Optimizing persistent surveillance missions for UAVs by column generation

William Stenberg[1][0009−0002−7837−4371], Caroline Granfeldt[2][0000−0003−3289−4902], Torbjörn Larsson[3][0000−0003−2094−7376], and Roghayeh Hajizadeh[3][0000−0002−7402−6292]

[1] Saab AB, 171 54 Solna, Sweden,
Corresponding author: william.stenberg@saabgroup.com
[2] Saab AB, 581 88 Linköping, Sweden
[3] Department of Mathematics, Linköping University, 581 83 Linköping, Sweden

Abstract. This paper explores the use of column generation methods to solve an extended vehicle routing problem tailored to surveillance missions using unmanned aerial vehicles (UAVs). The goal is to minimise the number of UAVs required by finding optimal routes while still meeting surveillance requirements for objects. A mission involves a fixed set of locations that must be persistently monitored and a number of depots from which UAVs can be deployed. The surveillance requirements are defined by specific maximum revisit intervals for the objects, and a single UAV may perform multiple depot tours within a route.
The method used to solve the problem is an extended set-partitioning formulation combined with a column generation problem, which is a resource-constrained, prize-collecting travelling salesman problem. The column generation problem generates tours for each depot and maximum revisit interval configuration, and the set-partitioning model select and combines these tours into routes. Lastly, a post-processing step is used to the obtained solutions to reduce total route times.
To evaluate the model, artificial instances with a varying number of depots and monitoring locations are constructed. The results show that the model can produce near-optimal solutions.

Keywords: mathematical optimization · column generation · decomposition methods · set partitioning · vehicle routing problem · travelling salesman problem · persistent · surveillance · UAV.

1 Introduction

The use of unmanned aerial vehicles (UAVs) for surveillance has become widespread in both civil and military applications, primarily due to their relatively low cost, ease of operation, and ability to reach otherwise inaccessible areas [3]. However, employing UAVs for such tasks raises the question of trying to make their use as efficient as possible.

Route planning for UAVs has been studied before, e.g., in [6] the goal is to minimise the number of UAVs needed to visit all locations in a given geographical region, where each UAV satisfies the maximum visiting times at the locations in a given tour. In the paper, all the UAVs are identical, and there are no depots considered. Hence, the start node in a tour is any given node which is included in the geographical region. Three different methods are investigated: 1) column generation, 2) heuristic column generation and 3) an ad hoc column generation method.

Other studies have approached UAV route planning and surveillance using variations of vehicle routing and coverage formulations. For instance, in [4] a mixed-integer programming model for multi-UAV routing with time windows is proposed, emphasising energy constraints and service frequency. Similarly, in [9] persistent UAV surveillance problems are examined, where revisit constraints in the form of refresh rates are explicitly modelled, formulating a MILP model and solved using a branch-and-cut algorithm to maintain a persistent coverage. In [12], a multi-robot persistent surveillance problem is explored, leveraging decomposition-based methods to handle heterogeneous sensing and motion constraints.

In addition to exact optimization methods, heuristic and decomposition-based approaches such as column generation [14], and logic-based Benders decomposition [10] are successfully applied to related routing and scheduling problems. These methods have been shown to be particularly effective for large-scale UAV or vehicle coordination problems, where combinatorial complexity makes exact enumeration infeasible.

Inventory routing poses a conceptually similar challenge to the one addressed in this paper. It considers a distribution network with a single supplier and multiple customers, where products must be delivered over a planning horizon to satisfy time-dependent demand. Continuous deliveries are required at specific intervals, and these are carried out by a fleet of vehicles, each with a fixed capacity [1]. As can be seen in [11] the similarities of the problem formulation, especially the focus on prioritising certain nodes prove the similarities of these two problem types. In this case, a central challenge lies in balancing the frequency of service (delivery or surveillance) with limited fleet resources.

Finally, the set partitioning and column generation problem described in [6, 5] has many elements which are incorporated in this paper. However, the UAV surveillance mission planning problem which was introduced in [13] is the paper on which this study is based upon. A certain set of locations require persistent surveillance, with differing requirements on how often there is a need for revisit, called maximum revisit interval. The goal is to minimise the number of UAVs needed while still meeting all surveillance demands of the locations. This paper focuses on UAVs, however, the results obtained are applicable for other autonomous agents as well since the focus is on route planning and the efficiency of vehicle use.

The remainder of this paper is structured as follows. Section 2 defines and models the UAV surveillance problem. Section 3 introduces a column generation model for solving the problem. Section 4 presents the computational results. Section 5 discusses the conclusions drawn from the results.

2 Problem Setting

A surveillance mission characterised by the following components is considered:

- A set of nodes $\mathcal{N}$ that require persistent surveillance,
- a set of depots $\mathcal{D}$ from which a fleet of UAVs operates,
- deterministic travel times, where the travel time t_{ij} for traversing an edge (i, j) is proportional to the Euclidean distance between nodes i and j. Effects such as wind and terrain are not considered,
- a *tour*, defined as a closed path that starts and ends at a depot and visits a subset of nodes,
- a fleet of UAVs subject to operational constraints, including
  - a maximum flight time T per tour,
  - a mandatory maintenance time ρ between consecutive tours, and
  - a UAV may be assigned to multiple tours,
- each node is visited by exactly one UAV,
- a *route*, defined as an ordered sequence of tours assigned to a single UAV, where a UAV may perform multiple tours over the mission horizon provided that maintenance constraints are satisfied,
- for each node $i \in \mathcal{N}$, a maximum revisit interval between consecutive visits τ_i, modelling the node's surveillance priority (smaller values indicate higher priority),
- a set of sufficient conditions is imposed to guarantee feasibility. In particular, if these conditions hold, a feasible solution always exists, at least under the trivial assignment of one UAV per node:
  - For each node $i \in \mathcal{N}$, $2t_{di} \leq T$, must hold for some depot $d \in \mathcal{D}$ to ensure all nodes are able to be visited,
  - and for each node $i \in \mathcal{N}$, $2t_{di} + \rho \leq \tau_i$, must hold for some depot $d \in \mathcal{D}$ to ensure that the maximum revisit interval is fulfilled.

In Figure 1, a feasible example solution is provided. In this example, two UAVs are utilised, where one UAV has been assigned two tours that combine into a route.

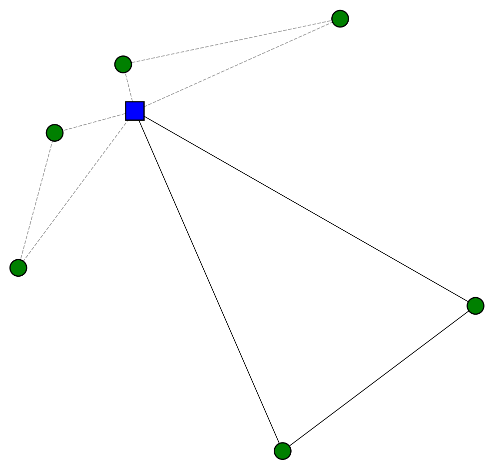

Fig. 1: Example solution, with 1 depot and 6 nodes.

The objective is to construct feasible UAV routes such that all nodes are visited within their maximum revisit interval, while respecting UAV flight time and maintenance constraints. Although a compact integer formulation is theoretically conceivable, the inherent structure of the problem is expected to lead to high complexity and limited practical solvability. Consequently, a column-oriented formulation is adopted.

The parameters and sets used for the column-oriented model are defined in Table 1.

Table 1: Definitions of sets and parameters in the master problem.

Symbol	**Indexed by**	**Definition**
Sets		
$\mathcal{N} := \{1, 2, \ldots, N\}$	i	Set of nodes.
$\mathcal{D} := \{1, 2, \ldots, D\}$	d	Set of depots.
$\mathcal{U}$	u	Set of UAVs available.
$\Delta := \{\tau_i \mid i \in \mathcal{N}\}$	δ	Set of maximum revisit intervals associated with the nodes.
$\Delta_{\geq \delta} := \{\tilde{\delta} \in \Delta \mid \tilde{\delta} \geq \delta\}$	$\tilde{\delta}$	Set of revisit intervals less strict than or equal to a given δ.

Symbol	Indexed by	Definition
$\mathcal{R}_{\delta d}$	r	Set of feasible tours for given δ and d.
Parameters		
a_{ir}		Binary parameter equal to 1 if node i is included in tour r, 0 otherwise.
ϕ_r		The total time it takes for a tour to be traversed, including the maintenance time at the depot.

The following decision variables are defined for the model. For each depot $d \in \mathcal{D}$ and UAV $u \in \mathcal{U}$, let

$$z_d^u = \begin{cases} 1, & \text{if some tour } r \in \mathcal{R}_{\tilde{\delta} d},\ \tilde{\delta} \in \varDelta_{\geq \delta}, \text{ is traversed,} \\ 0, & \text{otherwise,} \end{cases}$$

$$v_r^u = \begin{cases} 1, & \text{if UAV } u \in \mathcal{U} \text{ traverses tour } r \in \mathcal{R}_{\delta d}, \text{ where } \delta \in \varDelta, \\ 0, & \text{otherwise.} \end{cases}$$

Assuming that all tours are available, and therefore included in $\mathcal{R}_{\delta d}$, the model, which aims to minimise the number of UAVs needed is formulated as

$$Z_{\mathrm{IP}} := \min \quad \sum_{u \in \mathcal{U}} \sum_{d \in \mathcal{D}} z_d^u, \tag{1a}$$

$$\text{s.t.} \quad \sum_{u \in \mathcal{U}} \sum_{d \in \mathcal{D}} \sum_{\delta \in \varDelta} \sum_{r \in \mathcal{R}_{\delta d}} a_{ir} v_r^u = 1, \quad i \in \mathcal{N}, \tag{1b}$$

$$\sum_{\tilde{\delta} \in \varDelta_{\geq \delta}} \sum_{r \in \mathcal{R}_{\tilde{\delta} d}} \phi_r v_r^u \leq \delta z_d^u, \quad \delta \in \varDelta, d \in \mathcal{D}, u \in \mathcal{U}, \tag{1c}$$

$$\sum_{d \in \mathcal{D}} z_d^u \leq 1, \quad u \in \mathcal{U}, \tag{1d}$$

$$v_r^u \in \{0,1\}, \quad r \in \mathcal{R}_{\delta d}, \delta \in \varDelta, d \in \mathcal{D}, u \in \mathcal{U}, \tag{1e}$$

$$z_d^u \in \{0,1\}, \quad d \in \mathcal{D}, u \in \mathcal{U}. \tag{1f}$$

The objective function (1a) aims to minimise the number of routes used, and in effect, the number of UAVs used. Constraints (1b) are the set partitioning constraints ensuring that each node is visited by exactly one tour, each tour corresponds to one column. Constraints (1c) link the selection of tours to the corresponding routes by ensuring that if a specific route is chosen, the total traversal time of the selected tours does not exceed the maximum revisit interval for any node included in the route. Note that for each $\delta \in \Delta$, tours with less strict maximum revisit intervals are included in the constraint, since their inclusion in a route with a stricter maximum revisit interval does not break feasibility and consequently opens up for more possible routes. Constraints (1d) ensure that each UAV $u \in \mathcal{U}$ starts from only one depot. Constraints (1e) and (1f) are the binary restrictions of the decision variables.

Notably, in problem (1), an important structural relationship between tours (columns) and the objective function arises. Since multiple tours can be assigned to the same route, the routing of a UAV is represented by a combination of columns rather than a single one. As the objective function solely minimises the number of routes used, the tour decision variable does not appear explicitly in its formulation.

3 Solution Method

Model (1) belongs to the class of vehicle routing problems (VRPs) and is therefore NP-hard for non-trivial instance sizes, rendering the explicit enumeration of all feasible tours impractical. Consequently, a column generation approach is applied, considering only the routes that are likely to improve the solution. Figure 2 illustrates the overall algorithmic framework. Further details are provided later in this section.

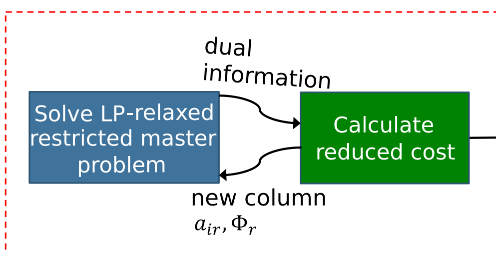

Fig. 2: Flowchart describing the algorithm used for solving the column generation problem.

Table 2: Definitions of sets and parameters used in the subproblems.

Symbol	Definition
Sets	

Symbol	**Definition**
$\Delta_{\leq\delta} := \{\tilde{\delta} \in \Delta \mid \tilde{\delta} \leq \delta\}$	Set of maximum revisit intervals that are more strict (i.e., smaller or equal) than a given $\delta \in \Delta$.
$\mathcal{V}_{d\delta} := \{i \in \mathcal{N} \mid \tau_i \geq \delta\} \cup \{d\}$	Set of nodes with a maximum maximum revisit interval greater than or equal to δ, including the depot d.

In Table 2 the sets used for the column generation are defined. Let $\overline{\mathcal{R}}_{\delta d} \subset \mathcal{R}_{\delta d},\ \delta \in \Delta,\ d \in \mathcal{D}$ denote the subset of routes obtained after performing the restriction. Let Z_{LP} be the LP-relaxed restricted version of model (1). Given a solution of Z_{LP}, the dual variables $\bar{\pi}_i,\ i \in \mathcal{N}$ and $\bar{\mu}^u_{\delta d},\ \delta \in \Delta,\ d \in \mathcal{D},\ u \in \mathcal{U}$ are obtained. Given this dual solution, the reduced cost is formulated such that the nested structure of Constraint (1c) is considered. For the strictest maximum revisit interval, i.e. the smallest element $\delta_{\min} \in \Delta$, all generated tours are included in its corresponding constraints. As δ is increased, only tours with weaker or equal maximum revisit intervals are included. Consequently, when generating a tour for a fixed δ, the dual variables that are necessary to include are $\bar{\mu}^u_{\tilde{\delta} d}, \tilde{\delta} \in \Delta_{\leq\delta}$, since a tour may appear in any of these constraints. The objective coefficient for the column generation problems will therefore include a term which is a sum over all such $\mu^u_{\tilde{\delta} d}$.

The overall set of feasible routes can be expressed as the union

$$\bigcup_{\delta \in \Delta} \bigcup_{d \in \mathcal{D}} \mathcal{R}_{\delta d},$$

where $\mathcal{R}_{\delta,d}$ contains all routes originating from depot d within maximum revisit interval δ. Fixing (δ, d) determines all route-specific parameters, yielding an independent pricing problem and thus a distinct column generation problem. Figure 3 illustrates this decomposition. Here, let $\delta_i \in \Delta$, $i = 1, 2, \ldots, |\Delta|$, denote the i-th maximum revisit interval.

Let the binary variables a_i and x_{ij} indicate node visits and direct travel from node i to node j, respectively. Furthermore, let the continuous variable ω_i represent the cumulative travel time from the depot to node i and is used in the Miller-Tucker-Zemlin (MTZ) constraints to prevent subtours. The pricing column generation problem is formulated to generate feasible UAV tours. Each feasible solution of the column generation problem corresponds to a tour r characterised by node-visit indicators a_{ir} and an associated tour cost contribution ϕ_r. These quantities are later used to construct new columns that are added to the master problem. Given that for each column generation problem, $\delta \in \Delta$ and

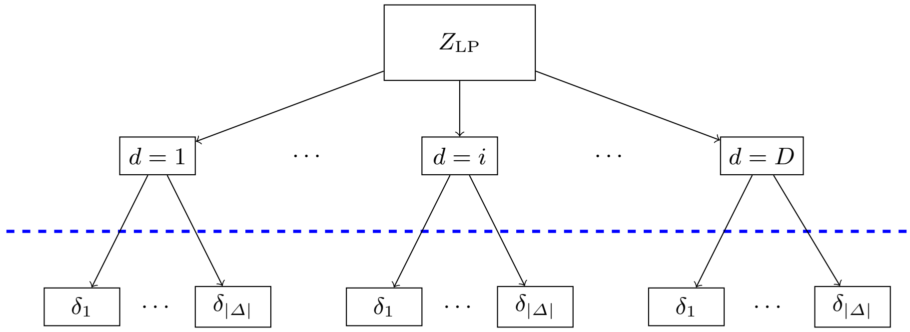

Fig. 3: Diagram illustrating the general structure of the problem. It shows the partitions made from Z_{LP} into subproblems. The first partition chooses one of the available depots in a mission. The second partition chooses a value $\delta \in \Delta$.

$d \in \mathcal{D}$ are fixed, the column generation problem can be formulated as

$$\min \quad -\sum_{i \in \mathcal{V}_{d\delta}} a_i \bar{\pi}_i - \sum_{\tilde{\delta} \in \Delta_{\leq \delta}} \bar{\mu}^u_{\tilde{\delta} d} \Big(\sum_{i \in \mathcal{V}_{d\delta}} \sum_{j \in \mathcal{V}_{d\delta} \setminus \{i\}} t_{ij} x_{ij} + \rho \Big), \tag{2a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{V}_{d\delta} \setminus \{i\}} x_{ij} = a_i, \qquad i \in \mathcal{V}_{d\delta}, \tag{2b}$$

$$\sum_{j \in \mathcal{V}_{d\delta} \setminus \{i\}} x_{ji} = a_i, \qquad i \in \mathcal{V}_{d\delta}, \tag{2c}$$

$$a_d = 1, \tag{2d}$$

$$\omega_j - \omega_i \geq t_{ij} - (T - t_{ij})(1 - x_{ij}), \qquad i, j \in \mathcal{V}_{d\delta}, j \neq d, \tag{2e}$$

$$\sum_{i \in \mathcal{V}_{d\delta}} \sum_{j \in \mathcal{V}_{d\delta} \setminus \{i\}} t_{ij} x_{ij} \leq T, \tag{2f}$$

$$a_i \in [0, 1], \qquad i \in \mathcal{V}_{d\delta}, \tag{2g}$$

$$x_{ij} \in \{0, 1\}, \qquad i, j \in \mathcal{V}_{d\delta}, i \neq j, \tag{2h}$$

$$\omega_i \leq T, \qquad i \in \mathcal{V}_{d\delta}. \tag{2i}$$

The objective function (2a) aims to minimise the reduced cost of a new tour. Constraints (2b) and (2c) ensure that if a node is visited, there are arcs which enter and exit the node. Constraint (2d) ensures that the depot is visited in the tour. Constraint (2e) is the MTZ formulation, aiming to remove subtours in the tour. Notice that for each ω_j, its minimum value is always the total time travelled before reaching node $j \in \mathcal{V}_{d\delta}$ if node j is included in the tour; this holds true for all nodes visited until the last node in the tour. The final edge travel time, between the last node in the tour and the depot, will not be added to any ω_j. This in turn necessitates Constraint (2f), which ensures that the tour does not exceed the maximum flight time of the UAV. Constraints (2g)–(2i) are the restrictions of the variables used. Note here that a_i is only restricted to the interval $[0,1]$, since the binary restriction of x_{ij}, coupled with Constraints (2b) and (2c), ensure that a_i takes the value of either 0 or 1.

Each feasible solution of Model (2) corresponds to a tour. Let that tour be denoted with r. The variables $a_i,\ i \in \mathcal{N}$ are used to construct the corresponding master problem parameters $a_{ir},\ i \in \mathcal{N},\ r \in \bar{\mathcal{R}}_{\delta d},\ \delta \in \Delta,\ d \in \mathcal{D}$. Given a solution, $\bar{x}_{ij},\ i,j \in \mathcal{V}_{d\delta},\ i \neq j$, the total tour time can be calculated as

$$\phi_r = \sum_{i \in \mathcal{V}_{d\delta}} \sum_{j \in \mathcal{V}_{d\delta} \setminus \{i\}} t_{ij} \bar{x}_{ij} + \rho, \tag{3}$$

where the parameters a_{ir} and ϕ_r obtained from the column generation problems solution are subsequently added back as a new tour to the master problem.

In order to possibly help the integer master problem find solutions, all columns found during the column generation step are added, even if their reduced cost is positive. After the column generation algorithm has terminated, Z_{IP} is re-solved using the final set of restricted routes $\overline{\mathcal{R}}_{\delta d},\ \delta \in \Delta,\ d \in \mathcal{D}$. If all costs c_p for a problem are integers, then it is known that if the following condition is satisfied;

$$\lceil Z_{\text{LP}}^* \rceil = \bar{Z}_{\text{IP}}, \tag{4}$$

the integer solution $\bar{Z}_{\text{IP}}$ is optimal.

3.1 Post-Processing

Since Z_{LP} focuses on minimising the number of UAVs used and does not account for tour or route efficiency, it is possible to get edges that cross each other, resulting in unintuitive and impractical sequences. Therefore, it is of interest to create a final solution that perhaps looks more "natural". Hence, a post-processing, route optimization step is introduced.

Given a solution of the LP-relaxed model, Z_{LP}^*, with an optimum found, let the route decision variables $\bar{z}_d^u = 1$, for some $d \in \mathcal{D}, u \in \mathcal{U}$, be the solution obtained. By fixing $\bar{z}_d^u$, the number of UAVs deployed will not increase, i.e. the original objective function cannot become better or worse. Therefore, by re-solving Z_{IP} with the routes $\bar{z}_d^u$ *fixed*, i.e. treated as parameters instead of variables, the focus can shift to reducing the *total* tour length. This is done by changing the objective function to optimize over v_r^u and minimising over the time taken for each tour. The resulting optimization problem is formulated as

$$\min \quad \sum_{u \in \mathcal{U}} \sum_{d \in \mathcal{D}} \sum_{\delta \in \Delta} \sum_{r \in \overline{\mathcal{R}}_{\delta d}} \phi_r v_r^u, \tag{5a}$$

$$\text{s.t.} \quad \sum_{u \in \mathcal{U}} \sum_{d \in \mathcal{D}} \sum_{\delta \in \Delta} \sum_{r \in \overline{\mathcal{R}}_{\delta d}} a_{ir} v_r^u = 1, \qquad i \in \mathcal{N}, \tag{5b}$$

$$\sum_{\tilde{\delta} \in \Delta_{\geq \delta}} \sum_{r \in \overline{\mathcal{R}}_{\tilde{\delta} d}} \phi_r v_r^u \leq \delta \bar{z}_d^u, \qquad \delta \in \Delta, d \in \mathcal{D}, u \in \mathcal{U}, \tag{5c}$$

$$v_r^u \in \{0,1\}, \qquad r \in \overline{\mathcal{R}}_{\delta d}, \delta \in \Delta, d \in \mathcal{D}, u \in \mathcal{U}, \tag{5d}$$

where the objective function (5a) focuses on minimising the total tour length. Constraints (5b) and (5c) are nearly identical to the constraints presented in Model (1), with the alteration that $\bar{z}_d^u$ is fixed. Any constraint containing only variables $\bar{z}_d^u$ is removed.

4 Results

All computation was done on a Dell precision 7680 with a 13th Gen Intel(R) Core(TM) i9-13950HX (32 CPUs), 2.2GHz and 64 GB RAM memory. Julia Version 1.11.3 was used [7], the MIP models were solved using Gurobi 12.0.1 [2] in conjunction with the modelling language JuMP [8].

4.1 Data and Problem Instances

The model is evaluated across multiple different sizes of missions, a mission size area is 3000 × 3000 meters and ranges from trivial cases with 3 nodes and 1 depot, up to 25 nodes and 5 depots. All missions are randomly generated regarding their depot and node placement, the values $\tau_i, i \in \mathcal{N}$ are chosen randomly from a list of different values. For each depot-node configuration, 10 instances are generated. Instance names follow the format `D`x`N`y`_`k, where x is the number of depots, y the number of nodes, and k the instance index within that configuration. Each time the column generation problem found an improving column, the master problem was re-solved and the duals were updated.

4.2 Illustrative Solution Example

To illustrate the structure of a feasible solution in greater detail, an example solution for the mission instance `D3N20_7`, comprising three depots and twenty nodes requiring visitation, is presented. Looking at Figure 4 the objective values for `D3N20_7` can be seen. Note that for this example, the integer model was solved using the restricted set at each iteration step as well. It is noticed that the first 8 iterations provide columns that improve the integer objective function as well. However, after that it fails to find new columns suitable for Z_{IP}, even though Z_{LP} obtains a value under 3. Based on Equation (4), it is not guaranteed that the current integer solution is optimal, since a better integer solution may still exist.

In Figure 5, a corresponding Gantt chart and the routing of the instance are shown, before and after the post processing step described in Model (5). Observe that three routes are very close to their maximum lap time. Except for one tour in the third route it is clear that the tours are short, which is a behaviour that can be seen for larger instances in general. There seems to be a preference for these short tours with a lot of depot visits. The comparison highlights the impact of the post-processing step on the overall route solution. Three of the routes have become slightly longer, the first one is shortened to the extent that it reduces the average time of all routes. Interestingly, it is observed

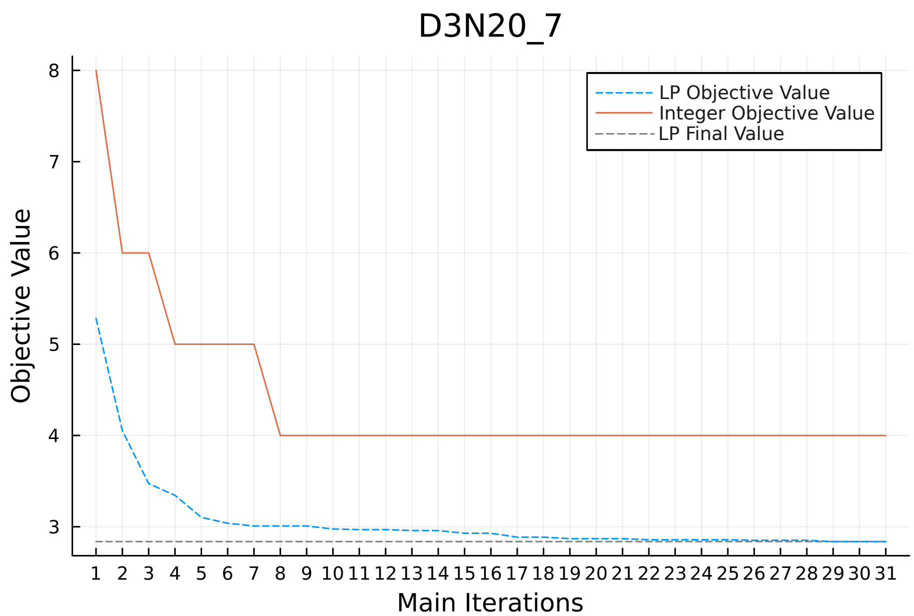

Fig. 4: Objective values of the integer and relaxed master problem across iterations for `D3N20_7`. The final integer objective value is 4, with a lower bound found at 2.84. The relative optimality gap equals 0.41.

that some nodes have switched their tour assignment. This behaviour stems from the fact that during the post-processing of the routes, where the total route time is minimised, the tours chosen for each route configuration are not fixed. As a consequence, different tours may be chosen, which in turn means that the nodes included in a route can change.

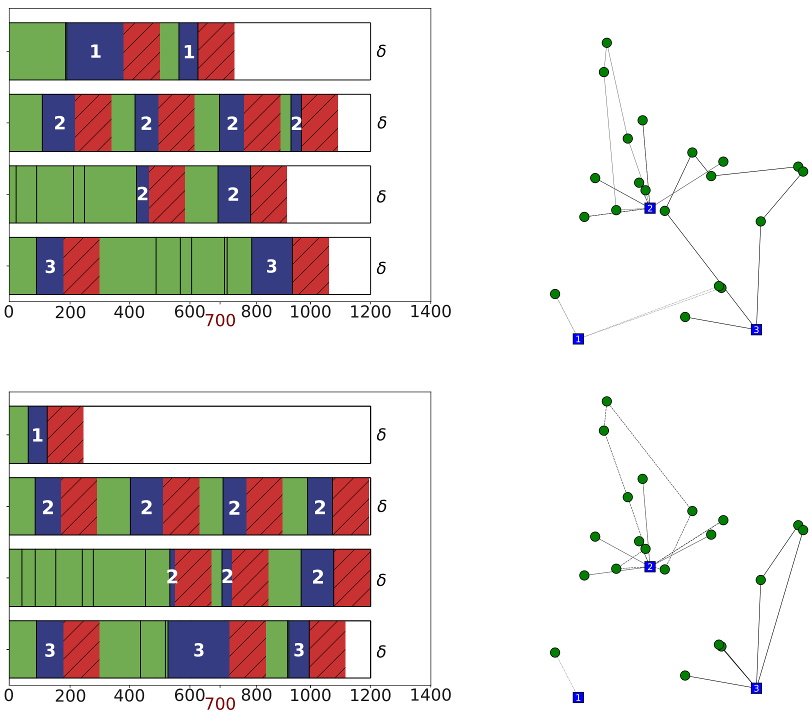

Fig. 5: Comparison of `D3N20_7` before and after post-processing. The left column shows Gantt charts, and the right column shows the corresponding routes. The top row is before post-processing and the bottom row is after. The objective value is 4.

4.3 Route Duration Analysis

The model tends to generate relatively short routes, even though longer routes are permitted. As shown in Figure 6, only a small number of solutions have routes longer than 20 minutes, highlighted in red. The average route duration is 15.2 minutes, indicated with a green line.

Furthermore, a tendency towards convergence around the average, where larger instances have less variance in their average route time can be observed.

In total, only four routes are longer than 20 minutes (1200 s.). This preference could be explained by the fact that the dual variables $\mu^u_{\tilde{\delta}d}$ associated with the smallest $\delta = 1200$ incorporates all possible maximum revisit intervals and could therefore become dominating. Another reason could be that it is simply more efficient using routes that include nodes with $\delta = 1200$, instead of having more routes and skipping those nodes.

It should be noted that, due to the randomised instance construction, the observed reduction in variance may be attributed to reduced variability in the generated larger instances rather than an inherent property of the model.

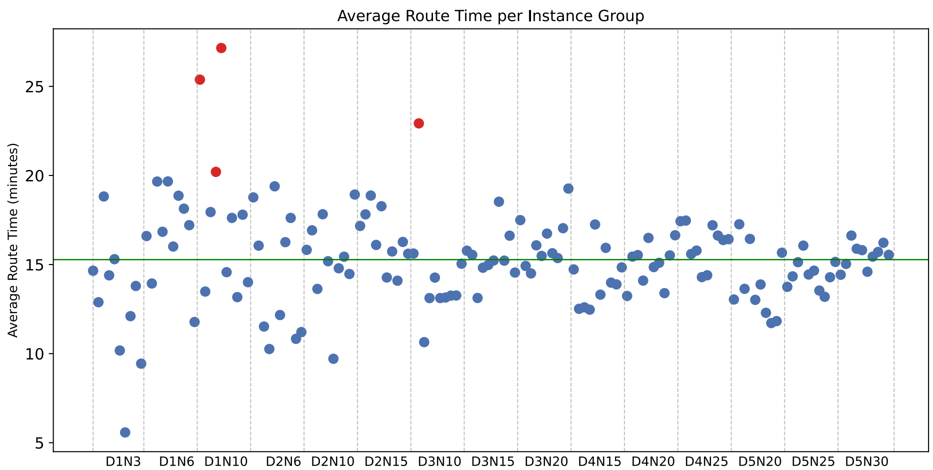

Fig. 6: Average route time per instance group. Average route time for all instances is 15.2 minutes.

4.4 Computation Times

Table 3 shows solution statistics for a few instances of different sizes. In Appendix A solution statistics for all instances solved are presented. The optimality gap is shown with percentages, using

$$\varepsilon_{\%} := 100 \cdot \varepsilon. \tag{6}$$

Small instances such as `D1N3_1` are solved almost instantaneously with very few columns and iterations, albeit with relatively large optimality gaps due to weak LP relaxations. As problem size increases, both the number of generated columns and the number of iterations grow substantially, which is reflected in longer runtimes.

Instances like `D2N10_6` and `D2N15_8` demonstrate a clear increase in complexity, with several hundred columns and iterations required before convergence. For the largest instances, such as D3N15_10, D5N20_9 and D5N30_2, the algorithm requires several minutes of computation time, indicating that column generation dominates the overall solution process. Despite this, the obtained optimality gaps remain moderate, suggesting that the LP relaxation provides useful bounds even for large-scale instances, though there is still room for strengthening the formulation.

Table 3: Representative sample of solution statistics for the model.

Instance	Time (s)	#Col.	#Iter.	Opt.	LP Opt.	$\varepsilon_{\%}$
D1N3_1	< 1	9	7	1	0.488	51.16
D1N10_8	5.49	94	85	2	1.317	34.14
D2N10_6	14.53	322	313	2	0.971	51.44
D2N15_8	31.53	327	313	3	2.114	29.53
D3N15_10	250.9	555	721	3	1.898	36.74
D4N20_6	6.58	124	209	5	3.672	26.55
D5N20_9	202.31	190	256	5	2.94	41.21
D5N30_2	221.54	240	421	7	5.246	25.05

Figure 7 shows the percentage of total computation time for different parts of the algorithm, the post-processing step in this model refers to the route optimization. The majority of time taken is in the column generation, however the integer master problem which gets solved at each iteration also takes up a considerable amount of time due to the additions of constraints on top of the typical set partition formulation which deals with the assignment of tours to routes. The "Other" category refers to steps such as pre-processing, the post-processing optimization step which take up a very small percentage of the time, as expected.

5 Conclusions

This article introduced a decomposition-based optimization model for persistent UAV surveillance routing. By combining column generation with an extended set partitioning problem, the approach focuses its computational effort on finding relevant and feasible tours while using a compact route-constructing master problem. The approach utilises a lexicographic optimization, in which the number of UAVs used is first minimised, followed by the minimisation of their total route length.

A key conclusion is drawn from how the post-processing step is constructed and how it affects already obtained solutions, it allows nodes to be reassigned from the solution obtained and distribute them to new routes. This reveals a flexibility in the solutions and may lead to improved route configurations.

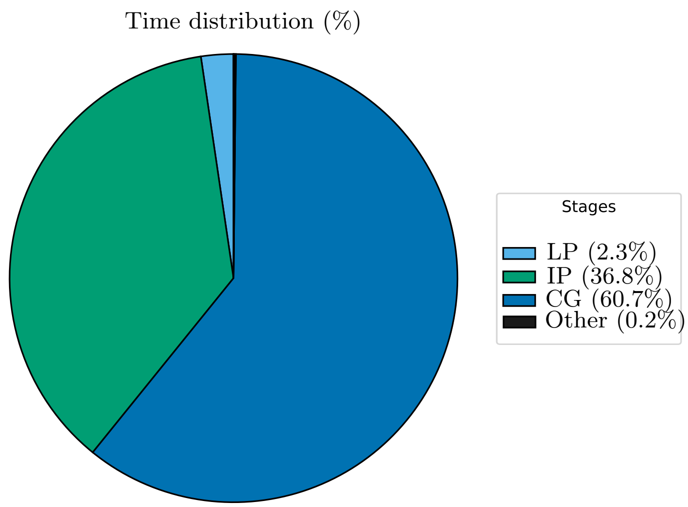

Fig. 7: Percentage of total computation time spent on each algorithm component, averaged over all instances.

The computational results show that the proposed strategy is effective for finding feasible and also near-optimal solutions. Moreover, it efficiently handles large-scale instances within reasonable computation times.

As expected, the column generation problems were computationally demanding and clearly dominate the overall time taken for solving instances. It would be of interest to solve the studied instances with a parallelisation step, where all column generation problems are solved simultaneously, to reduce their dominance.

Future work includes incorporating heterogeneous UAV fleets, and removing the limit of a node only being assigned to one unique tour in a solution, further expanding the possibilities for this type of application.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

A Appendix

In Table 4 the solution data about instances of the model is shown. In Table 5 the computation times for each algorithm step for all instances are shown. The optimality gap is shown with percentages, using Equation (6).

Table 4: MT instance data.

Instance	Time (s)	#Col.	#Iter.	Opt.	LP Opt.	$\varepsilon_{\%}$
D1N3_1	< 1	9	7	1	0.488	51.16
D1N3_2	3.04	11	9	1	0.215	78.52
D1N3_3	< 1	9	7	1	0.941	5.89
D1N3_4	< 1	4	4	2	1.439	28.03
D1N3_5	< 1	7	5	1	0.51	49.0
D1N3_6	< 1	9	7	2	1.017	49.13
D1N3_7	< 1	15	13	1	0.279	72.12
D1N3_8	< 1	12	10	1	0.605	39.49
D1N3_9	< 1	7	5	1	0.69	31.01
D1N3_10	< 1	9	7	1	0.315	68.54
D1N6_1	< 1	16	16	2	1.66	17.0
D1N6_2	< 1	21	16	2	1.394	30.31
D1N6_3	< 1	27	22	1	0.983	1.7
D1N6_4	< 1	33	28	1	0.842	15.8
D1N6_5	< 1	27	22	1	0.656	34.45
D1N6_6	< 1	7	5	3	2.402	19.94
D1N6_7	< 1	33	28	1	0.943	5.68
D1N6_8	< 1	14	13	3	2.342	21.94
D1N6_9	< 1	12	13	2	1.721	13.95
D1N6_10	< 1	38	33	2	1.178	41.12
D1N10_1	< 1	14	13	3	2.538	15.4
D1N10_2	2.74	85	101	3	2.023	32.58
D1N10_3	< 1	11	5	5	4.487	10.27
D1N10_4	< 1	18	25	3	1.972	34.27
D1N10_5	< 1	28	28	2	1.724	13.8
D1N10_6	1.79	90	81	2	1.457	27.14
D1N10_7	1.14	58	65	3	2.643	11.9
D1N10_8	5.49	94	85	2	1.317	34.14
D1N10_9	< 1	10	5	4	3.56	11.01
D1N10_10	2.34	78	69	2	1.401	29.95
D2N6_1	< 1	48	43	1	0.84	16.05
D2N6_2	< 1	10	7	3	2.409	19.69
D2N6_3	< 1	42	37	3	1.728	42.39
D2N6_4	< 1	30	25	2	1.026	48.71
D2N6_5	< 1	42	37	1	0.829	17.11
D2N6_6	< 1	42	37	2	1.163	41.87
D2N6_7	< 1	18	17	2	1.626	18.72
D2N6_8	< 1	62	57	1	0.881	11.92
D2N6_9	< 1	46	41	2	1.083	45.85
D2N6_10	< 1	38	33	2	1.121	43.96
D2N10_1	< 1	14	17	4	3.165	20.87
D2N10_2	< 1	12	7	4	3.384	15.39
D2N10_3	< 1	40	41	4	2.702	32.44
D2N10_4	< 1	10	9	4	3.565	10.88
D2N10_5	1.1	54	89	3	2.278	24.06
D2N10_6	14.53	322	313	2	0.971	51.44

Instance	Time (s)	#Col.	#Iter.	Opt.	LP Opt.	$\varepsilon_{\%}$
D2N10_7	< 1	26	33	4	2.958	26.05
D2N10_8	< 1	16	25	4	3.088	22.8
D2N10_9	< 1	16	25	4	2.894	27.65
D2N10_10	< 1	10	9	4	3.786	5.35
D2N15_1	< 1	51	49	5	4.28	14.39
D2N15_2	1.13	31	65	4	3.52	12.01
D2N15_3	23.47	191	177	2	1.887	5.64
D2N15_4	< 1	39	49	5	4.025	19.49
D2N15_5	1.02	39	49	4	3.312	17.21
D2N15_6	2.05	83	137	4	2.855	28.64
D2N15_7	1.4	55	81	6	4.72	21.33
D2N15_8	31.53	327	313	3	2.114	29.53
D2N15_9	< 1	15	9	6	4.881	18.65
D2N15_10	< 1	15	7	7	5.462	21.97
D3N10_1	< 1	16	25	3	2.178	27.39
D3N10_2	2.47	82	73	2	1.178	41.1
D3N10_3	8.86	214	205	2	1.064	46.78
D3N10_4	1.43	55	46	3	1.966	34.46
D3N10_5	< 1	55	61	3	2.112	29.6
D3N10_6	< 1	34	49	3	1.968	34.4
D3N10_7	< 1	28	37	4	2.598	35.05
D3N10_8	< 1	22	25	4	2.623	34.44
D3N10_9	8.08	250	241	2	1.326	33.69
D3N10_10	< 1	25	46	3	2.257	24.76
D3N15_1	< 1	15	13	5	3.945	21.1
D3N15_2	< 1	45	61	5	3.886	22.28
D3N15_3	< 1	15	13	6	3.937	34.38
D3N15_4	< 1	45	61	5	3.61	27.8
D3N15_5	< 1	42	109	5	3.632	27.35
D3N15_6	< 1	30	46	5	3.806	23.87
D3N15_7	26.81	195	181	2	1.824	8.81
D3N15_8	< 1	21	25	8	6.088	23.9
D3N15_9	4.9	99	169	4	2.853	28.69
D3N15_10	250.9	555	721	3	1.898	36.74
D3N20_1	1.75	53	133	6	5.129	14.52
D3N20_2	< 1	20	13	7	5.222	25.39
D3N20_3	< 1	20	13	8	5.801	27.49
D3N20_4	< 1	20	13	6	4.823	19.61
D3N20_5	26.42	116	193	5	3.871	22.58
D3N20_6	< 1	35	61	6	4.815	19.74
D3N20_7	40.97	290	406	4	2.837	29.06
D3N20_8	< 1	20	13	7	5.378	23.17
D3N20_9	< 1	20	13	7	5.964	14.8
D3N20_10	< 1	41	85	6	5.277	12.04
D4N15_1	< 1	27	49	5	3.681	26.37
D4N15_2	< 1	39	49	5	3.112	37.76
D4N15_3	< 1	39	49	5	3.15	37.01

Instance	Time (s)	#Col.	#Iter.	Opt.	LP Opt.	$\varepsilon_{\%}$
D4N15_4	< 1	15	17	8	4.987	37.66
D4N15_5	< 1	39	97	4	3.45	13.75
D4N15_6	< 1	47	129	5	3.272	34.56
D4N15_7	< 1	27	49	6	4.444	25.93
D4N15_8	2.07	79	129	4	2.797	30.07
D4N15_9	< 1	15	17	6	4.166	30.56
D4N15_10	< 1	55	81	5	3.673	26.54
D4N20_1	< 1	20	17	8	5.295	33.82
D4N20_2	< 1	36	65	8	5.962	25.47
D4N20_3	< 1	52	129	6	4.66	22.34
D4N20_4	< 1	28	33	8	5.639	29.51
D4N20_5	< 1	20	17	6	4.949	17.52
D4N20_6	6.58	124	209	5	3.672	26.55
D4N20_7	5.95	92	145	6	4.398	26.71
D4N20_8	< 1	20	17	7	4.688	33.03
D4N20_9	1.03	40	81	7	5.244	25.08
D4N20_10	< 1	20	17	6	4.993	16.78
D4N25_1	1.51	69	177	7	6.07	13.28
D4N25_2	2.39	65	161	8	6.984	12.7
D4N25_3	4.32	69	177	8	6.209	22.39
D4N25_4	< 1	41	65	7	5.45	22.14
D4N25_5	< 1	25	17	12	8.577	28.52
D4N25_6	< 1	25	17	12	8.639	28.0
D4N25_7	< 1	25	17	7	6.022	13.97
D4N25_8	< 1	25	17	8	6.651	16.86
D4N25_9	< 1	25	17	8	6.55	18.12
D4N25_10	< 1	29	17	8	6.57	17.87
D5N20_1	< 1	20	21	8	5.215	34.81
D5N20_2	1.79	70	101	6	5.178	13.69
D5N20_3	< 1	30	41	7	4.704	32.8
D5N20_4	< 1	20	21	7	5.754	17.8
D5N20_5	< 1	20	21	9	5.859	34.9
D5N20_6	< 1	45	101	8	5.541	30.74
D5N20_7	7.83	140	241	6	3.554	40.77
D5N20_8	< 1	20	21	8	4.686	41.42
D5N20_9	202.31	190	256	5	2.94	41.21
D5N20_10	2.8	60	161	6	4.549	24.18
D5N25_1	< 1	40	61	9	6.184	31.29
D5N25_2	< 1	25	21	10	7.169	28.31
D5N25_3	< 1	25	16	7	5.297	24.33
D5N25_4	< 1	25	21	8	6.425	19.68
D5N25_5	< 1	25	21	8	5.776	27.8
D5N25_6	1.86	70	181	8	5.863	26.71
D5N25_7	< 1	25	21	9	6.093	32.3
D5N25_8	< 1	25	21	8	5.276	34.05
D5N25_9	3.78	55	121	7	4.87	30.43
D5N25_10	7.81	105	161	8	6.06	24.26

Instance	Time (s)	#Col.	#Iter.	Opt.	LP Opt.	$\varepsilon_{\%}$
D5N30_1	< 1	30	21	10	7.214	27.86
D5N30_2	221.54	240	421	7	5.246	25.05
D5N30_3	45.6	150	481	7	5.562	20.54
D5N30_4	1.17	55	101	10	7.926	20.74
D5N30_5	< 1	50	81	9	7.071	21.43
D5N30_6	15.09	100	281	8	5.834	27.07
D5N30_7	< 1	30	21	10	7.723	22.77
D5N30_8	< 1	30	21	9	7.069	21.46
D5N30_9	< 1	40	41	11	8.922	18.89
D5N30_10	21.08	170	281	8	6.04	24.5

Table 5: Time taken per algorithm step for the model.

Instance	Preproc	LP	IP	Sub	Postproc	Alg	Total
D1N3_1	< 0.01	< 0.01	< 0.01	0.01	< 0.01	< 0.01	< 1
D1N3_2	0.05	2.25	0.15	0.44	0.08	0.07	3.04
D1N3_3	< 0.01	< 0.01	0.0	0.02	0.01	< 0.01	< 1
D1N3_4	< 0.01	< 0.01	< 0.01	0.0	< 0.01	< 0.01	< 1
D1N3_5	< 0.01	< 0.01	< 0.01	0.02	< 0.01	< 0.01	< 1
D1N3_6	< 0.01	< 0.01	0.01	0.02	0.0	< 0.01	< 1
D1N3_7	< 0.01	< 0.01	0.01	0.02	< 0.01	< 0.01	< 1
D1N3_8	0.0	< 0.01	0.03	< 0.01	< 0.01	< 0.01	< 1
D1N3_9	< 0.01	< 0.01	0.02	0.0	< 0.01	< 0.01	< 1
D1N3_10	< 0.01	< 0.01	< 0.01	0.02	< 0.01	< 0.01	< 1
D1N6_1	< 0.01	0.0	0.02	0.02	< 0.01	0.01	< 1
D1N6_2	< 0.01	< 0.01	0.05	0.04	0.0	< 0.01	< 1
D1N6_3	0.0	< 0.01	< 0.01	0.22	< 0.01	< 0.01	< 1
D1N6_4	< 0.01	0.08	0.08	0.44	0.08	0.03	< 1
D1N6_5	< 0.01	0.0	0.02	0.18	< 0.01	< 0.01	< 1
D1N6_6	< 0.01	< 0.01	< 0.01	< 0.01	0.01	< 0.01	< 1
D1N6_7	< 0.01	0.01	0.02	0.23	< 0.01	0.01	< 1
D1N6_8	< 0.01	0.0	< 0.01	0.03	< 0.01	< 0.01	< 1
D1N6_9	< 0.01	< 0.01	0.0	0.03	< 0.01	< 0.01	< 1
D1N6_10	< 0.01	0.01	0.06	0.35	0.02	< 0.01	< 1
D1N10_1	< 0.01	0.01	0.0	0.03	0.01	< 0.01	< 1
D1N10_2	< 0.01	0.17	0.12	2.45	< 0.01	0.0	2.74
D1N10_3	< 0.01	< 0.01	0.0	< 0.01	< 0.01	< 0.01	< 1
D1N10_4	< 0.01	0.0	0.01	0.19	< 0.01	< 0.01	< 1
D1N10_5	< 0.01	0.09	0.09	0.56	0.08	0.04	< 1
D1N10_6	< 0.01	0.07	0.23	1.49	< 0.01	< 0.01	1.79
D1N10_7	< 0.01	0.03	0.04	1.02	0.04	0.0	1.14
D1N10_8	< 0.01	0.1	0.29	5.08	0.02	< 0.01	5.49
D1N10_9	< 0.01	< 0.01	0.02	0.0	< 0.01	< 0.01	< 1
D1N10_10	< 0.01	0.06	0.19	2.09	< 0.01	0.01	2.34
D2N6_1	< 0.01	< 0.01	0.09	0.34	< 0.01	< 0.01	< 1
D2N6_2	< 0.01	0.03	0.02	< 0.01	< 0.01	< 0.01	< 1
D2N6_3	< 0.01	< 0.01	0.02	0.2	< 0.01	< 0.01	< 1
D2N6_4	< 0.01	< 0.01	0.02	0.12	< 0.01	< 0.01	< 1
D2N6_5	< 0.01	0.02	0.05	0.2	0.02	< 0.01	< 1
D2N6_6	< 0.01	0.04	0.02	0.57	< 0.01	< 0.01	< 1
D2N6_7	< 0.01	< 0.01	0.01	0.02	0.02	< 0.01	< 1
D2N6_8	< 0.01	0.01	0.03	0.32	< 0.01	< 0.01	< 1
D2N6_9	< 0.01	0.03	0.1	0.22	0.01	< 0.01	< 1
D2N6_10	< 0.01	0.02	0.01	0.23	< 0.01	< 0.01	< 1
D2N10_1	< 0.01	0.01	0.03	0.0	< 0.01	< 0.01	< 1
D2N10_2	< 0.01	0.0	0.02	< 0.01	< 0.01	< 0.01	< 1
D2N10_3	< 0.01	0.08	0.16	0.28	< 0.01	< 0.01	< 1
D2N10_4	< 0.01	< 0.01	0.03	0.0	< 0.01	< 0.01	< 1
D2N10_5	< 0.01	0.06	0.09	0.92	0.02	0.02	1.1
D2N10_6	< 0.01	0.61	1.85	12.05	0.02	0.01	14.53
D2N10_7	< 0.01	0.02	0.06	0.09	< 0.01	< 0.01	< 1

Instance	Preproc	LP	IP	Sub	Postproc	Alg	Total
D2N10_8	< 0.01	< 0.01	< 0.01	0.07	< 0.01	< 0.01	< 1
D2N10_9	< 0.01	0.01	0.0	0.07	0.0	< 0.01	< 1
D2N10_10	< 0.01	< 0.01	0.01	0.0	0.01	< 0.01	< 1
D2N15_1	< 0.01	0.05	0.11	0.75	0.02	< 0.01	< 1
D2N15_2	< 0.01	0.03	0.15	0.94	0.0	< 0.01	1.13
D2N15_3	< 0.01	0.58	1.9	20.98	0.0	< 0.01	23.47
D2N15_4	< 0.01	0.06	0.15	0.34	< 0.01	0.01	< 1
D2N15_5	< 0.01	0.03	0.38	0.58	0.02	0.02	1.02
D2N15_6	0.0	0.16	0.73	1.15	0.0	0.01	2.05
D2N15_7	< 0.01	0.06	0.31	0.99	0.02	0.02	1.4
D2N15_8	< 0.01	0.96	1.96	28.56	0.02	0.03	31.53
D2N15_9	< 0.01	< 0.01	0.02	0.02	< 0.01	< 0.01	< 1
D2N15_10	< 0.01	0.02	0.3	< 0.01	< 0.01	< 0.01	< 1
D3N10_1	< 0.01	0.02	< 0.01	0.08	< 0.01	< 0.01	< 1
D3N10_2	< 0.01	0.04	0.11	2.31	0.0	< 0.01	2.47
D3N10_3	< 0.01	0.23	0.48	8.15	< 0.01	< 0.01	8.86
D3N10_4	< 0.01	0.02	0.06	1.34	< 0.01	0.0	1.43
D3N10_5	< 0.01	0.02	0.07	0.43	0.03	< 0.01	< 1
D3N10_6	< 0.01	0.01	0.06	0.14	0.01	< 0.01	< 1
D3N10_7	< 0.01	0.02	0.05	0.1	< 0.01	< 0.01	< 1
D3N10_8	< 0.01	0.01	0.01	0.05	0.0	0.0	< 1
D3N10_9	< 0.01	0.28	0.6	7.2	< 0.01	0.0	8.08
D3N10_10	< 0.01	0.0	0.07	0.11	< 0.01	0.02	< 1
D3N15_1	< 0.01	0.03	0.01	0.01	0.0	< 0.01	< 1
D3N15_2	< 0.01	0.03	0.13	0.33	0.02	0.02	< 1
D3N15_3	< 0.01	0.01	0.03	0.02	0.0	< 0.01	< 1
D3N15_4	< 0.01	0.01	0.24	0.2	0.0	< 0.01	< 1
D3N15_5	0.0	0.03	0.08	0.55	< 0.01	< 0.01	< 1
D3N15_6	< 0.01	< 0.01	0.09	0.23	< 0.01	< 0.01	< 1
D3N15_7	< 0.01	0.67	1.39	24.7	0.02	0.02	26.81
D3N15_8	< 0.01	< 0.01	0.05	0.06	0.02	< 0.01	< 1
D3N15_9	< 0.01	0.18	2.61	2.09	0.0	0.01	4.9
D3N15_10	< 0.01	4.98	46.78	199.06	0.05	0.02	250.9
D3N20_1	< 0.01	0.1	0.65	0.95	0.02	0.02	1.75
D3N20_2	< 0.01	0.02	0.32	0.02	0.0	< 0.01	< 1
D3N20_3	< 0.01	< 0.01	0.18	0.04	0.0	< 0.01	< 1
D3N20_4	< 0.01	0.01	0.02	0.03	< 0.01	< 0.01	< 1
D3N20_5	< 0.01	0.37	2.4	23.65	0.01	0.0	26.42
D3N20_6	< 0.01	0.04	0.24	0.19	< 0.01	< 0.01	< 1
D3N20_7	< 0.01	2.17	6.28	32.47	0.03	0.01	40.97
D3N20_8	0.02	0.02	0.07	0.03	< 0.01	< 0.01	< 1
D3N20_9	< 0.01	< 0.01	0.03	0.02	< 0.01	< 0.01	< 1
D3N20_10	< 0.01	0.08	0.13	0.26	< 0.01	< 0.01	< 1
D4N15_1	< 0.01	0.04	0.01	0.16	< 0.01	< 0.01	< 1
D4N15_2	< 0.01	0.02	0.16	0.4	0.02	< 0.01	< 1
D4N15_3	< 0.01	0.05	0.05	0.09	< 0.01	< 0.01	< 1
D4N15_4	< 0.01	0.02	0.05	0.03	< 0.01	< 0.01	< 1
D4N15_5	< 0.01	0.05	0.14	0.36	< 0.01	< 0.01	< 1

Instance	Preproc	LP	IP	Sub	Postproc	Alg	Total
D4N15_6	< 0.01	0.06	0.17	0.47	< 0.01	0.02	< 1
D4N15_7	< 0.01	0.03	0.08	0.14	< 0.01	< 0.01	< 1
D4N15_8	< 0.01	0.21	0.56	1.27	0.02	0.01	2.07
D4N15_9	< 0.01	0.02	< 0.01	0.03	< 0.01	< 0.01	< 1
D4N15_10	< 0.01	0.06	0.18	0.56	0.0	< 0.01	< 1
D4N20_1	0.02	< 0.01	0.08	0.05	< 0.01	< 0.01	< 1
D4N20_2	< 0.01	0.05	0.14	0.19	< 0.01	< 0.01	< 1
D4N20_3	< 0.01	0.08	0.12	0.53	0.02	< 0.01	< 1
D4N20_4	< 0.01	0.03	0.15	0.07	0.02	0.02	< 1
D4N20_5	< 0.01	0.02	0.02	0.03	0.01	< 0.01	< 1
D4N20_6	< 0.01	0.34	1.15	5.05	0.02	0.02	6.58
D4N20_7	< 0.01	0.23	1.57	4.14	0.01	< 0.01	5.95
D4N20_8	< 0.01	0.02	0.01	0.05	0.01	< 0.01	< 1
D4N20_9	< 0.01	0.02	0.64	0.37	< 0.01	< 0.01	1.03
D4N20_10	< 0.01	0.01	0.04	0.04	0.0	< 0.01	< 1
D4N25_1	< 0.01	0.11	0.38	1.01	0.01	0.0	1.51
D4N25_2	< 0.01	0.1	0.39	1.89	0.01	0.0	2.39
D4N25_3	< 0.01	0.14	1.29	2.89	< 0.01	< 0.01	4.32
D4N25_4	< 0.01	0.08	0.25	0.27	0.0	< 0.01	< 1
D4N25_5	< 0.01	0.03	0.13	0.05	< 0.01	< 0.01	< 1
D4N25_6	< 0.01	< 0.01	0.44	0.04	0.02	< 0.01	< 1
D4N25_7	0.01	0.0	0.04	0.05	0.0	< 0.01	< 1
D4N25_8	< 0.01	0.01	0.07	0.05	0.0	< 0.01	< 1
D4N25_9	< 0.01	0.01	0.03	0.04	0.02	0.0	< 1
D4N25_10	< 0.01	0.01	0.03	0.04	0.0	< 0.01	< 1
D5N20_1	< 0.01	0.0	0.07	0.05	< 0.01	< 0.01	< 1
D5N20_2	0.0	0.09	0.55	1.15	0.0	< 0.01	1.79
D5N20_3	< 0.01	0.01	0.02	0.13	0.02	< 0.01	< 1
D5N20_4	< 0.01	0.01	0.04	0.04	0.0	< 0.01	< 1
D5N20_5	< 0.01	0.02	0.04	0.04	< 0.01	< 0.01	< 1
D5N20_6	< 0.01	0.02	0.4	0.5	0.01	0.02	< 1
D5N20_7	0.0	0.36	2.54	4.9	0.01	0.01	7.83
D5N20_8	< 0.01	0.01	0.11	0.04	0.0	0.01	< 1
D5N20_9	< 0.01	1.17	141.84	59.27	0.01	0.01	202.31
D5N20_10	< 0.01	0.08	0.35	2.36	0.02	< 0.01	2.8
D5N25_1	< 0.01	0.04	0.23	0.23	< 0.01	< 0.01	< 1
D5N25_2	< 0.01	0.02	0.11	0.07	< 0.01	< 0.01	< 1
D5N25_3	< 0.01	0.01	0.02	0.05	0.0	0.02	< 1
D5N25_4	< 0.01	0.01	0.04	0.07	< 0.01	< 0.01	< 1
D5N25_5	< 0.01	0.01	0.12	0.07	< 0.01	< 0.01	< 1
D5N25_6	< 0.01	0.14	0.64	1.07	< 0.01	0.0	1.86
D5N25_7	< 0.01	0.02	0.78	0.08	0.0	< 0.01	< 1
D5N25_8	< 0.01	0.01	0.08	0.07	0.0	< 0.01	< 1
D5N25_9	< 0.01	0.04	1.16	2.55	0.01	0.02	3.78
D5N25_10	< 0.01	0.23	2.97	4.58	0.02	0.0	7.81
D5N30_1	< 0.01	0.01	0.21	0.08	0.0	< 0.01	< 1
D5N30_2	< 0.01	3.05	139.79	78.66	0.03	0.01	221.54
D5N30_3	< 0.01	0.62	3.3	41.64	0.03	0.01	45.6

Instance	Preproc	LP	IP	Sub	Postproc	Alg	Total
D5N30_4	< 0.01	0.07	0.51	0.58	< 0.01	0.0	1.17
D5N30_5	< 0.01	0.07	0.21	0.42	0.01	< 0.01	< 1
D5N30_6	< 0.01	0.25	1.82	12.99	0.02	0.01	15.09
D5N30_7	< 0.01	0.03	0.82	0.08	0.0	0.02	< 1
D5N30_8	< 0.01	0.02	0.19	0.11	< 0.01	< 0.01	< 1
D5N30_9	< 0.01	0.03	0.57	0.15	0.01	0.02	< 1
D5N30_10	< 0.01	0.86	4.95	15.22	0.03	0.01	21.08

Bibliography

[1] Bertazzi, L., Speranza, M.G.: Inventory routing problems: an introduction. EURO Journal on Transportation and Logistics **1**(4), 307–326 (2012). https://doi.org/10.1007/s13676-012-0016-7

[2] Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2025), https://www.gurobi.com

[3] Hausamann, D., Zirnig, W., Schreier, G., Strobl, P.: Monitoring of gas pipelines—A civil UAV application. Aircraft Engineering and Aerospace Technology **77**(5), 352–360 (2005). https://doi.org/10.1108/00022660510617077

[4] Hong, Y., Jung, S., Kim, S., Cha, J.: Multi-UAV routing with priority using mixed integer linear programming. In: 2020 20th International Conference on Control, Automation and Systems (ICCAS). pp. 699–702. IEEE (2020). https://doi.org/10.23919/ICCAS50221.2020.9268329

[5] Hoppmann-Baum, K., Burdakov, O., Mexi, G., Casselgren, C.J., Koch, T.: Length-constrained cycle partition with an application to UAV routing. Optimization Methods and Software **37**(6), 2080–2116 (published 2022). https://doi.org/10.1080/10556788.2022.2053972

[6] Jonsson, E.: Route Planning for UAVs: A Column Generation Approach. Master's thesis, Linköping University (2025), https://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-211031

[7] Julia Computing, Inc.: The Julia Programming Language (nd), https://julialang.org/, accessed: 2025-10-17

[8] Lubin, M., Dowson, O., Dias Garcia, J., Huchette., J., Legat, B., Vielma, J.P.: JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. Mathematical Programming Computation (2023). https://doi.org/10.1007/s12532-023-00239-3

[9] Manyam, S.G., Rasmussen, S., Casbeer, D.W., Kalyanam, K., Manickam, S.: Multi-UAV routing for persistent intelligence surveillance & reconnaissance missions. In: 2017 international conference on unmanned aircraft systems (ICUAS). pp. 573–580. IEEE (2017). https://doi.org/10.1109/ICUAS.2017.7991314

[10] Mor, A., Speranza, M.G.: Vehicle routing problems over time: A survey. Annals of Operations Research **314**(1), 255–275 (2022). https://doi.org/10.1007/s10288-020-00433-2

[11] Raa, B., Aghezzaf, E.H.: A practical solution approach for the cyclic inventory routing problem. European Journal of Operational Research **192**(2), 429–441 (2009). https://doi.org/10.1016/j.ejor.2007.09.032

[12] Scherer, J., Rinner, B.: Multi-robot persistent surveillance with connectivity constraints. IEEE Access **8**, 15093–15109 (2020). https://doi.org/10.1109/ACCESS.2020.2967650

[13] Stenberg, W.: Optimizing Surveillance Missions for UAV Fleets by Column Generation. Master's thesis, Linköping University (2025), https://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-215111

[14] Zillies, J., Westphal, S., Thakur, D., Kumar, V., Pappas, G., Scheidt, D.: A column generation approach for optimized routing and coordination of a UAV fleet. In: 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). pp. 350–357. IEEE (2016). https://doi.org/10.1109/SSRR.2016.7784326