

A Real-time UAV Delivery System Considering Dock Selection and Spatial Conflict

Ziyi Hu^a, Yue Cao^{a,*}, Xinyu Li^b, Xu Zhang^c, Chuanke Zhang^d, Zhi Liu^e

^a*School of Cyber Science and Engineering, Wuhan University, Wuhan, China*

^b*Hubei Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan, China*

^c*School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, U.K.*

^d*School of automation, China University of Geosciences, Wuhan, China*

^e*Department of Computer and Network Engineering, University of Electro-Communications, Tokyo, Japan*

Abstract

The development of Unmanned Aerial Vehicle (UAV) has brought new vitality to logistics industry, especially for real-time delivery demand. Under real-time delivery model, the UAV transports parcel between docks (e.g., the designated location for UAV to drop off/pick up parcels), with vision that the fast response to demand is satisfied. Unlike traditional ground delivery services, the UAV model not only considers delivery cost, but also takes into account dock spatial conflicts during UAV takeoff and landing. Since the UAV delivery scheduling is a non-deterministic polynomial hard problem, the optimal solution is hard to obtain through off-the-shelf solvers within a limited response time under large-scale demand. Hence, we propose a real-time UAV delivery system, which provides the in-time UAV scheduling scheme under large-scale demand. In particular, the Ant Colony Algorithm and Q-value learning are utilized to select the optimal dock for UAV, so that the UAV flight time is reduced under spatial conflict avoidance. Besides, through solving the bi-level linear programming model, the optimal number of deployed docks and UAVs is determined, to reduce the delivery cost. Based on simulation experiments, our scheme outperforms other schemes on

*Corresponding author.

Email addresses: ZiyiHu@whu.edu.cn (Ziyi Hu), yue.cao@whu.edu.cn (Yue Cao), xinyuli@whu.edu.cn (Xinyu Li), x.zhang27@uea.ac.uk (Xu Zhang), ckzhang@cug.edu.cn (Chuanke Zhang), liu@ieee.org (Zhi Liu)

reducing UAV flight time and is robust to different geographical distribution of docks. The impact of spatial conflicts on UAV delivery is also reduced under our scheme.

Keywords:

UAV, Real-time delivery, delivery scheduling.

1. Introduction

As traffic congestion intensifies in urban areas, the transportation cost and delivery delay in traditional logistics industry has increased. Such dilemma is mainly caused by the utilization of vehicles to deliver parcels, as the delivery efficiency depends on ground transportation conditions and road infrastructure (Bhuiyan, Walker, & Ahmed, 2024). To reduce the impact of traffic congestion, the Unmanned Aerial Vehicle (UAV) is further applied to logistics industry (Kim, Kweon, Hwang, & Lee, 2024; Shao et al., 2023). Due to the high speed and avoidance of traffic disturbance (Wang, Leng, & Xiong, 2023), the UAV can satisfy the delivery demand with time requirements, including takeout delivery, medical resource allocation and emergency document transportation (Wang et al., 2023).

Although UAV can complete delivery tasks at high speed, the flight endurance of UAV is limited by battery capacity. Due to the limited flight endurance, the UAV may not be able to deliver all customers with different distance distributions at once (Mishra, & Tiwari, 2024). Instead, an improved solution is to deploy docks within the flight endurance of UAV, and merchants/customers can walk to docks to drop off/pick up parcels. Here, the dock is a place for UAV to drop off/pick up parcels. The parcels are transported by UAV between docks, e.g., the UAV picks up parcels at one dock and then fly to another dock to drop off. The work (Lemardelé, Estrada, Pagès, & Bachofner, 2021) demonstrates the effectiveness of dock-delivery models, while there is few works discussing the impact of dock and UAV deployment on delivery cost. Here, the delivery cost includes (1) system cost: deployment cost of docks and UAVs; (2) service cost: waiting time for UAV to start delivery task¹, flight time of UAV between docks to transport

¹The waiting time is caused when the subsequent task is received during UAV delivering the current task, as UAV must complete the current task before starting to deliver the subsequent one.

parcels, and customer/merchant walking time to docks.

In addition to dock/UAV deployment, the UAV task scheduling is also a crucial issue in reducing delivery cost. In general, the UAV task scheduling can be divided into two stages: the first stage is the matching of delivery task with docks, and the second stage is the matching of UAVs with tasks. Moreover, due to the complexity of UAV flight operations, spatial conflicts on docks during UAV takeoff/landing must be considered when optimizing the matching decisions under two stages (e.g., two UAVs cannot take off from the same dock simultaneously). Although work (Pei, Fang, Weng, & Yi, 2023) discusses the matching between UAVs and tasks of the second stage to avoid spatial conflicts, the dock selection of the first stage is not considered, thus increasing the flight time of UAVs between docks.

Based on above analyses, our main contributions are as follows:

(1) Although deploying docks has been proven efficiency in reducing delivery cost, there are few discussions on determination of dock/UAV deployment based on demand characteristics in delivery area. Here, a smaller number of docks and UAVs cannot meet the delivery demand, while an excessive number will increase the deployment cost of delivery system. Hence, a bi-level linear programming model is solved to obtain the optimal number of docks and UAVs deployed in delivery area, so that the system cost and service cost are both reduced.

(2) The work (Pei et al., 2023) discusses the UAV task scheduling of the second stage to avoid spatial conflicts, while the optimal dock selection under the first stage is not considered. This further increases the flight time of UAVs, as UAVs always fly between docks to deliver parcels. Therefore, we propose a delivery scheme considering both dock selection and spatial conflict avoidance. The optimal dock is selected through Ant Colony Algorithm and Q-value learning, to reduce the flight time and avoid spatial conflicts. In particular, the Ant Colony Algorithm is utilized to select the optimal dock in real-time, to reduce the average flight time based on UAV pheromones and heuristic information. Besides, the Q-value learning evaluates the impact of different avoidance strategies on current and future service cost, thereby reducing average service cost while avoiding dock spatial conflicts.

The rest of our paper is organized as follows: Section 2 reviews the work related to UAV delivery. Our system model is presented in Section 3. In Section 4, we propose the scheme to determine the optimal number of docks and UAVs deployed in delivery area. Section 5 details the strategy to schedule the UAV, select the optimal dock and avoid dock spatial conflicts. At

last, simulation results are shown in Section 6, and Section 7 draws the conclusions.

2. Related Works

2.1. Offline Demand UAV Delivery

Under offline demand model, all demand information is acquired before scheduling the UAV, and the UAV directly delivers parcels from warehouse to customers' location. Such model can be seen as the typical Parallel Drone Scheduling Traveling Salesman Problem Without Vehicle (PDSTSP-WV) discussed in work (Murray, & Chu, 2015). In general, the PDSTSP-WV problem is modelled through linear programming, and heuristic algorithms are common methods to solve the linear model. The work (Shi, Lin, Li, & Li, 2022) proposes a bi-objective optimization model for medical supplies delivery with the UAV, and solves the model through Non-dominated Sorting Genetic Algorithm II (NSGA-II) to reduce the delivery time. The genetic algorithm and simulated annealing can also be utilized to jointly optimize the solution, to promise convergence ability on solving large-scale instances demonstrated in work (Sajid, Mittal, Pare, & Prasad, 2022). The work (Raissa, Laurent, Dominique, Nathalie, & Alain, 2018) reduces the solution search time through dominance rule and upper bound analysis in a iterative two-step heuristic algorithm, thus converging quickly towards great solutions.

The works (Liu, Li, & Wu, 2023; Li, Hansen, & Zou, 2022) both discuss algorithms to solve UAV group formation and trajectory planning for delivery tasks. However, the delivery cost is not taken into account in above two works. Differently, the work (Liu, Li, & Wu, 2023) applies the pigeon algorithm to divide UAV groups, while work (Li, Hansen, & Zou, 2022) schedules UAV paths through batch optimization models. Another method is to model the urban environment as a 3D layered visibility graph network, and an improved ripple-spreading algorithm is developed to solve the multi-category multi-objective path optimization problem of UAVs (Hu et al., 2025). In particular, the work (Hu, Chen, Lyons, Solak, & Zink, 2024) discusses the impact of environmental factors such as weather on the UAV delivery cost, and the conclusion shows that the environment has a lower impact on large UAVs. To reduce the solving time of UAV task assignment, a computationally inexpensive two-phase approach is developed in work (Song, Park, & Park, 2022).

These heuristic algorithms rely on domain knowledge to propose rules for solution initialization and search, which need to be adjusted according to different scenarios. Instead, the out-performance of Reinforcement Learning (RL) in decision making has attracted research in UAV delivery scheduling, as RL learns the search strategy automatically without handcrafted designed heuristic (Wu, Fan, Shi, & Feng, 2023). The work (Li, Yan, Yu, Li, & Liu, 2024) proposes a neighborhood search based on reinforcement learning to improve the quality of the particles in the searching process. The physics-informed neural network framework is also utilized to solve the continuous traffic equilibrium model, and UAV flight routes are scheduled to avoid aerial conflicts (She, & Ouyang, 2024). Besides, the attention encoder-LSTM decoder hybrid model is proposed to determine the UAV routing (Bogyrbayeva et al., 2023).

The Deep Q-network (DQN) models are also used in multi-UAV path planning under natural disaster scenarios (Kutpanova, Kadhim, Zheng, & Zhakiyev, 2025), while the high training time cannot meet the time requirements for emergency demand. Although DQN can also be used to plan UAV flight operations to reduce energy consumption in work (Wang, Zhang, Du, Hua, & Zhong, 2024), the delivery task scheduling is not considered. Besides, the work (Liu et al., 2025) designs a neural network structure based on the attention mechanism from the UAV perspective, which process the features of both UAVs and requests as input for training. However, the network training requires a large amount of data, and the feature extraction may be inefficient. In addition to UAV path planning, the work (Khamidehi, Raeis, & Sousa, 2021) considers the matching of delivery tasks, UAVs and delivery centers based on double deep -Network. Although RL method can obtain satisfying results, the optimality conditions of these methods are often difficult to identify or explained (Mishra, & Tiwari, 2024).

2.2. Real-time Demand UAV Delivery

Under real-time demand model, each demand must be delivered as soon as possible through UAV delivering parcels between docks. Unlike the offline demand, the real-time demand arrives during UAV flight. Hence, the delivery system must decide whether to directly process each demand based on first-come-first-served, or wait for a time period to centrally process the collected demand amount. In the existing literature discussing real-time demand delivery (Pei et al., 2023; Sun, Fang, Guo, & Hu, 2024), the demand

within each decision-making period are centrally processed, so that heuristic algorithms for offline demand can be applied (Tao, Dai, Jiang, & Chen, 2020). However, this may not meet the delivery time requirements of real-time demand, because the demand is always processed at the end of each decision-making period.

Except for demand processing, the flight time is also not taken into account when selecting the optimal dock in works (Pei et al., 2023; Sun, Fang, Guo, & Hu, 2024), thus increasing the service cost. Moreover, the dock selection should consider spatial conflict avoidance, to adapt to practical scenarios. Although work (Pei et al., 2023) develops a linear model to schedule the takeoff/landing time of UAVs to avoid spatial conflicts, it does not consider the impact of conflict avoidance on dock selection. In fact, how to select the optimal dock under conflict avoidance is the crucial issue in reducing service cost.

Moreover, the optimal number of delivery UAV is also discussed in work (Pei, Dai, Yuan, Du, & Liu, 2021), where the optimal UAV number is determined through queuing theory with dynamic pricing. However, such method relies on adjusting the delivery price, while the customer acceptance of delivery price is not fully considered. In fact, there are significant differences in price acceptance among different groups of people demonstrated in work (Gao, Yang, & Qu, 2021).

2.3. Our Motivation

To reduce the delivery cost, we program a bi-level linear programming model to minimize both the system cost and service cost. Through solving the bi-level model, the optimal number of docks/UAVs deployed in delivery area is obtained. Besides, the optimal dock and spatial conflict avoidance strategy are selected based on Ant Colony Algorithm and Q-value learning, to reduce the service cost and avoid spatial conflicts.

3. System Model and Problem Definition

3.1. Overview

Fig. 1 shows the workflow of our system. Assuming that the entire city map is divided into different delivery areas, with one Delivery Center (DC) deployed in each delivery area. The demand in each delivery area follows a Poisson distribution. The DC determines the number of docks and UAVs

deployed in its delivery area through bi-level model (Decision 1-1 and 1-2). The customer will send delivery request to the nearest DC, including the demand merchant location and customer location. Upon receiving the request, the DC will allocate a UAV for the request immediately (Decision 2-1). After determining the UAV for delivery, the DC selects the optimal dock (Decision 2-2) and spatial conflict avoidance strategy (Decision 2-3) to minimize service cost. The DC then notifies the UAV of determined flight dock location and speed adjustment amount. The relevant entities are discussed below:

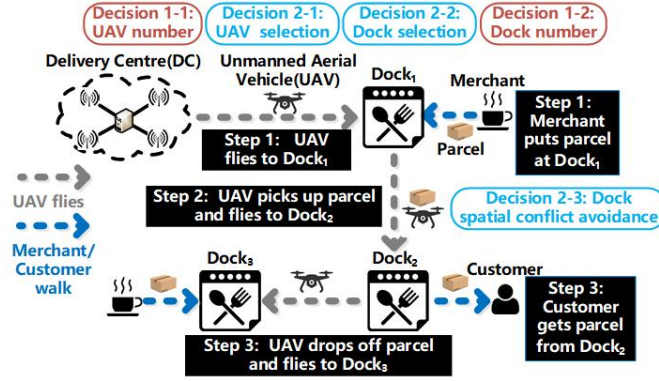


Figure 1: A typical workflow of UAV delivery System

- **Delivery Center (DC):** The DC determines the number of deployed docks and UAVs (in Section 4), and then processes delivery requests from customers. For each request, the DC will select the optimal UAV (in Section 5.1), flight dock (in Section 5.2) and spatial conflict avoidance strategy (in Section 5.3), to finish this request with the minimum service cost.
- **Unmanned Aerial Vehicle (UAV):** The UAV is responsible for delivering parcels between docks, where requests allocated from DC are delivered by UAV on a first-come-first-served basis.
- **Dock:** The merchant put parcels at one dock for UAV to pick up, and then the UAV flies to another dock to drop off parcels for customer. It is worth noting that it is not allowed to pick up and drop off the same parcel at the same dock. Even so, it is feasible to pick up one parcel after drop off another parcel at the same dock.

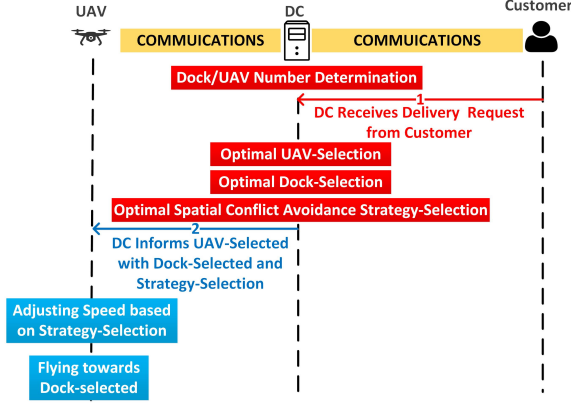


Figure 2: Time sequences of UAV delivery

3.2. Protocol for UAV delivery Process

Considering a customer with delivery demand, the protocol for UAV delivery is described below, with the time sequences shown in Fig. 2.

Step 1: Before processing the request, the DC will determine the number of deployed docks and UAVs discussed in Section 4, so that the system cost and service cost are both minimized.

When the request is received, the flight distance and available time of candidate UAVs is estimated by DC, to choose the optimal UAV based on Alg. 2 in Section 5. Then, the optimal dock is selected based on Ant Colony Algorithm (according to Alg. 3 in Section 5). Besides, the DC determines the optimal spatial conflict avoidance strategy through Q-value learning based on Alg. 4 in Section 5 (the time sequence is **1**).

Step 2: The DC informs the selected UAV with the optimal dock and spatial conflict avoidance strategy. Based on the optimal strategy, the UAV will decide whether to reduce the flight speed to the optimal dock (the time sequence is **2**).

3.3. Assumption

The assumptions in this paper are explained as follows:

3.3.1. Spatial Conflict Assumptions

We consider three types of dock spatial conflicts shown in Fig. 3, where δ is the minimum time difference for UAV to arrive/depart from the same dock without spatial conflicts.

- **Scenario (a):** The arrival time difference of two UAVs is within the minimum time difference δ .
- **Scenario (b):** The departure time difference of two UAVs is within δ .
- **Scenario (c):** The difference between departure time and arrival time of two UAVs is within δ .

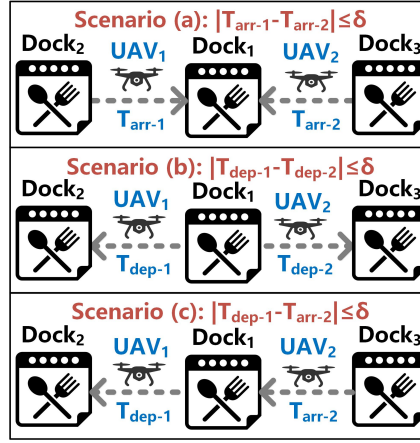


Figure 3: Three scenarios of spatial conflict

To avoid spatial conflict, we propose two strategies:

- **Strategy A (Speed Adjustment):** The UAV adjusts its flight speed, so that the arrival/departure time difference with other UAV is greater than the minimum time difference δ . For example, in Scenario (a) of Fig. 3, UAV₁ delays the arrival time T_{arr-1} to $T_{arr-2} + \delta$ by adjusting speed (the conflict Dock₁ is still selected).

It is worth noting that the route and speed of UAV are usually determined before takeoff, so it is difficult to change the speed during the flight. Therefore, for Strategy A, the UAV must determine the speed change amount before departure.

- **Strategy B (Conflict Dock Exclusion):** When selecting the optimal dock, docks with spatial conflicts will be excluded. The UAV will only fly to docks without spatial conflicts and do not adjust flight speed. For example in Fig. 3, UAV₁ will not select the conflict Dock₁.

3.3.2. UAV Assumptions

- **Payload Capacity:** The parcel weight of each delivery request is within the UAV payload capacity.
- **Flight Endurance:** We assume that the flight distance between DC and dock, as well as flight distance between docks, is within the UAV flight endurance. Besides, the UAV can replenish energy at docks.
- **Operation Time:** The UAV operations, including launching/retrieving, energy replenishment, and parcel pick-up/drop-off, are assumed to not require time.

3.3.3. Cost Assumptions

The delivery cost includes:

- **System Cost:** We assume that deploying a dock or UAV has a fixed cost, and the system cost is the summation of deployment cost for all docks as well as UAVs in delivery area.
- **Service Cost:** The cost of UAV completing a delivery request, including the waiting time for UAV to start delivering, the UAV flight time between docks to deliver parcels and merchant/customer walking time to docks. Since the service cost is defined as the summation of times, the service cost is equal to service time, and we use cost terminology uniformly.

3.4. Problem Definition

Our goals are: 1) Minimizing the delivery cost, and 2) Reducing the service cost of each delivery request. The relevant variables are shown in Table 1.

3.4.1. Minimizing the delivery cost

Since the delivery cost includes two costs: the system cost and service cost, we propose a bi-level linear programming model:

$$\text{Min } f_1(M, S) = c_1 \times M + c_2 \times S \quad (1)$$

$$\text{Min } f_2 = EWait + EFly + EWalk \quad (2)$$

Table 1: Table of Notations in Section 3

Symbol	Description
c_1/c_2	the deployment cost of dock/UAV
M/S	the deployment number of dock/UAV
M^*/S^*	the optimal deployment number of dock/UAV
Ω_{UAV}	the set of UAVs and indexed as i
$EWait$	expected waiting time for UAV to start delivering
$EFly$	expected UAV flight time between docks to deliver parcels
$EWalk$	expected merchant/customer walking time to docks
U^*	the optimal UAV for the delivery request
W_i	expected waiting time for UAV i to start the delivering
T_i	expected flight time between locations of UAV i and merchant
Ω_{dock}	the set of docks and indexed as j
D_1^*/D_2^*	the optimal dock for pick up/drop off parcel
$F_{j,j'}$	expected flight time among UAV location, dock j and dock j'
K_j	expected merchant walking time to dock j
$K_{j'}$	expected customer walking time to dock j'
$Stra^*$	the optimal spatial conflict avoidance strategy
δ	the minimum time difference to avoid spatial conflicts

where the notations are shown in Table 1. The upper-level problem of Eq. (1) aims to find the optimal number of docks and UAVs, denoted as M^*/S^* , to minimize the system cost.

In Eq. (2), $EWait$ is the expected waiting time for UAV to start delivering, $EFly$ is the expected UAV flight time between docks to deliver parcels, and $EWalk$ is the expected merchant/customer walking time to docks. The lower-level problem in Eq. (2) aims to minimize the expected service cost under the optimal dock and UAV number M^*/S^* .

3.4.2. Reducing the service cost of each delivery request

For a delivery request, the DC firstly selects the optimal UAV U^* to allocate the request according to Eq. (3):

$$U^* = \underset{i \in \Omega_{UAV}}{argmin} (W_i + T_i) \quad (3)$$

where Ω_{UAV} is the set of UAVs and indexed as i . W_i is the expected waiting time for UAV i to start this delivering, and T_i is the expected flight time between the last location of UAV i and demand merchant location.

Then, the DC determines the optimal dock for UAV to pick up and drop off parcel:

$$D_1^*, D_2^* = \underset{j, j' \in \Omega_{dock}}{argmin} (F_{j, j'} + K_j + K_{j'}) \quad (4)$$

where D_1^* is the optimal dock to pick up parcel, and D_2^* is the optimal dock to drop off parcel. Ω_{dock} is the set of docks and indexed as j . $F_{j, j'}$ is the expected flight time among UAV last location, dock j and dock j' . K_j is the expected merchant walking time to dock j , and $K_{j'}$ is the expected customer walking time to dock j' .

At last, the DC selects the optimal spatial conflict avoidance strategy S^* based on Q-value learning, where $Q - value$ is the state value corresponding to the selected strategy:

$$Stra^* = \underset{Stra.A, \quad Stra.B}{argmin} (Q - value) \quad (5)$$

The Q-learning is a typical algorithm in Reinforcement Learning, which repeatedly explores the optimal paths during iterations based on record value and current negative reward (Zhong, & Wang, 2025).

4. Determination of Dock and UAV Number

Fig. 4 depicts the overall operational logic of our scheme. In this section, we solve the bi-level linear programming model, to obtain the optimal number of docks and UAVs, so that the system cost and service cost are both minimized. The relevant variables are shown in Table 2.

4.1. Determination of optimal Dock Number

We firstly utilize two weight values α_1 and α_2 to combine Eq. (1) and Eq. (2):

$$Min \quad \alpha_1 \times (c_1 \times M + c_2 \times S) + \alpha_2 \times (EWait + EFly + EWalk) \quad (6)$$

where $0 < \alpha_1, \alpha_2 < 1$ and $\alpha_1 + \alpha_2 = 1$. We assume that α_1 and α_2 are designated by DC, to further prefer to reduce system cost or service cost. For

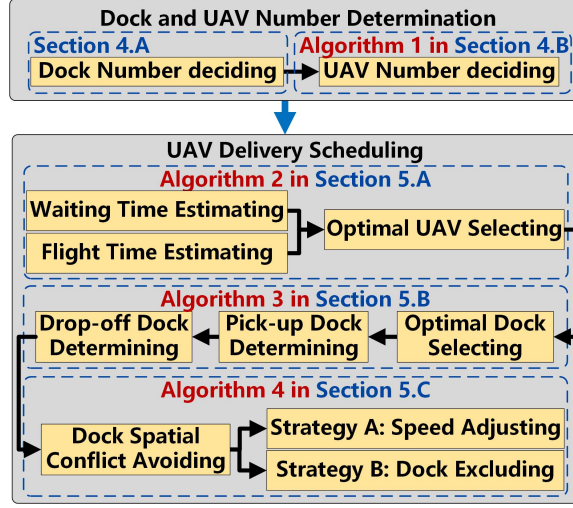


Figure 4: Algorithm flows in Section IV and Section V

example, if dc tends to optimize customer experience and reduce service cost, then α_2 will be set larger than α_1 (like $\alpha_2 = 0.7$ and $\alpha_1 = 0.3$). In general, we assume that the importance of two costs is equivalent, as DC needs to minimize both costs. The expected distance between merchant/customer location and dock is denoted as x , and the walking speed of merchant/customer is denoted as V_{user} . Then, the expected walking time is calculated as:

$$EWalk = x/V_{user} \quad (7)$$

According to Voronoi diagram discussed in work (Lemardelé et al., 2021), to ensure that all merchant/customer locations in the delivery area are within the expected distance x from the dock, the number of deployed docks is required based on Eq. (8):

$$M = \lceil \frac{A}{2x^2} \rceil \quad (8)$$

where M is the required number of docks, A is the square of delivery area, and $\lceil \cdot \rceil$ is an upward rounding function. According to the continuous approximation conclusion in work (Daganzo, 1984), the expected UAV flight time between docks to deliver parcel $EFly$ is estimated as:

Table 2: Table of Notations in Section 4

Symbol	Description
α_1/α_2	weight value of objective function
x	expected distance between merchant/customer location and dock
x^*	optimal distance between merchant/customer location and dock
A	square of delivery area
V_{user}	walking speed of merchant/customer
V_{UAV}	flight speed of UAV
$Function$	waiting time estimation function based on queuing theory
ρ^*	service intensity of UAV
LB/UB	the lower/upper bound for UAV number searching
Max_Num	the maximum UAV number

$$EFly = \frac{(\frac{2\sqrt{2}}{\sqrt{3}} + \frac{1}{2}) \times x}{V_{UAV}} \approx \frac{2.13x}{V_{UAV}} \quad (9)$$

where $(\frac{2\sqrt{2}}{\sqrt{3}} + \frac{1}{2}) \times x$ is the estimated flight distance for one delivery request through continuous approximation, and V_{UAV} is the flight speed of UAV. Based on queuing theory (Gross, Shortle, Thompson, & Harris, 2008), the expected waiting time for UAV to start delivering $EWait$ is related to arrival rate (fixed and known), service rate ($1/EFly$) and server number (deployed numebr of UAV, given as S). Therefore, the Eq. (6) can be rewritten as:

$$\begin{aligned} Min \quad & \alpha_1 \times \left(\frac{c_1 \times A}{2x^2} + c_2 \times S \right) \\ & + \alpha_2 \times \left(Function\left(\frac{V_{UAV}}{2.13x}, S\right) + \frac{2.13x}{V_{UAV}} + x/V_{user} \right) \end{aligned} \quad (10)$$

where $Function(\frac{V_{UAV}}{2.13x}, S)$ is the classical function for calculating waiting time in queuing theory. The Eq. (10) can be further divided into:

$$Min \quad \frac{1}{x^2} \times \left(\frac{\alpha_1 \times c_1 \times A}{2} \right) + x \times \left(\frac{2.13\alpha_2}{V_{UAV}} + \alpha_2/V_{user} \right) \quad (11)$$

$$\text{Min } S \times \alpha_1 \times c_2 + \text{Function}(\frac{V_{UAV}}{2.13x}, S) \times \alpha_2 \quad (12)$$

Note that Eq. (11) only contains one variable x (the values of other symbols are fixed and known), Eq. (12) only contains two variables x and S . Therefore, we obtain the value of x that minimizes Eq. (11), and then determine S to minimize Eq. (12). Let $f(x) = \frac{1}{x^2} \times (\frac{\alpha_1 \times c_1 \times A}{2}) + x \times (\frac{2.13\alpha_2}{V_{UAV}} + \alpha_2/V_{user})$, and derive $f(x)$ to obtain the first order derivative:

$$f'(x) = -\frac{\alpha_1 \times c_1 \times A}{x^3} + \frac{2.13\alpha_2}{V_{UAV}} + \alpha_2/V_{user} \quad (13)$$

It is obvious that $f'(x)$ monotonically increases. Therefore, the optimal x^* of minimizing $f(x)$ can be obtained by equating the first order derivative to zero:

$$f'(x^*) = 0 \rightarrow x^* = \sqrt[3]{\frac{\alpha_1 \times c_1 \times A}{\frac{2.13\alpha_2}{V_{UAV}} + \alpha_2/V_{user}}} \quad (14)$$

and the optimal dock number is obtained substitute into Eq. (8):

$$M^* = \lceil \frac{A}{2(x^*)^2} \rceil \quad (15)$$

4.2. Determination of optimal UAV Number

After determining the optimal dock number, the service rate can be estimated as $1/EFly$ according to Eq. (9). This means that the service intensity (arrival rate divided by service rate) of one UAV can also be estimated, denoted as ρ^* . Then, the expected waiting time for UAV to start delivering will only change with UAV number S (x is no longer an independent variable in *Function*, as it is fixed as the optimal value x^*):

$$\begin{aligned}
EW_{ait} = Function(S) &= \frac{1}{\text{arrival rate}} \times \left(\frac{(\rho^*)^S \times \rho^*/S}{S! \times (1 - \rho^*/S)^2} \right) \times \left(\frac{1}{\sum_{k=0}^{S-1} \frac{(\rho^*)^k}{k!} + \frac{(\rho^*)^S}{S! \times (1 - \rho^*/S)}} \right) \\
&= \frac{1}{\text{service rate}} \times \left(\frac{1}{S - \rho^*} - \frac{1}{S - \rho^* + \frac{1}{(S-1)! \sum_{k=0}^{S-1} \frac{(\rho^*)^{k-S}}{k!}}} \right) \\
&= \frac{2.13x^*}{V_{UAV}} \times \left(\frac{1}{S - \rho^*} - \frac{1}{S - \rho^* + \frac{1}{(S-1)! \sum_{k=0}^{S-1} \frac{(\rho^*)^{k-S}}{k!}}} \right)
\end{aligned} \tag{16}$$

Then Eq. (12) can be rewritten as:

$$\begin{aligned}
Min \quad S \times \alpha_1 \times c_2 + \alpha_2 \times \frac{2.13x^*}{V_{UAV}} \times \\
\left(\frac{1}{S - \rho^*} - \frac{1}{S - \rho^* + \frac{1}{(S-1)! \sum_{k=0}^{S-1} \frac{(\rho^*)^{k-S}}{k!}}} \right)
\end{aligned} \tag{17}$$

Although Eq. (17) only contains one variable S , it is difficult to obtain the optimal value through derivative, because the objective function includes the factorial calculation of S . Therefore, we propose a heuristic Alg. 1 to search for the optimal S^* . Due to the strict monotonic decrease of EW_{ait} as S increases, we traverse from the lower bound LB to the upper bound UB to find the optimal number of drones S^* , so that Eq. (17) is minimized (lines 15-22 in Alg. 1). The crucial issue is determining the appropriate upper and lower bounds to shorten search time and ensure solution quality:

4.2.1. The lower bound

According to queuing theory, the number of multiple servers must be greater than the service intensity, otherwise the waiting time will increase infinitely. Therefore, the lower bound of UAV number is determined by rounding up the service intensity (line 1 in Alg. 1).

4.2.2. The upper bound

The upper bound is obtained by binary search within the range of lower bound UB and the maximum UAV number Max_Num , which satisfies the following two rules (lines 2-14 in Alg. 1):

- **Rule 1:** Even if an additional UAV is added, the objective value (in Eq. (17)) cannot be reduced (line 5).
- **Rule 2:** Even if an UAV is reduced, the objective value cannot be reduced (line 6).

These two rules are proposed because the average waiting time decreases as the UAV deployment cost (proportional to UAV number) increases. For Rule 1, there is no need to increase UAV number when the cost of increasing one UAV is more than the average waiting time cost reduction. For Rule 2, if reducing the UAV number leads to an excessive increase in average waiting time, there is also no need for reduction².

²If it is not possible to find an upper bound that satisfies both two rules, we only consider the upper bound determination under Rule 1.

Algorithm 1 UAV Number Dtermination.

```
1:  $LB \leftarrow \lceil \rho^* \rceil$ 
2:  $high \leftarrow Max\_Num, low \leftarrow LB$ 
3: while( $k > LB$ )do
4:    $k \leftarrow \lceil \frac{high+low}{2} \rceil$ 
5:   if( $\alpha_2 \times Function(k) \leq \alpha_1 \times c_2$ ) and
6:     ( $\alpha_2 \times (Function(k-1) - Function(k)) \geq \alpha_1 \times c_2$ ) then
7:      $UB \leftarrow k$ 
8:     break
9:   else if( $\alpha_2 \times Function(k) > \alpha_1 \times c_2$ ) then
10:     $low \leftarrow k$ 
11:   else
12:     $high \leftarrow k$ 
13:   end if
14: end while
15:  $M\_Value \leftarrow MAX\_Integer$ 
16: for( $k = LB; k \leq UB; k++$ )do
17:   if( $k \times \alpha_1 \times c_2 + \alpha_2 \times Function(k) \leq M\_Value$ ) then
18:      $M\_Value \leftarrow k \times \alpha_1 \times c_2 + \alpha_2 \times Function(k)$ 
19:      $S^* \leftarrow k$ 
20:   end if
21: end for
22: return  $S^*$ 
```

5. UAV Delivery Scheduling

This section mainly discusses the optimal UAV, dock and spatial conflict avoidance strategy selection, to reduce the service cost of each delivery request. The relevant variables are shown in Table 3.

5.1. Determination of optimal UAV

Suppose DC receives a real-time delivery request, and the request information includes the location of demand merchant as well as customer, denoted as Loc_m/Loc_c . The DC stores the available time of each UAV U_i (denoted as t_i^{ava}), as well as the final arrival dock location after completing all allocated delivery requests (denoted as Loc_i^{last}). It is worth noting that

Table 3: Table of Notations in Section 5

Symbol	Description
Loc_m/Loc_c	location of merchant/customer
Loc_*^{last}	the last location of UAV
t_*^{ava}	the available time of UAV
DIS	straight-line distance between two locations
p_j	selection probability of dock j
τ	pheromone amount on flight edge
η	heuristic information on flight edge
β/γ	pheromone amount/heuristic information weight
H_{sum}	fixed total amount of pheromones
H_j	deposited pheromone on flight edge
φ	pheromone update weight
t_*^{dep}	UAV departure time from dock
t_*^{arr}	UAV arrival time at dock

the UAV always fly from the last location to the selected optimal pick-up dock, and then fly from the pick-up dock to the selected optimal drop-off dock.

5.1.1. Waiting time estimation

Once the request is received, the DC will estimate the waiting time for each UAV to start delivering this request (lines 4-7 in Alg. 2). If the UAV available time (t_i^{ava}) is later than the time of receiving the request (*Current Time*), the request will need to wait for time ($t_i^{ava} - \text{Current Time}$) before starting the delivery. Otherwise, when the available time is earlier than the current time, the request does not need to wait and will be immediately delivered by the UAV.

5.1.2. Flight time estimation

Besides, the flight time of UAV to merchant is also calculated, where DIS calculates the straight-line distance between two locations (line 3). This is because the UAV closer to merchants also fly closer to drop-off docks, as the drop-off dock is preferred as nearby the merchant to reduce walking time.

Then, the DC will select the earliest available and closest UAV to merchant as the optimal UAV, to reduce the waiting time and parcel pick-up time for delivery request (lines 9-11).

When allocating request to the optimal UAV, the last location and available time of U^* also need to be updated (denoted as Loc_*^{last} and t_*^{ava}), so that DC can select the optimal UAV for subsequent delivery request (lines 14-15 in Alg. 2).

Algorithm 2 Optimal UAV Selection.

```

1:  $Mini\_Value \leftarrow MAX\_Integer$ 
2: for ( $\forall U_i \in \Omega_{UAV}$ ) do
3:    $T_i \leftarrow \frac{DIS(Loc_m, Loc_i^{last})}{V_{UAV}}$ 
4:   if ( $Current\ Time < t_i^{ava}$ ) then
5:      $W_i \leftarrow t_i^{ava} - Current\ Time$ 
6:   else
7:      $W_i \leftarrow 0$ 
8:   end if
9:   if ( $T_i + W_i < Mini\_Value$ ) then
10:     $Mini\_Value \leftarrow T_i + W_i$ 
11:     $U^* \leftarrow U_i$ 
12:   end if
13: end for
14: select the optimal dock and strategy according to Alg.3/4
15: update  $Loc_*^{last}$  and  $t_*^{ava}$  in Alg. 4
16: return  $U^*$ 

```

5.2. Determination of optimal Dock

In this section, we consider the optimal dock selection under scenarios **without spatial conflicts**³. For each delivery request, the DC needs to choose two optimal docks: one for UAV to pick up parcel and the other to drop off parcel (two docks cannot be the same). A common approach is to traverse the dock set to find two docks that minimize the cost of UAV flight time and merchant/customer walking time. However, such method only considers the current request, while does not take into account the comprehensive

³The minimum time difference for UAV to arrive/depart from the same dock without spatial conflicts can be seen as zero.

optimization of future requests. To reduce the average cost of all requests, we apply Ant Colony Algorithm (Dorigo, & Blum, 2005) to determine the optimal dock. The Ant colony optimization is a well known probabilistic swarm intelligence optimization technique, and has been successful in dealing with problems of finding paths through graphs (Liu, Naoum-Sawaya, Gu, LecueF, & Shorten, 2019).

Algorithm 3 Optimal Dock Selection.

```

1: Input:  $Loc_*^{last}$ ,  $Loc_m$ ,  $\Omega_{dock}$ 
2: for ( $\forall D_j \in \Omega_{dock}$ ) do
3:   calculate  $p_j$  based on Eq. (18) and Eq. (19)
4: end for
5: select  $D^*$  based on  $p_j$ 
6: Output:  $D^*$ 

```

5.2.1. Selection of optimal pick-up dock

Consider Ant Colony Algorithm to select the optimal dock for picking up parcel in Alg. 3, and there are three input parameters: the final arrival dock location of selected UAV (Loc_*^{last}), location of merchant (Loc_m), and candidate dock set (Ω_{dock}). For each dock D_j in dock set, the selection probability p_j is calculated according to following Eq. (18):

$$p_j = \frac{[\tau_j]^\beta \times [\eta_j]^\gamma}{\sum_{k \in \Omega_{dock}} [\tau_k]^\beta \times [\eta_k]^\gamma} \quad (18)$$

where τ_j is the pheromone amount on edge $\langle Loc_*^{last}, D_j \rangle$, and β is the pheromone amount weight. η_j is the heuristic information on edge $\langle Loc_*^{last}, D_j \rangle$, and γ is the heuristic information weight. The heuristic information is inversely proportional to the flight time over edge $\langle Loc_*^{last}, D_j \rangle$ (denoted as $F_{Loc_*^{last},j}$) and walking time of merchant to dock D_j (denoted as K_j):

$$\eta_j = \frac{1}{F_{Loc_*^{last},j} + K_j} \quad (19)$$

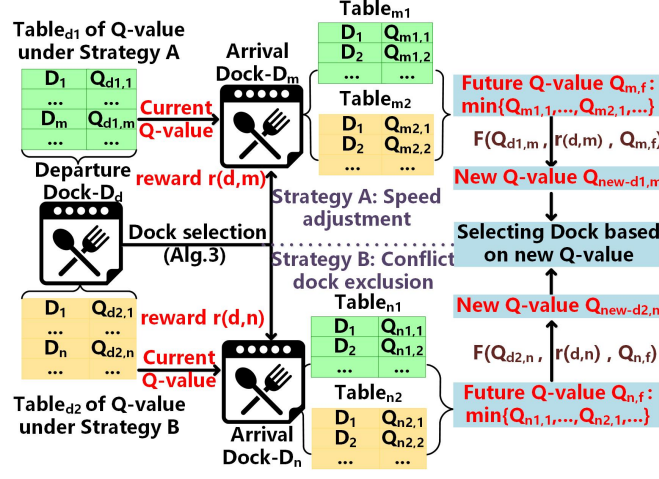


Figure 5: Dock selection under spatial conflict avoidance

once UAV chooses to fly over edge $\langle Loc_*^{last}, D_j \rangle$, the deposited pheromone by UAV on flight edge is calculated as follows:

$$H_j = \frac{H_{sum}}{DIS(Loc_*^{last}, Loc_j) + DIS(Loc_j, Loc_m)} \quad (20)$$

where H_{sum} is the fixed total amount of pheromones, and deposited pheromone is inversely proportional to flight distance and walking distance. After the flight, the pheromone amount on edge $\langle Loc_*^{last}, D_j \rangle$ is updated according to following Eq. (21) (φ is the update weight):

$$\tau_j = \varphi \times \tau_j + (1 - \varphi) \times H_j \quad (21)$$

5.2.2. Selection of optimal drop-off dock

After selecting the optimal dock for picking up parcel, our Alg. 3 can also be applied to select the other optimal dock for dropping off parcel by changing the input parameter value. When selecting the optimal drop-off dock, the final location of UAV becomes the selected pick-up dock location, the merchant location becomes customer location, and the candidate dock set needs to exclude the selected pick-up dock.

5.3. Determination of spatial conflict avoidance strategy

To avoid spatial conflicts, each dock has an arrival list denoted as *ArrList*, to store the expected arrival or departure time of UAV. The arrival list is sorted in descending order, so *ArrList.get(0)* is the latest arrival/departure time of UAV at the dock. As shown in Fig. 5, the UAV departs from the last location dock D_d and arrives at the optimal dock (D_m or D_n). To avoid spatial conflict, the departure time from D_d and arrival time at D_m or D_n should be both considered. Since Alg. 3 for selecting the optimal pick-up dock and drop-off dock is common except for input parameter values, the conflict avoidance scheme is also homogeneous no matter D_m/D_n is pick-up docks or drop-off docks. Therefore, we only explain the scheme when D_m/D_n is pick-up docks.

5.3.1. Spatial conflict on departure dock

The departure dock D_d is also the last location of UAV Loc_*^{last} , because the UAV will immediately deliver the subsequent request after completing the previous delivery (line 1 in Alg. 4). When there is no spatial conflict, the departure time of UAV from D_d , denoted as t_*^{dep} , is determined by UAV available time t_*^{ava} : if the available time of UAV is later than current time, the departure time will be UAV available time. Otherwise, the UAV can depart from D_d at current time (lines 2-6 in Alg. 4).

However, if there are spatial conflicts on dock D_d (lines 7-9 in Alg. 4), the UAV needs to delay the departure time to $D_d.ArrList.get(0) + \delta$, so that the time delay between the departure time (t_*^{dep}) and the latest departure/arrival time of other UAV ($D_d.ArrList.get(0)$) is greater than the minimum time difference δ . Then, the departure dock adds the expected departure time of UAV to $D_d.ArrList$, to avoid future spatial conflicts (lines 10-11 in Alg. 4).

5.3.2. Spatial conflict on arrival dock

We consider two strategies to avoid spatial conflicts on arrival dock: Strategy A adjusts UAV speed to delay the arrival time, while Strategy B will not select the conflict dock. This means that the optimal arrival dock chosen under different strategies may vary. As shown in Fig. 5, suppose the optimal dock selected under Strategy A is D_m , and the optimal dock selected under Strategy B is D_n (lines 12-13 in Alg. 4). Then, the optimal strategy is selected based on Q-value learning (Watkins, & Dayan, 1992).

When selecting the optimal dock D_m and D_n , the negative reward is calculated as the summation of UAV flight time and merchant walking time:

$$\begin{aligned} r(d, m) &= \frac{DIS(Loc_m, D_m)}{V_{user}} + \frac{DIS(D_d, D_m)}{V_{UAV}}, \\ r(d, n) &= \frac{DIS(Loc_m, D_n)}{V_{user}} + \frac{DIS(D_d, D_n)}{V_{UAV}} \end{aligned} \quad (22)$$

the Q-value of future state is estimated as the minimum value in Q-tables of D_m and D_n (each dock has two Q-tables to store the corresponding Q-value for selecting the next dock under different strategies):

$$\begin{aligned} Q_{m,f} &= \min\{Q_{m1,1}, \dots, Q_{m2,1}, \dots\}, \\ Q_{n,f} &= \min\{Q_{n1,1}, \dots, Q_{n2,1}, \dots\} \end{aligned} \quad (23)$$

$Q_{m1,1}/Q_{n1,1}$ is the Q-value when departing from D_m/D_n and select D_1 under Strategy A, and $Q_{m2,1}/Q_{n2,1}$ is the Q-value when departing from D_m/D_n and select D_1 under Strategy B. Based on negative reward and Q-value of future state, the new Q-value when departing from D_d and select D_m or D_n is calculated according to following Eq. (24) (line 14 in Alg. 4):

$$\begin{aligned} Q_{new-d1,m} &\leftarrow Q_{d1,m} + \lambda \times (r(d, m) + \mu \times Q_{m,f} - Q_{d1,m}), \\ Q_{new-d2,n} &\leftarrow Q_{d2,n} + \lambda \times (r(d, n) + \mu \times Q_{n,f} - Q_{d2,n}) \end{aligned} \quad (24)$$

where λ is the learning rate and μ represents the discount value implying the effect of the current selection on the future.

The dock and corresponding strategy with lower new Q-value are selected, to reduce UAV flight time and merchant walking time (lines 15-27 in Alg. 4):

- **If Strategy A is selected:** The optimal dock for UAV next arrival D^* will be D_m (line 16), and the corresponding Q-value should be updated (line 17). Under the scenario of no spatial conflict on D_m , the arrival time at D_m (denoted as t_*^{arr}) will be departure time t_*^{dep} plus flight time between departing dock D_d and arrival dock D_m denoted as $F_{d,m}$ (line 18).

However, if there are spatial conflicts on D_m (line 19), the UAV needs to adjust the speed (line 21). With the adjusted speed, the arrival time will delay to $D_m.ArrList.get(0) + \delta$, so that the time delay between the

arrival time (t_*^{arr}) and the latest departure/arrival time of other UAV ($D_m.ArrList.get(0)$) is greater than the minimum time difference (line 20).

- **Otherwise Strategy B is selected:** The optimal dock for UAV next arrival D^* will be D_n (line 24), and the corresponding Q-value should be updated (line 25). Since Strategy B only selects the dock without spatial conflict, the arrival time at D_n will be departure time t_*^{dep} plus flight time between departing dock D_d and arrival dock D_n (line 26).

Once determining the optimal dock for UAV next arrival D^* and arrival time t_*^{arr} , the optimal dock will add the expected arrival time of UAV to $D^*.ArrList$, to avoid future spatial conflicts (lines 28-29 in Alg. 4). Besides, the last location of UAV will be updated to D^* (line 31), and the available time of UAV will be updated to arrival time at D^* (in line 30).

Algorithm 4 Optimal Avoidance Strategy Selection.

```

1:  $D_d \leftarrow Loc_*^{last}$ 
2: if ( $Current\ Time < t_*^{ava}$ ) then
3:    $t_*^{dep} \leftarrow t_*^{ava}$ 
4: else
5:    $t_*^{dep} \leftarrow Current\ Time$ 
6: end if
7: if ( $|D_d.ArrList.get(0) - t_*^{dep}| \leq \delta$ ) then
8:    $t_*^{dep} \leftarrow D_d.ArrList.get(0) + \delta$ 
9: end if
10:  $D_d.ArrList.add(t_*^{dep})$ 
11: sort  $D_d.ArrList$  with descending order
12:  $D_m \leftarrow \text{Alg. 3}(D_d, Loc_m, \Omega_{dock})$ 
13:  $D_n \leftarrow \text{Alg. 3}(D_d, Loc_m, \Omega_{dock} \setminus \{conflict\ docks\})$ 
14: calculate  $Q_{new-d1,m}$  and  $Q_{new-d2,n}$  based on Eq. (24)
15: if ( $Q_{new-d1,m} \leq Q_{new-d2,n}$ ) then
16:    $D^* \leftarrow D_m$ 
17:    $Q_{d1,m} \leftarrow Q_{new-d1,m}$ 
18:    $t_*^{arr} \leftarrow t_*^{dep} + F_{d,m}$ 
19:   if ( $|D_m.ArrList.get(0) - t_*^{arr}| \leq \delta$ ) then
20:      $t_*^{arr} \leftarrow D_m.ArrList.get(0) + \delta$ 
21:      $V_{UAV} \leftarrow \frac{DIS(D_d, D_m)}{t_*^{arr} - t_*^{dep}}$ 
22:   end if
23: else
24:    $D^* \leftarrow D_n$ 
25:    $Q_{d2,n} \leftarrow Q_{new-d2,n}$ 
26:    $t_*^{arr} \leftarrow t_*^{dep} + F_{d,n}$ 
27: end if
28:  $D^*.ArrList.add(t_*^{arr})$ 
29: sort  $D^*.ArrList$  with descending order
30:  $t_*^{ava} \leftarrow t_*^{arr}$ 
31:  $Loc_*^{last} \leftarrow D^*$ 
32: return  $D^*$ 

```

6. CASE STUDY

The Opportunistic Network Environment (ONE) (Keränen, Ott, & Kärkkäinen, 2009) is used to implement a case study. In the default scenario, the street map is from the downtown area of Helsinki City with $4500 \times 3400 \text{ m}^2$ area in Finland shown in Fig. 6. We initialize 4 DCs through K-means clustering algorithm (Hartigan, & Wong, 1979), with square of each delivery area approximate to $4500 \times 3400 / 4 \text{ m}^2$. The default flight speed of UAV is 10.8 km/h , and the walking speed of merchant/customer is 1.5 m/s . The UAV departure location is initialized as DC, and then the UAV flies between docks. The minimum time difference for UAV to arrive/depart from the same dock without spatial conflicts is set as 180s.

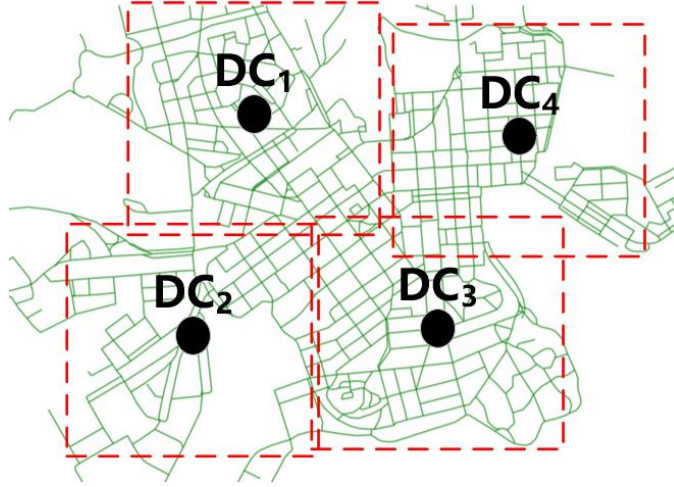


Figure 6: Simulation scenario of Helsinki City

The location of merchant/customer of each delivery request is randomly selected from a location on the map, and the delivery demand follows a Poisson distribution with an arrival rate of $1/80$. The average cost of deploying a dock is 100s, and the cost of deploying a UAV is 600s, where value of time is utilized to convert deployment cost unit (money) into time unit (second). It is worth noting that the service cost is estimated as the average cost of completing one delivery, so the system (deployment) cost is also set as the average value per task. The weight value α_1 and α_2 in Eq. (6) are both 0.5, with the initial upper limit of Alg. 1 set as 100. Besides, The weight

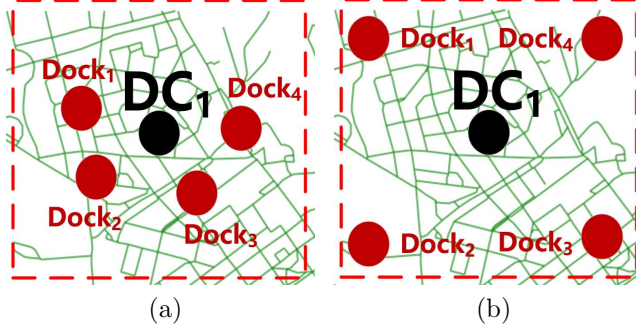


Figure 7: Dock geographical distribution: (a) centralized scenario (b) distributed scenario.

Table 4: TABLE OF Evaluation Attributions

Metrics (second unit)	Definitions
Deployment Cost	average cost for DC to deploy docks/UAVs
Average Waiting Time	average waiting time for UAV to start delivering
Average Walking Time	average walking time to docks
Average Flight Time	average flight time between docks

of pheromone amount and heuristic information (β/γ) is 3 and 7, with the pheromone update weight φ set as 0.5.

Although our scheme determines the optimal number of docks, the geographical distribution of docks is not considered. Therefore, we discuss two different scenarios of dock geographical distribution to evaluate performance shown in Fig. 7. Under the centralized scenario in Fig. 7 (a), the docks are centrally deployed and close to DC. While under the distributed scenario in Fig. 7 (b), the docks are distributed deployed and far from DC. The performance evaluation is the average result of two scenarios, and each experiment will be repeated five times. The evaluation attributes are shown in Table 4 (all units are second).

6.1. Influence of dock number

Based on our scheme, the optimal dock number is 4 docks at each delivery area. We compare the performance under 3 docks, 4 docks (Optimal), and 5 docks, where the number of UAVs are all fixed as 3. As shown in Fig. 8 (a), the average waiting time tends to zero under different numbers of docks, so only 3 docks and 3 UAVs are needed to meet the delivery demand. However, the average walking time is increased with the decrease in dock number in

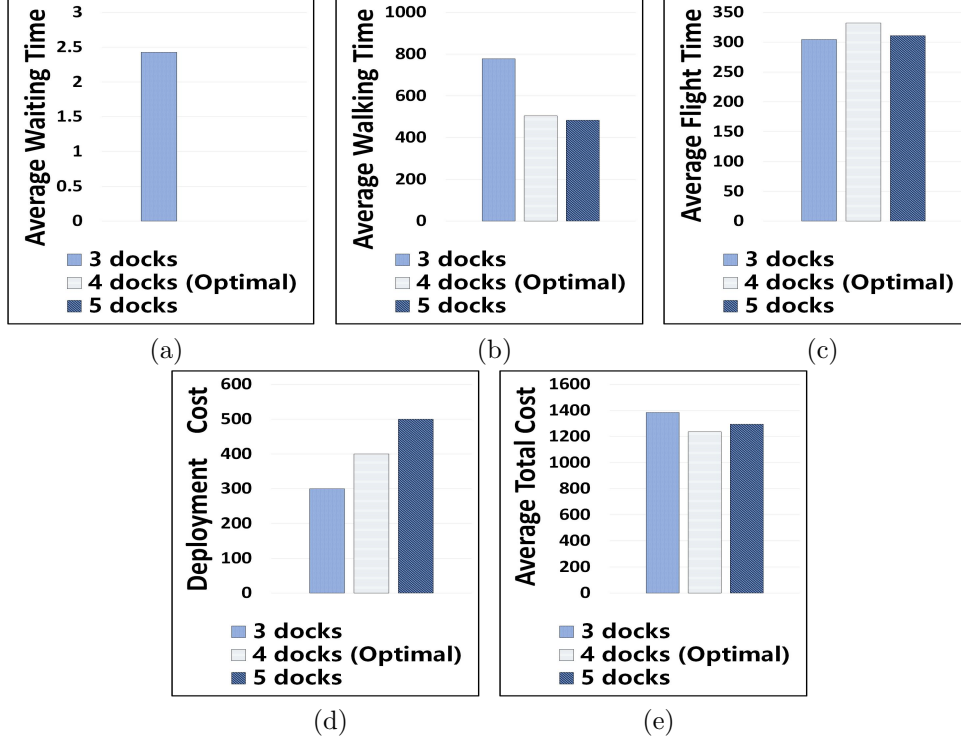


Figure 8: Influence of Dock Number: (a) Average Waiting Time (b) Average Walking Time (c) Average Flight Time (d) Deployment Cost (e) Average Total Cost (the summation of (a), (b), (c) and (d)).

Fig. 8 (b), which is consistent with the conclusion in Eq. (8). When the number of docks decreases from 4 to 3, the average walking time is increased by 54%.

Even though, when the number of docks increases from 4 to 5, the average walking time is just decreased by 4.26%. This means that the number of docks obtains a marginal effect on reducing average walking time, so adding more docks may be difficult to further reduce average service cost.

As shown in Fig. 8 (d), the deployment cost increases with the increase in dock number. Even if the average flight time increases under the optimal dock number in Fig. 8 (c), the average total cost is the minimum compared to 3 and 5 docks in Fig. 8 (e). This indicates that our scheme can achieve a balance between minimizing deployment cost and minimizing service cost, thereby reducing overall delivery cost.

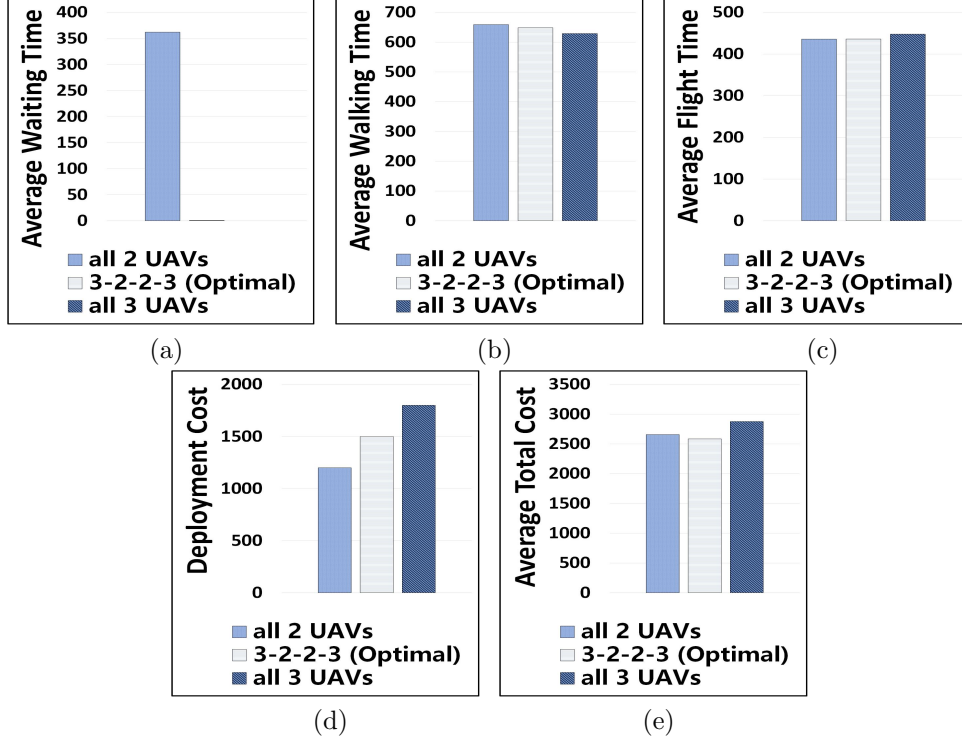


Figure 9: Influence of UAV Number: (a) Average Waiting Time (b) Average Walking Time (c) Average Flight Time (d) Deployment Cost (e) Average Total Cost (the summation of (a), (b), (c) and (d)).

6.2. Influence of UAV number

Through our Alg. 1, the optimal UAV number is determined as 3-2-2-3 at four delivery areas from DC_1 to DC_4 . We compare the performance under all 3 UAVs, all 2 UAVs and 3-2-2-3 UAVs (Optimal) deployed at DC_1 to DC_4 , where the dock number are all fixed as 4. As shown in Fig. 9 (a), the average waiting time is highly increased under all 2 UAVs due to reduction of available UAVs. Even though, compared with all 3 UAVs, the optimal UAV number 3-2-2-3 can also reduce the average waiting time to zero, with two UAVs reduced in total number. This is because our Alg. 1 can determine the optimal UAV number for each delivery area, rather than directly applying the homogeneous UAV number.

Besides, the average walking time in Fig. 9 (b) and the average flight time in Fig. 9 (c) is similar under different UAV numbers, because the average walking/flight time is mainly influenced by the dock number and

Table 5: TABLE OF Performance Gap under Different Dock Geographical Distribution

Gap (%)	Average Waiting	Average Walking	Average Flight	Average Total Cost
All 3	0	56.34	57.01	18.03
All 2	699.231705	53.92	64.29	59.36
3-2-2-3	145.94	57.13599351	62.48146461	21.31555956

geographical distribution. The deployment cost is also increased with the increase in UAV number in Fig. 9 (d). In overall, the average total cost is the minimum under the optimal UAV number compared to all 2 and 3 UAVs in Fig. 9 (e), because our Alg. 1 can determine the minimum UAV number that can reduce the average waiting time to zero.

We also evaluate the performance gap of UAV number under different dock geographical distribution scenarios in Fig. 7, and the gap is calculated as $(Worst - Best) / Worst * 100\%$. The results are shown in Table 5, where the key observation is that the increase in UAV number can reduce the impact of dock geographical distribution. When the UAV number decreases from all 3 to all 2, the average total cost gap increases by nearly three times (18% to 59%). Even though, our optimal solution only increases the performance gap of average total cost by 3%. In practice, the DC can set up redundant UAVs based on the optimal solution of UAV number, to meet the time constraints of delivery demand under distributed dock geographical distribution.

6.3. Influence of Dock Selection

This section evaluates the performance of dock selection without spatial conflicts. Under 3-2-2-3 UAVs, we compare our Ant Colony Algorithm (Ant-10⁴) with traversal algorithm (Tra-10), as well as the heuristic algorithm which optimizes the scheduling of UAV tasks based on greedy insertion and neighborhood search (Heu-10) in work (Sun et al., 2024). Since Heu-10 does not consider the dock selection, the optimal dock is selected based on our Alg. 2, while the last location of UAV is always DC. Moreover, we evaluate the performance of Ant Colony and traversal algorithms when the UAV number (3-2-2-2) is less than the optimal number, defined as Ant-9 and Tra-9. The deployment cost of Tra-10/Ant-10/Heu-10 are all 1500s, while the deployment cost of Tra-9/Ant-9 are both 1350s. We do not display the

⁴The number 10 behind the scheme denotes the number of deployed UAVs as 3-2-2-3, and the number 9 denotes the number of deployed UAVs as 3-2-2-2.

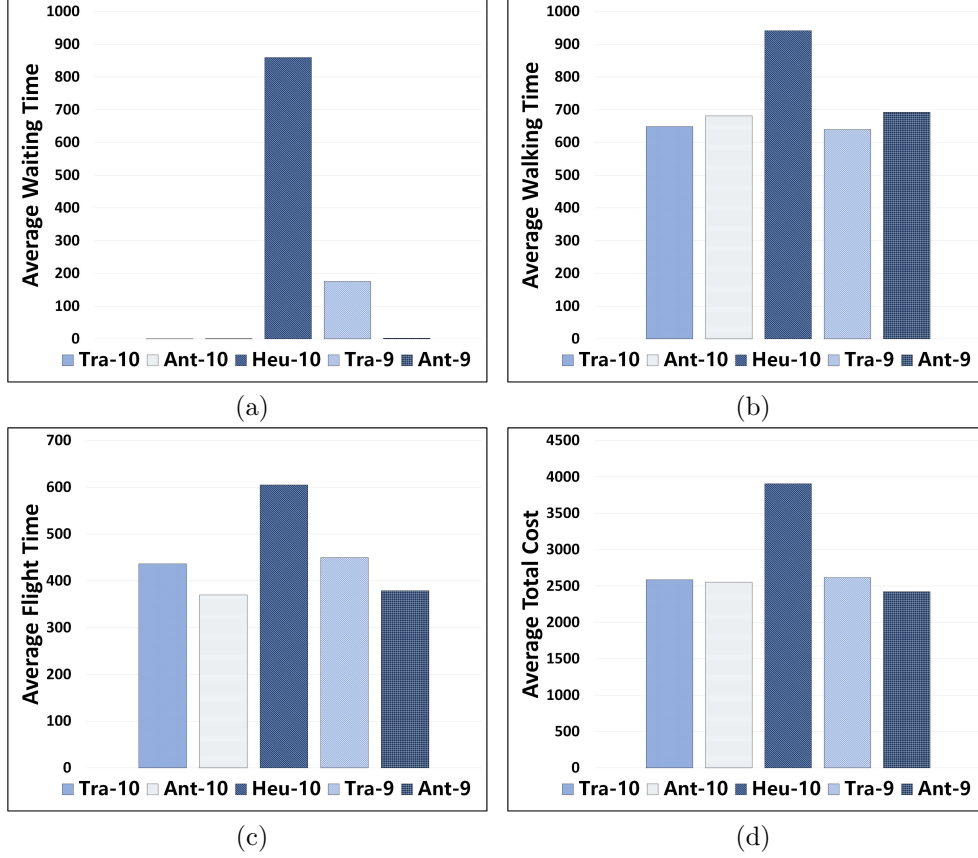


Figure 10: Influence of Dock Selection: (a) Average Waiting Time (b) Average Walking Time (c) Average Flight Time (d) Average Total Cost (the summation of (a), (b), (c) and deployment cost).

deployment cost in Fig. 10 due to limited space.

As shown in Fig. 10, the Heu-10 performs the worst in all evaluation attributes. This is because Heu-10 does not consider the impact of UAV location changes on the optimal dock selection. The optimal dock for the subsequent delivery depends on UAV location when completing the previous task, however, under Heu-10 the optimal dock is independent of UAV location. This highly increases the average cost of Heu-10.

Under 10 UAVs, the average waiting time of Ant-10 and Tra-10 are both zero in Fig 10 (a). When the number of UAVs decreases from 10 to 9, the average waiting time of Ant Colony Algorithm under Ant-9 still approaches to zero, while the average waiting time of traversal algorithm under Tra-9

Table 6: TABLE OF Performance Gap under Different Dock Geographical Distribution

Gap (%)	Average Waiting	Average Walking	Average Flight	Average Total Cost
Tra-10	145.94	57.13599351	62.48146461	21.31555956
Ant-10	316.04	41.3542087	76.2584687	18.929834
Heu-10	1640.717598	88.77335953	116.0057264	149.2883511
Tra-9	351.2	66.02	62.99	40.72
Ant-9	70.13052959	16.68918837	23.32997082	8.523939104

increases to 175s. This is because our Ant-10/Ant-9 can reduce the average flight time of UAV shown in Fig 10 (c), thus enabling UAV to become available faster. Although the average walking time of Ant-10/Ant-9 is increased in Fig 10 (b), the average flight time is decreased, and the total cost is the minimum compared to traversal algorithm and heuristic algorithm shown in Fig 10 (d). In addition, due to the reduction of flight time, the Ant Colony Algorithm can maintain waiting time around zero with the UAV number reduction, thus further decreasing the total cost from Ant-10 to Ant-9.

In particular, the performance gap of schemes under different dock geographical distributions are shown in Table 6. It is obvious that the performance gap of Heu-10 is the largest, because Heu-10 does not take into account the final location of UAV. Under our algorithm Ant-10/Ant-9, the optimal dock and UAV is selected based on a first-come-first-served principle. However, for Heu-10, the optimal docks are always selected for all requests in the decision pool before selecting the optimal UAVs. This means that the optimal dock selection for the subsequent request does not consider the connection with the previous task, thus increasing the flight time gap.

6.4. Influence of Spatial Conflict

The Ant Colony scheme without spatial conflicts (denoted as PureAnt) is used as the benchmark, to analyze the impact of spatial conflicts. Besides, to evaluate the performance of spatial conflict avoidance strategies, we compare the schemes that uses only Strategy A (denoted as SpeedA), Strategy B (denoted as AvoidB), and Q-value selection strategy (denoted as Q-valueAB). The Strategy A and Strategy B are both detailed in Section III.C, while Q-value selection strategy is proposed in Section V.C. The experiment results under scenario of Fig. 7 (a) are shown in Fig. 11.

As shown in Fig. 11 (a), the average waiting time of SpeedA, AvoidB and Q-valueAB are all increased. This is because that to avoid spatial conflicts, the UAV must decrease flight speed or delay the takeoff time, thus affecting

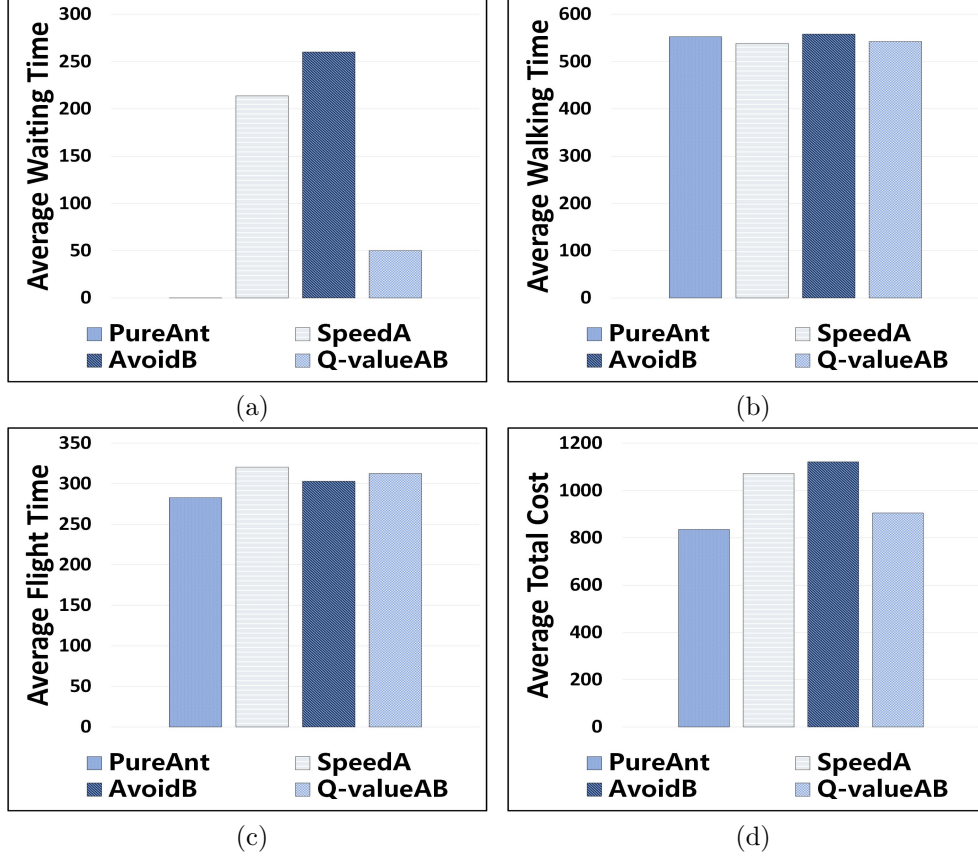


Figure 11: Influence of Spatial Conflict: (a) Average Waiting Time (b) Average Walking Time (c) Average Flight Time (d) Average Total Cost (the summation of (a), (b) and (c)).

UAV availability. Even so, the average waiting time under Q-valueAB is the lowest compared with SpeedA and AvoidB, as Q-valueAB will choose the optimal avoidance strategy based on current negative reward and future state.

Since SpeedA reduces the flight speed to avoid conflicts, the average flight time under SpeedA is the highest in Fig. 11 (c). However, for AvoidB, the average walking time is the highest shown in Fig. 11 (b), because AvoidB does not choose the conflict docks which may be the closest to merchant/customer. As a combination of SpeedA and AvoidB, Q-valueAB avoids spatial conflicts without significantly increasing the average walking and flight time, thus resulting in the lowest average total cost among the three strategies shown in Fig. 11 (d).

Table 7: TABLE OF Performance Gap of Solutions

Gap (%)	Average Waiting	Average Walking	Average Flight	Average Total Cost
SpeedA	90.03247742	1.374515143	7.848439587	14.99355237
AvoidB	230.0994638	8.116908967	0.586584093	27.41469552
Q-valueAB	87.00227236	5.467958375	5.627974983	9.084048109

We also evaluate the gap between the best and worst performance of these three avoidance strategies in repeated experiments. The results in Table 7 show that SpeedA obtains the highest performance gap in average flight time, while AvoidB has the highest performance gap in average walking time. This is consistent with the conclusion in Fig. 11: The highest performance gap in average flight time under Strategy A is caused by speed reduction, while conflicting dock avoidance leads to the highest performance gap in average walking time under Strategy B.

For Q-valueAB, its performance gap in average waiting time is the lowest, and it is stable in other performance gaps. Q-valueAB not only combines the advantages of SpeedA and AvoidB, but also considers both current negative reward and future state when choosing the strategy through Q-learning (Wang, 2020). This leads to a stable performance in avoiding spatial conflicts and reducing service cost.

6.5. Influence of External Disturbance

In this experiment, 50% of the docks are disturbed every 0.5 to 1 hour, and UAVs that choose to take off/land at that dock need to wait for 180s. The simulation results under 9 UAVs and 4 docks are shown in Table 8 (the unit is second).

It is obvious that Q-valueAB outperforms other strategies on reducing average waiting time, average driving time and average flight time. As the dock disturbance requires the UAV to wait for a time period before taking off/landing, the average waiting time of the three schemes all increase compared to no disturbance case. Furthermore, since AvoidB does not select disturbed docks, the average walking time increases the most. While for SpeedA, the average flight time is the highest because it avoids disturbances by reducing flight speed. Even though, the average walking time of Q-valueAB is similar to PureAnt under no disturbance case, and Q-valueAB obtains the lowest average total cost compared with SpeedA/AvoidB. This is because Q-valueAB selects the optimal dock based on historical record

Table 8: TABLE OF Performance Comparisons under External Disturbances

Strategy	Average Waiting	Average Walking	Average Flight	Average Total Cost
PureAnt (no disturbance)	0.0827	552.3958	282.82735	835.30585
SpeedA	273.7463	558.0574	350.4747	1182.278
AvoidB	330.0562	587.8433	341.1621	1259.062
Q-valueAB	69.99365	552.3223	330.6341	952.9501

values and current reward values. When the previous disturbance causes an increase in UAV costs, the Q-valueAB will record the historical cost values in the Q-table, to reduce the future selection of disturbed docks.

6.6. Static Numerical Study

In this section, the dock and demand locations are randomly generated in a matrix of $1000m * 1000m$. The number of docks is fixed at 4, and the UAV number is fixed at 3. Our scheme and heuristic scheme in work (Sun et al., 2024) are achieved in Matlab, while Gurobi in Python is utilized to obtain the optimal solution⁵. Since the waiting time is defined as the time difference between sending the request and starting delivering the request under our model, the waiting time is approximated estimated in Gurobi. This is because the waiting time cannot be calculated by a linear function, and the waiting time is approximated as the arrival time at the pickup dock minus the average flight time from other docks to that pickup dock. Each experiment is repeated 20 times, with an algorithm running time limit of 300 seconds. The objective value is the average delivery cost, which is calculated as the average of 20 experimental results. The performance gap evaluates the gap between the obtained solutions: $Gap = (the\ worst\ solution - the\ best\ solution) / (the\ worst\ solution) * 100\%$. The performance comparisons under different demand numbers without spatial conflicts are shown in Table 9.

It is obvious that our ant scheme is closer to the optimal solution obtained by Gurobi under 10 and 20 demand number compared to heuristic scheme. This is because our scheme can schedule UAVs in real-time, thereby reducing the waiting time cost of customers. When the demand number increases to 30, the ant scheme obtains a lower cost solution compared to Gurobi. This is because as the problem scale increases, it becomes difficult for the linear

⁵It is worth noting that if Gurobi cannot find the optimal solution, the sub-optimal solution stored in the solution pool will be outputted.

Table 9: TABLE OF Performance Comparisons under static scenario without spatial conflicts

Scheme	Ant.		Heu.		Gurobi	
Demand Number	Obj (s)	Gap (%)	Obj (s)	Gap (%)	Obj (s)	Gap (%)
10	893.5	31.4418	1206.76	52.3998	873.5284	-
20	1218.39	16.2046	1463.41	33.8043	1191	-
30	1285.8733	22.142	1646.1330	37.7481	1378.3468	6.2531

Table 10: TABLE OF Performance Comparisons under static scenario with spatial conflicts

Scheme	Ant+Q_value		Heu.		Gurobi	
Demand Number	Obj (s)	Gap (%)	Obj (s)	Gap (%)	Obj (s)	Gap (%)
10	1228.84	11.0759	1416.54	17.7373	1060.1	-
20	1650.76	16.7867	1947.63	23.0183	1361.5	-
30	2275.9067	10.352	2574.7943	14.1707	2317.1428	5.7423

solver to obtain the optimal solution within a limited computational time. In addition, the ant scheme outperforms on the optimal solution convergence, as the gap of ant scheme is lower than heuristic scheme under all demand number scenarios.

We also evaluate the performance considering spatial conflicts. As shown in Table 10, avoiding spatial conflicts increases the average delivery cost of all three schemes. When the demand number is 30, our solution can achieve the lowest average delivery cost with the help of the optimal conflict avoidance strategy selection. Another observation is that spatial conflicts reduce the performance gap between the optimal solutions of the three schemes. This may be because conflict avoidance limits the search range of the optimal solution, thereby accelerating the convergence of the solution.

6.7. Work Limitation Discussion

Although previous experimental results shows that our scheme outperforms other schemes on reducing delivery costs, there are still some work limitations that need to be discussed here. Firstly, our work does not consider the impact of UAV parcel payload and flight endurance. Secondly, although dock conflict avoidance is discussed, the collision avoidance in UAV flight routes is not taken into account. In future work, these limitations can be improved by using neural network models to capture demand and route features.

7. CONCLUSION

In this paper, we mainly propose a UAV delivery system considering both flight dock selection and spatial conflict avoidance. Based on the bi-level linear programming model, the optimal number of docks and UAVs deployed in delivery area is determined, to minimize both the system cost and service cost. In addition, the optimal dock is selected through Ant Colony Algorithm and Q-value learning, to reduce the service cost and avoid spatial conflict. Compared with other schemes, our proposed scheme can outperform on cost reduction as well as conflict avoidance, and is robust to different geographical distribution of docks.

References

- Bogyrbayeva, A., Yoon, T., Ko, H., Lim, S., Yun, H., & Kwon, C. (2023). A deep reinforcement learning approach for solving the traveling salesman problem with Drone, *Transportation Research Part C: Emerging Technologies*, 148, 103981. <https://doi.org/10.1016/j.trc.2022.103981>.
- Bhuiyan, T. H., Walker, V., Roni, M., & Ahmed, I. (2024). Aerial drone fleet deployment optimization with endogenous battery replacements for direct delivery of time-sensitive products. *Expert Systems with Applications*, 252, 124172. <https://doi.org/10.1016/j.eswa.2024.124172>.
- Daganzo, C. F. (1984). The length of tours in zones of different shapes. *Transportation Research Part B: Methodological*, 18(2), 135-145. [https://doi.org/10.1016/0191-2615\(84\)90027-4](https://doi.org/10.1016/0191-2615(84)90027-4).
- Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theor. Comput. Sci.*, 344(2-3), 43-278. <https://doi.org/10.1016/j.tcs.2005.05.020>.
- Gross, D., Shortle, J. F., Thompson, J.M., & Harris, C.M. (2008). *Fundamentals of Queueing*, Wiley-Interscience, New York, United States. <https://doi.org/10.1002/9781118625651>.
- Gao, K., Yang, Y., & Qu, X. B. (2021). Diverging effects of subjective prospect values of uncertain time and money. *Communications in Transportation Research*, 1, 100007. <https://doi.org/10.1016/j.commtr.2021.100007>.

- Hartigan, J. A., & Wong, M. A. (1979). A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 28(1), 100-108. <https://doi.org/10.2307/2346830>.
- Hu, X. B., Yang, C. S., Zhou, J., Zhang, Y. F., & Ma, Y. M. (2025). Research on 3D layered visibility graph route network model and multi-objective path planning for UAVs in complex urban environments. *Aerospace Science and Technology*, 159, 109947. <https://doi.org/10.1016/j.ast.2025.109947>.
- Hu, Z. C., Chen, H., Lyons, E., Solak, S., & Zink, M. (2024). Towards sustainable UAV operations: Balancing economic optimization with environmental and social considerations in path planning. *Transportation Research Part E: Logistics and Transportation Review*, 181, 103314. <https://doi.org/10.1016/j.tre.2023.103314>.
- Keränen, A., Ott, J., & Kärkkäinen, T. (2009). The ONE simulator for DTN protocol evaluation. 3rd ed. in *ICST SIMUTools 09*, Rome, Italy. <https://doi.org/10.4108/ICST.SIMUTOOLS2009.5674>.
- Khamidehi, B., Raeis, M., & Sousa, E.S. (2021). Dynamic Resource Management for Providing QoS in Drone Delivery Systems. 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), 3529-3536. <https://arxiv.org/abs/2103.04015>.
- Kim, J. S., Kweon, S. J., Hwang, S. W., & Lee, S. (2024). Crowdsourcing integration on the last mile delivery platform considering floating population data. *Expert Systems with Applications*, 248, 123312. <https://doi.org/10.1016/j.eswa.2024.123312>.
- Kutpanova, Z., Kadhim, M., Zheng, X., & Zhakiyev, N. (2025). Multi-UAV path planning for multiple emergency payloads delivery in natural disaster scenarios. *Journal of Electronic Science and Technology*, 100303. <https://doi.org/10.1016/j.jnlest.2025.100303>.
- Li, A., Hansen, M., & Zou, B. (2022). Traffic management and resource allocation for UAV-based parcel delivery in low-altitude urban space. *Transportation Research Part C: Emerging Technologies*, 143, 103808. <https://doi.org/10.1016/j.trc.2022.103808>.

- Liu, M., Naoum-Sawaya, J., Gu, Y., LecueF, L., & Shorten, R. (2019). A Distributed Markovian Parking Assist System. *IEEE Transactions on Intelligent Transportation Systems*, 20(6), 2230-2240. <https://doi.org/10.1109/TITS.2018.2865648>.
- Liu, Z. Y., Li, L. H., Zhang, X., Tang, W., Yang, Z., & Yang, X. M. (2025). Considering both energy effectiveness and flight safety in UAV trajectory planning for intelligent logistics. *Vehicular Communications*, 52, 100885. <https://doi.org/10.1016/j.vehcom.2025.100885>.
- Liu, Z. Y., Li, Y. Z., & Wu, Y. Q. (2023). Multiple UAV formations delivery task planning based on a distributed adaptive algorithm. *Journal of the Franklin Institute*, 360 (4), 3047-3076. <https://doi.org/10.1016/j.jfranklin.2023.01.008>.
- Lemardelé, C., Estrada, M., Pagès, L., & Bachofner, M. (2021). Potentialities of drones and ground autonomous delivery devices for last-mile logistics. *Transportation Research Part E: Logistics and Transportation Review*, 149, 102325. <https://doi.org/10.1016/j.tre.2021.102325>.
- Li, X. H., Yan, P. Y., Yu, K. Z., Li, P. F., & Liu, Y. C. (2024). Parcel consolidation approach and routing algorithm for last-mile delivery by unmanned aerial vehicles. *Expert Systems with Applications, Part E*, 238, 122149. <https://doi.org/10.1016/j.eswa.2023.122149>.
- Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54, 86-109. <https://doi.org/10.1016/j.trc.2015.03.005>.
- Mishra, D., & Tiwari, M. K. (2024). Integrated truck drone delivery services with an optimal charging stations. *Expert Systems with Applications*, 254, 124254. <https://doi.org/10.1016/j.eswa.2024.124254>.
- Pei, Z., Dai, X., Yuan, Y., Du, R., & Liu, C. (2021). Managing Price and Fleet Size for Courier Service with Shared Drones. *Omega*, 104(2), 102482. <https://doi.org/10.1016/j.omega.2021.102482>.
- Pei, Z., Fang, T., Weng, K., & Yi, W. (2023). Urban On-Demand Delivery via Autonomous Aerial Mobility: Formulation and Exact Algorithm. *IEEE*

- Transactions on Automation Science and Engineering, 20(3), 1675-1689.
<https://doi.org/10.1109/TASE.2022.3184324>.
- Raissa, M. S. G., Laurent, D., Dominique, F., Nathalie, G., & Alain, Q. (2018). An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks*, 72(4), 459-474.
<https://doi.org/10.1002/net.21846>.
- Sajid, M., Mittal, H., Pare, S., & Prasad, M. (2022). Routing and scheduling optimization for UAV assisted delivery system: A hybrid approach. *Applied Soft Computing*, 126, 109225.
<https://doi.org/10.1016/j.asoc.2022.109225>.
- Shao, S. J., Li, Y., Guo, S. Y., Wang, C. H., Chen, X. Y., & Qiu, X. S. (2023). Delay and Energy Consumption Oriented UAV Inspection Business Collaboration Computing Mechanism in Edge Computing Based Electric Power IoT. *Chinese Journal of Electronics*, vol. 32(1), 13-25.
<https://cje.ejournal.org.cn/article/doi/10.23919/cje.2021.00.312>.
- Shi, Y. H., Lin, Y., Li, B., & Li, R. Y. M. (2022). A bi-objective optimization model for the medical supplies' simultaneous pickup and delivery with drones. *Computers & Industrial Engineering*, 171, 108389-108389.
<https://doi.org/10.1016/j.cie.2022.108389>.
- She, R. F., & Ouyang, Y. F. (2024). Hybrid truck-drone delivery under aerial traffic congestion. *Transportation Research Part B: Methodological*, 185, 102970. <https://doi.org/10.1016/j.trb.2024.102970>.
- Song, B. D., Park, H., & Park, K. (2022). Toward flexible and persistent UAV service: Multi-period and multi-objective system design with task assignment for disaster management. *Expert Systems with Applications*, 206, 117855. <https://doi.org/10.1016/j.eswa.2022.117855>.
- Sun, X. T., Fang, M. H., Guo, S., & Hu, Y. (2024). UAV-rider coordinated dispatching for the on-demand delivery service provider. *Transportation Research Part E: Logistics and Transportation Review*, 186, 103571-103571. <https://doi.org/10.1016/j.tre.2024.103571>.
- Tao, J. W., Dai, H. Y., Jiang, H., & Chen, W. W. (2020). Dispatch optimisation in O2O on-demand service with crowd-sourced and in-house

- drivers. *International Journal of Production Research*, 59(20), 6054-6068.
<https://doi.org/10.1080/00207543.2020.1800120>.
- Watkins, C. J. C. H., & Dayan, P. (1992). Q-Learning. *Machine Learning*, 8(3-4), 279-292. <https://doi.org/10.1007/BF00992698>.
- Wang, F., Zhang, H. H., Du, S., Hua, M. Z., & Zhong, G. (2024). C-SPPO: A deep reinforcement learning framework for large-scale dynamic logistics UAV routing problem. *Chinese Journal of Aeronautics*, Available online.
<https://doi.org/10.1016/j.cja.2024.09.005>.
- Wang, L., Wu, X. W., Wang, Y. H., Xiao, Z., Li, L., & Fei, A. G. (2023). On UAV Serving Node Deployment for Temporary Coverage in Forest Environment: A Hierarchical Deep Reinforcement Learning Approach. *Chinese Journal of Electronics*, 32(4), 760-772.
<https://cje.ejournal.org.cn/article/doi/10.23919/cje.2021.00.326>.
- Wang, Y. F. (2020). Adaptive job shop scheduling strategy based on weighted Q-learning algorithm. *Journal of Intelligent Manufacturing*, 31, 417-432.
<https://doi.org/10.1007/s10845-018-1454-3>.
- Wang, Z. H., Leng, S. P., & Xiong, K. (2023). Multi-agent resource allocation strategy for UAV swarm-based cooperative sensing. *Chinese Journal on Internet of Things*, 1, 18-26. <https://doi.org/10.11959/j.issn.2096-3750.2023.00326>.
- Wu, G., Fan, M., Shi, J., & Feng, Y. (2023). Reinforcement Learning Based Truck-and-Drone Coordinated Delivery. *IEEE Transactions on Artificial Intelligence*, 4(4), 754-763. <https://doi.org/10.1109/TAI.2021.3087666>.
- Zhong, Y., & Wang, Y. H. (2025). Cross-regional path planning based on improved Q-learning with dynamic exploration factor and heuristic reward value. *Expert Systems with Applications*, 260, 125388.
<https://doi.org/10.1016/j.eswa.2024.125388>.