at its depot. On a daily basis, there are approximately 100 requests that arrive randomly. Among these requests, we suppose roughly two-thirds conform to the payload constraint of the drones, rendering them eligible for drone delivery, while the remaining requests should be served by truck delivery (Luo et al., 2022). Therefore, we generate the instances by setting the duration of each episode to 24 h and fixing the total number of requests to 100 in each episode. Within the given horizon, we randomly generate the receive time of each request and utilize this information to further generate the time windows. The weight of each request is determined from an even distribution between 5 and 20, with requests

weighing less than 15 kg eligible for drone delivery. The speed, cost, and endurance of truck and drone remain the same as the small-scale instances.

In addition to the single-day synchronized route instances described above, we extend the planning horizon to multiple days. We compare the route-based launch and retrieval (DCRP-eS) with the scenario involving pure truck (DVRP) and with the node-based launch and retrieval (DCRP).

7.2. Benchmark algorithms

The benchmark algorithms are described as follows:

A pure Mixed Integer Programming model (MIP): This method solves a mixed integer programming model defined with perfect request information. The integer programming model is shown in Section 5. In our model, the discrete points are generated from the road network, with a distance of 200 m between each point. While it requires approximately 20 s for a truck to reach adjacent discrete points, a drone accomplishes the same in about 10 s. We consider the truck and drone to have arrived synchronously at the same discrete point if the time difference between their arrivals is less than 10 s. The model is solved by Gurobi 10.0.1.

A Large Neighborhood Search Algorithm (LNS): As MIP is unable to procure a solution within an acceptable time for large-scale problems, we design LNS algorithm extensions to solve the full-information static problems (Shaw, 1998). Our algorithm involves several common types of destruction and repair operators. The destruction operators include random destruction operators and worst-case destruction operators, while the repair operators include greedy repair operators and regret repair operators.

A First-Come-First-Serve method (FCFS): This method is commonly used in practice for solving dynamic problems. It is a principle in which requests are catered to in the order that they arrive. It is characterized by fairness, and simplicity. From a queuing perspective, it operates on a strict sequential basis without priority considerations. If the agent and action are feasible, they may be assigned to the requests.

7.3. Benchmark scenarios

The benchmark scenarios are described as follows:

A node-based Dynamic Collaborative truck-drone Routing Problem (DCRP): This method shares the same settings as the DCRP-eS except that drones must take off and land from trucks at customer nodes or depot. Consequently, trucks may not arrive exactly on time when the drone reaches the retrieval positions, causing the drone to hover at the expense of the remaining battery. The waiting time induced by this asynchronism should be also considered as a part of the total cost for delivery operations. Note that if the truck arrives earlier than the drone at the retrieval position, the truck can simply turn off the engine without fuel costs. Therefore, we consider a waiting penalty on the early arrival of the drone at the retrieval position (Crisan et al., 2019; Cointreau et al., 2021). Let $\zeta_{ic}(s, a)$ denote the waiting time of drone i at customer node c resulted by taking action a at state s , and C_d denote the drone unit time

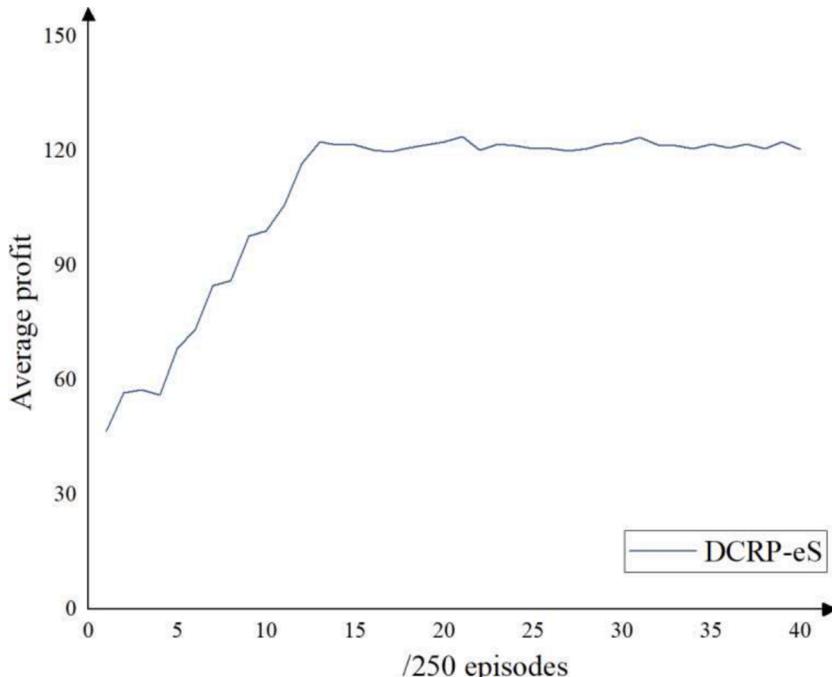


Fig. 4. RL training curves for small-scale instances (6 requests).

cost of waiting at the take-off and landing positions. The reward function for the DCRP for a paired agent i is defined as follows:

$$R_i(s, a) = \sum_{c \in C_i} (R_c w_{ic}(s, a) - C_d \zeta_{ic}(s, a)) - \eta(s, a) \quad (49)$$

Note that the DCRP-eS ensures the synchronization between trucks and drones when performing drone retrieval. Therefore, the DCRP-eS does not involve the waiting time penalty.

A Dynamic Truck and Drone Routing Problem (DTDRP): This method involves trucks and drones completing the random requests separately. Both truck and drone start and return to the depot independently without collaboration. Other settings are identical to DCRP-eS.

A Dynamic Vehicle Routing Problem (DVRP): This method only utilizes trucks to fulfill random requests and other settings are the same as DCRP-eS. Namely, the corresponding MDP and the RL solution approach merely involve trucks.

7.4. Results on small-scale instances

We evaluate and compare the solution quality and computational efficiency of the MIP, the LNS, the RL, and FCFS in a small-scale instance, aiming to assess their performance.

The RL model was subjected to off-line training over the course of 10,000 episodes in 2 h. An average profit curve, which stabilizes and converges around a value of approximately 120, can be observed in Fig. 4.

In the MIP model and LNS algorithm, all information is known in advance, while in RL and FCFS, requests arrive randomly. We can see in Table 5 that the proposed RL achieves a profit that is 18.56 % lower than the MIP model and 9.76 % lower than the LNS algorithm, despite the stochastic nature of request arrival in RL. This demonstrates that the profit obtained by our RL algorithm for dynamic problems is acceptable. While FCFS achieves a profit that is 37.78 % lower than the MIP model and 31.05 % lower than the LNS algorithm. Additionally, the LNS solution obtains a profit decrease of 9.75 % over the MIP solution, and the RL solution yields a reward increase of 30.89 % over the FCFS method. This demonstrates the superiority and efficiency of our algorithm.

We attempted to solve an instance encompassing seven requests, and discovered that it was beyond the operational memory capacity of computer. Therefore, we decided not to include the computational results from Gurobi in the larger instances, given the complexity of the model.

7.5. Results on large-scale instances

In large-scale instances, we compare the RL method with the MIP model, LNS algorithm, and FCFS heuristic algorithm. In addition, we compare varying operating scenarios performances including collaborative truck-drone route-based launch and retrieval (DCRP-eS), collaborative truck-drone node-based launch and retrieval (DCRP), separate truck and drone delivery (DTDRP), and pure truck delivery (DVRP).

7.5.1. Results for single-day instances

In off-line RL, the training is performed on the single-day instances set at 2000 episodes. Fig. 5 shows the average profit of the DCRP-eS, DCRP, DTDRP, and DVRP during training.

In Fig. 5, it is evident that all model algorithms converge. DCRP-eS achieves the highest average profit of 4400; while DCRP, DTDRP, and DVRP reach 4200, 4100, and 3200, respectively. For the training speed, due to the fact that each tested model has a similar structure and number of parameters, the required number of iterations for convergence is essentially equivalent.

Furthermore, to examine the impact of endurance, we compared the proposed solution with the benchmark algorithms regarding the performance under various endurance levels, as shown in Table 6. Subsequently, we compared the proposed DCRP-eS scenario with other benchmark scenarios regarding the performance under various endurance levels, as shown in Table 7.

Table 6 presents it is not possible to obtain a feasible solution within a reasonable timeframe using the MIP model with the Gurobi solver. Moreover, as an algorithm where all information is known in advance, the LNS does not significantly outperform the RL. The performance improvement of LNS over RL is 15.82 % on average. The LNS requires 20 min to compute the results, while the RL only needs 8 s, further proving the suitability of RL in practice. Furthermore, the RL algorithm outperforms the FCFS algorithm by an average profit margin of 11.90 %, which shows in the same dynamic problem, the proposed RL is superior.

We discover the following findings from Table 7 and Fig. 6: (i) DVRP only includes truck transportation and thus is unaffected by changes in drone endurance levels. In contrast, we can witness growth in profits in the other three transportation scenarios involving drones, which exhibit a trend where profits stabilize when the endurance reaches around 50 km, indicating that drone endurance no longer impacts profitability. This phenomenon can be attributed to the maximum distance in the Singapore road network being approximately 60 km; hence, a drone range of 50 km is sufficient to meet the delivery requirements for nearly all requests. (ii) For most of the endurance levels, we observe that the route-based launch and retrieval (DCRP-eS) consistently demonstrate the highest average

Table 5

Results obtained by RL, FCFS, LNS, and MIP in DCRP-eS in small-scale instances.

Requests	RL	FCFS	LNS	MIP
6	127.51	97.42	141.30	156.57

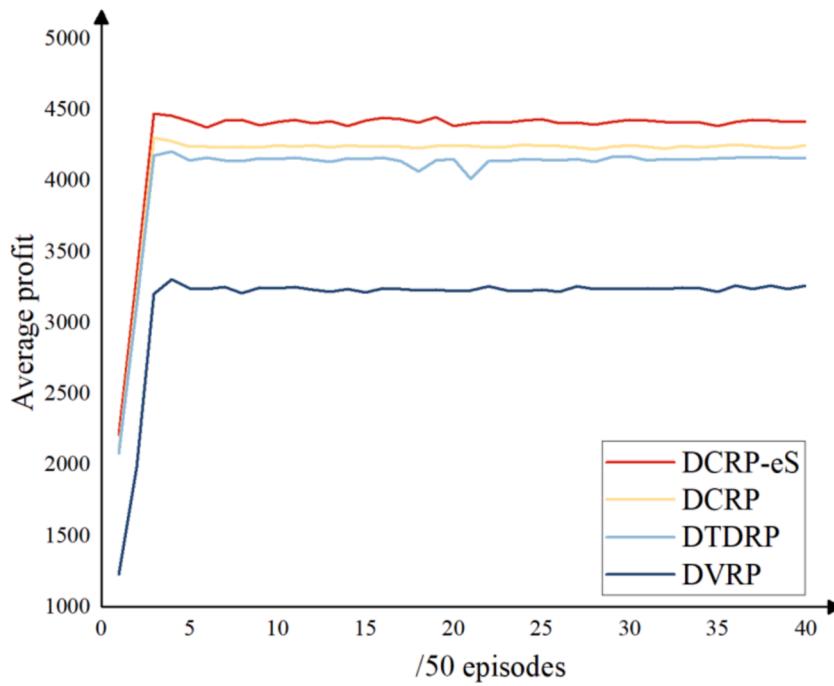


Fig. 5. Training curves of RL under various scenarios (100 requests).

Table 6
Results obtained by RL, FCFS, LNS, and MIP in DCRP-eS for different endurance.

Endurance	RL		FCFS		LNS		MIP
	RL	FCFS	RL	FCFS	RL	FCFS	—
10	4055.09		3814.65		4641.22		—
20	4278.76		3850.68		4863.20		—
30	4386.37		3899.46		5097.50		—
40	4439.24		3877.96		5226.46		—
50	4513.22		4028.15		5373.06		—
60	4512.55		4007.18		5488.30		—
70	4511.20		4011.80		5435.31		—
80	4511.20		4011.80		5443.97		—
90	4511.20		4011.80		5488.30		—
100	4511.20		4011.80		5485.21		—

“-” indicates the algorithm cannot obtain a solution in 2 h.

Table 7
Results obtained by RL and FCFS in various scenarios for different endurance.

Endurance	DCRP-eS		DCRP		DTDRP		DVRP	
	RL	FCFS	RL	FCFS	RL	FCFS	RL	FCFS
10	4055.09	3365.15	3814.65	3019.62	3728.89	2967.32	3642.09	2922.04
20	4278.76	3619.63	3850.68	3105.04	3719.74	3156.61	3642.09	2922.04
30	4386.37	3801.41	3899.46	3123.25	4065.27	3636.61	3642.09	2922.04
40	4439.24	3878.05	3877.96	3154.40	4220.02	3769.75	3642.09	2922.04
50	4513.22	3964.57	4028.15	3391.80	4355.06	3840.88	3642.09	2922.04
60	4512.55	3951.18	4007.18	3353.32	4355.06	3840.88	3642.09	2922.04
70	4511.20	3951.18	4011.80	3335.01	4355.06	3840.88	3642.09	2922.04
80	4511.20	3951.18	4011.80	3335.01	4355.06	3840.88	3642.09	2922.04
90	4511.20	3951.18	4011.80	3335.01	4355.06	3840.88	3642.09	2922.04
100	4511.20	3951.18	4011.80	3335.01	4355.06	3840.88	3642.09	2922.04

profit among all the tested scenarios, followed by the scenario of separate truck and drone delivery (DTDRP), node-based launch and retrieval (DCRP), and the absence of drones (DVRP). Moreover, compared to higher endurance levels, the DCRP-eS shows a more pronounced advantage than other scenarios for lower endurance levels (10 ~ 20 km). (iii) For the DTDRP, its profits have increased

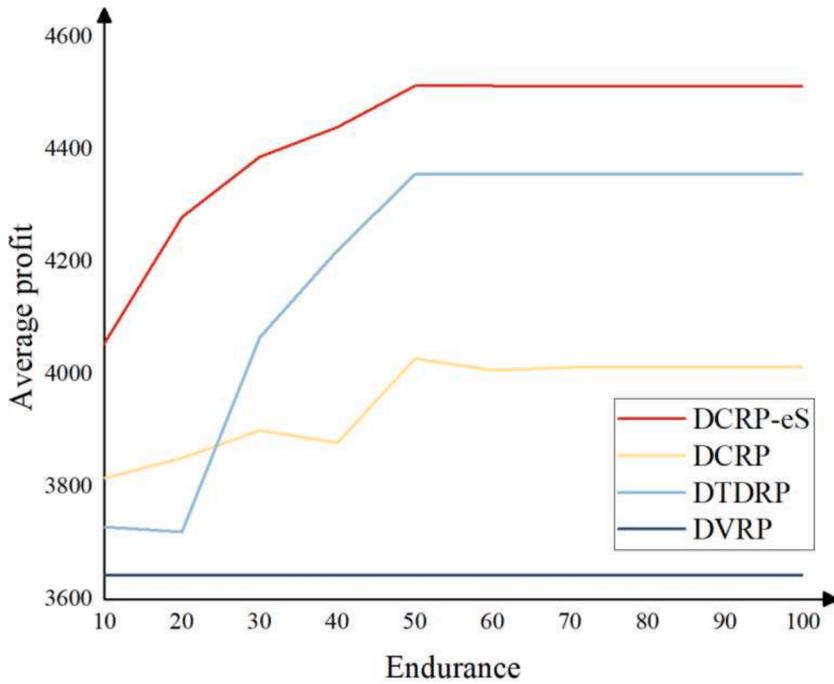


Fig. 6. Comparison of RL results in various scenarios under different endurance.

significantly from over 3700 to more than 4300, marking the largest growth observed among the four scenarios. This indicates that the endurance of drone has a significantly greater impact on DTDRP compared to other scenarios. This is because the DTDRP model does not involve coordination between trucks and drones, and the endurance of the drone directly determines the scope of customer service by drone. (iv) Under any endurance level or scenario, the proposed RL algorithm outperforms the FCFS algorithm, with both methods utilizing roughly the same amount of CPU time. Overall, the above findings show the superiority of the proposed DCRP-eS scenario and RL algorithm.

7.5.2. Results for multi-day instances

We extend the planning horizon to multiple days and solve the multi-day instances compare the solution with benchmark methods. For each of the single-day instances generated in Section 7.1, we extend the planning horizon to 2, 3, 4, 5, 6, and 7 days, and generate 100 requests each day. We choose an endurance of 10 km as our operational standard, in order to ensure battery constraints are effective. We solve the instances using RL, MIP, LNS, and FCFS in DCRP-eS and Table 8 reports the computational results. We compare various scenarios using RL and FCFS in Table 9. The maximum runtime for each instance is capped at 2 h.

The results of MIP are not shown in Table 8, due to the impractical computation time of the MIP model in the current instances. Moreover, when the number of requests reaches 500, the performance of the LNS algorithm is notably constrained. Within a 2-hour timeframe and limited iterations, it fails to find a solution that is superior to the initial one. As expected, the CPU time increases with the growth in instances. However, even in a seven-day instance, the maximum time required by RL is only 347 s, which indicates that our proposed method is effective. In Fig. 7 comparing the results of varying algorithms, we observe that the average rewards generated by RL are consistently larger by an average of 28.03 % than those generated by FCFS for assorted horizons. In the initial four days, the reward gained by RL is less compared to the LNS algorithm. However, after five days, RL surpasses LNS due to the constraints of the LNS algorithm. This pattern holds across multiple days, indicating the stability and effectiveness of our RL algorithm in large datasets.

Table 8

Results obtained by RL, FCFS, LNS, and MIP in DCRP-eS with assorted planning horizons (endurance = 10).

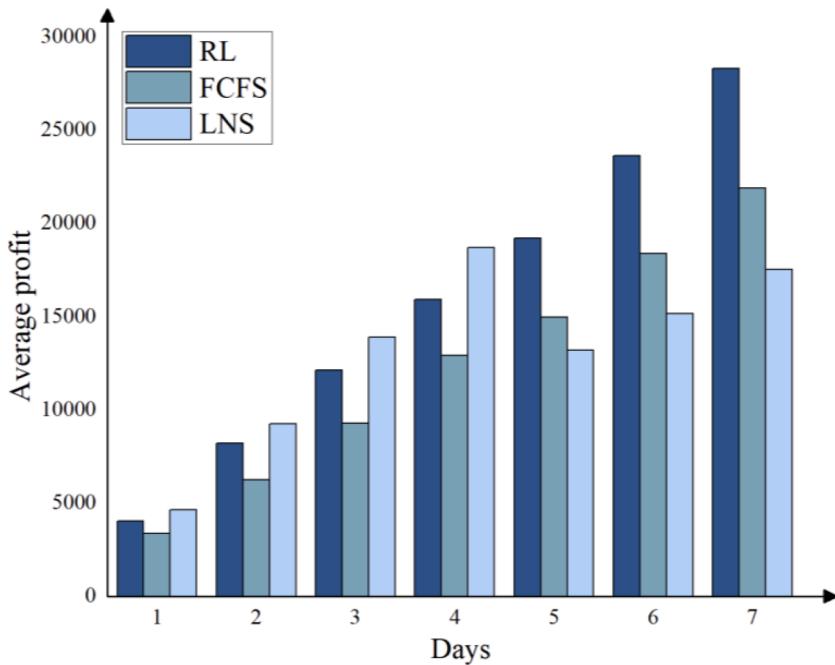
Day	RL	FCFS	LNS	MIP
1	4055.09	3365.15	4641.22	—
2	8191.48	6248.28	9253.34	—
3	12109.85	9274.32	13895.76	—
4	15919.44	12903.29	18703.35	—
5	19189.19	14962.74	13194.44	—
6	23635.55	18364.92	15152.33	—
7	28298.23	21889.11	17516.54	—

“—” indicates the algorithm cannot obtain a solution in 2 h.

Table 9

Profits obtained with assorted planning horizons in various scenarios (endurance = 10).

Day	DCRP-eS		DCRP		DTDRP		DVRP	
	RL	FCFS	RL	FCFS	RL	FCFS	RL	FCFS
1	4055.09	3365.15	3814.65	3019.62	3728.89	2967.32	3642.09	2922.04
2	8191.48	6248.28	7594.12	5999.13	7558.80	5991.64	7222.31	5713.84
3	12109.85	9274.32	11545.57	8944.01	10981.77	8903.49	10663.95	8756.07
4	15919.44	12903.29	15223.77	12252.84	14854.88	12394.26	14628.04	12069.90
5	19189.19	14962.74	18647.02	13717.14	17921.87	14568.59	17263.72	14129.12
6	23635.55	18364.92	22467.87	17765.17	21870.38	17672.07	21246.22	17354.03
7	28298.23	21889.11	26670.91	20179.87	25745.77	21001.28	25108.82	20377.06

**Fig. 7.** Comparison of varying algorithms profits for multi-day instances in DCRP-eS.

Furthermore, when examining the performance of the RL algorithm across various scenarios in [Table 9](#) and [Fig. 8](#), we observed that average profit exhibited variations, highlighting the notion that route-based launch and retrieval (DCRP-eS) outperformed node-based launch and retrieval (DCRP) and separate truck and drone delivery (DTDRP), and both of these were more effective than scenarios without drones (DVRP). Under similar RL algorithms, the profit of DCRP-eS was on average 5.13 % higher than that of DCRP, 8.51 % higher than DTDRP, and a significant 11.65 % higher than DVRP. On average, the profit of DCRP-eS outperformed the other benchmark scenarios by 8.43 %.

From an economic perspective, we compared the costs detailed in [Table 10](#) and the profits outlined in [Table 9](#). Upon comparison, it is observed that under the RL algorithm, the average profit of DCRP-eS is 4.60 times the cost. Next, DCRP follows with profits standing at 3.67 times the cost; DTDRP comes third at 3.24 times, and finally, DVRP, which is only 2.91 times. This indicates that our scenario is economically viable and can significantly enhance the output-to-input ratio of businesses. In addition, the RL algorithm, as compared to the FCFS algorithm, demonstrates an average profitability rate 2.16 times higher across various scenarios, thus reinforcing the economic viability of our algorithm. Overall, the route synchronization of our scenario and the proposed RL algorithm notably contribute to financial enhancements for businesses.

[Table 11](#) presents the total waiting time of drones under the DCRP-eS and DCRP at the customer node for truck retrieval, which intuitively shows the benefits of our en-route takeoff and landing strategy. This route-based launch and retrieval strategy can save an average of 0.90 h per drone per day.

7.6. Discussion

This section systematically compares the results for single-day instances of drones with varying endurance and load capacity in DCRP-eS solved by RL in [Table 12](#). The endurance range is 10 to 50 km, and the load capacity range is 5 to 20 kg. The endurance range

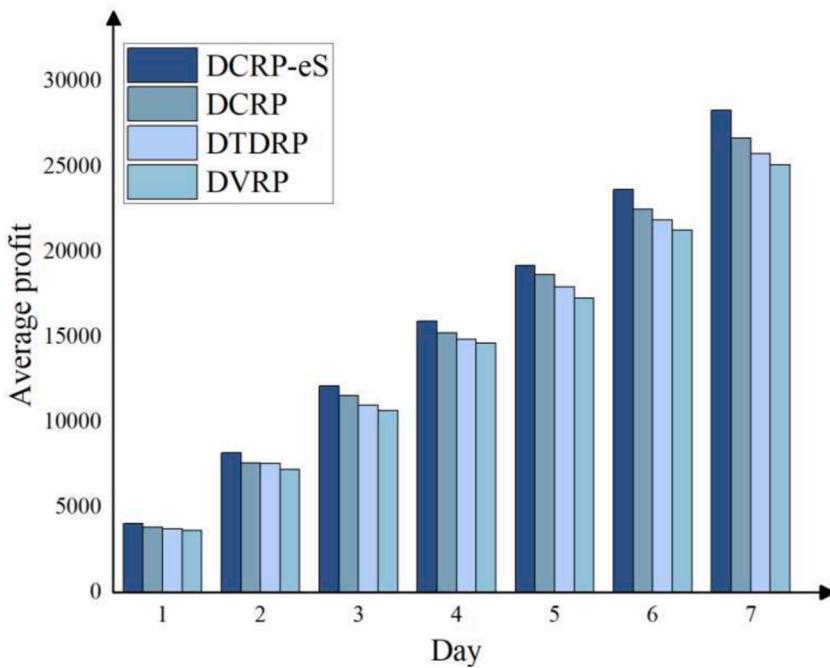


Fig. 8. Comparison of RL profits under various scenarios for multi-day instances.

Table 10

Costs with assorted planning horizons in various scenarios (endurance = 10).

Day	DCRP-eS		DCRP		DTDRP		DVRP	
	RL	FCFS	RL	FCFS	RL	FCFS	RL	FCFS
1	962.89	1599.32	1012.40	1753.91	1098.16	1629.51	1184.96	1674.78
2	1546.39	3434.41	2101.11	3696.09	2136.42	3605.12	2384.21	3622.78
3	2477.09	5264.52	3000.02	5601.58	3398.53	5488.86	3833.54	5636.27
4	3748.98	6648.85	4172.24	7097.52	4495.48	6956.10	4680.92	7048.39
5	4565.28	8576.61	5050.23	9950.01	5743.88	8942.91	6270.42	9149.90
6	5409.28	10261.08	6373.86	10941.26	6875.18	10788.54	7429.40	11019.72
7	5503.14	11853.66	7130.46	13531.64	7965.34	12402.42	8552.47	12590.60

Table 11

Wait time obtained by RL in DCRP-eS and DCRP (hour).

Day	DCRP-eS		DCRP	
	Profit	Wait time	Profit	Wait time
1	4055.09	0	3814.65	10.11
2	8191.48	0	7594.12	23.23
3	12109.85	0	11545.57	25.51
4	15919.44	0	15223.77	29.67
5	19189.19	0	18647.02	37.33
6	23635.55	0	22467.87	54.10
7	28298.23	0	26670.91	64.20

of 10 to 50 km for drone analysis is selected because battery performance studies have shown that this range significantly impacts operational outcomes, with little benefit for endurance beyond 50 km in terms of cost-effectiveness and operational efficiency. The load capacity of 5 to 25 kg covers all request weights, ensuring the drone system can adapt to various customer demands efficiently, thus optimizing the performance of the truck-drone system.

The analysis of the profit in the DCRP-eS reveals a significant relationship between increased drone endurance and load capacity with the profitability of the truck-drone system. Table 12 and Fig. 9 suggest that for drones with higher endurance, the profit increase caused by augmenting the load capacity is also higher. For instance, by increasing the load capacity from 5 kg to 20 kg, using drones with 10 km endurance yields a profit increase of 372.66 (4203.29—3830.63), while using 50 km as the drone endurance results in a

Table 12

The gained profit with varying drone endurance and load capacities.

Load (kg)	Endurance (km)				
	10	20	30	40	50
5	3830.63	3896.91	3890.19	3880.81	3904.93
6	3868.86	3916.77	3921.56	3958.22	3967.59
7	3845.94	3956.48	4010.14	4035.81	4039.39
8	3928.91	3983.52	4058.94	4007.55	4039.32
9	3895.38	4008.52	4118.61	4091.79	4117.56
10	3952.20	4155.69	4190.07	4195.16	4237.86
11	4013.07	4157.67	4228.46	4271.27	4315.41
12	4016.37	4161.52	4262.55	4264.40	4295.27
13	4076.77	4210.01	4360.79	4398.83	4422.26
14	4021.56	4265.91	4386.37	4439.24	4513.22
15	4055.09	4278.76	4487.58	4525.41	4581.55
16	4168.87	4329.63	4496.26	4594.62	4606.52
17	4246.22	4332.36	4563.33	4637.70	4688.03
18	4177.46	4454.16	4575.70	4701.55	4754.27
19	4165.84	4483.94	4635.48	4742.12	4833.90
20	4203.29	4506.89	4647.38	4756.27	4843.48
21	4203.29	4506.89	4647.38	4756.27	4843.48
22	4203.29	4506.89	4647.38	4756.27	4843.48
23	4203.29	4506.89	4647.38	4756.27	4843.48
24	4203.29	4506.89	4647.38	4756.27	4843.48
25	4203.29	4506.89	4647.38	4756.27	4843.48

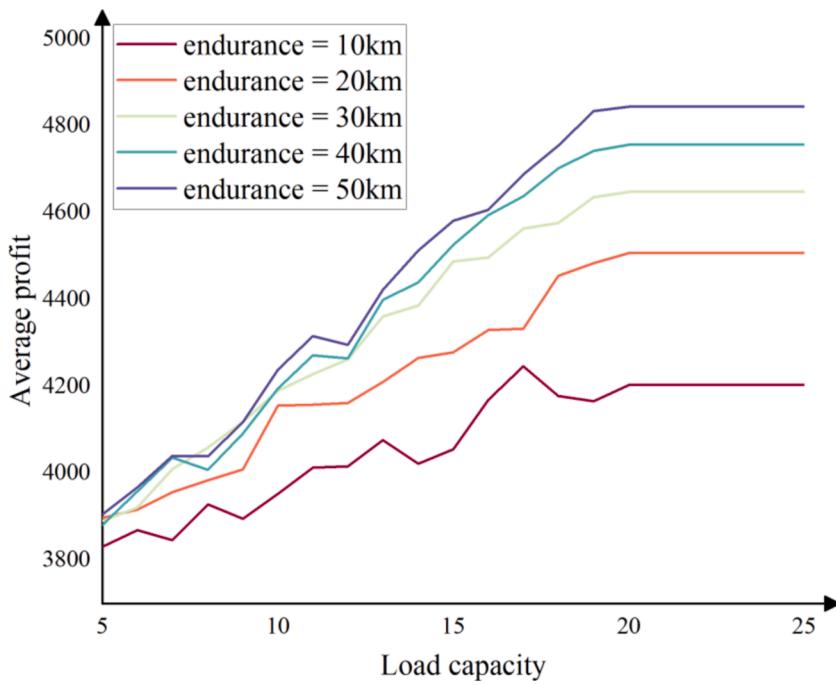


Fig. 9. Comparison of profits under various endurance and load capacities.

profit increase of 938.55 (4843.48—3904.93). Similarly, in situations where drones are capable of carrying heavier loads, augmenting their endurance has a more significant impact on the outcome. This dynamic and nonlinear interplay between endurance and load capacity unveils a synergistic effect, where the optimization of one parameter independently boosts performance and potentiates the benefits of the other, culminating in a synergistic surge in the overall efficiency and financial success of the truck-drone system. The reason may be straightforward that as drones with higher endurance are capable of covering greater distances, they can maintain operations over a larger area without the need for frequent stops. When these drones are also equipped with increased load capacities, they can transport more goods per trip, optimizing each journey and maximizing the utility of their extended range. Furthermore, we observe that after the load capacity reaches 20 kg, the increase in profit does not continue to grow as it does with lower capacities. This plateau may be because a load capacity of 20 kg already accommodates the maximum weight demands of the delivery requests being

considered. Beyond this point, further increases in load capacity might not translate into additional profits, as the drones are already capable of carrying all required goods in a single trip. In such cases, the surplus capacity may go unused, thereby diminishing the impact of further capacity enhancements on overall performance.

Based on the above findings, we supplement several managerial insights regarding the selection of drones for delivery tasks. While it is advantageous to invest in drones with higher load capacities and endurance, it is important to recognize that increasing these parameters beyond a certain point—such as a load capacity exceeding 20 kg and an endurance exceeding 50 km—may not yield proportional profit gains. Therefore, managers should strategically balance these factors, selecting drones that meet delivery demands without incurring unnecessary costs for excessive capacity. Additionally, the specific road network conditions and the average weight of delivery requests should be carefully considered to ensure that drones are optimized for the most common scenarios, maximizing operational efficiency. By investing in drones with higher load capacities and endurance within a cost-effective range, companies can enhance the profitability of their truck-drone systems while maintaining the flexibility to adapt to diverse delivery requirements.

8. Conclusions

This paper studies a dynamic collaborative truck-drone routing problem with en-route synchronization (DCRP-eS), where a logistics operator manages trucks and drones to handle requests that arrive at random times. The operator should respond dynamically to requests with the objective of maximizing profits. We formulate an MDP model to determine the optimal policy for DCRP-eS and create an MIP model for static CRP-eS to facilitate comparison. To solve the MDP model, we develop an RL method that enables real-time responses to requests and provides a novel FLR model to solve the opportunity take-off and landing positions. To address the CRP-eS, we employ the Gurobi and LNS algorithm. We generate problem instances from the Solomon dataset and actual operational information from a logistics operator in Singapore. We assess the effectiveness of RL in solving DCRP-eS compared to benchmark methods including MIP, LNS, and FCFS, as well as benchmark scenarios like DCRP, DTDRP, and DVR. Our computational results demonstrate that the proposed method outperforms the traditional FCFS decision rule for instances of various sizes. Compared to the LNS algorithm for static CRP-eS problems, we also achieve considerable profits. Furthermore, the route-based collaborative launch and retrieval is superior to both traditional truck transportation, separate truck and drone delivery, and node-based collaborative truck-drone delivery.

The proposed model assumes constant speeds for trucks and drones, ignoring various practical factors that may impact the traveling speed (such as the strength of wind and road traffic density). More flexible delivery scenarios, such as coordinated pickup and delivery of goods, distribution area boundaries, drones being able to land on other trucks, or a single truck accommodating multiple drones, are out of the scope of our current problem setting. There are also some exact algorithms and other reinforcement learning algorithms can be designed for our problem. We leave the study on these extensions to future research.

CRediT authorship contribution statement

Haipeng Cui: Writing – original draft, Project administration, Methodology, Conceptualization. **Keyu Li:** Writing – original draft, Methodology, Conceptualization. **Shuai Jia:** Writing – review & editing, Methodology, Funding acquisition, Conceptualization. **Qiang Meng:** Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study is supported by the National Natural Science Foundation of China (72401198, 72371218), the Ministry of Education of Singapore via the project MOE-000458-00, and the Guangdong Basic and Applied Basic Research Foundation (2024B1515020012). Any opinions, findings and conclusions or recommendations expressed in this study are those of the author(s) and do not reflect the views of the Ministry of Education of Singapore.

Appendix A. State transition of the paired agent

In this appendix, we describe the state transition when an action a was applied at a state s . For each $i \in G$ such that $m_i^{(1)} = m_i^{(2)} = 0$ (i.e., truck i is currently idle), $n_i^{(1)} = n_i^{(2)} = 0$ (i.e., drone i is currently idle), let $u(l) = m_i^{(3)}$ be the present location of truck i , $d(l) = n_i^{(3)}$ be the present location of truck i . Present time is t , the subsequent state s' is constructed in the following five cases:

- (i) If $a_2 = 1$ (i.e., drone i is assigned to p request c), then in the subsequent state s' , the states of drone i is updated by setting $n_i^{(1)} \leftarrow 2$, $n_i^{(2)} \leftarrow c$, $n_i^{(3)} \leftarrow l(c)$, $n_i^{(4)} \leftarrow \text{time}(c) - t$, $n_i^{(5)} \leftarrow s_dis$, where $l(c)$ represents the coordinate of the destination for requests c , while $\text{time}(c)$

denotes the starting time for serving the requests c , and s_dis represents the flight distance of the drone serving the request. The state of the truck remains unchanged in the current situation.

- (ii) If $a_3 = 1$ (i.e., truck i is assigned to present request c), then in the subsequent state s' , the states of truck i is updated by setting $m_i^{(1)} \leftarrow 2$, $m_i^{(2)} \leftarrow c$, $m_i^{(5)} \leftarrow m_i^{(3)}$, $m_i^{(6)} \leftarrow m_i^{(4)}$, $m_i^{(3)} \leftarrow l(c)$, $m_i^{(4)} \leftarrow time(c) - t$. The state of the drone remains unchanged in the current situation.
- (iii) If $a_4 = 1$ (i.e., drone i is assigned to retrieve to truck), then in the subsequent state s' , the states of drone i is updated by setting $n_i^{(1)} \leftarrow 1$, $n_i^{(2)} \leftarrow u$, $n_i^{(3)} \leftarrow rel(u)$, $n_i^{(4)} \leftarrow time(u) - t$, $n_i^{(5)} \leftarrow n_i^{(5)} + r_dis$, where $rel(u)$ represents the retrieval location, while r_dis the distance from the drone service node to the retrieval location. The state of the truck remains unchanged in the current situation.
- (iv) If $a_5 = 1$ (i.e., truck i is assigned to return to depot), then in the subsequent state s' , the states of truck i is updated by setting $m_i^{(1)} \leftarrow 1$, $m_i^{(2)} \leftarrow N_0$, $m_i^{(5)} \leftarrow m_i^{(3)}$, $m_i^{(6)} \leftarrow m_i^{(4)}$, $m_i^{(3)} \leftarrow l(N_0)$, $m_i^{(4)} \leftarrow time(N_0) - t$. The state of the drone remains unchanged in the current situation.
- (v) If $a_6 = 1$ (i.e., truck i is assigned to wait in customer node), then in the subsequent state s' , the states of truck i is updated by setting $m_i^{(1)} \leftarrow 0$, $m_i^{(2)} \leftarrow m_i^{(3)}$, $m_i^{(5)} \leftarrow m_i^{(3)}$, $m_i^{(6)} \leftarrow m_i^{(4)}$, $m_i^{(3)} \leftarrow m_i^{(3)}$, $m_i^{(4)} \leftarrow 0$

Appendix B. Algorithm structure and hyperparameters

Fig. 10 illustrates the structure of the DQN algorithm for the reinforcement learning, accompanied by the relevant hyperparameters.

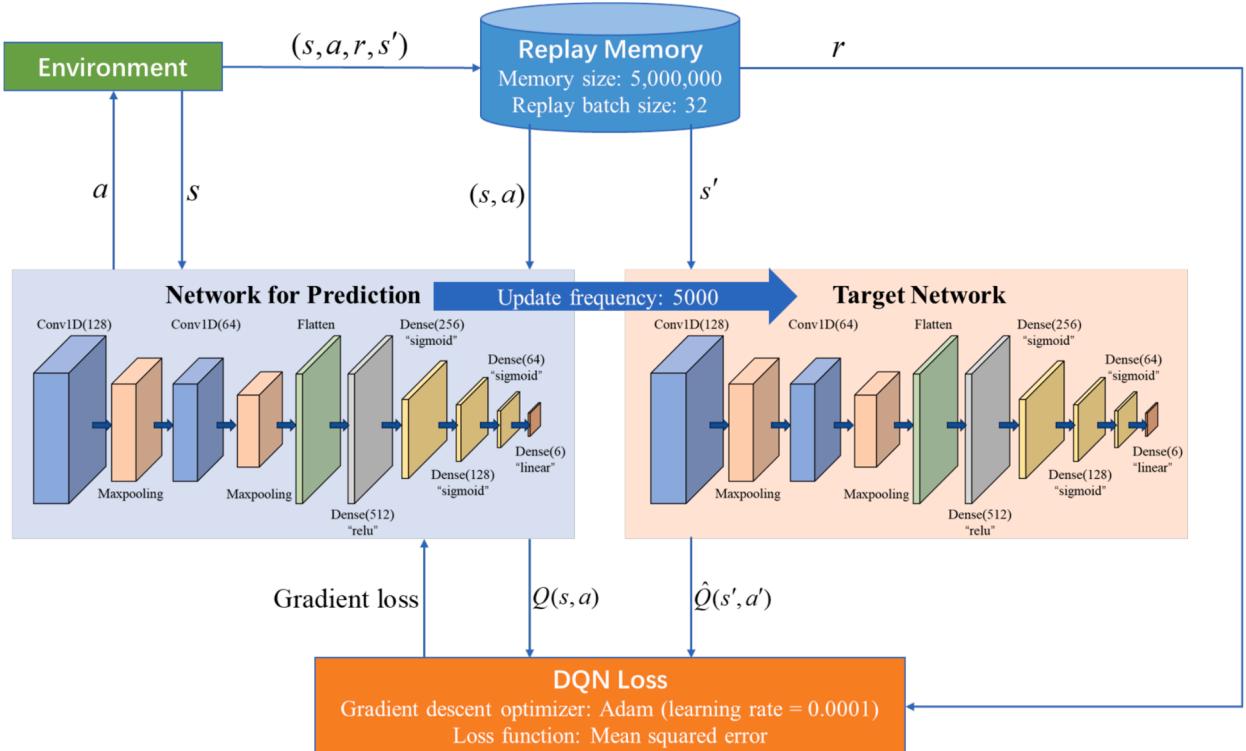


Fig. 10. Structure of the proposed DQN algorithm (Discount factor: 0.99; Initial epsilon: 1; Lower limit of epsilon: 0.05).

References

- AbdAllah, A., Essam, D.L., Sarker, R.A., 2017. On solving periodic re-optimization dynamic vehicle routing problems. *Appl. Soft Comput.* 55, 1–12.
- Accorsi, L., Vigo, D., 2020. A hybrid metaheuristic for single truck and trailer routing problems. *Transp. Sci.* 54 (5), 1351–1371.
- Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. *Transp. Sci.* 52 (4), 965–981.
- Boccia, M., Masonne, A., Sforza, A., Sterle, C., 2021. A column-and-row generation approach for the flying sidekick travelling salesman problem. *Transp. Res. Part C Emerg. Technol.* 124, 102913.
- Carlsson, J.G., Song, S., 2018. Coordinated logistics with a truck and a drone. *Manage. Sci.* 64 (9), 4052–4069.
- Cavani, S., Iori, M., Roberti, R., 2021. Exact methods for the traveling salesman problem with multiple drones. *Transp. Res. Part C Emerg. Technol.* 130, 103280.
- Chen, X.W., Ulmer, M.W., Thomas, B.W., 2022. Deep q-learning for same-day delivery with vehicles and drones. *Eur. J. Oper. Res.* 298 (3), 939–952.
- Chen, X.W., Wang, T., Thomas, B.W., Ulmer, M.W., 2023. Same-day delivery with fair customer service. *Eur. J. Oper. Res.* 308 (2), 738–751.

- Chung, S.H., Sah, B., Lee, J., 2024. Pseudo node insertion method for synchronization in drone-truck combined operations. *Comput. Ind. Eng.* 189, 109963.
- Coindeau, M.A., Gallay, O., Zufferey, N., 2021. Parcel delivery cost minimization with time window constraints using trucks and drones. *Networks* 78 (4), 400–420.
- Crisan, G.C., Nechita, E., On a cooperative truck-and-drone delivery system. 23rd KES International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES). Budapest, HUNGARY, 2019, 38-47.
- Cui, H., Chen, S., Chen, R., Meng, Q., 2022. A two-stage hybrid heuristic solution for the container drayage problem with trailer reposition. *Eur. J. Oper. Res.* 299 (2), 468–482.
- Dastpak, M., Errico, F., Jabali, O., 2023. Off-line approximate dynamic programming for the vehicle routing problem with a highly variable customer basis and stochastic demands. *Comput. Oper. Res.* 159, 106338.
- Dayarian, I., Savelsbergh, M., Clarke, J.P., 2020. Same-day delivery with drone resupply. *Transp. Sci.* 54 (1), 229–249.
- Dell'Amico, M., Montemanni, R., Novellani, S., 2022. Exact models for the flying sidekick traveling salesman problem. *Int. Tran. Oper. Res.* 29 (3), 1360–1393.
- Dohn, A., Rasmussen, M.S., Larsen, J., 2011. The vehicle routing problem with time windows and temporal dependencies. *Networks* 58 (4), 273–289.
- Es Yurek, E., Ozmutlu, H.C., 2018. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transp. Res. Part C Emerg. Technol.* 91, 249–262.
- Frifita, S., Afzar, H.M., Hnaien, F., 2022. An efficient mat-heuristic algorithm for the dynamic disassembly assembly routing problem with returns. *Eur. J. Ind. Eng.* 16 (5), 584–617.
- Gao, J.J., Chen, L., Laporte, G., He, X.T., 2023. Scheduling trucks and drones for cooperative deliveries. *Transp. Res. Part E Logist. Transp. Rev.* 178, 103267.
- Gu, R.X., Poon, M., Luo, Z.H., Liu, Y., Liu, Z., 2022. A hierarchical solution evaluation method and a hybrid algorithm for the vehicle routing problem with drones and multiple visits. *Transp. Res. Part C Emerg. Technol.* 141, 103733.
- Gu, R.X., Liu, Y., Poon, M., 2023. Dynamic truck-drone routing problem for scheduled deliveries and on-demand pickups with time-related constraints. *Transp. Res. Part C Emerg. Technol.* 151, 104139.
- Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H., 2018. On the min-cost traveling salesman problem with drone. *Transp. Res. Part C Emerg. Technol.* 86, 597–621.
- Han, J., Liu, Y.Q., Li, Y., 2023. Vehicle routing problem with drones considering time windows and dynamic demand. *Appl. Sci.-Basel* 13 (24), 13086.
- Harbison, T., 2020. A hybrid genetic algorithm for the traveling salesman problem with drone. *J. Heuristics* 26 (2), 219–247.
- Jeong, H.Y., Song, B.D., Lee, S., 2019. Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones. *Int. J. Prod. Econ.* 214, 220–233.
- Jia, S., Cui, H., Chen, R., Meng, Q., 2022. Dynamic container drayage with uncertain request arrival times and service time windows. *Transp. Res. Part B Methodol.* 166, 237–258.
- Jiang, J., Dai, Y., Yang, F., Ma, Z., 2024. A multi-visit flexible-docking vehicle routing problem with drones for simultaneous pickup and delivery services. *Eur. J. Oper. Res.* 312 (1), 125–137.
- Kim, D., Ko, C.S., Moon, I., 2023. Coordinated logistics with trucks and drones for premium delivery. *Transportmetrica A: Transport Sci.* 1–29.
- Kitjacharoenchai, P., Min, B.C., Lee, S., 2020. Two echelon vehicle routing problem with drones in last mile delivery. *Int. J. Prod. Econ.* 225, 107598.
- Kuo, R.J., Lu, S.H., Lai, P.Y., Mara, S.T.W., 2022. Vehicle routing problem with drones considering time windows. *Expert Syst. Appl.* 191, 116264.
- Lan, B., Suzuki, Y., 2024. Using intermediate points in parcel delivery operations with truck-based autonomous drones. *Decis. Sci.* 1–16.
- Lei, D.M., Cui, Z.Z., Li, M., 2022. A dynamical artificial bee colony for vehicle routing problem with drones. *Eng. Appl. Artif. Intell.* 107, 104510.
- Li, H., Chen, J., Wang, F., Zhao, Y., 2022. Truck and drone routing problem with synchronization on arcs. *Nav. Res. Logist.* 69 (6), 884–901.
- Lin, L.J., 1992. Self-Improving reactive agents based on reinforcement learning, planning and teaching. *Mach. Learn.* 8 (3), 293–321.
- Liu, S., Luo, Z.X., 2023. On-demand delivery from stores: Dynamic dispatching and routing with random demand. *M&Som* 25 (2), 595–612.
- Long, Y., Xu, G., Zhao, J., Xie, B., Fang, M., 2024. Dynamic truck-UAV collaboration and integrated route planning for resilient urban emergency response. *IEEE Trans. Eng. Manage.* 71, 9826–9838.
- Lu, Y., Yang, J., Yang, C., 2023. A humanitarian vehicle routing problem synchronized with drones in time-varying weather conditions. *Comput. Ind. Eng.* 184, 109563.
- Luo, Q., Wu, G., Ji, B., Wang, L., Suganthan, P.N., 2022. Hybrid multi-objective optimization approach with pareto local search for collaborative truck-drone routing problems considering flexible time windows. *IEEE Trans. Intell. Transp. Syst.* 23 (8), 13011–13025.
- Luo, Q.Z., Wu, G.H., Trivedi, A., Hong, F.Y., Wang, L., Srinivasan, D., 2023. Multi-objective optimization algorithm with adaptive resource allocation for truck-drone collaborative delivery and pick-up services. *IEEE Trans. Intell. Transp. Syst.* 24 (9), 9642–9657.
- Marinelli, M., Caggiani, L., Ottomanelli, M., Dell'Orco, M., 2018. *En route* truck-drone parcel delivery for optimal vehicle routing strategies. *IET Intel. Transport Syst.* 12 (4), 253–261.
- Masmoudi, M.A., Mancini, S., Baldacci, R., Kuo, Y.H., 2022. Vehicle routing problems with drones equipped with multi-package payload compartments. *Transp. Res. Part E Logist. Transp. Rev.* 164, 102757.
- Masone, A., Poikonen, S., Golden, B.L., 2022. The multivisit drone routing problem with edge launches: An iterative approach with discrete and continuous improvements. *Networks* 80 (2), 193–215.
- Meng, S., Guo, X., Li, D., Liu, G., 2023. The multi-visit drone routing problem for pickup and delivery services. *Transp. Res. Part E Logist. Transp. Rev.* 169, 102990.
- Moghaddam, B.F., Ruiz, R., Sadjadi, S.J., 2012. Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Comput. Ind. Eng.* 62 (1), 306–317.
- Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transp. Res. Part C Emerging Technol.* 54, 86–109.
- Pan, W.X., Liu, S.Q., 2023. Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Appl. Intell.* 53 (1), 405–422.
- Panadero, J., Juan, A.A., Bayliss, C., 2020. Maximising reward from a team of surveillance drones: A simheuristic approach to the stochastic team orienteering problem. *Eur. J. Ind. Eng.* 14 (4), 485–516.
- Pina-Pardo, J.C., Silva, D.F., Smith, A.E., Gatica, R.A., 2024. Dynamic vehicle routing problem with drone resupply for same-day delivery. *Transp. Res. Part C Emerg. Technol.* 162, 104611.
- Ramos, T.R.P., Vigo, D., 2023. A new hybrid distribution paradigm: Integrating drones in medicines delivery. *Expert Syst. Appl.* 234, 120992.
- Rothenbächer, A.K., Drexl, M., Irnich, S., 2018. Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows. *Transp. Sci.* 52 (5), 1174–1190.
- Sacramento, D., Pisinger, D., Ropke, S., 2019. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transp. Res. Part C Emerg. Technol.* 102, 289–315.
- Salama, M.R., Srinivas, S., 2022. Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites. *Transp. Res. Part E Logist. Transp. Rev.* 164, 102788.
- Schermer, D., Moeini, M., Wendt, O., 2019a. A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Comput. Oper. Res.* 109, 134–158.
- Schermer, D., Moeini, M., Wendt, O., 2019b. A matheuristic for the vehicle routing problem with drones and its variants. *Transp. Res. Part C Emerg. Technol.* 106, 166–204.
- Shaw, P., Using constraint programming and local search methods to solve vehicle routing problems. MAHER, M., PUGET, J.F. Principles and Practice of Constraint Programming - Cp98. 1998, 417-431.
- Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* 35 (2), 254–265.
- Sutton, R.S., 1988. Learning to predict by the methods of temporal differences. *Mach. Learn.* 3, 9–44.
- Thomas, T., Srinivas, S., Rajendran, C., 2023. Collaborative truck multi-drone delivery system considering drone scheduling and en route operations. *Ann. Oper. Res.* 1–47.
- Ulmer, M.W., Goodson, J.C., Mattfeld, D.C., Hennig, M., 2019. Offline-online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transp. Sci.* 53 (1), 185–202.

- Van Hasselt, H.G., Arthur; Silver, David, Deep reinforcement learning with double q-learning. 30th AAAI Conference on Artificial Intelligence. 2016, 2094-2100.
- Vásquez, S.A., Angulo, G., Klapp, M.A., 2021. An exact solution method for the TSP with drone based on decomposition. *Comput. Oper. Res.* 127, 105127.
- Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L., Velasco, N., 2013. A matheuristic for the truck and trailer routing problem. *Eur. J. Oper. Res.* 230 (2), 231–244.
- Wang, Y., Peng, S.G., Zhou, X.S., Mahmoudi, M., Zhen, L., 2020. Green logistics location-routing problem with eco-packages. *Transp. Res. Part E Logist. Transp. Rev.* 143, 102118.
- Wang, Y., Luo, S.Y., Fan, J.X., Zhen, L., 2024. The multidepot vehicle routing problem with intelligent recycling prices and transportation resource sharing. *Transp. Res. Part E Logist. Transp. Rev.* 185, 103503.
- Wang, X.Y., Poikonen, S., Golden, B., 2017. The vehicle routing problem with drones: Several worst-case results. *Optim. Lett.* 11 (4), 679–697.
- Wang, Z., Sheu, J.-B., 2019. Vehicle routing problem with drones. *Transp. Res. Part B Methodol.* 122, 350–364.
- Wang, Y., Wang, Z., Hu, X.P., Xue, G.Q., Guan, X.Y., 2022. Truck-drone hybrid routing problem with time-dependent road travel time. *Transp. Res. Part C Emerging Technol.* 144, 103901.
- Yin, Y., Yang, Y., Yu, Y., Wang, D., Cheng, T.C.E., 2023. Robust vehicle routing with drones under uncertain demands and truck travel times in humanitarian logistics. *Transp. Res. Part B Methodol.* 174, 102781.
- Zhang, J., Van Woensel, T., 2023. Dynamic vehicle routing with random requests: A literature review. *Int. J. Prod. Econ.* 256, 108751.