https://dx.doi.org/10.1016/j.trc.2023.104406

Unmanned aerial vehicle service network design for urban monitoring

Bolong Zhou[a], Wei Liu[b,*], Hai Yang[a]

[a]*Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China*

[b]*Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, China*

Abstract

This study examines the multi-depot location-routing problems of unmanned aerial vehicles (UAVs) for urban monitoring (MDLRP-UM). MDLRP-UM arises in various practical applications, including daily police patrols in urban residential areas, forest fire patrols, urban infrastructure status monitoring and data collection, traffic flow monitoring at important intersections, and monitoring of urban temperature and humidity, among others. These diverse applications can be modeled as a general mixed-integer quadratically constrained problem (MIQCP), where we jointly plan the service routes of the UAVs, the frequency on each route, and the location of the depots to minimize the total cost. To solve the proposed problem, we decompose it into a master problem and sub-problems. We then propose an iterative algorithm (termed as "Frequency-Time-Frequency Strategy") to solve the sub-problems, which is to find the optimal frequency and corresponding single service time for a given single route. The "Frequency-Time-Frequency Strategy" is further nested within a tailored adaptive large neighborhood search (ALNS) based heuristic algorithm to solve the master problem. The efficiency and effectiveness of the proposed solution method are demonstrated by a series of numerical studies.

Keywords: UAV, urban monitoring, routing, location, ALNS

1. Introduction

Over the past decade, there has been a growing interest in unmanned aerial vehicles (UAVs) or drones, with the commercial UAV market projected to surpass USD 15,624.7 million by 2026 according to a recent study by Polaris Market Research (Polaris Market Research, 2021). The continuous development of UAV technology, coupled with its features of flexible deployment, high mobility, strong hovering ability, and versatility in installing various equipment, has attracted attention in various large-scale civilian applications, including wireless coverage, search and rescue operations, parcel delivery, infrastructure inspections, security, and surveillance (Hayat et al., 2016). Among these applications, the use of UAVs as monitoring vehicles in urban environments is emerging

*Corresponding author. Email address: wei.w.liu@polyu.edu.hk

as a promising application (Barmpounakis & Geroliminis, 2020). The UAV flies to a specific location and uses integrated sensors to detect or monitor crowds or objects within a designated area while hovering in the air, all without human intervention on site.

The increasing complexity and scale of urban environments necessitate innovative solutions for effective monitoring and management (Engin et al., 2020; Kashef et al., 2021). UAVs offer valuable applications in urban monitoring, such as enhancing surveillance and security for law enforcement agencies (Daniel & Wietfeld, 2011). Equipped with high-resolution cameras and thermal imaging sensors, UAVs can monitor public spaces, identify security threats, and respond swiftly to emergencies. UAVs also play a crucial role in inspecting and maintaining urban infrastructure (Máthé & Buşoniu, 2015), using specialized sensors like LiDAR or multispectral cameras to efficiently detect structural defects in bridges, buildings, roads, and utility networks. Moreover, UAVs have the potential to contribute to environmental monitoring (Carminati et al., 2019) and traffic monitoring (Barmpounakis et al., 2016; Mahajan et al., 2023). By providing real-time data and accurate spatial models, UAVs support informed decision-making for urban planners and environmental agencies, fostering sustainable and resilient cities.

Establishing a safe and effective monitoring network using UAV requires solving many technical challenges, such as how to obtain accurate and stable positioning information by GPS in urban environment (Marais et al., 2014), design effective UAV control and path planning algorithms to bypass obstacles and avoid collisions or property loss (Sathyaraj et al., 2008), and manufacture easy-to-carry sensors and machine learning-based vision methods (Redmon & Farhadi, 2018), as well as efficient data collection and process systems. While there are many studies related to overcoming these technical issues and achieving substantial results, to the best of our knowledge, there are limited studies addressing the service network design challenges for UAV monitoring systems.

This study considers a scenario where a single operator provides urban monitoring services using UAVs. In this scenario, a single operator is responsible for providing specific urban monitoring services, such as daily patrols, utilizing UAVs. The city consists of multiple points or areas that require monitoring, referred to as demand points. The operator, or the monitoring service provider, must ensure that each demand point meets the minimum monitoring requirements. Unlike the more commonly studied UAV-based delivery problem (Li et al., 2022), monitoring services in this context are repetitive, continuous, and long-term in nature (Hari et al., 2021). The monitoring requirement of each demand point is typically considered as the proportion of time a UAV spends monitoring the designated area. This involves determining both the duration of monitoring during each visit to a demand point and the frequency at which the UAV visits that specific demand point. Moreover, UAVs have limited range due to battery capacity and require designated locations to stay when not performing tasks. Therefore, the operator can set up several depots at multiple locations to dispatch UAVs and provide them with charging, repairs, and routine maintenance work. Besides, unlike delivery problems, monitoring services are repetitive in nature. Therefore, it is necessary to consider the network's complexity. Each additional service route significantly increases

the network's complexity and presents operational challenges for the service provider, including scheduling individual UAVs and managing additional software and hardware resources. At the network design or planning level, the operator faces the challenge of solving the inherent combinatorial optimization problem, which involves determining the following key factors. (i) The optimal number and locations of depots that effectively cover the monitoring area. (ii) The number of UAV flight routes to be developed, including which demand points to visit on each route and their sequencing. (iii) The single monitoring time allocated to each demand point on each route, along with the frequency of dispatching UAVs on each route. Solving this optimization problem is crucial for achieving efficient and effective monitoring operations in the UAV-based urban monitoring system.

The main contributions of this study are summarized below. (i) To the authors' best knowledge, this paper is the first to study the location-routing joint decision-making and optimization problem for UAV monitoring in urban environments, i.e., the multi-depot location-routing problem of UAVs for urban monitoring (MDLRP-UM). We address the challenge of jointly determining the optimal service routes of the UAVs, frenquency on each route, and the location of the depots to minimize the total cost. (ii) The studied problem was modeled as a mixed integer quadratically constrained problem (MIQCP), where the quadratic constraints properly reflect the relationship between the UAV's single monitoring time, service intensity on the routes, and total monitoring requirement, which has rarely been modeled in existing UAV-based service network design studies. (iii) We design a tailored heuristic algorithm based on problem decomposition and the adaptive large neighborhood search (ALNS) approach for solving the MDLRP-UM. Computational experiments show that for small-scale instances, our algorithm can find solutions of the same quality as existing commercial solvers but at a faster speed. For real-world large-scale instances that solvers cannot solve, our algorithm can provide high-quality feasible solutions in acceptable time. We also carry out a sensitivity analysis in the numerical studies to assess how the changes in demand distribution, cost, and battery capacity will affect the optimal service design and system cost.

The remainder of the paper is organized as follows. Section 2 provides a brief literature review on related topics. The problem setting and the mathematical formulation are discussed in Section 3. Section 4 presents a comprehensive description of the tailored ALNS-based heuristic. Computational experiments for small and large instances are conducted and analyzed in Section 5 before we conclude and discuss potential directions for future research in Section 6.

2. Literature review

2.1. UAV-based monitoring

In recent years, there has been an increasing trend in utilizing UAV-based systems for monitoring and exploring vast areas of interest. These applications primarily focus

on the surveillance and analysis of individuals, groups, behaviors, activities, infrastructure, or buildings (Zhang & Huang, 2021). This emerging field of research leverages the capabilities of UAVs to collect valuable data and provide insights for various purposes.

The widespread utilization of UAVs in monitoring applications can be attributed to the continuous refinement of their configuration and performance, as highlighted by Yang et al. (2022). To cater to the specific monitoring requirements of different objects, researchers have developed a range of sensors that can be installed on specific types of UAVs (Baldini et al., 2020; Amarasingam et al., 2022). These sensors enable the collection of remote sensing data with varying levels of spatial and temporal resolutions. Furthermore, significant progress has been made in UAV data processing through the implementation of advanced image processing algorithms based on machine learning (Lu et al., 2017; Kako et al., 2020; Barmpounakis et al., 2019; Zhu et al., 2018). Additionally, there are researchers who are dedicated to providing precise and consistent position information for UAVs, enabling them to conduct monitoring operations in complex urban environments, such as caves and densely built urban areas (Petráček et al., 2021; Tiemann & Wietfeld, 2017; Zhang & Hsu, 2018).

Another popular research direction involves the collaborative application of UAVs and wireless sensor networks (WSNs). For instance, a study by Valente et al. (2011) proposes an air-ground wireless sensor network for crop monitoring, where UAVs are utilized as aerial nodes to monitor the temperature of the entire crop. This approach enables comprehensive monitoring and analysis of crop conditions. In terms of traffic management, Bashir et al. (2022) proposed a UAV and WSN network closed-loop control architecture for highway traffic monitoring. This architecture aims to capture speeding drivers on the highway and improve overall traffic safety. However, it is important to note that while these studies excel in addressing specific challenges and proposing effective frameworks, they primarily lack focus on the strategic-level design of UAV monitoring networks.

In the realm of monitoring applications, several optimization models and algorithms have been proposed to address UAV cruise route optimization and deployment challenges. Liu et al. (2016) presented a multi-objective optimization model that minimizes the number of drones in use while reducing the total drone cruising distance. They employed an evolutionary algorithm to solve this problem effectively. Liu et al. (2013) transformed the UAV traffic monitoring problem, without maximum flight distance constraints, into the traveling salesman problem and utilized a simulated annealing algorithm to find solutions. Mersheeva & Friedrich (2012) introduced a mixed-integer programming model for UAV route planning, considering gas station constraints, and solved it using a metaheuristic variable neighborhood search algorithm. Liu et al. (2022) tackled the mission planning problem in a scenario where only one drone is available without wireless charging equipment in a military reconnaissance mission. They proposed a hybrid framework that combines evolutionary algorithms and deep reinforcement learning methods to optimize the solution. Mersheeva & Friedrich (2015) addressed the issue of limited energy resources when using multiple drone fleets to monitor crime scenes in open-air incidents. An insertion heuristic with a negotiation

mechanism for energy resources was proposed to solve this problem.

However, existing research on UAV monitoring only focuses on one of the aspects of route planning or UAV deployment. Moreover, these studies often overlook the repetitive, continuous, and long-term nature of monitoring services, leading to a lack of service planning tools considering service frequency. To the best of the authors' knowledge, there is currently no available research that explores the joint optimization of all these factors with the goal of reducing overall costs. This includes optimizing the service routes of UAVs, determining the appropriate frequency for each route, and strategically locating depots to ensure efficient operations.

2.2. Location-routig problems

The aforementioned problem is relevant to existing studies on the location routing problems (LRP) in the context of delivery and logistics, where several research has shown that if the strategic decision of the location of facilities and the tactical design of vehicle routes are not considered simultaneously, the system's total cost can be excessive. Although these studies cannot directly solve our problem, readers interested in LRP may refer to Nagy & Salhi (2007), Prodhon & Prins (2014), a review of variants and extensions of the LRP provided by Drexl & Schneider (2015). Additionally, Tadaros & Migdalas (2022) provides a review of bi-and multi-objective LRP. In particular, existing studies have focused on minimizing the total cost of the UAV-based delivery network by jointly designing cost-effective infrastructure and routes. For example, Pinto et al. (2020) developed a drone-based food delivery service network, where they aimed to balance demand coverage and the number of departure stations. Pinto & Lagorio (2022) proposed an optimization model aimed at designing a network for drone delivery, where each point to be served is assigned to a designated departure station, and the setting of intermediate charging stations extends the reach of the drone. A survey conducted by Li et al. (2021) examined the routing problems of ground vehicles and UAVs from a two-echelon scheme perspective. Likewise, Wen & Wu (2022) investigated the two-echelon routing problem in the context of large drones carrying multiple small drones for cooperative parcel delivery. They proposed a divide and conquer framework (DCF)-based three-stage iterative optimization algorithm to address this problem effectively.

Moreover, several articles have extensively addressed the issue of collaboration between UAVs and trucks in delivery operations, recognizing it as a promising mode for parcel delivery services (Murray & Chu, 2015; Agatz et al., 2018; Wang et al., 2017; Poikonen et al., 2017; Schermer et al., 2019). Notably, numerous studies have specifically focused on optimizing the placement of docking hubs, strategically positioned intermediate points that enhance the interaction efficiency between trucks and UAVs (Wang & Sheu, 2019; Xia et al., 2023). The careful selection of hub locations can lead to substantial reductions in distribution costs and significantly improve the overall collaboration effectiveness between trucks and UAVs.

However, designing a network of UAV monitoring services is different from UAV delivery and logistics problems and involves additional/different complexities. For the proposed UAV monitoring service design problem, the location of the depot, the service

route, frequency on the route, and service time at each monitoring site (i.e., demand point in this paper) should be jointly optimized, which are not involved in existing UAV delivery problems. Moreover, the UAV monitoring service may involve multiple UAVs with different routes to serve the same monitoring site, i.e., the same demand point is served by coordinating different UAVs, which has rarely been considered in existing UAV-based service network design studies. In addition, the UAVs have limited flying ranges (subject to battery capacity). The decisions in relation to location, routing, and service time at each monitoring site interact with each other and are jointly subject to the flying range constraints.

3. Problem statement and mathematical formulation

3.1. Problem Description

Daily police patrols in cities play a critical role in ensuring the safety and security of city residents and facilities. In densely populated urban areas, patrols are an effective way to detect potential illegal activities in a timely manner, reducing crime rates and possible financial losses. Unmanned vehicles have been used for this purpose in some cities, such as the 5G-based smart police unmanned patrol car in Shenzhen, China, and the smart robot unmanned patrol car purchased by Hong Kong Airport for patrolling restricted areas. In this study, we consider a similar application scenario where UAVs are used as monitoring vehicles to conduct daily patrols in an urban area. To design a monitoring network at the planning level, we typically need three key components: demand points (with monitoring demand), depots, and routes.

In the urban area under consideration, locations with high levels of crowd gathering that require frequent monitoring are represented as demand points denoted by i. Each demand point is characterized by a single coordinate pair (a_i, b_i). The monitoring needs of these zones are influenced by various inherent factors, including population density, facility types, and land characteristics. These factors are taken into account and reflected in the form of a monitoring requirement h_i. The monitoring requirement is defined as the ratio of the time being monitored by UAVs to the total time. For instance, a busy transportation hub downtown, such as a subway station, may have a high monitoring requirement with a value of $h_i = 0.75$. This indicates that UAVs should perform patrol tasks for at least three-quarters of the whole time. On the other hand, recreational parks might have lower monitoring requirements compared to busy urban hubs.

In addition to the monitoring requirement, the bounds for single monitoring time are also pre-specified to satisfy potential operational requirements. Once a UAV visits a demand point i, the time it spends there for a single stay to perform monitoring tasks must be between the minimum time l_i and the maximum time u_i. The minimum time required for one inspection depends on the area and shape of different monitored zones and is reflected in the lower bound l_i. It is noteworthy that in practice for a patrol network the upper bound on the time duration of a single stay often helps increase the mobility of police or UAVs in this paper (Mazerolle et al., 2002; Koper et al., 2013).

Due to many physical or practical conditions, there can be only a limited number of locations that might be used as depots for the UAVs, where S denotes the set of candidate depots at pre-determined possible locations. For instance, existing police stations in the target area might serve as depots $s \in S$ to dispatch UAVs, replace batteries and provide routine maintenance. We consider that each depot has sufficient space to accommodate multiple UAVs and perform maintenance work, and do not consider the capacity limit of the depot (given that UAVs are often not too large).

The UAVs in the network will perform tasks along designed routes following certain service patterns. Typically, a UAV starts from a specific depot, flies to one or more crowded gathering zones, stays in each zone for a certain period for monitoring, and returns to the same depot. Both the path and the service time together form a route $r \in R$ to be determined. Note that the route segment between two points is not the actual flight trajectory of the UAV but rather the path of the flight, or more precisely, the flight time based on the Euclidean distance. For each route, the operator will jointly determine the frequency F_r on which the UAV is dispatched, and the single monitoring time E_{ir} for each demand point i on the route r.

The frequency F_r represents the number of UAVs dispatched per unit time on service route r, with the reciprocal of this frequency representing the interval between dispatching two consecutive UAVs on the route. For example, if we consider "minute" as the unit of single monitoring time, a value of $F_r = 0.05$ indicates that 0.05 UAVs are dispatched every minute. In other words, the interval between dispatching two consecutive UAVs on the route is 20 minutes. By continuously dispatching UAVs based on the established F_r on each route, a "static" monitoring service network is designed by the operator. This ensures a more evenly distributed coverage over the entire day, aligning with safety requirements and enhancing monitoring capabilities.

When designing the monitoring service network, we take into account the following aspects. (i) While few studies on UAV-based delivery networks have considered the impact of factors such as wind direction or load on energy consumption (Zhang et al., 2021; Xia et al., 2023; Dukkanci et al., 2021; Cheng et al., 2020), the majority of existing research assumes that battery discharge is proportional to the travel distance or time (Pinto & Lagorio, 2022). In this study, we consider that the power consumption of the UAV is directly proportional to its operational time, which includes both the en-route time between two points and the monitoring time at a demand point. This assumption is based on the understanding that a "demand point" in the model represents an area where the UAV moves within the zone to carry out monitoring tasks, rather than remaining stationary or hovering in one spot. (ii) The UAVs that provide monitoring services are homogeneous and have a constant battery-constrained maximum flight time q. When a UAV returns to the depot after performing a mission along a determined route, it is immediately replaced with a fully charged battery. (iii) For decision-making by the monitoring service provider, selecting depots, designing routes, and optimizing the frequency and service time on each route are interrelated. The selection of depots and served demand points determines the routes to design as well as the frequency and total operation time on each route. The objective is to minimize the overall cost, which

comprises the costs associated with infrastructure, management, and operational cost. The first component depends on the selection of active depots, the second component on active routes, while the third component is collectively determined by the frequency and flight time on the routes.

3.2. Model formulation

We now proceed to the mathematical formulation of the proposed problem concerning the design of a service network with UAVs for urban monitoring. This formulation aims to simultaneously optimize the selection of depots, activation of routes, frequencies on each route, and single monitoring times for demand points along the routes. Decisions regarding vehicle routing and active depots entail a significant number of binary variables. The presence of both continuous variables, such as the single monitoring time at a demand point on the route, and the frequency of the route, gives rise to inevitable bilinear terms. To address this, we propose a Mixed Integer Quadratically Constrained Program (MIQCP) model for tackling the multi-depot location-routing problem of UAVs for urban monitoring (MDLRP-UM).

For clarity, we adopt the unit of "minute" for time throughout the model and subsequent chapters, without sacrificing generality. Additionally, we pre-convert the distance between two points into flight time by dividing the Euclidean distance by the nominal average speed. This flight time is denoted as d_{ij}. Similarly, when we mention the "battery level", we are referring to the remaining flight time that the UAV can sustain with a specific amount of battery power.

To facilitate understanding, we provide a list of indices, sets, parameters, and decision variables used in the MDLRP-UM model in Table 1.

Table 1 Notations.

Indices and sets	Description
i, j	Indices of demand points
r	Index of a route
s	Index of a candidate depot
I	Set of demand points
R	Set of routes
S	Set of candidate depots
Parameters	**Description**
n_i	Number of demand points
n_r	Number of maximum routes
n_s	Number of candidate depots
(a_i, b_i)	Coordinates of point i
d_{ij}	Distance-determined travelling time from point i to point j (unit: minute)

h_i	Monitoring requirement of demand point i (ratio)
l_i	Minimum single monitoring time for a UAV visiting demand point i (unit: minute)
u_i	Maximum single monitoring time for a UAV visiting demand point i (unit: minute)
q	Battery-constrained maximum flight time (unit: minute)
w	Coefficient converting operation costs from a minute to a week
c^R	Monetary Cost of one new route (the life-cycle cost is converted in to a weekly-equivalent cost) (unit: dollar)
c^S	Monetary Cost of activating one new depot (the life-cycle cost is converted in to a weekly-equivalent cost) (unit: dollar)
c^C	Monetary Cost of charging and inspecting a UAV for one time (unit: dollar)
c^O	Monetary Cost of operating one UAV for one minute (unit: dollar)
M	A sufficiently large number
Decision Variables	**Description**
E_{ir}	Single monitoring time at demand point i on route r (unit: minute)
F_r	Frequency of dispatching UAVs on route r (unit: flight per minute)
$P_{ir,in}$	Battery-determined maximum operation time when arrives at point i on route r (unit: minute)
$P_{ir,out}$	Battery-determined maximum operation time when leaves from point i on route r (unit: minute)
T_r	Operation time of route r (unit: minute)
X_{ijr}	Binary variables; 1 if route r goes to point j after point i; 0, otherwise.
A_r	Binary variables; 1 if route r is used; 0, otherwise.
B_s	Binary variables; 1 if depot s is activated; 0, otherwise.

Based on the notations in Table 1, the MDLRP-UM is written as the following MIQCP.

$$\min \sum_{r\in R} A_r c^R + \sum_{s\in S} B_s c^S + w \sum_{r\in R} (F_r c^C + F_r T_r c^O) \tag{1}$$

s.t. Routing Constraints

$$\sum_{s\in S} \sum_{\substack{j\in I\cup S \\ j\neq s}} X_{sjr} = A_r, \forall r \in R \tag{2}$$

$$\sum_{\substack{i \in I \cup S \\ i \neq s}} \sum_{s \in S} X_{isr} = A_r, \forall r \in R \tag{3}$$

$$\sum_{\substack{i \in I \cup S \\ i \neq s}} X_{isr} - \sum_{\substack{j \in I \cup S \\ j \neq s}} X_{sjr} = 0, \forall s \in S, \forall r \in R \tag{4}$$

$$\sum_{\substack{j \in I \cup S \\ j \neq i}} X_{ijr} - \sum_{\substack{j \in I \cup S \\ j \neq i}} X_{jir} = 0, \forall i \in I, \forall r \in R \tag{5}$$

Battery Monitoring Constraints

$$P_{jr,in} \leq P_{ir,out} - X_{ijr} d_{ij} + q(1 - X_{ijr}), \forall i \in I \cup S, \forall j \in I \cup S, i \neq j, \forall r \in R \tag{6}$$

$$P_{sr,out} = qB_s, \forall s \in S, \forall r \in R \tag{7}$$

$$P_{ir,out} = P_{ir,in} - E_{ir}, \forall i \in I, \forall r \in R \tag{8}$$

$$P_{ir,in} \geq 0, \forall i \in I \cup S, \forall r \in R \tag{9}$$

$$T_r \geq q - P_{sr,in}, \forall s \in S, \forall r \in R \tag{10}$$

Demand Fulfilment Constraints

$$\sum_{r \in R} E_{ir} F_r \geq h_i, \forall i \in I \tag{11}$$

Active Depot, Active Route Constraints

$$B_s \geq X_{isr}, \forall i \in I \cup S, \forall s \in S, i \neq s, \forall r \in R \tag{12}$$

$$A_r \geq \frac{F_r}{M}, \forall r \in R \tag{13}$$

Single Monitoring Time Constraints

$$E_{ir} \leq u_i \sum_{\substack{j \in I \cup S \\ j \neq i}} X_{ijr}, \forall i \in I, \forall r \in R \tag{14}$$

$$E_{ir} \geq l_i \sum_{\substack{j \in I \cup S \\ j \neq i}} X_{ijr}, \forall i \in I, \forall r \in R \tag{15}$$

Range and Binary Constraints of Decision Variables

$$E_{ir} \geq 0, \forall i \in I, \forall r \in R \tag{16}$$

$$T_r \geq 0, \forall r \in R \tag{17}$$

$$F_r \geq 0, \forall r \in R \tag{18}$$

$$0 \leq P_{ir,in} \leq q, \forall i \in I \cup S, \forall r \in R \tag{19}$$

$$0 \leq P_{ir,out} \leq q, \forall i \in I \cup S, \forall r \in R \tag{20}$$

$$X_{ijr} \in \{0,1\}, \forall i \in I \cup S, \forall j \in I \cup S, i \neq j, \forall r \in R \tag{21}$$

$$A_r \in \{0,1\}, \forall r \in R \tag{22}$$

$$B_s \in \{0,1\}, \forall s \in S \tag{23}$$

The non-linear objective function (1) seeks to minimize the overall cost of the monitoring service network at the strategic level. The constant cost c^S is assigned to each active depot, representing the one-off infrastructure cost of building a depot. Similarly, c^R represents the costs brought by each additional service route since it will significantly increase the network's complexity and present operational challenges for the service provider, including scheduling individual UAVs and managing additional software and hardware resources. Note that in our model, c^S and c^R are pre-converted weekly-equivalent costs (the life-cycle cost is converted in to a weekly-equivalent cost). In terms of operation and maintenance, c^C is the cost of charging and inspecting a UAV for one time. After completing a monitoring task and returning to the depot, a UAV requires battery replacement and inspection, which incurs maintenance costs. Additionally, in an ongoing service network, UAVs that reach the end of their lives are continuously replaced, and the amortized UAV acquisition cost, c^O, is introduced to represent the cost of operating a UAV per unit of time (minute). To convert the operational costs to a weekly basis, a coefficient of $w = 10^4$ is used, considering that one week is approximately 10^4 minutes. Therefore, all costs, including infrastructure investment costs, management costs, and operating costs, are considered in a weekly basis. Since the total operating cost is determined by the product of the operation time and frequency of the UAVs on the routes, the product term $F_r T_r c^O$ results in a nonlinear objective function.

The cost minimization is subject to five groups of constraints. The first group relates to UAV routes and is adapted from a typical vehicle routing problem. For each used route, constraints (2) require the route to start from a depot exactly once, while constraints (3) ensure that the route ends at a depot exactly once. Constraints (4) indicate that the ending depot and the starting depot of a route must be the same. Constraints (5) state that a route visiting demand point i must also depart from i.

The second group of constraints is concerned with battery tracking, and constraints (6) also provide subtour elimination. Constraints (6) record a UAV's battery level based on the sequence of nodes (demand points and depots) visited. If node j is visited by a UAV on route r after node i ($X_{ijr} = 1$), the battery-determined maximum operation time upon arriving at node j is reduced based on the distance-determined travelling time between nodes i and j. Otherwise, constraints (6) are relaxed. Note that because a UAV consumes battery when performing monitoring tasks at a demand point it visits, and its almost-dead battery is replaced with a fully charged one immediately after returning to the depot, two essential variables $P_{ir,in}$ and $P_{ir,out}$ are used to represent the battery levels while entering or leaving a node i, respectively. Constraints (7) reset

the battery level to q when a UAV leaves an active depot. Constraints (8) reduce the battery level when a UAV leaves a demand point according to the time it spends performing monitoring tasks there. Constraints (9) and (10) ensure that each UAV has sufficient battery to visit the remaining demand points on the route and return to the depot, respectively.

The third group of constraints (11) is the demand fulfillment constraints. By setting the single monitoring time and route frequency, these constraints ensure that the service level to each demand point in the network reaches at least its monitoring requirement. Note that this formulation allows multiple UAVs on different routes serving the same monitoring site or demand point, and this scenario will be referred to as a "split visit". Since both the single monitoring time E_{ir} and the frequency F_r are continuous variables, so the product terms are nonconvex bilinear. This nature causes an inherent nonlinearity in the model, making the problem a nonlinear program (NLP), specifically, a quadratically constrained program (QCP).

The fourth group of constraints relates to the selection of depots and routes. Constraints (12) ensure that only active depots are used to dispatch UAVs, replace batteries, and provide routine maintenance, while constraints (13) ensure that UAVs are dispatched only on used routes with positive frequencies.

The final group of constraints (14) and (15) links the single monitoring time and the path on each route. If route r visits demand point i, then the single monitoring time at demand point i must be within the lower and upper bounds; otherwise, it is set to zero.

Table 2 provides a summary of the key characteristics of the proposed MIQCP model. In this table, we can observe that variables such as E_{ir}, T_r, F_r, $P_{ir,in}$, and $P_{ir,out}$ are continuous variables, while X_{ijr}, A_r and B_s are binary variables. Constraints (6) and (11), as well as the objective function, involve nonlinearity. The complexity of the proposed model is influenced by the number of demand points, candidate depots, and routes. For instance, in the case of a small network with 4 demand points, 2 candidate depots, and a maximum of 5 service routes, our model would consist of 90 continuous variables, 187 binary variables, 265 linear constraints, and 4 nonlinear constraints.

Table 2 Characteristics of the proposed MIQCP.

Variables/Constraints/Objective function	Size ($\|I\| = 4, \|S\| = 2, \|R\| = 5$)	Type
E_{ir}	$\|I\|\|R\| = 20$	Continuous
T_r, F_r	$\|R\| = 5$	Continuous
$P_{ir,in}, P_{ir,out}$	$\|I \cup S\|\|R\| = 30$	Continuous
X_{ijr}	$\|I \cup S\|^2\|R\| = 180$	Binary
A_r	$\|R\| = 5$	Binary
B_s	$\|S\| = 2$	Binary
Objective function (1)	1	Non-linear

Constraints (2),(3),(13)	$\|R\| = 5$	Linear
Constraints (4),(7),(10)	$\|S\|\|R\| = 10$	Linear
Constraints (9)	$\|I \cup S\|\|R\| = 30$	Linear
Constraints (5),(8),(14),(15)	$\|I\|\|R\| = 20$	Linear
Constraints (6)	$\|I\|(\|I\| - 1)\|R\| = 60$	Linear
Constraints (11)	$\|I\| = 4$	Non-linear
Constraints (12)	$\|S\|(\|I \cup S\| - 1)\|R\| = 50$	Linear

4. Solution approach

The joint optimization model of MDLRP-UM in Section 3 in nature is an NP-hard problem, as it is an extension of location routing problems (LRP), and there is no algorithm for solving it in polynomial time. Moreover, in our studied problem, the frequency on the routes and single monitoring times for demands on the route interact with each other and jointly determine the overall system cost. Additionally, the problem under study incorporates the multiplication of continuous variables. These factors contribute to the inherent complexity of the problem, making it highly challenging to solve. Off-the-shelf commercial non-linear solvers, such as Gurobi, are only able to solve small-sized instances of the studied problem. Consequently, we propose a heuristic approach to solve real-world large-scale cases.

The basic idea to solve MDLRP-UM is to decompose it into a master problem and sub-problems. Figure 1 describes the framework of the proposed algorithm. The master problem is the location-routing problem of UAV, where we jointly determine the active depots, path of the routes, and how to split the monitoring requirement of a demand point so that it can be cooperatively satisfied by different UAVs from multiple routes. The sub-problem is to determine the frequency and service time of each route given the path and demands to serve. We use a customized Adaptive Large Neighborhood Search (ALNS) algorithm to solve the master problem and nest within it a proposed exact algorithm termed as "Frequency-Time-Frequency strategy" for solving the sub-problems. Furthermore, whenever a new best solution is found during the ALNS search process, we utilize the commercial solver Gurobi to exactly solve a small linear programming problem. This process is referred to as "route-collaboration improvement", which enables us to enhance the solution's quality even further (see section 4.2.5).

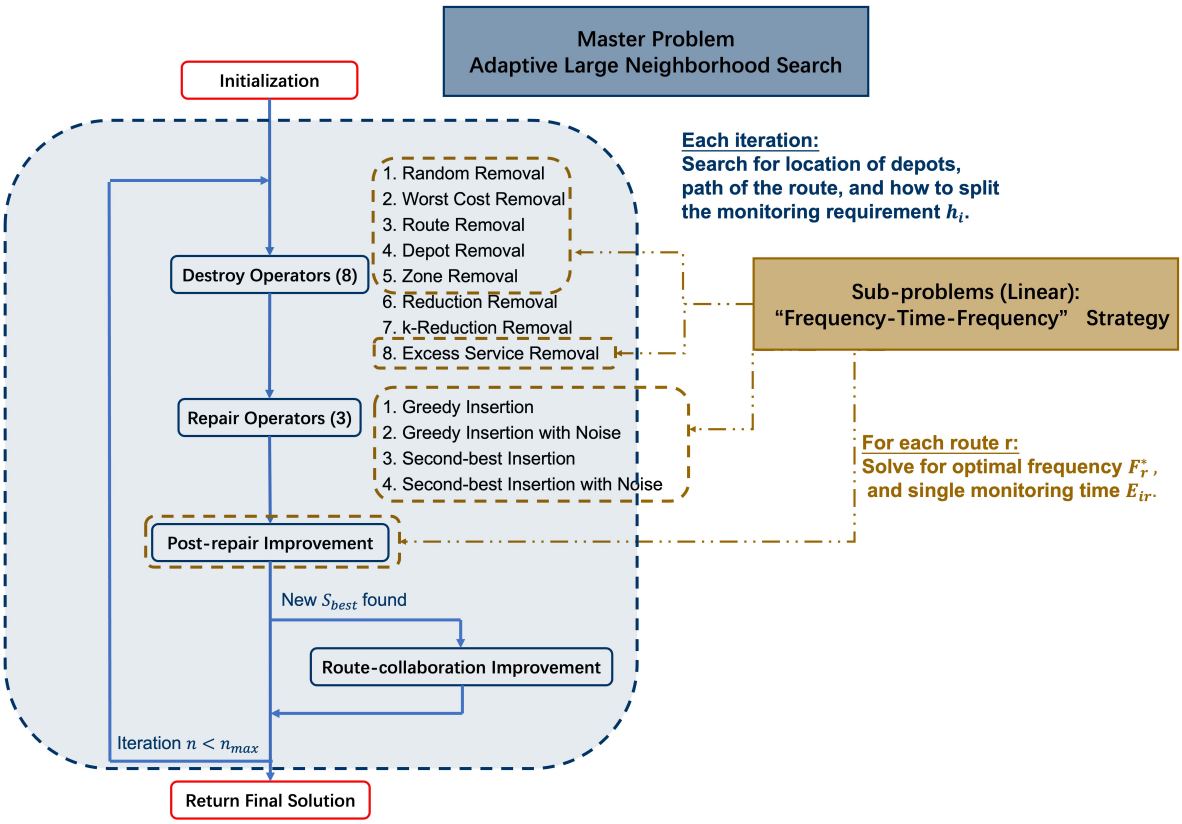


Fig. 1. Overall framework of the proposed solution approach.

4.1. Solving the subproblem - Frequency-Time-Frequency strategy

To obtain the optimal frequency F^* and corresponding single monitoring time $E_i, i \in I_r$ for a given route, we propose an iterative algorithm (termed as "Frequency-Time-Frequency strategy"). Consider a route r severing a series of demand point $i, j, ..., \in I_r$, and each demand point i has a monitoring requirement h_i to be served by this route. The lower bound l_i and upper bound u_i of the single monitoring time is also given, and the visiting sequence is assumed determined. We now describe the four steps of the proposed algorithm.

Step-1: Preparation. First, check if this route is feasible. The feasibility indicates that $t_{min} = \sum d_{ij} + \sum_{i \in I_r} l_i \leq q$, where $\sum d_{ij}$ is the total flight time between the nodes on the route, and q is the battery-constrained maximum flight time. For a feasible route, we can write $t_{allowed} = q - \sum d_{ij}$. An in-determined note list I is created with its item to be $i, j, ..., \in I_r$.

Step-2: Initial solution generation. For each of the demand point $i \in I_r$, calculate the minimum frequency $F_{i,min} = \frac{h_i}{u_i}$. Set the frequency of the route to the max-min frequency $F^* = \max_{i \in I} F_{i,min}$. The corresponding optimal single monitoring time for each

demand point on this route $i, j, ..., \in I$ is calculated by $E_i = \begin{cases} l_i, \frac{h_i}{F^*} < l_i \\ \frac{h_i}{F^*}, l_i \leq \frac{h_i}{F^*} \leq u_i \end{cases}$. Check if this solution is feasible. If yes, this is the optimal solution; if not, go to Step-3.

Step-3: Fix the single monitoring time for the "low-demand point". We call a demand point a "low-demand point" if it satisfies $\frac{h_i}{F^*} < l_i$. For any low-demand points, set $E_i = l_i$ and $t_{allowed} = t_{allowed} - l_i$, and remove it from the in-determined note list I. This means the single monitoring time of is kind of demand points will not be changed later and it will not affect the frequency nor other demand points' single monitoring time.

Step-4: Obtain the new solution. Calculate the new frequency $F^* = \frac{\sum_{i \in I_r} h_i}{t_{allowed}}$, and the corresponding single monitoring time $E_i = \frac{h_i}{F^*}$. Check if there is new "low-demand point" occurred. If yes, go back to Step-3; if no, we get the optimal solution.

Since this algorithm iteratively calculates and updates the frequency of the route and the corresponding single monitoring times, it is called the Frequency-Time-Frequency strategy. The pseudocode for this algorithm is provided in Algorithm 1. The optimality of the solution obtained by the Frequency-Time-Frequency strategy is explained in Appendix A, through a linearization of the non-linear optimization problem and writing its Karush-Kuhn-Tucker conditions. The Frequency-Time-Frequency strategy is nested in the Adaptive Large Neighborhood Search (ALNS) framework to solve the master problem, and thus will be used intensively.

Algorithm 1 Frequency-Time-Frequency Strategy

Feasibility check: $t_{min} = \sum d_{ij} + \sum_{i \in I_r} l_i \leq q$
$t_{allowed} \leftarrow q - t_{min}$
for all $i \in I_r$ **do**
  $F_{i,min} \leftarrow \frac{h_i}{u_i}$
end for
$F^* \leftarrow \max_{i \in I} F_{i,min}$
for all $i \in I_r$ **do**
  if $\frac{h_i}{F^*} < l_i$ **then**
    $E_i \leftarrow l_i$
  else
    $E_i \leftarrow \frac{h_i}{F^*}$
  end if
end for
while $\sum_{i \in I} E_i > t_{allowed}$ **do**
  for all $i \in I_r$ **do**
    if $\frac{h_i}{F_*} < l_i$ **then**
      $E_i \leftarrow l_i$
      $t_{allowed} \leftarrow t_{allowed} - l_i$
      $I_r \leftarrow I_r \setminus i$
    end if
  end for
  $F^* \leftarrow \frac{\sum_{i \in I_r} h_i}{t_{allowed}}$
  for all $i \in I_r$ **do**
    $E_i \leftarrow \frac{h_i}{F^*}$
  end for
end while
return F^* and $E_i, \forall i \in I_r$

4.2. Solving the master problem - ALNS

We design a tailored adaptive large neighborhood search (ALNS) algorithm for the master problem of MDLRP-UM, where we determine the location of active depots, path of the routes, and how to split the monitoring requirement of the demand points.

The Large Neighborhood Search (LNS) framework, originally introduced by Shaw (1998), is a technique that enhances solutions by repeatedly destroying and repairing them in large neighborhoods. Ropke & Pisinger (2006a) extended this approach to the Adaptive Large Neighborhood Search (ALNS), which employs two classes of destroy and repair operators to allow for the adaptive use of multiple large neighborhoods. The ALNS records the historical performance of each neighborhood and dynamically adjusts the operator selection process (Pisinger & Ropke, 2007). This adaptive mechanism provides flexibility and extra freedom and is widely used to solve optimization problems in various fields, including transportation and logistics (Ropke & Pisinger, 2006b; Demir

et al., 2012; Ma et al., 2023). For a comprehensive overview of the ALNS metaheuristic framework and its applications, we refer readers to the recent survey by Windras Mara et al. (2022). In the following, we present the components and overall framework of our algorithm.

4.2.1. Initialization

A feasible initial solution $S_{initial}$ is essential for the adaptive large neighborhood search. During the initialization phase, we employ a four-step greedy approach to generate such a feasible initial solution. The pseudocode for this initialization process is presented in Algorithm 2.

Step-1: Active a new depot. Let the battery level be q. Calculate the number of unserved demand points covered within a radius of $r_c q$ from each candidate station, where $0 < r_c \leq 1$ (1/3 in the implementation). The depot with the most unserved demand points around will be selected to be activated.

Step-2: Create a new route. A new route is created starting from the selected depot. It goes to the nearest unserved demand point within the battery allowance, and goes back to the depot. The battery allowance is defined as the remaining power enough for the UAV to go from the current position to the demand point, stay there for the upper bound of the single monitoring time u_i and return to the depot.

Step-3: Insert a nearest unserved demand point. One unserved demand point closest to the last served demand point on the route will inserted to this route if it satisfies the battery allowance. This process is repeated until there are no reachable demand points for this route. Calculate the frequency on the route and the single monitoring time for the demand points on the route accordingly using the Frequency-time-Frequency strategy.

Step-4: Check whether the current depot has reachable unserved demand points. If yes, go back to Step-2, if not, go back to Step-1. Repeat this process until all demand points are served.

The reader may be aware of that in the initial solution, each demand point is only served by one assigned route. However, in the following stage, some of the destroy operators introduced in 4.2.2 will allow the "split visits" scenario (one demand point is served by multiple routes) by decomposing the monitoring requirement of a demand point.

Algorithm 2 Generate a feasible initial solution

while there exists an unserved demand point **do**
  for all non-activated depots **do**
    Calculate the number $n_{i(s)}$ of unserved demand points covered within a radius of $r_c q$ from the depot s.
  end for
  Activate the depot s with the largest $n_{i(s)}$.
  repeat
    Create a new route r from depot s.
    while battery is enough to visit another unserved demand points $i, j, ...$ **do**
      Insert the nearest i to route r.
    end while
    Calculate F_r and E_{ir} using Frequency-time-frequency strategy.
  until no reachable demand point from depot s
end while
return a feasible initial solution $S_{initial}$

4.2.2. Destroy operators

The destroy stage removes k demand points from the current solution either partially or completely using the destroy operators. The removed demand points are added into an unserved demand list. During this process, unless otherwise specified, any changes to a route require updating its frequency and corresponding single monitoring time for the demand points using the Frequency-Time-Frequency strategy. After determining the unserved demand list, we calculate the unserved demand $h_{i,unserved}$ of each demand point i by subtracting the existing services provided from the original monitoring requirement, that is, $h_{i,unserved} = h_i - \sum_{r \in R} E_{ir} F_r$.

We adopt eight types of destroy operators in our implementation, with the first five inspired and modified from previous literature (Ropke & Pisinger, 2006a,b; Demir et al., 2012; Wang et al., 2020), while the last three are introduced in this work. We describe each operator below:

1. Random Removal (RR): This operator randomly removes one demand point on a randomly-selected route in the current solution and adds it to the unserved demand list. The idea is to increase the randomness of the search process and diversify the search space.

2. Worst Cost Removal (WCR): This operator selects the demand point that generates the maximum cost saving or maximum benefit from removing it in the current solution. If a demand point is served by multiple routes in the current solution, the operator treats them as different demand points on different routes and removes the one with the highest cost saving. We calculate the cost savings for each demand point i on a route as $\Delta i = cost(S_{current}) - cost(S_{current} \backslash i)$, remove the demand point with the highest Δ_i and add to the unserved demand list.

3. Route Removal (ROR): This operator randomly removes a full route in the current solution. All demand points served by this route are removed sequentially and added to the unserved demand list. This operator is widely used in vehicle routing problems to remove a series of associated nodes at once from a solution.

4. Depot Removal (DR): This operator randomly selects an activated depot and removes all routes originating from that depot in the current solution. This operator plays a critical role in determining the location of depots.

5. Zone Removal (ZR): This operator removes a series of demand points based on their 2D Cartesian Coordinates. It first computes the corner points of the entire region, randomly select an area accounting for a certain predefined proportion (9% in the implementation) of the total region, and remove all demand points within this area sequentially and add them to the unserved demand list. If the area does not contain any demand point, it will do nothing in that iteration.

6. Reduction Removal (RER): This operator splits the monitoring requirement of a certain demand point and reduces part of the demand served in the current solution. Recall that for a route r severing a series of demand point $i, j, ..., \in I_r$, if its operation time has not achieved the battery-limited maximum time, then the frequency of this route is determined by the "most-needed" demand point, that is $F_r = \max_{i \in I_r} F_{i,min}$, where $F_{i,min} = \frac{h_i}{u_i}$ is the minimum frequency of a demand point. Such a case is likely to lead to a waste of resources and an increase in cost, because other demand points on the route may not actually need such a high frequency to serve.

   The idea of this operator is to reduce the impact of this situation by finding and splitting the "most-needed" demand point on a route. It will randomly select one route r which serves at least two demand points, and set the new frequency on the route $F_{r,new}$ to the minimum frequency of the "second most-needed" demand point. In that case, the "most-needed" demand point will be put into the unserved demand list with a decomposed unserved demand, since the actual monitoring service provided to the "most-needed" demand point is reduced.

   For the route whose operation time has achieved the battery-limited maximum time, this operator still perform the same strategy to set the frequency and decompose the monitoring demand. Note that this decomposition is not always a perfect split, and the potential sub-optimality brought by such decomposition could be reduced or eliminated by the route-collaboration improvement to be introduced later in subsection 4.2.5.

7. k-Reduction Removal (k-RER): This operator is a variant of Reduction Removal (RER). It randomly selects one route r which serves at least $k + 1$ demand points, and set the new frequency on the route $F_{r,new}$ to the minimum frequency of the "$(k + 1)^{th}$ most-needed" demand point. In that case, k demand points will be split

and put into the unserved demand list together during one iteration, which increases the efficiency of decomposition. In our implementation, k is set to three.

8. Excess Service Removal (ESR): This operator is based on the situation that for some demand points, the monitoring service provided by the network may exceed the monitoring requirement required. Such a scenario is called an excess service and it is likely to occur when a demand point with a relatively low monitoring requirement is served by a route with a relatively high frequency. To see that, consider a route r with its max-min frequency $F_r = \max_{i \in I_r} F_{i,min}$, if a demand point i on the route has a monitoring requirement h_i such that $\frac{h_i}{F_r} < l_i$, then the monitoring service provided to this demand point $h_{i,served} = E_{ir} F_r \geq l_i F_r$ is greater than the monitoring requirement h_i. This situation is also possible to occur when a demand point is severed by multiple routes since the monitoring requirement is decomposed and for each route, the monitoring requirement to serve is thus relatively low.

   The intuition of this operator is to identify and reduce this mismatching of the demand and frequency in the current solution. It will randomly select one demand point with excess service, and remove it from every route which serves this demand point. It also provides the ability to "merge" the demand points served by multiple routes since the single demand point i will be put into the unserved demand list with its original monitoring requirement h_i. This can be seen as a reverse process of the split demand function performed by Reduction Removal (RER) and k-Reduction Removal (k-RER), reducing the possibility of excessively unnecessary splitting of demand points.

4.2.3. Repair operators

In the repair stage, the demand points i in the unserved demand list with its unserved demand $h_{i,unserved}$ will be inserted to the current destroyed solution. Four repair operators are designed in our algorithm, all of which are inspired from the implementation in literature (Ropke & Pisinger, 2006a,b; Ghilas et al., 2016; Wang et al., 2020).

1. Greedy Insertion (GI): This operator iterates over all demand points in the unserved demand list and inserts them in the way and position that produces the minimum incremental cost in the current solution. The unserved demand list is randomly shuffled before selecting demand points sequentially. Possible ways to insert the demand point include a) inserting it into an existing route that serves it and merging the demand, b) inserting it into an existing route that has not served it and choosing the best position, c) creating a new route from an active depot and inserting the demand point into that new route, or d) activating a new depot and inserting the demand point into a new route from that active depot. If any route is changed during the insertion process, its frequency and corresponding single monitoring time for the demand points on that route should be updated using the Frequency-Time-Frequency strategy.

2. Greedy Insertion with Noise (GIN): This operator is an extension of GI that incorporates a noise term $random(0.1, 1)$ to increase the randomness of the search space when inserting greedily. The cost of insertion is evaluated using $CostWithNoise = ActualCost \times random(0.1, 1)$ instead of the actual cost unless using the $ActualCost$.

3. Second-best Insertion (SI): This operator is similar to the Greedy Insertion (GI), but it differs in the way it selects the insertion position for each demand point. Specifically, SI chooses the position that results in the second-least incremental cost in the current solution. This approach helps to expand the search space within certain limits and reduces the risk of getting trapped in a local optimum.

4. Second-best Insertion with Noise (SIN): This operator is an extension of SI that incorporates a noise term $random(0.1, 1)$ to increase the randomness of the search space when inserting greedily. The cost of insertion is evaluated using $CostWithNoise = ActualCost \times random(0.1, 1)$ instead of the actual cost unless using the $ActualCost$.

4.2.4. Post-repair improvement

We introduce a post-repair improvement procedure to further improve the quality of the repaired solution. The procedure aims to identify unnecessary or redundant visits to demand points on some routes. Under the "split visits" scenario (one demand point is served by multiple routes), when the monitoring requirement of a demand point is decomposed and re-inserted into another route, it is possible that the services provided by the new route can meet the entire monitoring requirement individually, rendering the remaining services provided by the old route redundant. Therefore, the procedure scans all demand points on each route and removes every unnecessary visit. Specifically, it removes a demand point i on a particular route r^* if $\sum_{r \in R \setminus r^*} E_{ir} F_r \geq h_i$. If any demand point on a route is removed during this process, the frequency on that route and the corresponding single monitoring time for the remaining demand points on that route should be updated using the Frequency-Time-Frequency strategy. Finally, a new solution S_{new} is generated.

4.2.5. Route-collaboration improvement

The previously introduced customized destroy operators have the ability to heuristically determine the decomposition of the monitoring requirement for a specific demand point, assigning it to different routes and enabling multiple routes to serve the same demand point simultaneously. However, the solutions obtained through this approach often have room for improvement in terms of quality. Therefore, we propose a route-collaboration improvement.

The route-collaboration improvement process can be seen as an extension of the subproblem discussed in section 4.1. Firstly, we identify all demand points that fall into "split visits" scenarios, indicating they are served by at least two routes within the current solution. We then find the subset of routes that visit these demand points. For each route in this subset, the visiting sequence is fixed. The lower/upper bound of the single monitoring time, and the total (original) monitoring requirement of each

demand point on the route are known. The route-collaboration improvement aims to optimally adjust the frequency F_r^* and corresponding single monitoring time E_{ir} for all these routes simultaneously. Essentially, this process implicitly searches for the optimal decomposition of the monitoring requirement for a demand point among the given subset of routes.

We formulate this problem as a small linear programming problem, as outlined in Appendix B. Whenever a new best solution is found during the ALNS search process, we employ the commercial solver Gurobi to precisely solve this small linear programming problem. This further enhances the quality of the solution obtained during the search process by our algorithm.

4.2.6. Adaptive mechanisms and acceptance criterion

In the ALNS, the destroy operators and repair operators are applied iteratively to generate new solutions. We adopt the Roulette Wheel principle as the adaptive mechanism during the search process. Initially, each destroy or repair operator is assigned an equal probability of selection, 1/8 and 1/4 respectively, since the potential performance of operators is unknown. The probability is updated to the weighted score gained every η iterations. Let $f(S_{best})$, $f(S_{current})$, and $f(S_{new})$ denote the costs of the objective functions for the best solution, current solution, and new solution, respectively. If a new best solution is generated, both the destroy and repair operator used in that iteration are assigned a score of $\sigma_1 + (f(S_{new}) - f(S_{current})) \times 100/f(S_{best})$. If a solution is found better than the current but not the best solution, the score of the operator is assigned a score of $\sigma_2 - (f(S_{new}) - f(S_{best})) \times 100/f(S_{best})$. If the solution is worse than the current solution but accepted by the acceptance criterion, the score is assigned a score of σ_3. Under this mechanism, operators that contribute more in every η iterations will have a higher probability of being selected in the next η iterations.

Metropolis criteria is applied as the acceptance criterion, which provides the opportunity to avoid getting trapped in a local optimum. When $f(S_{new}) < f(S_{current})$, the new solution is accepted immediately. Otherwise, the ALNS algorithm assigns a probability of $e^{(f(S_{new})-f(S_{current}))/T}$ to accept the solution, where T is a non-negative current temperature. T is initially set to T_0, and it decreases at a determined cooling rate $0 < \alpha < 1$ after every iteration. As the iterative process progresses and the temperature cools down, the probability of accepting a poor solution decreases.

When the preset maximum number of iterations is reached, the algorithm returns the best solution found. The general structure of our tailored ALNS algorithm is presented in algorithm 3.

Algorithm 3 The general structure of tailored ALNS algorithm

Initialization: A feasible initial solution $S_{initial}$
Initialize probabilities associated with the operators
$S_{best} \leftarrow S_{current} \leftarrow S_{initial}, i \leftarrow 1, T \leftarrow T_0$
repeat
  Select a destroy and a repair operator using the roulette wheel selection
  $S_{to\ improve} \leftarrow repair(destroy(S_{current}))$
  $S_{new} \leftarrow post-repair\ improve(S_{to\ improve})$
  if $f(S_{new}) < f(S_{current})$ **then**
    $S_{current} \leftarrow S_{new}$
    if $f(S_{new}) < f(S_{best})$ **then**
      $S_{best} \leftarrow S_{new}$
      if at least one demand point is served by multiple routes **then**
        $S_{best} \leftarrow route-collabaration\ improve(S_{best})$
      end if
    end if
  else
    if $rand[0,1] < e^{(f(S_{new})-f(S_{current}))/T}$ **then**
      $S_{current} \leftarrow S_{new}$
    end if
  end if
  if $i = \eta$ **then**
    Update probabilities associated with the operators by adaptive mechanisms
    $i = 0$
  end if
  $T = \alpha T, i = i + 1$
until total number of iterations is reached
return S_{best}

5. Numerical experiments

This section presents the results of computational experiments performed on both small and large instances to evaluate the performance of our tailored ALNS-based heuristic. For the small-scale instances, off-the-shelf commercial solvers can reach exact optimality, and we use these solutions to assess the quality of solutions obtained from Algorithm 3. For large-scale problems, the solver cannot provide any feasible results, and we can only obtain solutions from our algorithm.

To gain further insights, we conducted a sensitivity test by varying the cost and the UAV battery-constrained maximum flight time q of UAV. This allows us to examine the impact of parameter changes on both the total cost and the number of service routes employed in the solution. Additionally, we also investigate the effect of allowing "split visits" (multiple UAVs on different routes serving the same monitoring site or demand point) on the cost and solution pattern for both uniform demand and cluster demand.

The proposed algorithm is implemented in Python. All experiments are conducted on a machine with Apple M1 3.2 GHz processor and 16GB RAM. Gurobi v9.5.2 is used in the Python environment for obtaining the exact optimal solutions.

5.1. Paramater Setting

Referring to the Consumer Drones from DJI (2023), the battery-constrained maximum flight time of the UAV is limited to 30 minutes. Considering that the real trajectory of the UAV is longer than the Euclidean distance on the map, the "nominal" average flight speed between two points is assumed 10 kilometers per hour. The cost-related constants are $(c^R, c^S, c^C, c^O) = (12 \times 10^4, 20 \times 10^4, 15, 6)$.

We refer to the parameter tables from Koç (2016) and Wang et al. (2020) to fine-tune the parameters of our tailored ALNS algorithm. The parameters we use in our algorithm are provided in Table 3.

Table 3 Parameters used in the tailored ALNS.

Description	Typical values
Maximum number of iteration N_{max}	10000
Number of iterations for roulette wheel η	50
Roulette wheel parameter ρ	0.3
Score for new best solution σ_1	5
Score for improving the current solution σ_2	3
Score for accepting a worse solution σ_3	1
Startup temperature T_0	1000
Cooling rate α	0.999

We denote the above set of parameters as "Setting A". In the subsequent computational experiments, we also employed two additional parameter sets. In "Setting B", all parameters remain the same as in "Setting A", except for the battery-constrained maximum flight time of the UAV, which is doubled to 60 minutes. In "Setting C", all parameters remain the same as in "Setting A", except that the cost of a new route is reduced to one-sixth of the original value, i.e., $c^R = 2 \times 10^4$.

5.2. Small-scale problems

Initially, we generate small-scale instances by selecting densely populated areas in urban Hong Kong. Each instance covers a rectangular area of approximately four square kilometers. The target demand points to be monitored are metro stations within the area, while existing police stations serve as potential depots for UAVs. The latitude and longitude coordinates of these points are obtained from Google Maps and then converted into Cartesian coordinates (x, y).

We create four small-scale instances, varying in the number of demand points from four to eight and the number of depots from two to six. The monitoring requirements

for each demand point are randomly generated between zero and one. The lower bound of the single monitoring time of the UAV at each demand point is randomly selected from one to three minutes, while the upper bound is a random integer between two and seven minutes (should ensure that it is larger than the lower bound).

To expand our experimental dataset, we empoly the following approach. First, we randomly generate coordinates for demand points and depots within a rectangular area of the same size as the previously mentioned instances, thereby creating additional instances. Secondly, for instances with no more than six demand points, we test three different parameter settings (i.e., "Setting A", "Setting B", "Setting C"). This results in a total of twenty-six small-scale instances, named as "DxSy-nz", where x represents the number of demand points, y denotes the number of candidate depots, n indicates the instance index, and z signifies the parameter setting (as shown in Table 4).

Table 4 presents the total cost and CPU time for the test instances using Gurobi and our ALNS-based heuristic. It also reports the number of routes used in the optimal solution found. Due to the stochastic nature of the search process, each instance is solved five times using the ALNS-based heuristic, and the table displays the best solution values and total CPU time for these five runs. Out of the twenty-six instances tested, Gurobi fails to find or prove the optimal solution within 10 hours for five of them. However, our ALNS algorithm consistently produces solutions of at least the same quality as those obtained by Gurobi for all twenty-six instances.

The computation time using Gurobi tends to exhibit exponential growth with the number of demand points and candidate depots. Additionally, even for the same instance, the time required to obtain the exact optimal solution can vary from several seconds to several thousand seconds or longer. In contrast, our ALNS algorithm demonstrates stable running times, all within half a minute for the instances tested.

In instance D9S6-1A, we observe that our algorithm could even find a better solution compared to the best solution obtained by running the commercial solver for 10 hours. Both solutions share the same decisions regarding activated depots and route paths, but they slightly differ in the frequency and single monitoring time of some demand points on a specific route. This discrepancy can be attributed to the effectiveness of the Frequency-Time-Frequency Strategy proposed in Section 4.1, which enables us to identify the exact optimal frequency and single monitoring time for each route, thereby significantly reducing the search space. Conversely, when Gurobi is employed to directly solve the MIQCP in Section 3.2, it may struggle to fully comprehend the problem's characteristics, decompose it appropriately, and recognize that the subproblem is a linear problem. The result of both solutions is depicted in Figure 2.

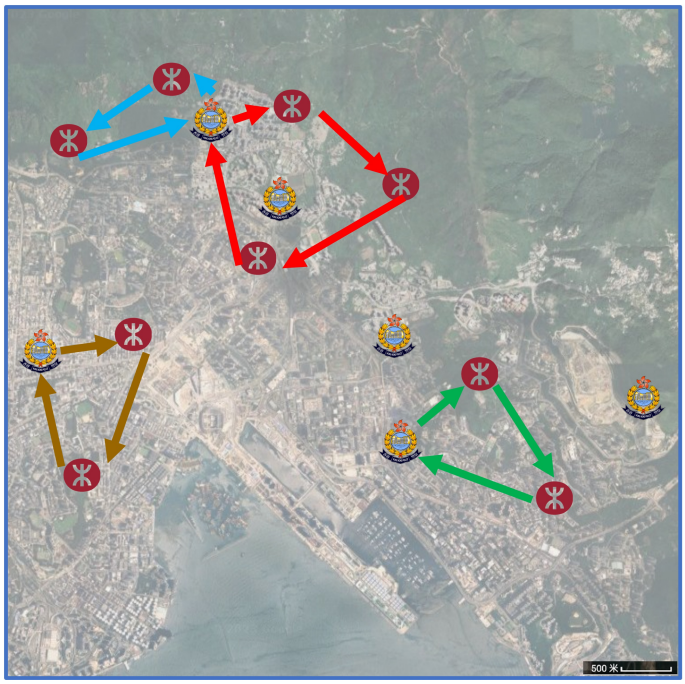
Table 4 Comparison of the results for small-scale instances.

Instance	Gurobi total cost (10^4)	Gurobi CPU time (sec)	Gurobi # of routes	ALNS total cost (10^4)	Cost gap (%)[c]	ALNS CPU time (sec)[d]	ALNS # of routes
D4S2-1A[a]	66.12	1.59	2	66.12	0.00	13.08	2
D4S2-1B[a]	63.67	2.08	1	63.67	0.00	13.91	1
D4S2-1C[a]	45.10	201.94	3	45.10	0.00	14.93	3
D4S2-2A	167.37	8.48	3	167.37	0.00	13.81	3
D4S2-2B	137.68	5119.89	2	137.68	0.00	13.54	2
D4S2-2C	132.78	27.07	4	132.78	0.00	13.49	4
D4S2-3A	123.55	14.87	2	123.55	0.00	13.48	2
D4S2-3B	100.59	285.49	2	100.59	0.00	13.97	2
D4S2-3C	92.87	25.74	4	92.87	0.00	13.37	4
D4S2-4A	118.49	7.97	2	118.49	0.00	13.15	2
D4S2-4B	99.94	94.45	2	99.94	0.00	14.20	2
D4S2-4C	88.75	27.44	3	88.75	0.00	13.97	3
D4S2-5A	117.78	12.97	2	117.78	0.00	14.40	2
D4S2-5B	97.22	111.65	2	97.22	0.00	13.89	2
D4S2-5C	87.70	48.04	4	87.70	0.00	14.35	4
D5S2-1A	164.22	60.53	4	164.22	0.00	15.49	4
D5S2-1B	117.49[b]	**10 hr**	2	117.49	0.00	15.46	2
D5S2-1C	124.22	141.48	4	124.22	0.00	15.79	4
D5S2-2A	92.37	667.35	2	92.37	0.00	14.96	2
D5S2-2B	84.95	3186.69	1	84.95	0.00	15.07	1
D5S2-2C	70.54[b]	**10 hr**	3	70.54	0.00	16.76	3
D6S2-1A[a]	122.66	381.45	3	122.66	0.00	16.43	3
D6S2-1B[a]	99.35[b]	**10 hr**	2	99.35	0.00	18.99	2
D6S2-1C[a]	91.93	895.42	4	91.93	0.00	18.91	4
D8S3-1A[a]	126.37[b]	**10 hr**	3	126.37	0.00	21.54	3
D9S6-1A[a]	**176.40[b]**	**10 hr**	4	**176.26**	**-0.08**	24.46	4

[a] Coordinates are generated based on Hong Kong real-world scenarios.
[b] Values obtained at the end of the 10 h running time.
[c] Cost gap (%) $= \frac{Cost(ALNS)-Cost(Gurobi)}{Cost(Gurobi)} * 100\%$.
[d] For each instance, we report the total time for 5 runs of our ALNS algorithm.

Figure 3 illustrates the results obtained for another typical small-scale instance, D8S3-1A. For this instance, both Gurobi and ALNS yield consistent solutions. In this

network with eight demand points and three candidate depots, one depot is chosen to be activated, and three UAV service routes originate from it. It can also be observed that one demand point with a high monitoring requirement ($h_6 = 0.82$) is served by two routes together, aiming to enhance UAV operational efficiency and reduce costs.



Demand Point
Depot
Route 1
Route 2
Route 3
Route 4

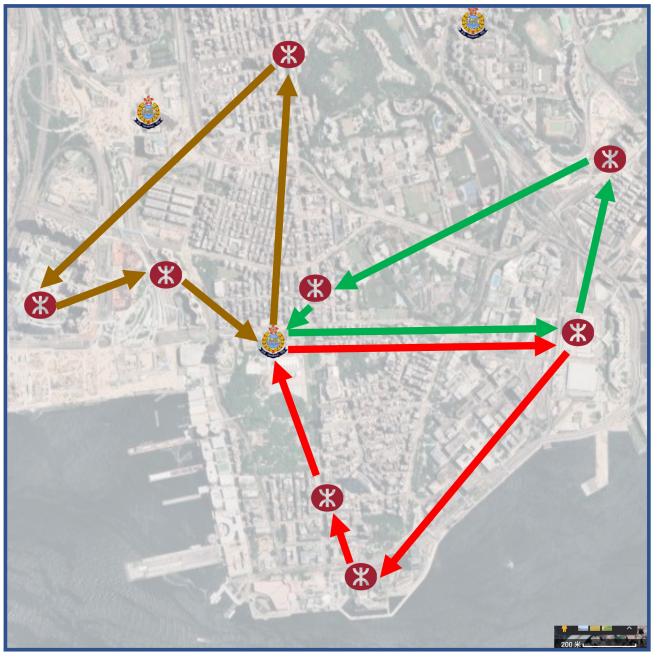
Total cost: 176.40×10^4 (Gurobi, 10hours)

Route	Node(time)	Frequency	Total time
1	S1 - D6(5.80) - D7(6.0) - S1	0.0983	26.28
2	S5- D8(6.0) - D9(2.0) - S1	0.055	23.74
3	S2- D3(1.0) -D4(**3.09**) - D5(**4.91**) - S1	**0.054**	**29.31**
4	S2- D2(3.0)- D1(2.0) - S1	0.27	17.45

Total cost: 176.26×10^4 (ALNS)

Route	Node(time)	Frequency	Total time
1	S1 - D6(5.80) - D7(6.0) - S1	0.0983	26.28
2	S5- D8(6.0) - D9(2.0) - S1	0.055	23.74
3	S2- D3(1.0) -D4(**3.15**) – D5(**5.0**) - S1	**0.055**	**29.16**
4	S2- D2(3.0)- D1(2.0) - S1	0.27	17.45

Fig. 2. Result of small-scale instance D9S6-1A. The solutions obtained from Gurobi and ALNS differ in the frequency and single monitoring time of some demand points on route 3.



Demand Point
Depot
Route 1
Route 2
Route 3

Total cost: 126.37×10^4 (both Gurobi and ALNS)

Route	Node(time)	Frequency	Total time
1	S2 - D5(**5.45**) - D6(**2.23**) - D7(**3.0**) - S2	**0.1569**	30
2	S2 - D7(**3.24**) - D8(**4.64**) - D3(**3.06**) - S2	**0.1078**	30
3	S2 - D3(5.51) - D1(1.77) - D2(2.0) - S2	0.0961	30

Fig. 3. Result of small-scale instance D8S3-1A. Both Gurobi and ALNS yield consistent solutions. Demand point D6 is served by two routes.

5.3. Large-scale case study

In addition to the small-scale instances, we conduct a large-scale case study to monitor the traffic flow in the Kowloon district of Hong Kong. Given the district's extensive network of arterial roads, monitoring the entrances and exits of these roads is crucial for mitigating congestion and reducing traffic accidents. Unmanned aerial vehicles (UAVs) offer an efficient means of monitoring these areas, as they provide a bird's-eye view of the traffic flow that is difficult to achieve with ground-based monitoring systems. Moreover, the use of UAVs can provide a clear view of the surrounding traffic flow, particularly in areas where entrances and exits are located on overpasses. Our objective in this case study is to identify an optimal UAV monitoring network that can effectively monitor all demand points while minimizing total cost.

We collect the real longitude and latitude coordinates of all 119 entrances and exits of these arterial roads from Google Maps as demand points, while all 24 police stations in the area are designated as candidate depots (Figure 4). All parameter settings are consistent with "Setting A" mentioned in previous sub-chapters.

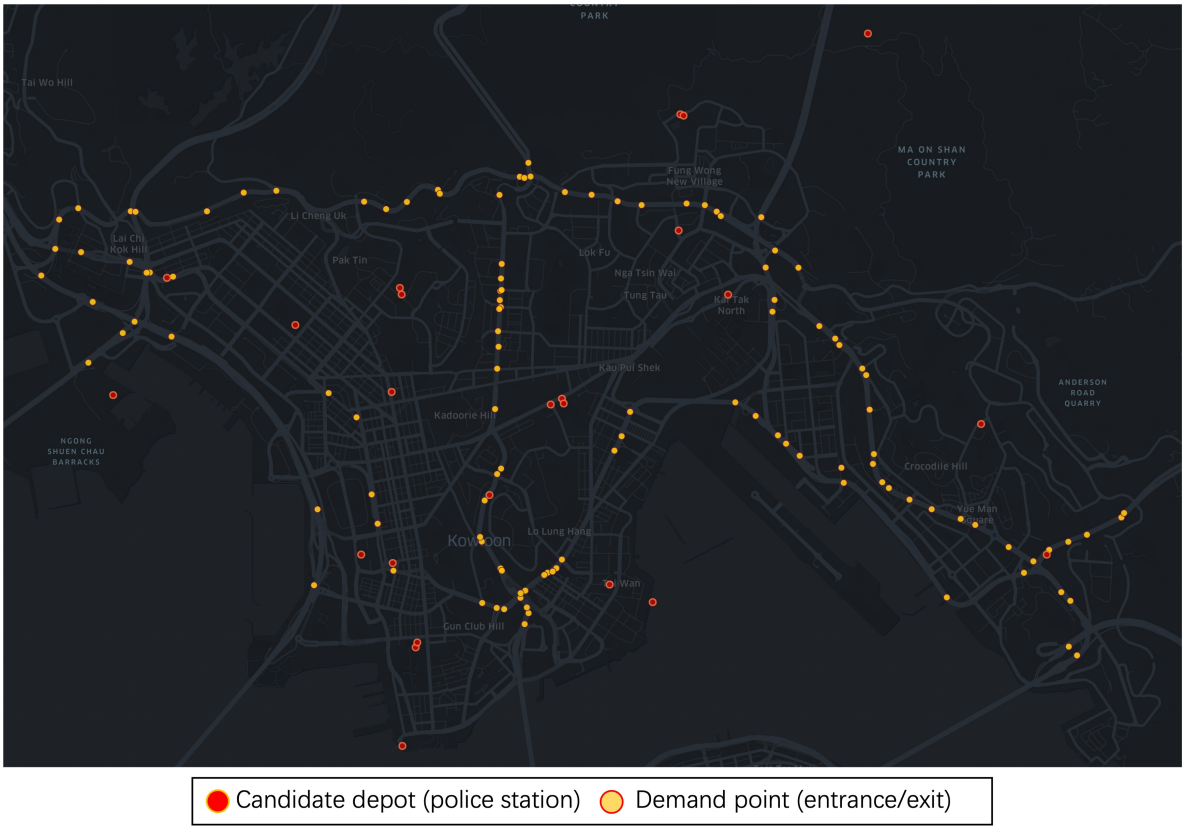


Fig. 4. Distribution of entrances/exits and police stations in Kowloon, Hong Kong.

Given a time limit of 10 hours, the commercial solver Gurobi fails to provide any feasible solutions for this large-scale case. We used the proposed ALNS-based heuristic to solve this problem 10 times. The best and average values of the objective function are 940.95×10^4 and 969.56×10^4, respectively, with a standard deviation of 17.99, indicating that our algorithm's performance is relatively stable. The average CPU time is 147.63 seconds, with a standard deviation of 3.21. Upon analyzing the final results of the 10 runs, we find that these solutions activate five to six depots and use 14 to 17 service routes, with the number of demand points under a split-visit scenario varying from 9 to 23. A typical example of a solution generated by our proposed ALNS-based heuristic is presented in Figure 5. These results demonstrate the effectiveness of our heuristic in solving real-world large-scale UAV monitoring design problems.

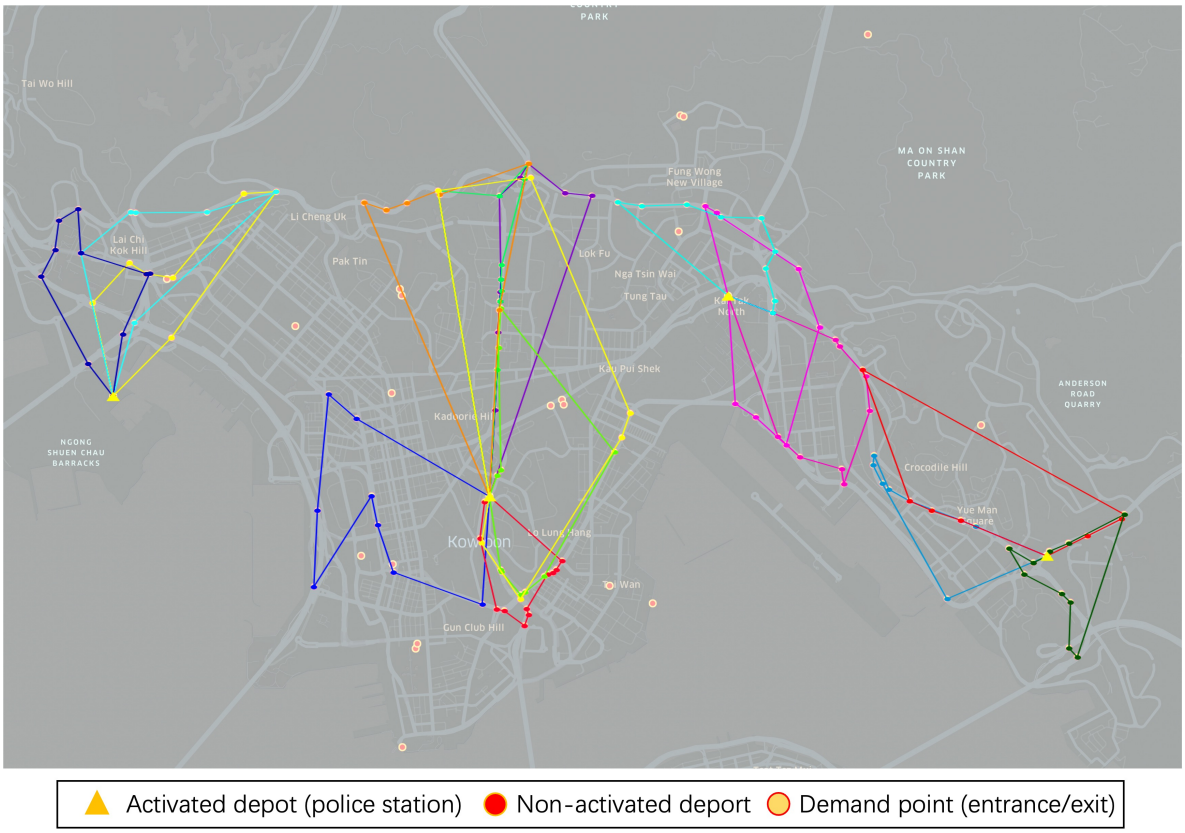


Fig. 5. An example of solution generated by ALNS-based heuristic for the large-scale case. In this solution, 4 depots are activated and 16 routes are used. 14 demand points are under a split-visit scenario. The objective function value is 940.95×10^4.

We also assess the efficiency of each operator individually in our ALNS algorithm through the large-scale case study. The approach used by Naccache et al. (2018) is adopted as our reference method. We start with removal operators by executing the algorithm for 10 runs with all removal operators except one at a time. The outcomes of this analysis are detailed in Table 5. Similarly, we conduct the same analysis for the repair operators, and the corresponding results are presented in Table 6. It is evident that the omission of any one of the destroy or repair operators will result in a decline in the solution's quality, therefore, we conclude that no operator appears to be redundant within our algorithm. The results clearly indicate that our selection of operators effectively balances the search process by combining strength and diversity. Additionally, the success of our informatively proposed operators underscores their reliance on the problem's feature design.

Table 5 Quality of the solution when using all but one destroy operator.

Used removal operator	Best Total Cost (10^4)	Average Total Cost (10^4)	Average CPU Time (sec)
All	940.95	969.56	147.63
No Random Removal	959.25	993.75	195.58
No Worst Cost Removal	943.39	981.63	138.46
No Route Removal	959.58	973.26	173.81
No Depot Removal	945.99	986.94	159.50
No Zone Removal	963.52	992.35	136.67
No Reduction Removal	953.21	977.68	171.50
No k-Reduction Removal	958.27	981.06	152.32
No Excess Service Removal	957.07	982.70	170.65

Table 6 Quality of the solution when using all but one repair operator.

Used repair operator	Best Total Cost (10^4)	Average Total Cost (10^4)	Average CPU Time (sec)
All	940.95	969.56	147.63
No Greedy Insertion	981.42	1012.67	180.67
No Greedy Insertion with Noise	951.04	981.29	223.98
No Second-best Insertion	964.89	983.94	146.30
No Second-best Insertion with Noise	948.41	980.04	140.48

5.4. Battery and Cost

In this subchapter, we proceed to conduct sensitivity tests regarding the cost and UAV battery-constrained maximum flight time q.

We first examine the impact of UAV battery capacity on the monitoring network. From the results table for small-scale instances (Table 4), it is observed that doubling the battery-constrained maximum flight time of the UAV to 60 minutes resulted in total cost savings ranging from 3.7% to 28.5% across the eight small-scale cases. These savings could be attributed to three key factors. Firstly, a larger battery capacity enhances the maneuverability and flexibility of the UAV, enabling it to provide services to more distant demand points or extend the duration of service for a specific demand point. Secondly, longer flight times potentially result in fewer routes needed in the network. Among the eight small-scale instances, when q is doubled, the best-obtained solutions for five instances utilize one to two fewer routes, while the number of routes remains unchanged for the remaining three instances. This streamlines the service network and reduces route costs. Furthermore, longer flight times lead to less frequent battery replacements and maintenance, resulting in lower operating costs.

Figure 6 illustrates the overall trend of the total cost and the number of routes used by the network as the battery-constrained maximum flight time of the UAV increases. It is based on the large-scale case discussed in subchapter 5.3, and each case is run for 10 times.

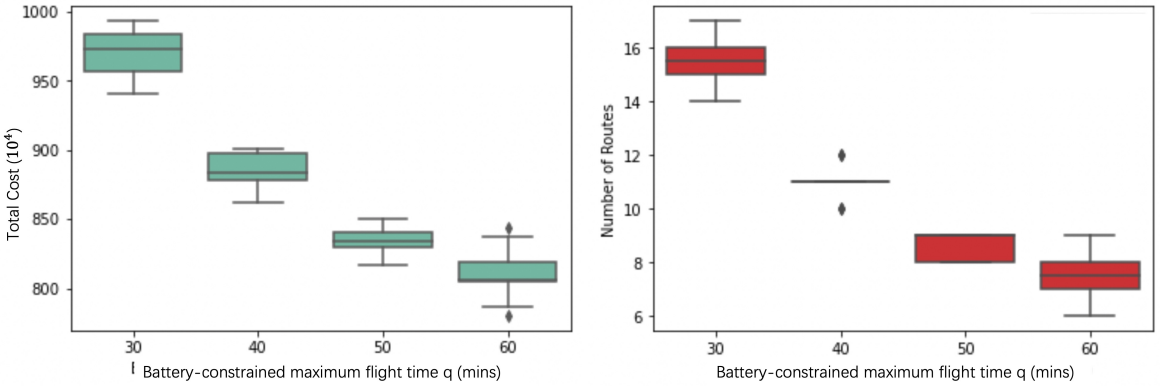


Fig. 6. Total cost and number of routes in the large-scale network at different setting of battery-constrained maximum flight time (q).

We also present the impact of route cost (c^R), which represents the management cost of each new route in the network, on the monitoring network. As shown in table 4, when the route cost is reduced from 12×10^4 to 1/6 of the original value (i.e., 2×10^4), the eight small-scale instances exhibit cost savings ranging from 20.7 percent to 31.8 percent. Simultaneously, six instances utilize 1 or 2 additional routes, while the remaining two instances maintain the same number of routes used. Note that without considering the route cost, the model may yield a solution with more routes, which can significantly increase the complexity of the network and pose substantial operational challenges for the service provider. Figure 7 illustrates the overall trend of the total cost and the number of routes used by the network as the route cost increases. It is also based on the large-scale case discussed in subchapter 5.3, and each case is run for 10 times.

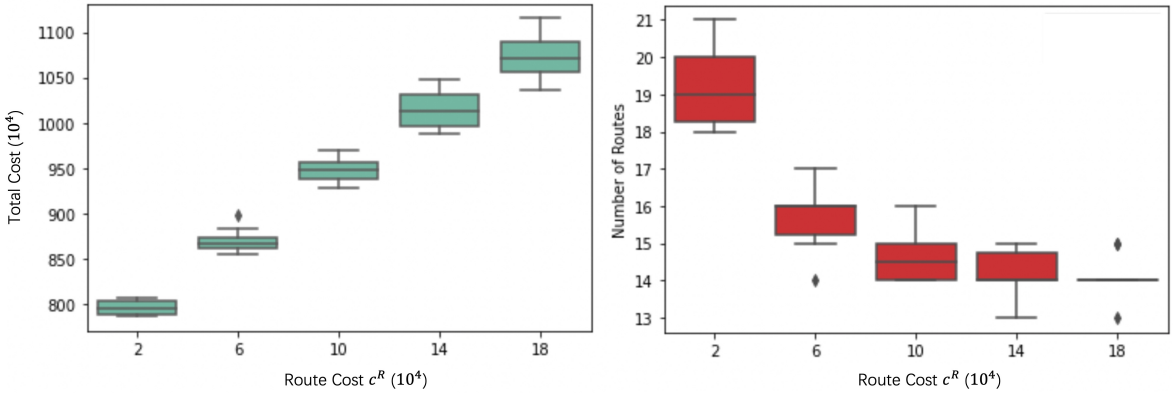


Fig. 7. Total cost and number of routes in the large-scale network at different setting of route cost (c^R).

5.5. Split visits and cluster demand

As mentioned in previous chapters, our model allows for the decomposition and satisfaction of a demand point's monitoring service by UAVs from multiple routes, including routes originating from different depots. We refer to this type of service scenario as a "split visit". However, in certain practical applications, due to objective conditions and operational requirements, some service forms do not permit "split visits". This means that each demand point can only be served by a UAV on a single route. Such service requirements impose limitations on the flexibility of monitoring network design.

In the following discussion, we aim to compare these two service scenarios and explore the potential efficiency loss in UAV operations resulting from a service form that prohibits "split visits". We examine the percentage of cost savings and the impact on the solution pattern that "split visits" offer, thereby gauging the extent of efficiency reduction.

We consider both uniform and cluster distribution for the demand points. In the cluster distribution case, 25 demand points and three candidate depots are scattered in each of the four quadrants, resulting in a total of 100 demand points and 12 candidate depots. Each quadrant is a 6 km $\times$ 6 km square, and the demand points and candidate depots are randomly located in a circle with a radius of 2 km around the center of the quadrant. In the uniform distribution case, 100 demand points and 12 candidate depots are randomly scattered in a circle with a radius of 4 km.

The monitoring requirement and the lower/upper bound of the single monitoring time are determined using the same way as before. All parameter settings remain consistent with "Setting A" as discussed in previous sub-chapters, except for the battery-constrained maximum flight time of the UAV. Specifically, it is set to 1.2 times the required flight time between the two farthest-spaced points in the network. This adjustment ensures that each randomly generated instance yields feasible solutions.

We generated 50 instances for each distribution case. For each instance, we considered both service scenarios: allowing "split visits" and disallowing "split visits"

(meaning a demand point could only be served by UAVs from the same routes). We solved each scenario 10 times using our proposed ALNS-based heuristic algorithm.

The boxplots of Figures 8 and 9 show the best total cost and average total cost of 10 runs from the 50 instances of uniform and cluster distribution, respectively. In each graph, the first boxplot corresponds to the scenario without "split visits", and the second one is the scenario with "split visits". On average, allowing split visits leads to the best total cost savings of approximately 2.37% and 5.14%, and average total cost savings of approximately 2.17% and 5.00% for uniformly and cluster distributed demand points respectively. It is not difficult to understand that the instances with cluster distribution demonstrate more significant cost savings through the implementation of "split visits". In comparison to the uniform distribution, each cluster in the cluster distribution contains fewer demand points for a route to select, and the monitoring requirements exhibit greater variation. Consequently, when "split visits" are not allowed, a less optimal demand point must be chosen to be visited by the route, resulting in a greater loss of UAV operational efficiency.

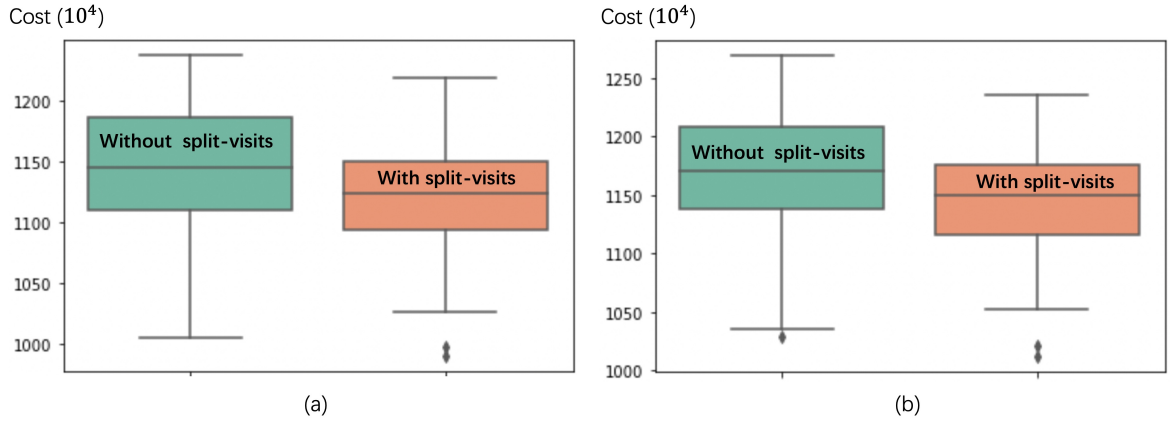


Fig. 8. Split visits are allowed vs. not allowed in terms of (a) best total cost (b) average total cost for uniformly distributed demand points.

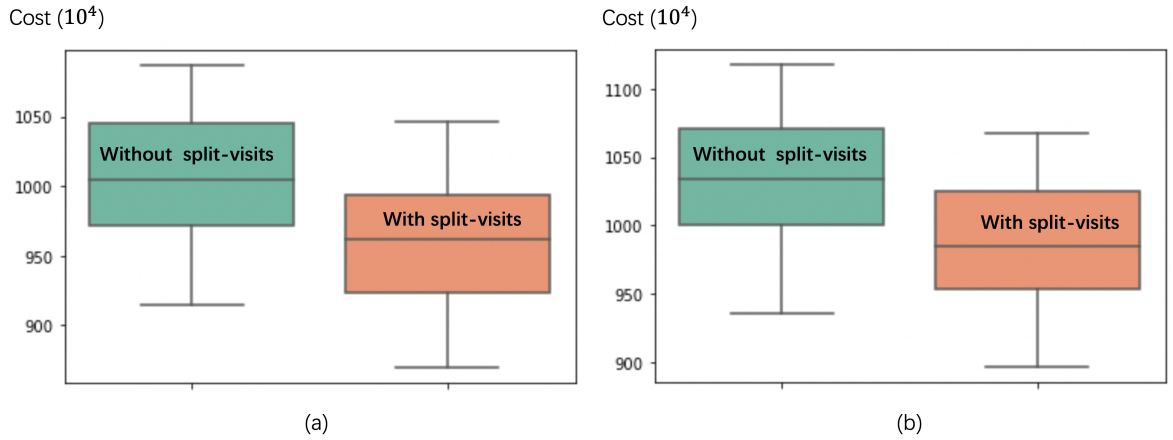


Fig. 9. Split visits are allowed vs. not allowed in terms of (a) best total cost (b) average total cost for cluster distributed demand points.

Upon examining the UAV service route pattern in the optimal solution for each instance of cluster-distributed demand points, we discover that regardless of whether split visits are allowed or not, each route exclusively connects depots and demand points within the same quadrant. There are no cross-region routes identified in any of the instances. This pattern can be attributed to the fact that the operating cost of UAVs in the air is primarily determined by battery consumption or flight time. To maximize UAV efficiency, the monitoring service provider aims to serve demand points within their respective surrounding areas, in order to optimize the ratio of monitoring time to total flight time. Deploying a UAV to fly to another cluster may not be cost-effective, especially when the clusters of demand points are located far apart. Figure 10 displays the results of a representative instance with cluster-distributed demand points, obtained using our ALNS-based heuristic algorithm for both scenarios – with and without split visits. The total cost is 908.35×10^4 and 1005.7×10^4, respectively.

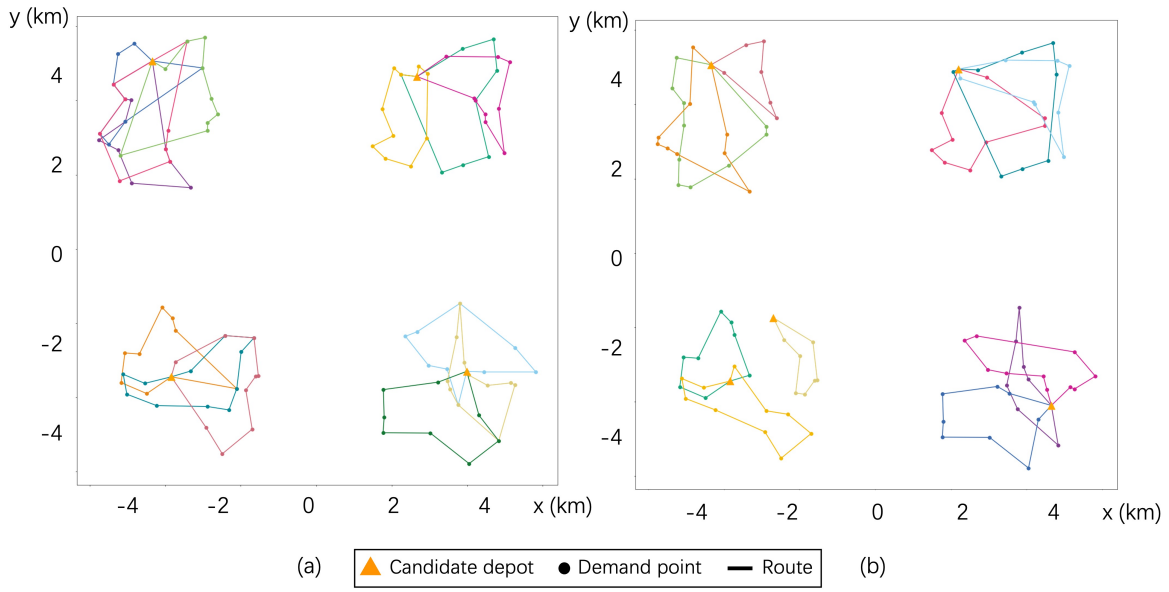


Fig. 10. Result of a instance with cluster-distributed demand points for scenarios (a) with split visits (total cost: 908.35×10^4) (b) without split visits (total cost: 1005.70×10^4).

6. Conclusion and future research

In this paper, we proposed a mixed-integer quadratically constrained programming model for the problem of designing an urban UAV monitoring network. The model considers the monitoring requirement and the single monitoring time of each demand point, as well as various aspects of construction, management, and operation costs. This research can help the monitoring service providers in cities to find a set of service routes and the corresponding depots that minimize the total cost while satisfying the monitoring requirement of each demand point.

To solve the problem, we decomposed it into a master problem and sub-problems. We proposed the Frequency-Time-Frequency Strategy, which is proven to find the optimal frequency and corresponding single service time for a route, nested within a proposed adaptive large neighborhood search (ALNS) based heuristic algorithm to solve the master problem. A series of computational experiments have shown the effectiveness of the proposed algorithm. We also demonstrated the applicability of our model and algorithm in a real-world case study of monitoring traffic flow in the Kowloon district of Hong Kong.

We conducted sensitivity tests to examine the effects of parameter changes on the total cost and the number of service routes in the solution, focusing on the route cost and the UAV battery-constrained maximum flight time. Additionally, we investigated the impact of split visits on the solution pattern. Results showed that allowing split visits can result in an average cost savings of approximately 2% and 5% for uniformly and cluster distributed demand points respectively. We also observed that when the demand point is cluster distribution, the routes tend to only provide monitoring services

to the demand points within one cluster in order to optimize the ratio of monitoring time to total flight time.

This paper illustrates the potential of using UAVs for daily monitoring networks in urban environments, and future research can expand in several directions. Firstly, the proposed model and algorithm assume that the monitoring requirements of demand points are deterministic and homogeneous in time. However, future research can modify the model and algorithm to handle dynamic monitoring scenarios, where monitoring requirements at demand points change over time. This can be achieved by incorporating predictive models of demand point behavior and using real-time data to adjust the monitoring network accordingly. Secondly, although this paper focuses on network design at the planning level, future research can extend the approach to route optimization in the operation phase to determine the operating schedule and the number of UAVs required for route coordination. Thirdly, the integration with other existing ground monitoring facilities can be considered, and the entire network can be optimized by using a multi-modal optimization method. These research directions can further enhance the effectiveness and efficiency of UAV monitoring network design in urban areas and contribute to the development of more efficient and effective solutions for real-world applications.

Acknowledgment. We would like to thank the anonymous referees for their constructive and helpful comments, which helped us to improve both the technical quality and exposition of the manuscript significantly. This study was partially supported by the Research Grants Council of Hong Kong through NSFC/RGC Joint Research Scheme (N_PolyU521/22), and The Hong Kong Polytechnic University (P0040900, P0041316).

Declaration of Generative AI and AI-assisted technologies in the writing process. During the preparation of this work, the authors used ChatGPT in order to polish some of the words, sentences, and grammar to improve the overall readability of the article. After using this tool/service, the authors reviewed and edited the content as needed and takes full responsibility for the content of the publication.

Appendix A Optimality of Frequency-Time-Frequency strategy

By denoting the total en-route time between the nodes as $\sum d_{ij}$ (since the vehicle routing is fixed) and the time for monitoring $t_{allowed} = q - \sum d_{ij}$, The sub-problem can be formulated as a minimization problem as follows:

$$\begin{aligned} \min w(c^C F + c^O F T) &= \min w(c^C + c^O \sum d_{ij}) F + \sum_{i \in I_r} w c^O E_i F \\ &= \min \alpha F + \sum_{i \in I_r} \beta E_i F \end{aligned} \tag{24}$$

where we let $\alpha = w(c^C + c^O \sum d_{ij}), \beta = wc^O$ and is subject to

$$-E_i F + h_i \leq 0, \forall i \in I_r \tag{25}$$

$$l_i - E_i \leq 0, \forall i \in I_r \tag{26}$$

$$E_i - u_i \leq 0, \forall i \in I_r \tag{27}$$

$$\sum_{i \in I_r} E_i - t_{allowed} \leq 0 \tag{28}$$

$$-F \leq 0 \tag{29}$$

Note that since the Hessian matrix is not semi-positive definite, the objective function (24) is non-convex (also non-linear). However, by simply denoting $Y_i = E_i F, \forall i \in I_r$, it can be converted to the following linear problem.

$$\min \alpha f + \sum_{i \in I_r} \beta Y_i \tag{30}$$

s.t.

$$-Y_i + h_i \leq 0, \forall i \in I_r \tag{31}$$

$$l_i F - Y_i \leq 0, \forall i \in I_r \tag{32}$$

$$Y_i - u_i F \leq 0, \forall i \in I_r \tag{33}$$

$$\sum_{i \in I_r} Y_i - t_{allowed} F \leq 0 \tag{34}$$

$$-F \leq 0 \tag{35}$$

A linear optimization problem satisfies the linear constraint qualification, thus its optimal solution must satisfy the Karush–Kuhn–Tucker conditions, i.e. equation (31) - (35) and,

$$\lambda_{1,i}(-Y_i + h_i) = 0, \forall i \in I_r \tag{36}$$

$$\lambda_{2,i}(l_i F - Y_i) = 0, \forall i \in I_r \tag{37}$$

$$\lambda_{3,i}(Y_i - u_i F) = 0, \forall i \in I_r \tag{38}$$

$$\lambda_4(\sum_{i \in I_r} Y_i - t_{allowed} F) = 0 \tag{39}$$

$$\lambda_5(-F) = 0 \tag{40}$$

$$\frac{\partial L}{\partial F} = \alpha + \sum_{i \in I_r} \lambda_{2,i} l_i - \sum_{i \in I_r} \lambda_{3,i} u_i - \lambda_4 t_{allowed} - \lambda_5 = 0 \tag{41}$$

$$\frac{\partial L}{\partial Y_i} = \beta - \lambda_{1,i} - \lambda_{2,i} + \lambda_{3,i} + \lambda_4 = 0, \forall i \in I_r \tag{42}$$

where $\lambda_{1,i}, \lambda_{2,i}, \lambda_{3,i}, \lambda_4, \lambda_5 \geq 0, , \forall i \in I_r$

First of all, we divide the problem into two cases according to whether λ_4 is greater than 0.

Case 1. $\lambda_4 > 0$.

This implies that $\sum_{i \in I_r} Y_i - t_{allowed} F = 0$ from equation (39) and we can generate a series of solutions. In such a case, the objective function can be modified as

$$\min \alpha f + \sum_{i \in I_r} \beta Y_i = \alpha F + \beta t_{allowed} F$$

Within this set of solutions, the objective function is only strictly increasing with the frequency of the route F, therefore we need to minimize the frequency F **within the feasible region** to get the candidate optimal solution.

Case 2. $\lambda_4 = 0$.

Under this case, we first observe that it is necessary that $\lambda_5 = 0$, since otherwise $F = 0$ and constraints (33) will be violated. This indicates that $F > 0$.

Moreover, it can be readily derived that it is not possible for all of the $\lambda_{3,i}, \forall i \in I_r$ to be zero simultaneously, since otherwise equation (41) will never hold.

With the above in mind, we only need to consider two cases: some $\lambda_{3,i}$ are zero and some are not; and all $\lambda_{3,i}$ are larger than zero. This further divides Case 2 into two sub-cases namely Case 2a and Case 2b.

Case 2a. Some of the $\lambda_{3,i}, i \in I_r$ are zero, but not all of them. For the demand point i with $\lambda_{3,i} > 0$, we have $Y_i = u_i F$ from equation (38). For the rest of the demand point i with $\lambda_{3,i} = 0$, to make the corresponding equation (42) hold, we must have either $\lambda_{1,i} > 0$ and that indicates $Y_i = h_i$ from equation (36) or $\lambda_{2,i} > 0$ and that indicates $Y_i = l_i f$ from equation (37) . Therefore, we generate a series of solutions.

In such a case, the objective function can be modified as

$$\min \alpha F + \sum_{i \in I_r} \beta Y_i = \alpha F + \beta \sum_{i \in I_r : \lambda_{3,i} = 0} u_i F + \beta \sum_{i \in I_r : \lambda_{1,i} = 0} h_i + \beta \sum_{i \in I_r : \lambda_{2,i} \neq 0} l_i F$$

Within this set of solutions, the objective function is only strictly increasing with the frequency of the route F, therefore we need to minimize the frequency F **within the feasible region** to get the candidate optimal solution.

Case 2b. All of the $\lambda_{3,i}, i \in I_r$ are larger than zero. This indicates that $F = Y_i = u_i F, \forall i \in I_r$. In such a case, the objective function can be modified as

$$min \alpha F + \sum_{i \in I_r} \beta Y_i = \alpha F + \beta \sum_{i \in I_r} u_i F$$

Within this set of solutions, the objective function is also only strictly increasing with the frequency of the route F, therefore we need to minimize the frequency F **within the feasible region** to get the candidate optimal solution.

To conclude, the optimal solution is the solution minimizing the frequency F within the feasible region.

We now explain why the solution obtained by the Frequency-Time-Frequency Strategy in Section 4.1 is optimal. First, Step-1 and Step-2 together generate an initial solution with its frequency to be the minimum frequency considering all constraints except for the battery-allowance (28). This initial solution is optimal solution if it is feasible in terms of battery-allowance. If this solution is not feasible, the algorithm then moves between adjacent candidate optimal solutions by iteratively increasing the frequency through Step-3 and Step-4 and stops when it achieves the optimum. Here is a easy way to further check the optimality of the solution obtained by the Frequency-Time-Frequency Strategy with a frequency F^*. The reader can easily derive that, any solution with a frequency $F < F^*$ is infeasible, and any solution with a frequency $F > F^*$ will encounter a higher cost in the objective function.

Appendix B The route-collaboration improvement problem

We begin by identifying all demand points that are served by at least two routes. We denote the set of all these routes as R_c. The demand points visited by route $r \in R_c$ are represented as $i, j, \ldots \in I_r$. The union of these sets I_r is denoted as I_c, which includes all the demand points under consideration. Additionally, we define the total en-route time between nodes as $(\sum d_{ij})r$ (since the vehicle routing is fixed) and the time allocated for monitoring as $Tallowed, r = q - (\sum d_{ij})_r$.

The route-collaboration improvement problem can then be formulated as a minimization problem, stated as follows:

$$\begin{aligned}\min \sum_{r\in R_c} w(F_r c^C + F_r T_r c^O) &= \min \sum_{r\in R_c} w(c^C + c^O(\sum d_{ij})_r)F_r + \sum_{r\in R_c}\sum_{i\in I_r} wc^O E_{ir}F_r \\ &= \min \sum_{r\in R_c} \alpha_r F_r + \sum_{r\in R_c}\sum_{i\in I_r} \beta E_{ir}F_r\end{aligned} \tag{43}$$

where we let $\alpha_r = w(c^C + c^O(\sum d_{ij})_r), \beta = wc^O$ and is subject to

$$-\sum_{r\in R_c} E_{ir}F_r + h_i \le 0, \forall i \in I_c \tag{44}$$

$$l_i - E_{ir} \le 0, \forall r \in R_c, \forall i \in I_r \tag{45}$$

$$E_{ir} - u_i \le 0, \forall r \in R_c, \forall i \in I_r \tag{46}$$

$$E_{ir} = 0, \forall r \in R_c, \forall i \in (I_c \setminus I_r) \tag{47}$$

$$\sum_{i\in I_r} E_{ir} - t_{allowed,r} \le 0, \forall r \in R_c \tag{48}$$

$$-F_r \leq 0, \forall r \in R_c \tag{49}$$

By simply denoting $Y_{ir} = E_{ir}F_r, \forall r \in R_c, \forall i \in I_c$, it can be converted to the following linear problem.

$$\min \sum_{r \in R_c} \alpha_r F_r + \sum_{r \in R_c} \sum_{i \in I_r} \beta Y_{ir} \tag{50}$$

subject to

$$-\sum_{r \in R_c} Y_{ir} + h_i \leq 0, \forall i \in I_c \tag{51}$$

$$l_i F_r - Y_{ir} \leq 0, \forall r \in R_c, \forall i \in I_r \tag{52}$$

$$Y_{ir} - u_i F_r \leq 0, \forall r \in R_c, \forall i \in I_r \tag{53}$$

$$Y_{ir} = 0, \forall r \in R_c, \forall i \in (I_c \setminus I_r) \tag{54}$$

$$\sum_{i \in I_r} Y_{ir} - t_{allowed,r} F_r \leq 0, \forall r \in R_c \tag{55}$$

$$-F_r \leq 0, \forall r \in R_c \tag{56}$$

This linear problem is solved by the commercial solver Gurobi whenever a new best solution is found during the ALNS search process.

References

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, *52*, 965–981.

Amarasingam, N., Salgadoe, A. S. A., Powell, K., Gonzalez, L. F., & Natarajan, S. (2022). A review of UAV platforms, sensors, and applications for monitoring of sugarcane crops. *Remote Sensing Applications: Society and Environment*, *26*, 100712.

Baldini, F., Anandkumar, A., & Murray, R. M. (2020). Learning pose estimation for UAV autonomous navigation and landing using visual-inertial sensor data. In *2020 American Control Conference (ACC)* (pp. 2961–2966). IEEE.

Barmpounakis, E. N., & Geroliminis, N. (2020). On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment. *Transportation Research Part C: Emerging Technologies*, *111*, 50–71.

Barmpounakis, E. N., Vlahogianni, E. I., & Golias, J. C. (2016). Unmanned aerial aircraft systems for transportation engineering: Current practice and future challenges. *International Journal of Transportation Science and Technology*, *5*, 111–122.

Barmpounakis, E. N., Vlahogianni, E. I., Golias, J. C., & Babinec, A. (2019). How accurate are small drones for measuring microscopic traffic parameters? *Transportation Letters*, *11*, 332–340.

Bashir, N., Boudjit, S., & Zeadally, S. (2022). A closed-loop control architecture of UAV and WSN for traffic surveillance on highways. *Computer Communications*, *190*, 78–86.

Carminati, M., Kanoun, O., Ullo, S. L., & Marcuccio, S. (2019). Prospects of distributed wireless sensor networks for urban environmental monitoring. *IEEE Aerospace and Electronic Systems Magazine*, *34*, 44–52.

Cheng, C., Adulyasak, Y., & Rousseau, L.-M. (2020). Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B: Methodological*, *139*, 364–387.

Daniel, K., & Wietfeld, C. (2011). Using public network infrastructures for UAV remote sensing in civilian security operations. *Homeland Security Affairs, Supplement*, *3*.

Demir, E., Bektaş, T., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, *223*, 346–359.

DJI (2023). Consumer drones comparison. http://https://www.dji.com/hk-en/products/comparison-consumer-drones?site=brandsite&from=nav/, Last accessed on 2023-02-25.

Drexl, M., & Schneider, M. (2015). A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, *241*, 283–308.

Dukkanci, O., Kara, B. Y., & Bektaş, T. (2021). Minimizing energy and cost in range-limited drone deliveries with speed optimization. *Transportation Research Part C: Emerging Technologies*, *125*, 102985.

Engin, Z., van Dijk, J., Lan, T., Longley, P. A., Treleaven, P., Batty, M., & Penn, A. (2020). Data-driven urban management: Mapping the landscape. *Journal of Urban Management*, *9*, 140–150.

Ghilas, V., Demir, E., & Van Woensel, T. (2016). A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands. *Transportation Research Part B: Methodological*, *91*, 34–51.

Hari, S. K. K., Rathinam, S., Darbha, S., Kalyanam, K., Manyam, S. G., & Casbeer, D. (2021). Optimal UAV route planning for persistent monitoring missions. *IEEE Transactions on Robotics*, *37*, 550–566.

Hayat, S., Yanmaz, E., & Muzaffar, R. (2016). Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Communications Surveys & Tutorials*, *18*, 2624–2661.

Kako, S., Morita, S., & Taneda, T. (2020). Estimation of plastic marine debris volumes on beaches using unmanned aerial vehicles and image processing based on deep learning. *Marine Pollution Bulletin*, *155*, 111127.

Kashef, M., Visvizi, A., & Troisi, O. (2021). Smart city as a smart service system: Human-computer interaction and smart city surveillance systems. *Computers in Human Behavior*, *124*, 106923.

Koç, Ç. (2016). A unified-adaptive large neighborhood search metaheuristic for periodic location-routing problems. *Transportation Research Part C: Emerging Technologies*, *68*, 265–284.

Koper, C. S., Taylor, B. G., & Woods, D. J. (2013). A randomized test of initial and residual deterrence from directed patrols and use of license plate readers at crime hot spots. *Journal of Experimental Criminology*, *9*, 213–244.

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.

Li, H., Chen, J., Wang, F., & Bai, M. (2021). Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: A review. *European Journal of Operational Research*, *294*, 1078–1095.

Liu, W., Zhang, T., Huang, S., & Li, K. (2022). A hybrid optimization framework for UAV reconnaissance mission planning. *Computers & Industrial Engineering*, *173*, 108653.

Liu, X., Gao, L., Guan, Z., Song, Y., & Zhang, R. (2016). A multi-objective optimization model for planning unmanned aerial vehicle cruise route. *International Journal of Advanced Robotic Systems*, *13*, 116.

Liu, X., Gao, L., Guang, Z., & Song, Y. (2013). A UAV allocation method for traffic surveillance in sparse road network. *Journal of Highway and Transportation Research and Development (English Edition)*, *7*, 81–87.

Lu, H., Fu, X., Liu, C., Li, L., He, Y., & Li, N. (2017). Cultivated land information extraction in UAV imagery based on deep convolutional neural network and transfer learning. *Journal of Mountain Science*, *14*, 731–741.

Ma, W., Zeng, L., & An, K. (2023). Dynamic vehicle routing problem for flexible buses considering stochastic requests. *Transportation Research Part C: Emerging Technologies*, *148*, 104030.

Mahajan, V., Barmpounakis, E. N., Alam, M. R., Geroliminis, N., & Antoniou, C. (2023). Treating noise and anomalies in vehicle trajectories from an experiment with a swarm of drones. *IEEE Transactions on Intelligent Transportation Systems*, .

Marais, J., Meurie, C., Attia, D., Ruichek, Y., & Flancquart, A. (2014). Toward accurate localization in guided transport: Combining GNSS data and imaging information. *Transportation Research Part C: Emerging Technologies*, *43*, 188–197.

Máthé, K., & Buşoniu, L. (2015). Vision and control for UAVs: A survey of general methods and of inexpensive platforms for infrastructure inspection. *Sensors*, *15*, 14887–14916.

Mazerolle, L., Hurley, D., & Chamlin, M. (2002). Social behavior in public space: An analysis of behavioral adaptations to CCTV. *Security Journal*, *15*, 59–75.

Mersheeva, V., & Friedrich, G. (2012). Routing for continuous monitoring by multiple micro AVs in disaster scenarios. In *ECAI 2012* (pp. 588–593). IOS Press.

Mersheeva, V., & Friedrich, G. (2015). Multi-UAV monitoring with priorities and limited energy resources. In *Proceedings of the International Conference on Automated Planning and Scheduling* (pp. 347–355). volume 25.

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.

Naccache, S., Côté, J.-F., & Coelho, L. C. (2018). The multi-pickup and delivery problem with time windows. *European Journal of Operational Research*, *269*, 353–362.

Nagy, G., & Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research*, *177*, 649–672.

Petráček, P., Krátkỳ, V., Petrlík, M., Báča, T., Kratochvíl, R., & Saska, M. (2021). Large-scale exploration of cave environments by unmanned aerial vehicles. *IEEE Robotics and Automation Letters*, *6*, 7596–7603.

Pinto, R., & Lagorio, A. (2022). Point-to-point drone-based delivery network design with intermediate charging stations. *Transportation Research Part C: Emerging Technologies*, *135*, 103506.

Pinto, R., Zambetti, M., Lagorio, A., & Pirola, F. (2020). A network design model for a meal delivery service using drones. *International Journal of Logistics Research and Applications*, *23*, 354–374.

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, *34*, 2403–2435.

Poikonen, S., Wang, X., & Golden, B. (2017). The vehicle routing problem with drones: Extended models and connections. *Networks*, *70*, 34–43.

Polaris Market Research (2021). Commercial UAV (unmanned aerial vehicle) market share, size, trends & industry analysis report by type (rotary blade UAV, fixed wing UAV); by end-use (agriculture, energy & public utilities, construction, media & entertainment, government); by region; segment forecast, 2021 - 2028. https://www.polarismarketresearch.com/industry-analysis/commercial-uav-market/request-for-sample.

Prodhon, C., & Prins, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research*, *238*, 1–17.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *Computer Vision and Pattern Recognition*, *1804*, 1–6.

Ropke, S., & Pisinger, D. (2006a). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, *40*, 455–472.

Ropke, S., & Pisinger, D. (2006b). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, *171*, 750–775.

Sathyaraj, B. M., Jain, L. C., Finn, A., & Drake, S. (2008). Multiple UAVs path planning algorithms: A comparative study. *Fuzzy Optimization and Decision Making*, *7*, 257–267.

Schermer, D., Moeini, M., & Wendt, O. (2019). A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C: Emerging Technologies*, *106*, 166–204.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *International Conference on Principles and Practice of Constraint Programming* (pp. 417–431). Springer.

Tadaros, M., & Migdalas, A. (2022). Bi-and multi-objective location routing problems: classification and literature review. *Operational Research*, *22*, 4641–4683.

Tiemann, J., & Wietfeld, C. (2017). Scalable and precise multi-UAV indoor navigation using TDOA-based UWB localization. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (pp. 1–7). IEEE.

Valente, J., Sanz, D., Barrientos, A., Del Cerro, J., Ribeiro, Á., & Rossi, C. (2011). An air-ground wireless sensor network for crop monitoring. *Sensors*, *11*, 6088–6108.

Wang, X., Poikonen, S., & Golden, B. (2017). The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, *11*, 679–697.

Wang, Y., Lei, L., Zhang, D., & Lee, L. H. (2020). Towards delivery-as-a-service: Effective neighborhood search strategies for integrated delivery optimization of e-commerce and static O2O parcels. *Transportation Research Part B: Methodological*, *139*, 38–63.

Wang, Z., & Sheu, J.-B. (2019). Vehicle routing problem with drones. *Transportation research part B: methodological*, *122*, 350–364.

Wen, X., & Wu, G. (2022). Heterogeneous multi-drone routing problem for parcel delivery. *Transportation Research Part C: Emerging Technologies*, *141*, 103763.

Windras Mara, S. T., Norcahyo, R., Jodiawan, P., Lusiantoro, L., & Rifai, A. P. (2022). A survey of adaptive large neighborhood search algorithms and applications. *Computers & Operations Research*, *146*, 105903.

Xia, Y., Zeng, W., Zhang, C., & Yang, H. (2023). A branch-and-price-and-cut algorithm for the vehicle routing problem with load-dependent drones. *Transportation Research Part B: Methodological*, *171*, 80–110.

Yang, Z., Yu, X., Dedman, S., Rosso, M., Zhu, J., Yang, J., Xia, Y., Tian, Y., Zhang, G., & Wang, J. (2022). UAV remote sensing applications in marine monitoring: Knowledge visualization and review. *Science of The Total Environment*, *838*, 155939.

Zhang, G., & Hsu, L.-T. (2018). Intelligent GNSS/INS integrated navigation system for a commercial UAV flight control system. *Aerospace Science and Technology*, *80*, 368–380.

Zhang, J., Campbell, J. F., Sweeney II, D. C., & Hupman, A. C. (2021). Energy consumption models for delivery drones: A comparison and assessment. *Transportation Research Part D: Transport and Environment*, *90*, 102668.

Zhang, J., & Huang, H. (2021). Occlusion-aware UAV path planning for reconnaissance and surveillance. *Drones*, *5*, 98.

Zhu, J., Sun, K., Jia, S., Li, Q., Hou, X., Lin, W., Liu, B., & Qiu, G. (2018). Urban traffic density estimation based on ultrahigh-resolution UAV video and deep neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*, 4968–4981.