Transportation Research Part E

Persistent Monitoring with Drones: A Column Generation-Based Approach

--Manuscript Draft--

Manuscript Number:	
Article Type:	Full Length Article
Keywords:	Aerial surveillance, Drone scheduling, Persistent monitoring, Column generation
Corresponding Author:	Jun Xia Shanghai Jiao Tong University Shanghai, CHINA
First Author:	Tangfei Li
Order of Authors:	Tangfei Li
	Jun Xia
	Biao Yuan
	Silong Zhang
	Sixiang Zhao
Abstract:	We investigate a drone scheduling problem arising from aerial surveillance, where a fleet of homogeneous drones is deployed to persistently monitor a set of ground targets within a cyclic planning time horizon. An important feature of the problem is that each target has a requirement on the monitoring frequency, namely, the time gap between consecutive visits to a target must not exceed a given time length. The objective is to find a set of scheduled flight trips for drones with minimum total duration time, such that the monitoring frequency requirements of all targets can be satisfied. We introduce a cyclic time-expanded network to represent the problem, based on which a route-based integer linear programming formulation is proposed. To solve this formulation, we develop a column generation-based approach to obtain an optimal or a near-optimal solution. Our numerical study shows that the proposed column generation-based approach can effectively obtain high-quality solutions.
Suggested Reviewers:	Lingxiao Wu Assistant Professor, The Hong Kong Polytechnic University lingxiao-leo.wu@polyu.edu.hk Dr Lingxiao Wu is an expert in transportation planning, especially in aviation operations modelling and optimization.
	Ning Zhu Professor, Tianjin University zhuning@tju.edu.cn Prof Ning Zhu is an expert in transportation and is familar with optimization of drone operations.

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY
安泰经管学院
ANTAI COLLEGE
Economics·Management

Jun Xia
Associate Professor
Shanghai Jiao Tong University
Antai College of Economics & Management
lgtxiaj@sjtu.edu.cn
86-21-62932393

April 24, 2024

Dear Professor Wang,

We wish to submit a research article entitled "Persistent Monitoring with Drones: A Column Generation-Based Approach" for its possible publication in *Transportation Research Part E*. We confirm that this work is original and has not been published elsewhere, nor is it currently under consideration for publication elsewhere.

The paper was submitted to the "AI-Driven Mobility" special issue of *Transportation Research Part E* with Manuscript ID TRE-D-24-00975. However, it has been returned with a suggestion to resubmit the paper for consideration in a regular issue.

In this study, we investigate a drone scheduling problem arising from a persistent monitoring application, in which a fleet of drones is routed to collaboratively monitor a set of targets with certain monitoring frequency requirements. For the study of this problem, we have made the following contributions:

- First, we introduce a more general drone routing scenario for persistent monitoring, wherein drones are scheduled possibly with multiple trips for collaborative monitoring and target observations adhere to specified frequency requirements. To model the problem, we employ a cyclic time-expanded network and propose a route-based integer linear programming formulation.
- Second, we develop a column generation-based approach to solve the proposed formulation. In this approach, we first employ the column generation technique to solve the linear relaxation of the formulation. Utilizing the columns and the valid lower bound computed from column generation, we apply a hybrid strategy combining route enumeration and branch-and-price to compute an integral solution for the original problem.
- Third, we conduct extensive computational experiments to validate the effectiveness of our proposed solution approach. Specifically, we solve the proposed formulation with up to 24 time segments and up to 35 targets, achieving optimality for most small-size instances and near-optimality for medium-size instances. The impact of the monitoring frequency requirements on the obtained solution is also analyzed.

We believe that this manuscript is appropriate for publication by *Transportation Research Part E* as it focuses on applying operations research techniques to tackle a real problem arising from the logistics and transportation industry, which highly relates to the journal's scope.

Jun Xia
Assistant Professor
Shanghai Jiao Tong University
Antai College of Economics & Management
lgtxiaj@sjtu.edu.cn
86-21-62932393

This work was supported by National Natural Science Foundation of China under Grant (72171147, 72301170, and 72031006). We have no conflicts of interest to disclose. Thanks for your consideration of this manuscript.

With my warmest regards,
Jun Xia.

Comments from the editors and reviewers:
The guest editor considers this paper out of scope of the SPECIAL issue. I am willing to consider the paper if it is resubmitted to a regular issue.

Thank you for submitting your manuscript "Persistent Monitoring with Drones: A Column Generation-Based Approach" to our special issue on Emerging AI-Driven Smart and Sustainable Mobility. I have carefully read your paper and appreciate the high quality of your work and writing, which reflects solid research efforts. However, I regret to inform you that the research content of this paper does not align well with the theme of our special issue.

Our response: Thank you for the suggestion. We have resubmitted this manuscript to the regular issue of *Transportation Research Part E*. Many thanks for your reconsideration.

Research Highlights

- Introduce a drone scheduling problem for persistent monitoring.
- Construct a cyclic time-expanded network to represent the drone scheduling problem.
- Propose a route-based integer linear programming formulation and develop a column generation-based approach to solve the formulation.
- Conduct numerical experiments to demonstrate the effectiveness and efficiency of the proposed solution approach.

Title Page (with Author Details)

Persistent Monitoring with Drones: A Column Generation-Based Approach

Tangfei Li[a], Jun Xia[a], Biao Yuan[a], Silong Zhang[b], Sixiang Zhao[a]

[a] *Data-Driven Management Decision-Making Lab, Antai College of Economics & Management, Shanghai Jiao Tong University, China*
[b] *Department of Logistics and Maritime Studies, Faculty of Business, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

Abstract

We investigate a drone scheduling problem arising from aerial surveillance, where a fleet of homogeneous drones is deployed to persistently monitor a set of ground targets within a cyclic planning time horizon. An important feature of the problem is that each target has a requirement on the monitoring frequency, namely, the time gap between consecutive visits to a target must not exceed a given time length. The objective is to find a set of scheduled flight trips for drones with minimum total duration time, such that the monitoring frequency requirements of all targets can be satisfied. We introduce a cyclic time-expanded network to represent the problem, based on which a route-based integer linear programming formulation is proposed. To solve this formulation, we develop a column generation-based approach to obtain an optimal or a near-optimal solution. Our numerical study shows that the proposed column generation-based approach can effectively obtain high-quality solutions.

Keywords: Aerial surveillance, Drone scheduling, Persistent monitoring, Column generation

Email address: `lgtxiaj@sjtu.edu.cn` (Jun Xia)

Persistent Monitoring with Drones: A Column Generation-Based Approach

Abstract

We investigate a drone scheduling problem arising from aerial surveillance, where a fleet of homogeneous drones is deployed to persistently monitor a set of ground targets within a cyclic planning time horizon. An important feature of the problem is that each target has a requirement on the monitoring frequency, namely, the time gap between consecutive visits to a target must not exceed a given time length. The objective is to find a set of scheduled flight trips for drones with minimum total duration time, such that the monitoring frequency requirements of all targets can be satisfied. We introduce a cyclic time-expanded network to represent the problem, based on which a route-based integer linear programming formulation is proposed. To solve this formulation, we develop a column generation-based approach to obtain an optimal or a near-optimal solution. Our numerical study shows that the proposed column generation-based approach can effectively obtain high-quality solutions.

Keywords: Aerial surveillance, Drone scheduling, Persistent monitoring, Column generation

1. Introduction

Aerial surveillance, involving monitoring operations using passive and active remote sensors, is widely used in commercial and military applications such as border patrol and environmental monitoring/enforcement. Over the past twenty years, the demand for surveillance activities has surged. According to Insights (2021), the global aerial surveillance market size is expected to grow to \$7.70 billion by 2028, at a compound annual growth rate of 6.14%.

With the rapid development of unmanned aerial vehicles (UAVs) or drones, surveillance systems based on drones have attracted much attention (Zandieh et al., 2023). Drones can easily be equipped with sophisticated sensors to conduct remote monitoring and survey a relatively large-scale area from altitude (Zhang et al., 2021). Moreover, drones have fewer geographic limitations and require less manual labor, making them ideal for boring or dangerous tasks, such as information collection for military purposes (Xia et al., 2017; Moskal et al., 2023) and damage assessment in humanitarian relief (Zhang et al., 2023).

This study focuses on a drone scheduling problem for aerial surveillance, where a predefined set of targets must be monitored persistently by a fleet of homogeneous drones within a cyclic planning time horizon. Each target has a monitoring frequency requirement, meaning that a target needs to be visited repeatedly and there is a maximum time gap between two consecutive visits. The importance of targets varies, and different monitoring frequency levels are assigned accordingly. Higher priority targets require more frequent visits, leading to shorter required time gaps between consecutive visits. A set of drone trips is performed to execute the persistent monitoring tasks. Each trip begins at a depot, visits a sequence of targets, and returns to the depot. The duration time of a trip depends on the sequence of targets visited and is confined to the drone's battery power limit. A drone can perform multiple trips by replacing its battery at the depot. The goal of this drone scheduling problem is to select a set of feasible trips for the drones, such that the total duration time of the selected trips is minimized and all targets' monitoring frequency requirements are satisfied within the cyclic planning time horizon.

The problem investigated in this paper differs from conventional drone routing optimiza-

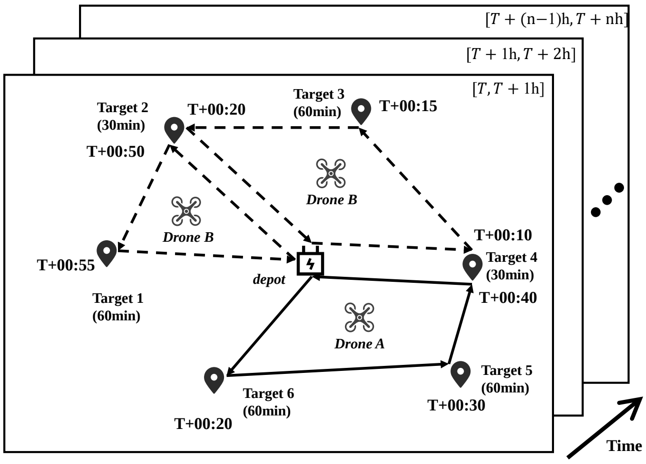

Fig. 1. Illustration of persistent monitoring with drones over an infinite time horizon.

tion problems in existing literature, mainly in considering monitoring frequency requirements. To achieve persistent monitoring, a target may be visited multiple times by either the same drone or different drones within a single trip or across multiple trips. Such collaborative monitoring with a drone fleet greatly enhances monitoring efficiency. However, the underlying optimization becomes more complicated. When monitoring frequency requirements are overlooked, the drone scheduling problem can be seen as a multi-trip vehicle routing problem, which is NP-hard (Cattaruzza et al., 2016). To address the persistent monitoring under an infinite time horizon, we consider the drone scheduling problem on a cyclic planning time horizon, as illustrated in Figure 1. Thus, any feasible solution to the problem provides a cyclic solution for persistent monitoring over an infinite time horizon.

This study offers the following contributions and main results. First, we introduce a more general drone routing scenario for persistent monitoring, wherein drones are scheduled possibly with multiple trips for collaborative monitoring and target observations adhere to specified frequency requirements. To model the problem, we employ a cyclic time-expanded network and propose a route-based integer linear programming formulation. Second, we develop a column generation-based approach to solve the proposed formulation. In this approach, we first employ the column generation technique to solve the linear relaxation of the formulation. Utilizing the columns and the valid lower bound computed from column generation, we apply a hybrid strategy combining route enumeration and branch-and-price to compute an integral solution for the original problem. Third, we conduct extensive computational experiments to validate the effectiveness of our proposed solution approach. Specifically, we solve the proposed formulation with up to 24 time segments and up to 35 targets, achieving optimality for most small-size instances and near-optimality for medium-size instances. The impact of the monitoring frequency requirements on the obtained solution is also analyzed.

The remainder of this paper is organized as follows. Section 2 comprehensively reviews the related work. In Section 3, we describe the problem based on a cyclic time-expanded network and present the mathematical formulation. In Section 4, we introduce a column generation-based approach and elaborate its main steps. Computational results are reported and discussed in Section 5. Finally, Section 6 presents the conclusion and future directions of this study.

2. Literature Review

Optimization and planning problems associated with drone applications have attracted increasing attention in recent years. Otto et al. (2018) provide a comprehensive overview of drone-related optimization problems, categorizing them into several distinct types of operations, such as area coverage, search operations, location routing, etc. Our study focuses on routing drones among a predefined set of candidate locations for monitoring purposes, which is relevant to the drone routing problem (DRP) studied in the literature.

Most DRPs studied in the literature focus on parcel delivery, aiming to plan efficient working paths for drone-only or truck-drone cooperative operations while addressing last-mile delivery challenges. One may refer to Wang et al. (2020), Macrina et al. (2020), Eskandaripour and Boldsaikhan (2023), and Liang and Luo (2022) for a broader survey of relevant studies.

Target monitoring is another important application of the DRP, where most existing studies assume each monitored target to be visited exactly once. Shen et al. (2020) investigate a dynamic multi-UAV path planning problem for detecting ship emissions in ports. The positions and quantity of the ships to be detected are dynamic, as well as the time required to detect each ship. A fast particle swarm optimization-based heuristic method is developed to tackle the dynamic detection and plan the UAVs' route for detecting each vessel once at near-optimal cost and time. Zeng et al. (2022) focus on a truck-drone routing problem for surveillance tasks, where all targets are monitored by the drone exactly once. The truck is allowed to perform a battery swap and ship the drone from one location to the other. They consider the synchronization constraint of both vehicles during scheduled rendezvous and propose an efficient neighborhood search heuristic to solve the problem.

Instead of a requirement to monitor all targets, some literature has explored the maximization of profits achieved through selective target monitoring, particularly when the number of available drones is limited. Xia et al. (2019) investigate a drone scheduling problem that involves monitoring real-time sailing vessels using multiple drones and depots while allowing a drone to perform multiple trips. The objective is to maximize the total weighted number of vessels monitored. They model the dynamics of the sailing vessels using a deterministic time-dependent location function and formulate the problem as a flow-based optimization based on a time-expanded network. To obtain near-optimal solutions, they develop a Lagrangian relaxation-based solution method. Dasdemir et al. (2022) focus on a drone routing problem aimed at maximizing total prizes. The prizes associated with each target depend on its visiting time. They formulate the problem as a mixed-integer linear program and devise a hybrid algorithm to generate approximate Pareto-optimal frontier solutions. Fang et al. (2023) investigate a team orienteering problem for routing multi-UAVs in landslide monitoring, where areas are divided into mandatory and optional groups based on their emergency levels. They aim to maximize the geological information collected from the monitored areas and develop a neural network-based heuristic method.

In persistent monitoring, it is essential to monitor targets repetitively, resulting in multiple visits to a single target. However, the requirement for the time gap between two consecutive visits is rarely considered. Zhou et al. (2023) study a multi-depot location-routing problem of UAVs, where each demand point's monitoring requirement is considered the minimum proportion of time a UAV spends in urban monitoring. The UAVs are confined to a maximum time for monitoring each target, and a target can be monitored by multiple UAVs. In the traffic monitoring scenario explored by Li et al. (2018), traffic edges require repetitive drone monitoring to capture and update information on the fluctuating traffic demands. To incorporate the visiting time gap requirement in persistent monitoring, Hari et al. (2021) consider minimizing the maximum time gap between consecutive visits to a target, thus optimizing the performance of monitoring tasks. They consider a single drone to visit multiple targets with equal priority, which is a generalization of the traveling salesman problem (TSP) by allowing the drone to visit the depot within a cycle.

The DRP literature concerning persistent monitoring has recently considered the requirement for a target's monitoring frequency. Drucker et al. (2019) study the drone cyclic-routing problem for persistent monitoring of targets under a monitoring frequency requirements constraint. A target can be visited multiple times during a trip or by different drones, but each drone only performs a single trip. To determine the minimum number of drones, they propose a portfolio approach that combines the strengths of constraint solving and model checking. Scott et al. (2023) consider a multi-depot TSP with monitoring frequency requirements constraint of targets. To simplify the problem, each drone performs a single trip, and each target is monitored exactly once by the same drone. The length of each drone's trip is bounded by monitoring the frequency of the targets visited in each trip. The drone team's objective is to minimize the total travel time of all UAVs. Lagrangian relaxation-based technique is applied, and three dual update methods are utilized to find tight upper bounds. Fauske et al. (2020) study how to route heterogeneous agents to sufficiently monitor targets to satisfy the frequency requirements of targets during the planning time horizon as much as possible, whose objective is to maximize the total value of the covered observations. To tackle the problem, they introduce a novel time-indexed integer linear programming formulation and develop a delayed row and column generation method.

Table 1 summarizes the essential features considered in the drone routing optimization for persistent monitoring. In this study, we consider a fleet of multiple drones, allowing for a flexible drone-target monitoring relationship. Moreover, we address the challenge of accommodating multiple trips, considering monitoring targets with diverse priorities, and

Table 1. Summary on relevant works to drone routing for persistent monitoring

Papers	Fleet	Target	Route	Priority	Frequency	Plan	Objective	Method
Li et al. (2018)	n	n-1	S	No	Null	Acyclic	Min total cost/time	Local branching
Fauske et al. (2020)	n	n-n	S	Yes	Soft	Acyclic	Max total profit	Column generation
Hari et al. (2021)	1	1-n	M	No	Soft	Cyclic	Min-max time gap	MIP solver
Drucker et al. (2019)	n	n-n	S	Yes	Hard	Cyclic	Min drone number	Portfolio approach
Zhou et al. (2023)	n	n-1	M	No	Null	Acyclic	Min total cost/time	ALNS heuristic
Scott et al. (2023)	n	1-1	S	Yes	Hard	Cyclic	Min total cost/time	Lagrangian relaxation
This study	n	n-n	M	Yes	Hard	Cyclic	Min total cost/time	Column generation

Notes. **Fleet**: number of drones considered (1/n); **Target**: drone-target monitoring relationship (1-1/1-n/n-1/n-n); **Route**: number of routes performed by each drone (S/M: Single-trip/Multiple-trip); **Priority**: whether to consider diverse monitoring priorities (Yes/No); **Frequency**: monitoring frequency constraints (Hard/Soft/Null); **Plan**: planning time horizon (Cyclic/Acyclic).

treating monitoring frequency requirements as hard constraints. Incorporating these features helps explore more effective drone operations for persistent monitoring.

3. Problem Description and Mathematical Formulation

We formally describe the drone scheduling problem in Section 3.1. A cyclic time-expanded network is introduced to represent the problem in Section 3.2. A route-based integer programming formulation is proposed in Section 3.3.

3.1. Problem Description

The drone scheduling problem is described on a directed graph $G = (V_0, E)$, where V_0 represents the vertex set and E represents the edge set. The vertex set $V_0 = V \cup \{0\}$ consists of a set $V = \{1, 2, ..., |V|\}$ of target vertices and a depot vertex $\{0\}$. The edge set is defined by $E = \{(u, v) : u \in V_0, v \in V_0, u \neq v\}$. Each target vertex $v \in V$ has a non-negative monitoring frequency level denoted by Δ_v, which specifies an upper limit on the time gap between consecutive visits to target v. A cyclic planning time horizon $[0, H]$ is considered for the target monitoring. Without loss of generality, we assume that each target must be visited at least once during the cyclic planning time horizon. A fleet set $K = \{1, 2, ..., |K|\}$ of homogeneous drones initially available at the depot is deployed to perform flight trips to monitor the targets. The time taken by a drone to traverse an edge $(u, v) \in E$ is denoted as τ_{uv}. A feasible trip begins at the depot vertex, visits a sequence of target vertices, and ultimately returns to the depot vertex. Each drone trip is constrained by a maximum duration time of T_d due to the battery power limit.

The following conditions are considered to generalize the drone routing scenario and enhance the monitoring efficiency. (i) A drone is allowed to undertake multiple trips within the planning time horizon. This flexibility enables the drones to revisit targets as needed. (ii) During each trip, a drone can visit a target multiple times. This ensures that the target is adequately monitored to meet the monitoring frequency requirements. (iii) A target can be visited multiple times by different drones or by different trips of the same drone. This allows for increased coverage and efficient utilization of available drones.

In the problem, we assume no operating time at each vertex. Namely, we disregard the time needed for battery replacement, target monitoring, or the drone launching/landing times within a drone trip. These operating times at vertices can be easily incorporated by adjusting the time assigned to each edge.

The problem aims to design a set of scheduled routes for drones to monitor targets and to minimize the flight time of all drones' routes during a cyclic planning time horizon while satisfying the following constraints: (i) Each route starts and ends at the depot. (ii) The frequency requirement of each target should be satisfied. (iii) The drone duration time and fleet size constraints must be satisfied.

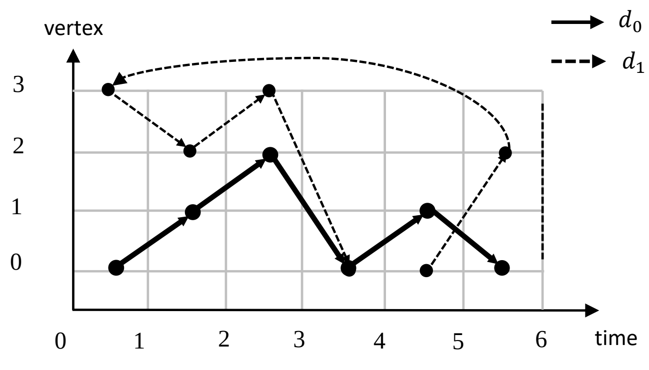

Fig. 2. A feasible solution of the illustrated example.

An illustrative example of the proposed drone scheduling problem is as follows. Consider a cyclic planning time horizon $[0, 6]$ with time length $H = 6$ and a graph that consists of a single depot vertex $\{0\}$ and three target vertices $V = \{1, 2, 3\}$. Two drones are deployed, i.e., $|K| = 2$. The maximum duration time of a trip is set as $T_d = 5$. The flight times on all edges are set to ones, i.e., $\tau_{uv} = 1$ for $(u, v) \in E$. The monitoring frequency levels are set as $\Delta_1 = \Delta_2 = 3$ and $\Delta_3 = 4$. A feasible solution of the illustrative example mentioned above is presented in Fig. 2. In the solution, drone d_0 performs two trips, and drone d_1 performs one trip. Target 1 is monitored by drone d_0 in two trips, while target 3 is monitored twice by drone d_1 within a single trip. Target 2 is monitored collaboratively by the two drones.

3.2. Modeling the Problem on Cyclic Time-expanded Network

We model the drone scheduling problem on a cyclic time-expanded network. The planning time horizon $[0, H]$ is discretized into a set $T = \{1, 2, ..., T_{\max}\}$ of discrete time segments, where each time segment $t \in T$ corresponds to a time interval $[t - 1, t)$ of length $H/T_{\max}$. Note that $T_{\max}$ is a positive integer number that can be set arbitrarily, which influences the accuracy of the discretization. In the network, all targets and the depot are associated with the time segments, yielding a set of target-time nodes $N_0 = \{(v, t) : v \in V_0, t \in T\}$. For any pair of nodes (v, t) and (v', t'), an arc is formed from (v, t) to (v', t') if $t' - t = \tau_{vv'}, t' > t$ or $T_{\max} - t + t' = \tau_{vv'}, t' < t$ holds in the cyclic time-expanded network. For clarity, let i be the index of the target-time node, with the corresponding target (or depot) and time segment denoted by $v(i)$ and $t(i)$, respectively. Let $A = \{(i, j) : i \in N_0, j \in N_0\}$ be the set of all the generated arcs. For arc $a \in A$, its left start node and right end node are indexed by $l(a)$ and $r(a)$, respectively. The flight time associated with arc $a \in A$ is thus computed by $t_{l(a)r(a)}$. A cyclic time-expanded network, denoted by $G' = (N_0, A)$, is then constructed.

Let R be a complete route set, where $r = (i_1, i_2, ..., i_r) \in R$ corresponds to a feasible route with the flight time c_r that does not exceed the drone's maximum duration time T_d and $v(i_1) = v(i_r) = 0$. For each $v \in V$, it is required that the number of time segments between consecutive visits to target v is no greater than Δ_v, implying that target v must be visited at least once for every Δ_v continuous time segments in G'.

Let $N = \{(v, t) : v \in V, t \in T\}$ represent the set of nodes concerning candidate target observations. To address the monitoring frequency requirement, we introduce $W(t, v)$ as a set of time segments for $(v, t) \in N$, such that $W(t, v) = \{t - \Delta_v + 1, ..., t\}$ if $t \geq \Delta_v$ and $W(t, v) = \{1, ..., t\} \bigcup \{T_{\max} - \Delta_v + 1 + t, ..., T_{\max}\}$ if $t < \Delta_v$. Based on $W(t, v)$, define $N'_i = \{(v(i), t) : t \in W(t(i), v(i))\}$ as the subset of target observations covered by the monitoring at node i. A target observation (v, t) is covered by a route if it contains at least one node that covers (v, t). Let N_r be the set of target observations covered by route r. Let $N(\tilde{R})$ denote the set of target observations covered by at least one route in $\tilde{R}$, i.e., $N(\tilde{R}) = \bigcup_{r \in \tilde{R}} N_r$. The problem of finding the optimal drone routes for all targets $v \in V$ being monitored at least once for every Δ_v continuous time segments in G' is equivalent to a problem of selecting a subset $R^* \subseteq R$ of routes, such that $N(R^*) = N$ and the total flight time of the selected routes in R^* is minimized.

3.3. Mathematical Model

By representing the drone scheduling problem on a cyclic time-expanded network G', we develop a route-based formulation. For $r \in R$ and $t \in T$, define a binary parameter $\gamma_t^r = 1$ if a route r is utilized at time t, and $\gamma_t^r = 0$ otherwise. For node $(v, t) \in N$, define a binary parameter $\delta_{vt}^r = 1$ if node (v, t) is covered by route r, and $\delta_{vt}^r = 0$ otherwise. The selection of route $r \in R$ is decided by a binary decision variable π_r, which equals to one if route r is selected and zero otherwise.

Given the route set R, a route-based integer linear programming formulation F_R is presented as follows.

$$z(F_R) = \min \sum_{r \in R} c_r \pi_r \tag{1.a}$$

$$\text{s.t.} \quad \sum_{r \in R} \delta_{vt}^r \pi_r \geq 1 \quad \forall v \in V, t \in T \tag{1.b}$$

$$\sum_{r \in R} \gamma_t^r \pi_r \leq K \quad \forall t \in T \tag{1.c}$$

$$\pi_r \in \{0, 1\} \quad \forall r \in R \tag{1.d}$$

The objective function (1.a) minimizes the total duration time of the selected routes. Constraints (1.b) ensure that the monitoring frequency requirements of all the targets are satisfied. Constraints (1.c) restrict that the number of drones operating at each time segment is no more than the total number of available drones. Constraints (1.d) define the domain of the decision variables.

4. Solution Approach

Solving F_R is difficult as it may involve exponentially many variables. To tackle this challenge, we introduce a column generation-based approach (CGBA) to solve F_R.

4.1. Overview of CGBA

The overall framework of CGBA is described as follows.

Step 1: *LP relaxation computation via column generation (Section 4.2).*

Solve the LP relaxation of F_R (i.e., LF_R) by column generation and obtain its optimal value $z(LF_R)$. Set $z_{LB} = z(LF_R)$. Let R_1 be the set of routes generated when LF_R is optimally solved. Solve F_{R_1} based on the obtained route set R_1 by a general-purpose MIP solver with a time limit of T_{MIP}. If a feasible solution is found, we set $z_{UB} = z(F_{R_1})$, where $z(F_{R_1})$ denotes the objective value of the obtained solution; Otherwise, set $z_{UB} = +\infty$.

Step 2: *Integral solution computation via route enumeration (Section 4.3).*

Given z_{LB} and z_{UB}, let $z_{UB}^{guess} = \min\{z_{LB}(1 + gap_{guess}), z_{UB}\}$ be an upper bound guessed according to a preset percentage gap gap_{guess}. With the dual solution of LF_R, generate route set R_2 by enumerating all routes with their reduced costs no greater than $z_{UB}^{guess} - z_{LB}$. If the running time of route enumeration exceeds T_{RE}, go to Step 3; Otherwise, solve F_{R_2} using a general-purpose MIP solver with a time limit of T_{MIP}, and perform one of the following cases according to the solution status:

- If F_{R_2} is solved with a solution of objective value $z(F_{R_2}) > z_{UB}^{guess}$, CGBA terminates with a feasible solution found.

- If F_{R_2} is solved with a feasible solution of objective value $z(F_{R_2}) \leq z_{UB}^{guess}$, update $z_{UB} = z(F_{R_2})$ and then return to the beginning of Step 2 for another round of computation.
- If F_{R_2} is solved with an optimal solution of objective value $z(F_{R_2}) \leq z_{UB}^{guess}$, CGBA terminates with an optimal solution found.
- If no feasible solution is found, CGBA terminates with no feasible solution.

Step 3: *Integral solution computation via branch-and-price (Section 4.4).*

Taking the solution of LF_R obtained in Step 1 as the root node solution, implement a branch-and-bound search to update z_{LB} and z_{UB}. When the gap between z_{UB} and z_{LB} is smaller than a tolerance threshold ε or the total running time exceeds T_{CGBA}, CGBA terminates with a near-optimal solution found.

The CGBA consists of three main steps. In Step 1, we adopt column generation to solve the LP relaxation of F_R, denoted by LF_R, to compute a valid lower bound z_{LB}. Utilizing the set R_1 of routes generated after column generation, we attempt to obtain a feasible solution by solving F_{R_1}, which obtains a valid upper bound z_{UB}. In Step 2, with the dual solution of LF_R obtained at the end of column generation, we enumerate the set R_2 of all routes with reduced costs no greater than $z_{UB}^{guess} - z_{LB}$, where z_{UB}^{guess} is a guessed upper bound according to a preset percentage gap gap_{guess}. If the time of route enumeration exceeds the time limit T_{RE}, we move to Step 3. Otherwise, we solve F_{R_2} to obtain an optimal or feasible solution of F_R. In Step 3, we consider the solution of LF_R obtained by the column generation in Step 1 as the root node solution of an enumeration tree and employ branch-and-price further to solve F_R. The CGBA is executed with a maximum running time of T_{CGBA}.

4.2. Step 1: LP Relaxation Computation via Column Generation

We employ a column generation algorithm to solve LF_R to compute a valid lower bound. The algorithm starts with a restricted master problem (RMP) based on a temporary route set R' that initially includes only a dummy column. In each iteration of column generation, the RMP is solved by an LP solver to obtain its optimal dual solution, and a pricing problem is then solved based on the obtained dual solution to find routes with negative reduced costs, these routes being added to R' to proceed the iterations. The column generation stops when no route with a negative reduced cost can be found from the pricing problem.

Let p_{vt} be the dual variable associated with constraint (1.b) for $v \in V$ and $t \in T$. Let q_t be the dual variable associated with constraint (1.c) for $t \in T$. Based on the values of p_{vt} and q_t in any dual solution of RMP, the reduced cost $RC(r)$ of variable π_r for $r \in R$ is

$$RC(r) = c_r - \sum_{v \in V} \sum_{t \in T} p_{vt} \delta_{vt}^r - \sum_{t \in T} q_t \gamma_t^r$$

The pricing problem aims to find a route with minimal reduced cost, known as an elementary shortest path problem with resource constraints (ESPPRC). In this study, we develop a labeling algorithm to solve the pricing problem.

Each label L represents a state of a forward partial path starting from the depot. A label consists of five dimensions, i.e., $L = (i, j, \{\sigma_v : v \in V\}, \rho, \phi)$, where i is the index of the last visited node, j is the index of the predecessor node of i, $\{\sigma_v : v \in V\}$ records the set of time differences from time $t(i)$ to the time of last visit for each target $v \in V$, ρ is the reduced cost value computed based on the partial path, and ϕ is the total flight time of the partial path. The label $L = (i, j, \{\sigma_v : v \in V\}, \rho, \phi)$ is feasible if and only if $\phi \leq T_d$. We call L a complete label if $v(i) = 0$ and the set of nodes visited by the path is non-empty. Otherwise, L is called an incomplete label.

The labeling algorithm is a label extension procedure that starts from a set of initial feasible incomplete labels, i.e., $\{(i_0, null, \{\sigma_v : \sigma_v = \Delta_v, v \in V\}, 0, 0) : v(i_0) = 0, t(i_0) \in T\}$. By extending a label $L = (i, j, \{\sigma_v : v \in V\}, \rho, \phi)$ toward a new node w, the newly generated label $L' = (i', j', \{\sigma'_v : v \in V\}, \rho', \phi')$ is determined by the following state transition equations: $i' = w$; $j' = i$; $\sigma'_v = \sigma_v + t_{iw}$ for $v \in V \setminus \{v(i)\}$ and $\sigma'_v = t_{iw}$ for $v = v(i)$; $\rho' = \rho + t_{iw} - \sum_{t=t_{\min}(w)}^{t(w)} p_{v(w)t} - \sum_{t=t(w)-t_{iw}}^{t(w)-1} q_t$, where $t_{\min}(w) = t(w) - \min\{\sigma'_{v(w)}, \Delta_{v(w)}\} + 1$; $\phi' = \phi + t_{iw}$. During the labeling algorithm, we apply the pruning technique based on the dominance rule presented in Proposition 1 to accelerate the label extension process and reduce the number of labels generated.

Proposition 1. *Let* $L_1 = (i_1, j_1, \{\sigma_{v,1} : v \in V\}, \rho_1, \phi_1)$ *and* $L_2 = (i_2, j_2, \{\sigma_{v,2} : v \in V\}, \rho_2, \phi_2)$ *be two incomplete labels.* L_1 *dominates* L_2 *if the following conditions are satisfied simultaneously: (i)* $i_1 = i_2$*, (ii)* $\phi_1 \leq \phi_2$*, (iii)* $\rho_1 \leq \rho_2$*, and (iv)* $\min\{\sigma_{v,1}, \Delta_v\} \geq \min\{\sigma_{v,2}, \Delta_v\}$ *for all* $v \in V$*.*

Proof. Given an incomplete label L, let $\omega(L)$ be a backward partial path starting from the last visited node of L and ending at a depot node, such that L and $\omega(L)$ can be connected to obtain a complete label. According to condition (i), any backward partial path $\omega(L_2)$ for label L_2 can be connected to the forward partial path of label L_1 to form a complete label. According to condition (ii), the complete label created by combining L_1 and $\omega(L_2)$ corresponds to a feasible route with total flight time not exceeding the drone's maximum duration time. Let r_1 and r_2 be the two feasible routes after extending L_1 and L_2 with the backward partial path $\omega(L_2)$, respectively. According to the definition of $RC(r)$ and conditions (iii-iv), it can be derived that $RC(r_1) \leq RC(r_2)$ always holds. Therefore, given any possible backward partial path $\omega(L_2)$ for label L_2, it can always be connected to the forward partial path of label L_1 to obtain a complete label with a lower reduced cost, which thus proves that L_1 dominates L_2. $\square$

In the early stage of the column generation process, we adopt a heuristic approach instead of seeking exact solutions for the pricing problems. Specifically, we select a subset of incomplete labels with smaller reduced costs for extension. This heuristic strategy accelerates the identification of complete labels with negative reduced costs, thereby speeding up the column generation iterations. When the heuristic labeling algorithm fails to find routes with a negative reduced cost, we turn off the heuristic strategy and use the exact labeling algorithm to find the optimal solution for the pricing problem to prove the optimality.

At the end of the column generation performed in Step 1 of CGBA, we obtain the optimal solution of LF_R that provides a valid lower bound z_{LB}. Let R_1 be the set of routes obtained by combining R' with the routes generated by the labeling algorithm in the last CG iteration. We employ a MIP solver to solve F_{R_1} to attempt to find a feasible solution that can provide a valid upper bound z_{UB}.

4.3. Step 2: Integral Solution Computation via Route Enumeration

Based on the valid upper bound z_{UB} computed in Step 1 of CGBA, we update the value of the guessed upper bound by $z_{UB}^{guess} = \min\{z_{LB}(1 + gap_{guess}), z_{UB}\}$. Given the guessed upper bound z_{UB}^{guess}, the valid lower bound z_{LB} and its corresponding dual solution computed at the end of the column generation, we generate route set R_2 by enumerating all routes $r \in R$ with reduced cost $RC(r) \leq z_{UB}^{guess} - z_{LB}$.

The route enumeration relies on a dynamic programming process using a state representation similar to the labeling algorithm described in Section 4.2. In route enumeration, we utilize a new dominance rule, as shown in Proposition 2, to avoid generating routes that will not be selected in the optimal solution of F_R.

Proposition 2. *Let $L_1 = (i_1, j_1, \{\sigma_{v,1} : v \in V\}, \rho_1, \phi_1)$ and $L_2 = (i_2, j_2, \{\sigma_{v,2} : v \in V\}, \rho_2, \phi_2)$ be two incomplete labels. L_1 dominates L_2 if the following conditions are satisfied simultaneously: (i) $i_1 = i_2$, (ii) $\phi_1 \leq \phi_2$, (iii) $N(L_1) \supseteq N(L_2)$, where $N(L)$ represents the set of target observations covered by the partial path of label L.*

Proof. Given an incomplete label L, let $\omega(L)$ be a backward partial path starting from the last visited node of L and ending at a depot node, such that L and $\omega(L)$ can be connected to obtain a complete label. According to condition (i), any backward partial path $\omega(L_2)$ for L_2 can always be connected to the forward partial path for label L_1 to form a complete label. Let r_1 and r_2 be the two feasible routes after extending L_1 and L_2 with the backward partial path $\omega(L_2)$, respectively. According to condition (ii), the total flight time of r_1 must not exceed that of r_2. According to condition (iii), the target observations covered by r_2 must also be covered by r_1. Therefore, in any optimal solution of F_R, r_2 must not be selected as it is dominated by r_1. This further proves that L_1 dominates L_2. □

In addition to the dominance rule, we apply a completion-bound pruning technique to speed up the route enumeration. For any incomplete label, we compute a lower bound on the minimum increment of reduced cost for extending it to a complete label. For a label $L = (i, j, \{\sigma_v : v \in V\}, \rho, \phi)$, if the lower bound plus the reduced cost ρ exceeds $z_{UB}^{guess} - z_{LB}$, there is no need to extend L further because it cannot lead to a route with its reduced cost satisfying $RC(r) \leq z_{UB}^{guess} - z_{LB}$. In the time-expanded network, such a lower bound on the minimum increment of reduced cost for extending an incomplete label to a complete label can be calculated by solving the shortest path problem from the last visited node of the incomplete label to a depot node.

To avoid generating too many routes, we set a time limit T_{RE} for the route enumeration process. If the running time of route enumeration exceeds T_{RE}, we directly move to Step 3 to compute the integral solution through branch-and-price. Otherwise, we solve F_{R_2} using a general-purpose MIP solver with a time limit of T_{MIP}, which results in the following cases for the output. (i) If F_{R_2} is solved with a feasible solution of objective value $z(F_{R_2}) > z_{UB}^{guess}$, we terminate CGBA and output the best feasible solution found. (ii) If F_{R_2} is solved with a feasible solution of objective value $z(F_{R_2}) \leq z_{UB}^{guess}$, we return to update z_{UB}^{guess} and reduce the route set R_2 by eliminating the routes with $RC(r) > z_{UB}^{guess} - z_{LB}$. Solve F_{R_2} again for a new solution. (iii) If F_{R_2} is solved with an optimal solution of objective value $z(F_{R_2}) \leq z_{UB}^{guess}$, we terminate CGBA and output the obtained solution as the optimal solution. (iv) If no feasible solution is found, we terminate CGBA with no feasible solution.

4.4. Step 3: Integral Solution Computation via Branch-and-Price

When the running time of route enumeration in Step 2 of CGBA exceeds the given time limit T_{RE}, we move to Step 3 of CGBA and adopt branch-and-price to compute the integral solution of F_R further. In the branch-and-price, we take the solution of LF_R computed by column generation in Step 1 of CGBA as the root node solution and perform a branch-and-bound search based on an enumeration tree explored using a best-bound strategy. That is, the tree node corresponding to the current lower bound is selected for further branching.

At each node, the variable branching strategy is based on the concept of node clusters. Given $t \in T$, let $T(t, n)$ be a set of n time segments, such that $T(t, n) = \{t - n + 1, \ldots, t\}$ if $t \geq n$ and $T(t, n) = \{1, \ldots, t\} \cup \{T_{max} - n + t + 1, \ldots, T_{max}\}$ if $t \leq n$. For any node (v, t) of the time-expanded network, we define $\Gamma(v, t) = \{(v, t') : t' \in T(t, n)\}$ as a node cluster including n consecutive nodes of target v on or before time segment t. Given $\Gamma(v, t)$, define $R(\Gamma(v,t))$ as a subset of routes where each route passes at least one node included by $\Gamma(v,t)$. Let $\hat{\pi} = \{\hat{\pi}_r : r \in R\}$ be the optimal solution of the LP relaxation problem obtained from the current tree node. The variable branching is to find a node cluster $\Gamma(v,t)$ among $(v,t) \in N$, such that the decimal part for the value of $\sum_{r \in R(\Gamma(v,t))} \hat{\pi}_r$ is closest to 0.5.

Based on the solution $\hat{\pi}$, the branching creates two new tree nodes with additional constraints, one with $\sum_{r \in R(\Gamma(v,t))} \pi_r \leq \lfloor \sum_{r \in R(\Gamma(v,t))} \hat{\pi}_r \rfloor$ and the other with $\sum_{r \in R(\Gamma(v,t))} \pi_r \geq \lceil \sum_{r \in R(\Gamma(v,t))} \hat{\pi}_r \rceil$. Note that the new constraints created by the branching depend on the nodes of the time-expanded network and thus will not influence solving the pricing problem in column generation. Moreover, one can adjust the value of n to scale the size of the node cluster, aiming for a balanced tree search. A node of the enumeration tree can be fathomed if the lower bound computed is greater than or equal to a valid upper bound.

5. Computational Experiments

This section reports on the computational results of the problem studied in this paper. Our algorithm is implemented in C++ programming language, and CPLEX 12.8 is utilized as the LP and MIP solver. The experiments are performed on a computer with a 3.00 GHz Intel(R) Core (TM) i9-13900K CPU and 128 GB memory.

5.1. Test Instances

We generate the test instances based on the coordinate information of the Solomon instances proposed by Solomon (1987). Specifically, four types of instances, namely C1, C2, R, and RC, are selected from the Solomon's dataset. We choose the coordinate data of the first m customers as the targets and then build the Euclidean distance matrix $\{D_{uv} : u, v \in V_0\}$ to compute the flight time between targets, where $m \in \{10, 15, 20, 25, 30, 35\}$. The number of time segments $T_{\max}$ considered to build the time-expanded network is chosen from $\{6, 12, 18, 24\}$. For types C1, C2, and R, we set the planning time horizon as $H = 108$ and set the drone's maximum duration time as $T_d = \lfloor 90 * H/T_{\max} \rfloor$. For type RC, whose instances include more diversified customer locations, we set the planning time horizon as $H = 132$ and set the drone's maximum duration time as $T_d = \lfloor 110 * H/T_{\max} \rfloor$. The flight time from target u to target v is $\tau_{uv} = \lceil D_{uv}H/T_{\max} \rceil$. For each target $v \in V$, the monitoring frequency level Δ_v is selected randomly from $\{1, 3, 6, 12, 18, 24\}$.

We use "*Type*-Nx-Ty" to denote the name of an instance class associated with instance type (*Type*), number of targets (x), and number of time segments (y). To attest to the performance of our solution approach under different problem scales, we put the generated instances into two separate groups, one for small-size instances and the other for medium-size instances. For the small-size instances, we set $x \in \{10, 15, 20\}$ and $y \in \{6, 12\}$. For the medium-size instances, we set $x \in \{25, 30, 35\}$ and $y \in \{18, 24\}$. The number of available drones in small-size and medium-size instances is set to $|K| = 20$ and $|K| = 50$, respectively.

5.2. Solution Performance of CGBA

We initialize the guessed percentage gap as $gap_{guess} = 5\%$. The time limit for route enumeration in Step 2 is set as $T_{RE} = 200$ seconds. The time limit for the MIP solver is set as $T_{MIP} = 1800$ seconds. The time limit for the overall algorithm is set as $T_{CGBA} = 7200$ seconds. The size of the node cluster for variable branching in the branch-and-price is set as $n = 6$. The tolerance threshold in the branch-and-price is set as $\varepsilon = 1$.

The computational results obtained by CGBA for the small-size and medium-size instances are presented in Table 2 and Table 3, respectively. In both tables, the solution performance of the three main steps of CGBA are reported separately under columns "Step 1", "Step 2" and "Step 3". Columns "LB" and "UB" show the lower bound and upper bound found in each step. Columns "ColNum" show the number of routes generated and enumerated in Step 1 and Step 2, respectively. Column "NodeNum" shows the number of nodes explored in the branch-and-price. The running time used by each step is presented in column "CPU(s)". Column "Gap(%)" shows the percentage gap between the lower bound and upper bound values achieved at Step 1, i.e., $(z_{UB} - z_{LB})/z_{UB}$. Column "FinalGap(%)" shows the percentage gap between the best lower bound and the best-known upper bound at the end of CGBA, which is marked as "OPT" if an optimal solution has been found.

The results presented in Table 2 show that CGBA can successfully solve most small-size instances to optimality within the time limit. Among the 16 instances solved to optimality, CGBA can obtain optimal solutions for 13 instances at Step 2 of CGBA, indicating that the integral solution computation via route enumeration can effectively solve small-size instances. Such effectiveness is attributed to the small optimality gaps achieved after the computation of column generation, as shown by "Gap(%)" under Step 1. Branch-and-price is activated for computing two instances, i.e., C1-N10-T12 and C1-N15-T12, from which it can be seen that the branch-and-bound search can effectively find better solutions to improve the upper bounds, by contrast with the upper bounds computed in Step 1.

Table 2. Computational results for small-size instances

Instance	Step 1						Step 2			Step 3				FinalGap(%)
	LB	CPU(s)	ColNum	UB	CPU(s)	Gap(%)	UB	ColNum	CPU(s)	LB	UB	CPU(s)	NodeNum	
C1-N10-T6	60.0	0.2	1969	60	0.1	0.00	$\Longrightarrow$							OPT
C2-N10-T6	61.3	0.2	2384	62	0.1	1.21	62	2864	0.23	$\Longrightarrow$				OPT
R-N10-T6	73.0	0.1	1825	73	0.1	0.00	$\Longrightarrow$							OPT
R-N15-T6	58.0	0.2	2121	58	0.1	0.00	$\Longrightarrow$							OPT
R-N20-T6	109.0	0.4	2619	110	0.1	0.91	110	3097	0.2	$\Longrightarrow$				OPT
RC-N10-T6	72.5	0.2	1882	74	0.1	2.03	74	2284	0.3	$\Longrightarrow$				OPT
RC-N15-T6	115.0	0.3	2216	118	0.1	2.54	118	2894	0.4	$\Longrightarrow$				OPT
C1-N10-T12	50.0	6.8	10533	54	5.0	7.41	N/A	25832	200.5	50.0	50	38.3	33	OPT
C1-N15-T12	148.0	2.1	6550	157	4.2	1.33	N/A	39262	200.9	148.0	150	6992.4	17887	1.33
C2-N10-T12	182.0	0.6	3315	184	0.3	1.09	184	6343	1.0	$\Longrightarrow$				OPT
C2-N15-T12	185.3	1.4	3828	186	0.1	0.36	186	5476	1.1	$\Longrightarrow$				OPT
C2-N20-T12	225.0	6.6	4724	227	0.4	0.88	227	8864	1.8	$\Longrightarrow$				OPT
R-N10-T12	190.0	0.2	2366	192	0.2	1.04	192	3422	0.2	$\Longrightarrow$				OPT
R-N10-T12	204.0	0.2	2786	215	0.5	5.12	214	4682	22.0	$\Longrightarrow$				OPT
R-N20-T12	216.4	2.1	4921	219	0.5	1.21	219	13629	2.5	$\Longrightarrow$				OPT
RC-N10-T12	97.1	0.9	3159	98	0.1	0.87	98	6319	0.8	$\Longrightarrow$				OPT
RC-N15-T12	203.3	2.1	3243	209	0.5	2.71	209	20111	1,806.6	$\Longrightarrow$				2.71
RC-N20-T12	208.0	59.7	12651	217	0.4	4.15	217	28587	937.0	$\Longrightarrow$				OPT

Note. "N/A" under Step 2 means that a valid upper bound is not computed due to the premature termination of route enumeration. "N/A" under Step 3 means that a valid upper bound cannot be obtained by the branch-and-bound search.

Table 3 reports on the computational results for the medium-size instances. Due to the increased problem size, more computation times are consumed in each step of CGBA. Note that route enumeration cannot be completed within the given time limit for more than half of the tested instances, which requires branch-and-price for computing the integral solutions. Among all the medium-size instances, three instances are solved to optimality within the given time limit. As shown by column "FinalGap(%)", for those instances not solved to optimality, CGBA can yield near-optimal solutions with their optimality gaps ranging from 0.96% to 6.66%, which shows the strength of CGBA in obtaining high-quality solutions.

From both Table 2 and Table 3, it reveals that the running time of route enumeration process grows with the number of time segments and number of customers. When the running time of route enumeration exceeds the time limit, we do not attempt to compute a new valid upper bound by solving F_{R_2}. Different from the results for small-size instances, branch-and-price exhibits limited efficacy in finding better integer solutions, however, it indeed can promote the lower bounds to further close the final optimality gaps.

Table 3. Computational results for medium-size instances

Instance	Step 1						Step 2			Step 3				FinalGap(%)
	LB	CPU(s)	ColNum	UB	CPU(s)	Gap(%)	UB	ColNum	CPU(s)	LB	UB	CPU(s)	NodeNum	
C1-N25-T18	488.7	304.3	24276	503	56.7	2.83	N/A	43012	208.5	494.5	N/A	6637.0	1591	1.69
C1-N30-T18	619.8	57.5	15682	645	1804.1	3.91	N/A	29475	213.3	623.1	N/A	5127.4	1465	3.40
C1-N35-T18	756.5	63.6	11940	771	1803.7	1.88	N/A	21128	216.3	758.5	N/A	5117.4	1467	1.62
C2-N25-T18	611.5	16.6	9462	615	1.0	0.57	615	126396	17.8	$\Longrightarrow$				OPT
C2-N30-T18	739.8	27.7	14473	753	57.6	1.76	N/A	34545	208.58	745.5	N/A	6906.2	6007	1.00
C2-N35-T18	860.0	74.8	20290	870	700.7	0.97	N/A	820300	580.5	861.6	N/A	1833.9	587	0.96
R-N25-T18	675.0	9.1	9353	678	0.4	0.44	678	15797	2.9	$\Longrightarrow$				OPT
R-N30-T18	792.5	18.6	10451	805	97.1	1.55	807	396497	1,861.9	$\Longrightarrow$				1.55
R-N35-T18	852.8	29.4	10359	864	18.2	1.35	864	145203	1,828.0	$\Longrightarrow$				1.35
RC-N25-T18	741.5	24.8	6462	752	17.9	1.40	752	44292	353.3	$\Longrightarrow$				OPT
C1-N25-T24	876.5	469.6	69301	911	1804.6	3.79	N/A	84044	219.2	876.6	N/A	4708	119	3.78
C1-N30-T24	705.7	236.6	40948	744	1802.2	5.15	N/A	51523	231.0	705.7	N/A	4932.5	179	5.15
C2-N25-T24	890.5	36.2	16213	954	1804.4	6.66	N/A	91656	213.7	890.5	N/A	5145.9	1931	6.66
C2-N30-T24	722.5	129.7	30510	767	1804.5	5.80	N/A	56688	221.6	722.5	N/A	5046.1	185	5.80
C2-N35-T24	719.9	172.3	16469	747	278.9	3.63	N/A	44761	226.7	720.4	N/A	6524.9	249	3.63
R-N25-T24	793.1	14.7	10398	820	786.4	3.40	819	104454	1812.9	$\Longrightarrow$				3.27
R-N30-T24	561.0	24.9	11312	614	1801.2	8.63	601	638336	1949.9	$\Longrightarrow$				6.66
RC-N25-T24	690.2	92.6	13959	715	21.4	3.47	714	419247	1936.6	$\Longrightarrow$				3.47

Note. "N/A" under Step 2 means that a valid upper bound is not computed due to the premature termination of route enumeration. "N/A" under Step 3 means that a valid upper bound cannot be obtained by the branch-and-bound search.

5.3. Impact of Monitoring Frequency Requirement on the Solution

This section studies the impact of the monitoring frequency requirement on the solutions obtained. Experiments are conducted based on four original instances with 24 time segments, namely R-N25-T24, R-N30-T24, C1-N25-T24, and C1-N30-T24. By varying the selection range of the monitoring frequency levels, we consider three ranges (i.e., V1, V2, and V3) for each of the four original instances. The instances of V1 are exactly the original instances with the monitoring frequency levels randomly selected from $\{1, 3, 6, 12, 18, 24\}$. For the instances of V2 and V3, the monitoring frequency levels are chosen randomly from $[6, 12]$ and $[18, 24]$, respectively.

Table 4. Computational results under different monitoring frequency requirements

Instance	Range	Step 1				Step 2			Step 3				FinalGap(%)
		LB	CPU(s)	UB	CPU(s)	UB	ColNum	CPU(s)	LB	UB	CPU(s)	NodeNum	
R-N25-T24	V1	793.1	14.7	820	786.4	819	94056	1812.9	$\Longrightarrow$				3.27
R-N25-T24	V2	356.1	18.74	371	307.1	369	69426	3623.7	$\Longrightarrow$				3.32
R-N25-T24	V3	156.2	138.3	213	3.6	212	36256	1808.2	$\Longrightarrow$				26.14
R-N30-T24	V1	561.0	24.9	614	1801.2	601	627024	1949.9	$\Longrightarrow$				6.66
R-N30-T24	V2	442.1	37.7	468	1,801.0	476	783456	1977.3	$\Longrightarrow$				5.50
R-N30-T24	V3	140.6	333.2	152	0.1	145	57360	15.39	$\Longrightarrow$				OPT
C1-N25-T24	V1	876.5	469.6	911	1804.6	N/A	14743	219.2	876.6	N/A	4708	119	3.78
C1-N25-T24	V2	274.5	1913.5	300	1808.2	N/A	14913	220.9	274.5	N/A	3200.7	134	8.49
C1-N25-T24	V3	111.4	1901.8	154	1805.2	N/A	1774	220.7	111.4	N/A	3280.3	120	27.66
C1-N30-T24	V1	705.7	236.6	744	1802.2	N/A	10575	231.0	705.7	N/A	4932.5	179	5.15
C1-N30-T24	V2	287.8	1913.5	346	1809.2	N/A	10815	234.9	287.8	N/A	3250.7	198	16.81
C1-N30-T24	V3	112.4	2077.6	167	1805.8	N/A	2275	234.2	112.4	N/A	3020.3	202	32.68

Note. "N/A" under Step 2 means that a valid upper bound is not computed due to the premature termination of route enumeration. "N/A" under Step 3 means that a valid upper bound cannot be obtained by the branch-and-bound search.

Table 4 reports on the computational results obtained under different monitoring frequency requirements, using the same reporting information as previous tables. Table 4 reveals that the solution efficiency of CGBA is relatively stable when solving the problems under different monitoring frequency levels. For those instances with less restrictive monitoring frequency requirements (i.e., V3), the lower and upper bounds reported are smaller than those with tighter monitoring frequency requirements (i.e., V1 and V2), because the problem with less restrictive monitoring frequency requirements usually has a larger solution region, resulting in a lower total duration time of the operated drone trips. From the perspective of solution optimality, it can be seen from column "FinalGap" that the instances with tighter monitoring frequency requirements are solved with smaller optimality gap values, which demonstrates that CGBA is advantageous in tackling the monitoring frequency requirements when optimizing drone operations.

6. Conclusions

This paper investigates a drone scheduling problem arising from persistent monitoring applications. To address the general drone routing scenario and enhance the monitoring efficiency, we introduce a time-expanded network representation of the problem and propose an integer linear programming formulation. To solve this formulation, we develop a column generation-based approach. Through extensive numerical experiments conducted on small-size and medium-size instances derived from the Solomon dataset, we demonstrate the effectiveness of our overall solution approach in finding optimal or near-optimal solutions.

In future work, a possible research direction is to explore the utilization of heterogeneous drones with varying flight speeds or monitoring coverage levels. These new considerations can facilitate broader applications of drone surveillance. Additionally, the collaboration between drones and a mothership in persistent monitoring scenarios would also be worth investigating for further research.

References

Cattaruzza, D., Absi, N., and Feillet, D. (2016). Vehicle routing problems with multiple trips. *4OR*, 14:223–259.

Dasdemir, E., Batta, R., Köksalan, M., and Öztürk, D. T. (2022). UAV routing for reconnaissance mission: A multi-objective orienteering problem with time-dependent prizes and multiple connections. *Computers & Operations Research*, 145:105882.

Drucker, N., Ho, H.-M., Ouaknine, J., Penn, M., and Strichman, O. (2019). Cyclic-routing of unmanned aerial vehicles. *Journal of Computer and System Sciences*, 103:18–45.

Eskandaripour, H. and Boldsaikhan, E. (2023). Last-mile drone delivery: Past, present, and future. *Drones*, 7(2):77.

Fang, C., Han, Z., Wang, W., and Zio, E. (2023). Routing UAVs in landslides monitoring: A neural network heuristic for team orienteering with mandatory visits. *Transportation Research Part E: Logistics and Transportation Review*, 175:103172.

Fauske, M. F., Mannino, C., and Ventura, P. (2020). Generalized periodic vehicle routing and maritime surveillance. *Transportation Science*, 54(1):164–183.

Hari, S. K. K., Rathinam, S., Darbha, S., Kalyanam, K., Manyam, S. G., and Casbeer, D. (2021). Optimal UAV route planning for persistent monitoring missions. *IEEE Transactions on Robotics*, 37(2):550–566.

Insights, F. B. (2021). Airborne surveillance market size, share & covid-19 impact analysis, by component(radars,sensors, lmaging system, others), by technology (unmanned system, mannedsystem), by platform (fixed-wing aircraft, rotary-wing aircraft, uav), by applicationcommercial, military & government), and regional forecast, 2021-2028.

Li, M., Zhen, L., Wang, S., Lv, W., and Qu, X. (2018). Unmanned aerial vehicle scheduling problem for traffic monitoring. *Computers & Industrial Engineering*, 122:15–23.

Liang, Y.-J. and Luo, Z.-X. (2022). A survey of truck–drone routing problem: Literature review and research prospects. *Journal of the Operations Research Society of China*, 10(2):343–377.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., and Laporte, G. (2020). Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, 120:102762.

Moskal, M. D., Dasdemir, E., and Batta, R. (2023). Unmanned aerial vehicle information collection missions with uncertain characteristics. *INFORMS Journal on Computing*, 35(1):120–137.

Otto, A., Agatz, N., Campbell, J., Golden, B., and Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, 72(4):411–458.

Scott, D., Manyam, S. G., Casbeer, D. W., and Kumar, M. (2023). A Lagrangian algorithm for multiple depot traveling salesman problem with revisit period constraints. *IEEE Transactions on Automation Science and Engineering*, 20(1):690–702.

Shen, L., Wang, Y., Liu, K., Yang, Z., Shi, X., Yang, X., and Jing, K. (2020). Synergistic path planning of multi-UAVs for air pollution detection of ships in ports. *Transportation Research Part E: Logistics and Transportation Review*, 144:102128.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.

Wang, K., Yuan, B., Zhao, M., and Lu, Y. (2020). Cooperative route planning for the drone and truck in delivery services: A bi-objective optimisation approach. *Journal of the Operational Research Society*, 71(10):1657–1674.

Xia, J., Wang, K., and Wang, S. (2019). Drone scheduling to monitor vessels in emission control areas. *Transportation Research Part B: Methodological*, 119:174–196.

Xia, Y., Batta, R., and Nagi, R. (2017). Controlling a fleet of unmanned aerial vehicles to collect uncertain information in a threat environment. *Operations Research*, 65(3):674–692.

Zandieh, F., Ghannadpour, S. F., and Mazdeh, M. M. (2023). Integrated ground vehicle and drone routing with simultaneous surveillance coverage for evading intentional disruption. *Transportation Research Part E: Logistics and Transportation Review*, 178:103266.

Zeng, F., Chen, Z., Clarke, J.-P., and Goldsman, D. (2022). Nested vehicle routing problem: Optimizing drone-truck surveillance operations. *Transportation Research Part C: Emerging Technologies*, 139:103645.

Zhang, G., Jia, N., Zhu, N., Adulyasak, Y., and Ma, S. (2023). Robust drone selective routing in humanitarian transportation network assessment. *European Journal of Operational Research*, 305(1):400–428.

Zhang, G., Zhu, N., Ma, S., and Xia, J. (2021). Humanitarian relief network assessment using collaborative truck-and-drone system. *Transportation Research Part E: Logistics and Transportation Review*, 152:102417.

Zhou, B., Liu, W., and Yang, H. (2023). Unmanned aerial vehicle service network design for urban monitoring. *Transportation Research Part C: Emerging Technologies*, 157:104406.

Declaration of interests

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

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: