Monitoring Ships' Emissions Using Unmanned Aerial Vehicles

Anastasios Biblias
School of Informatics,
Aristotle University of Thessaloniki
Greece
mpimplias@csd.auth.gr

Emmanouil S. Rigas
School of Medicine,
Aristotle University of Thessaloniki
Greece
erigas@auth.gr

Nick Bassiliades
School of Informatics,
Aristotle University of Thessaloniki
Greece
nbassili@csd.auth.gr

ABSTRACT

The use of Unmanned Aerial Vehicles (UAVs) to monitor airborne emissions from ships has become quite widespread, especially in maritime areas where stricter restrictions have been imposed to minimize air pollution caused by ships. This paper focuses on this application of drones and specifically aims to schedule the trips that will be made by a set of drones in order to monitor as many ships as possible given a set of spatial and temporal constraints. The general drone scheduling problem is broken down into two sub-problems: In the first sub-problem, the number of ships and the set of requests for ship surveillance, (i.e., the positions and times that require surveillance) are known in advance. This problem is modeled as an Integer Linear Programming (ILP) one and is solved offline and optimally. In the second sub-problem, similarly to the first, the number of ships is known in advance; however, the set of requests is not, thereby requiring that, during the period the drones are either on their trips or idle, new monitoring requests may be generated. To solve this sub-problem, an online heuristic algorithm was developed. Both optimal solution and heuristic algorithm are evaluated on various sets of realistic data, and their efficiency is verified, with the main conclusion being that the optimal solution always ends up monitoring more ships than the heuristic, while the heuristic algorithm is significantly faster than the optimal and scales to larger problems.

CCS CONCEPTS

• **Computing methodologies → Planning and scheduling**; **Search methodologies**; *Modeling and simulation*; **Artificial intelligence**.

KEYWORDS

drone, unmanned aerial vehicle, emissions, monitoring, ILP, optimization, heuristic

ACM Reference Format:
Anastasios Biblias, Emmanouil S. Rigas, and Nick Bassiliades. 2024. Monitoring Ships' Emissions Using Unmanned Aerial Vehicles. In *13th Conference on Artificial Intelligence (SETN 2024), September 11–13, 2024, Piraeus, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3688671.3688777

SETN 2024, September 11–13, 2024, Piraeus, Greece

ACM ISBN 979-8-4007-0982-1/24/09
https://doi.org/10.1145/3688671.3688777

1 INTRODUCTION

The brothers Jacques and Louis Bréguet, who worked for Charles Richet, a controversial Nobel Prize winner, were the first inventors to create a quadcopter in 1907, the first design to resemble a modern drone.[1] An unmanned aerial vehicle (UAV), unmanned aerial system (UAS), remotely piloted aircraft system (RPAS), or simply a *drone*, is any type of flying vehicle that does not have a pilot in the fuselage, but flies either autonomously or by remote control.

The development of drones in recent years has been rapid, and as a result their use has become diverse. Amongst others, drones have emerged as one of the most versatile and powerful tools for monitoring environmental and ecological change [2, 10]. From tracking wildlife populations to assessing the impact of deforestation, allowing researchers to easily monitor forests, wetlands and other ecosystems that were previously difficult to access, drones have been proven invaluable. In the case of wildfires, drones equipped with thermal imaging can pinpoint hotspots and guide firefighting efforts, and in coastal regions, drones can monitor coral reefs, which are sensitive indicators of ocean health. Another major application of drones in this domain is the monitoring of air pollution [8]. The monitoring can cover wide areas such as a city neighbourhood, or specific infrastructures, or vehicles such as a factory, or a ship.

In this paper, we focus on the use of drones in monitoring air pollutant emissions from ships. Specifically, we aim to solve the problem of scheduling drones to monitor ships given a set of temporal and spatial constraints, a problem that can be traced back to the Drone Scheduling Problem (DSP) [6]. We simulate a sea area with a graph whose nodes are either points that ships will pass through at some points in time, or the stations of the drones. The aim is to maximize the number of ships that are monitored given a time span. To solve it, we develop two approaches: The first is an optimal one based on Integer Linear Programming (ILP) techniques, and concerns the scenario where we know in advance the position and the time that each ship will need to be monitored. The second is a heuristic algorithm, which concerns the scenario in which the monitoring requests are produced dynamically over time.

2 RELATED WORK

The problem that we solve in this paper shares similarities with a number of problem categories such as the Multiple Travelling Salesman Problem (MTSP) [1], the Electric Vehicle Routing Problem (EVRP) [4], and more importantly the Drone Scheduling Problem (DSP) [6]. The Drone Scheduling Problem (DSP) is a generic problem category and several research approaches can be classified under it, many of them being related to the use of drones to monitor emissions. For example, in [14] the authors treat the DSP problem of

[1]https://www.internationaldroneassociation.com/general-6

scheduling drones to monitor vessels' emissions as an NP-hard problem, aiming to inspect as many ships as possible, also considering the possibility of prioritizing the inspection of ships. In contrast, [12] and [3] propose an ant colony-based algorithm for solving the problem of scheduling drones for ships' emissions monitoring, which according to experimental results outperforms others both in terms of solution quality and solution speed. However, in [11] the authors approach the same problem taking also into account the variations in the speed of ships and drones due to weather conditions, and deal with the dynamic dispatch of drones for surveillance of a ship based using Reinforcement Learning (RL). Moreover, the authors in [9], use a cooperative multi-UAV unmanned aerial vehicle (UAV) algorithm that incorporates a tabu matrix into particle swarm optimization (PSO) to schedule drones for ships' monitoring in ports. In a different domain, [13] focus on the DSP problem which is addressed using a Fast Path Planning with Rules (FPPWP) algorithm, where a basic framework for aerial data collection is designed, which includes the following five elements: network deployment, node placement, anchor point search, fast drone path planning, and network data collection. Finally, an interesting approach from a different domain, to solving a problem very similar to the one studied in this paper, is that of [5] where the authors simply aim to collect data on the energy consumption of a company's customers, using an ant colony optimisation system.

This work has been inspired by these works and the problems that they solved, and is stepping upon the work by Rigas et al. [7] where they propose algorithms to schedule the monitoring of locations by drones. In contrast to [7], here we consider drones with a limited range which is renewed by replacing the battery with a fully charged one at the drones' stations, and we also propose an online algorithm to solve the problem in hand. In contrast to the rest of the state of the art, we propose a holistic approach where drones powered by an electric motor are scheduled to monitor the gas emissions of ships given specific spatial and temporal constraints. In doing so, we solve the problem both offline and to optimality assuming full knowledge of the demand in advance and online where the demand appears dynamically over time. In the first case we use an exact solution (i.e., MILP), while in the second one an approximate solution (i.e., heuristic search). In all cases, we schedule the charging of the drones' batteries in order to maximize their availability.

3 PROBLEM DEFINITION

The purpose of this work is to schedule the flying and the charging of a fleet of drones $d \in D \subseteq \mathbb{N}$ aiming to monitor the emissions of a set of ships $v \in V \subseteq \mathbb{N}$. Please note that terms *ship* and *vessel* are used interchangeably reffering to the same entity. We model the area of interest as an undirected graph $G(N, E)$ where the nodes $n \in N \subseteq \mathbb{N}$ are either stations for the drones where they can park and charge $c \in C \subseteq N$, or locations in the sea where ships must be monitored and the edges $e \in E \subseteq \mathbb{N}$ are the straight-line links between the nodes. Time is divided into a set of discrete points $t \in T \subseteq \mathbb{N}$ and is the same for both drones and ships. Between two nodes, a distance $dist_{n,n'} \in T$ in terms of points in time a drone needs to fly across the two locations exists. In this context, monitoring requests $demand_{v,n,t} \in \{0, 1\}$ exist, where the monitoring for a ship v should take place at a particular node n, a particular point in time (or time slot) t. Each drone has a current $b_t^d \in \mathbb{N}$ and a maximum b_{max}^d battery level and drones can recharge their batteries in any of the stations c_n.

Each ship traverses a set of nodes in the sea area at different points in time, forming a route. A scheduling algorithm selects for each drone the nodes it will visit and determines the times to monitor the ships it has chosen for inspection. Furthermore, a single drone can monitor multiple ships, provided that the monitoring occurs at different times. The drone stations are nodes from which drones launch to begin their inspection routes and where they land to recharge their batteries. It should be noted that from this problem, we have identified two possible scenarios: The first scenario involves a situation where the problem data is known in advance, and we solve it optimally using ILP. The second scenario arises when the problem data is dynamically updated during the execution of the scenario, and we have developed a heuristic algorithm to address it.

The problem data are summarized as follows:

- The number of vessels to be inspected.
- The nodes through which each ship will pass, and the time point this passing will happen.
- The number of drones.
- The number of nodes-stations, nodes that are the initial and final position of the drones and where they can charge their battery.
- Number of points in time, i.e., the time interval for which the optimal paths of the drones have to be calculated,
- The autonomy of drones, i.e., how many points in time they can fly until they need recharging.
- A set of coordinates that constitute the nodes through which ships pass, as well as the station nodes.

4 OPTIMAL OFFLINE SOLUTION

When solving the aforementioned problem offline and to optimality, we assume full knowledge of all problem data as mentioned in the previous section. This problem is modeled as a system of linear equations, a set of constraints, and an objective function, which is given to IBM ILOG CPLEX library to calculate the optimal solution. CPLEX is considered the state of the art is solving large scale and hard optimization problems.

Four sets of decision variables are used to compile the constraints and objective function, which relate to drones, time slots, and ship control. Specifically, the variable $x_{n,t} \in \{0, 1\}$
is a two-dimensional matrix of nodes by time slots, with each cell being a decision variable indicating whether at least one drone is at a node in a time slot. The variable $y_{d,n,t} \in \{0, 1\}$ is a three-dimensional matrix of drones by nodes by time slots, with each cell being a decision variable indicating whether a drone is at a node in a time slot. The variable $k_{d,t,n,n1} \in \{0, 1\}$ is a four-dimensional matrix of drones by time slots by nodes by nodes, with each cell being a decision variable indicating whether a drone is traveling from one node to another node in a time slot. Finally, the variable $start_{d,t,n,n1} \in \{0, 1\}$ is a four-dimensional matrix of drones by time slots by nodes by nodes by nodes, with each cell being a decision

variable indicating whether a drone starts traveling from one node to another node in a time slot.

The objective function (1) has two parts. Our goal is to maximise the first part, which is the number of ships monitored, while we want to minimise the second part, where the second part is the number of time slots that the drones travel (i.e., avoiding unnecessary traveling), to ensure that we find the best route between the nodes that the drones travel through. The second part of the objective function is multiplied by a very small number μ in order to ensure that it will never become larger than the first one and, thus executing monitoring tasks is not affected.

Objective function:

$$Maximize \sum_{v=0}^{V}\sum_{n=0}^{N}\sum_{t=0}^{T}(demand_{v,n,t} \times x_{n,t}) - \mu \times \sum_{d=0}^{D}\sum_{n=0}^{N}\sum_{n1=0}^{N}\sum_{t=0}^{T-1}|k_{d,t+1,n,n1} - k_{d,t,n,n1}| \tag{1}$$

Temporal, spatial, and routing constraints:

$$\forall d, \sum_{c=0}^{C} y_{d,c,0} = 1 \tag{2}$$

$$\forall d, \sum_{c=0}^{C} y_{d,c,T-1} = 1 \tag{3}$$

$$\forall n \forall t, \sum_{d=0}^{D} y_{d,n,t} \geq x_{n,t} \tag{4}$$

$$\forall d \forall t, \sum_{n=0}^{N} y_{d,n,t} \leq 1 \tag{5}$$

$$\forall d \forall t, \sum_{n=0}^{N}\sum_{n1=0}^{N} k_{d,t,n,n1} \leq 1 \tag{6}$$

$$\forall d \forall t, \sum_{n=0}^{N}\sum_{n1=0}^{N} k_{d,t,n,n1} \leq 1 \tag{7}$$

$$\forall d \forall t, \sum_{n=0}^{N} y_{d,n,t} + \sum_{n=0}^{N}\sum_{n1=0}^{N} k_{d,t,n,n1} = 1 \tag{8}$$

$$\forall d \forall t, \sum_{n=0}^{N} y_{d,n,t} + \sum_{n=0}^{N}\sum_{n1=0}^{N} start_{d,t,n,n1} \leq 1 \tag{9}$$

$$\forall d \forall t \forall n \forall n1 \left(t + dist_{n,n1} \geq T\right), k_{d,t,n,n1} = 0 \tag{10}$$

$$\forall d \forall t \forall n \forall n1 \left(t + dist_{n,n1} \geq T\right), start_{d,t,n,n1} = 0 \tag{11}$$

The symbols of the optimal solution	
demand	It represents the node and the time at which each ship requiring monitoring
x	Decision variable indicating whether a drone is present at a node at a given time
μ	A very small number, namely 0.001
k	Decision variable indicating for each drone whether at some point in time it goes from one node to another
y	Decision variable indicating which node each drone is in at any given time
start	Decision variable indicating for each drone whether at some point in time it starts going from one node to another
V	Representing all vessels
N	Represents the set of nodes
T	It represents the set of moments in time
dist	Variable representing the distance between two nodes

Table 1: The symbols of the optimal solution

$$\forall d \forall t \forall n \forall n1 \quad t + dist_{n,n1} \geq T \in \left(t, t + dist_{n,n1}\right), k_{d,t,n,n1} \geq start_{d,t,n,n1} \tag{12}$$

$$\forall d \forall t \forall n \forall n1 \left(t + dist_{n,n1} + 1 \geq T\right), y_{d,n,t-1} \geq start_{d,t,n,n1} \tag{13}$$

$$\forall d \forall t \forall n \forall n1 \left(t + dist_{n,n1} + 1 \geq T\right), y_{d,n,t+dist_{n,n1}} \geq start_{d,t,n,n1} \tag{14}$$

$$\forall d \forall t : t + b_t^d < timeSlots, \sum_{c=0}^{C}\sum_{n1=0}^{N}\sum_{t1=t}^{t+b_t^d} y_{d,c,t1} \geq \sum_{c=0}^{C}\sum_{n1=0}^{N} start_{d,t,n,n1} \tag{15}$$

$$\sum_{t=0}^{T}\sum_{n=0}^{N}\sum_{d=0}^{D}\sum_{n1=0}^{N} k_{d,t,n,n1} = \sum_{t=0}^{T}\sum_{n=0}^{N}\sum_{d=0}^{D}\sum_{n1=0}^{N} \left(dist_{n,n1} \times start_{d,t,n,n1}\right) \tag{16}$$

$$\sum_{t=0}^{T}\sum_{n=0}^{N}\sum_{d=0}^{D} \left|y_{d,n,t+1} - y_{d,n,t,}\right| = \sum_{t=0}^{T}\sum_{n=0}^{N}\sum_{d=0}^{D}\sum_{n1=0}^{N} start_{d,t,n,n1} \tag{17}$$

In order to achieve the goal of the objective function, we impose some constraints both on the correct behavior of the drones and on the correct cooperation between the decision variables. First, we describe the constraints that belong to the first category: Each drone must be at a node station both at the beginning (Eq. 2) and at the end (Eq. 3) of the period in which the scenario is examined, and each drone can be at a maximum of one node at any given time (Eq. 5). Furthermore, each drone can only fly from one node to one other node at any given time (Eq. 6), but can also only initiate one transition from one node to another at any given time (Eq. 7) Also, before a drone can start a trip, some conditions have to be met: (a) a trip should not take place (Eq. 10) or be started (Eq. 11) if there is not enough time left to complete the trip, (b) to start a trip the drone has to be in the correct node at the previous point in time (Eq. 13), and (c) for each trip the drone has to be in the correct node after the transition (Eq. 14). Of course, each drone should fly as long as its battery autonomy allows and should have returned to a station to recharge (Eq. 15). As for the second class of constraints, one had to be defined in such a way that the decision variables x and y communicate correctly, i.e., if there is at least one drone at a node at a given time, then the sum of y at that node and at that time for all available drones should be greater than 1 (Eq. 4). Furthermore, it must be checked that the two decision variables (y and k) cannot be 1 at the same time, i.e., a drone cannot be at a node (y[i][j][l]=1) and at the same time fly from one node to another (k[i][j][l][m]=1) (Eq. 8), but neither the decision variables start and y are 1 at the same time, i.e., a drone cannot be at a node and start a trip at the same time (Eq. 9). Also, the variable k must have a value greater than or equal to the value of start (Eq. 12), and this is because in the case start = 1, the drone has started a trip, so k must be 1. Finally, the total time a drone flies must be equal to the number of starts times the relative distance of the executed trips (Eq. 16), but the number of times y changes value should be twice the number of times a trip is started (this is because y changes value when it starts and when it ends - so 2 times) (Eq. 17).

5 GREEDY ONLINE ALGORITHM

As far as the second variation of the problem is concerned, an online heuristic algorithm is developed. This algorithm takes as input the set of nodes, drones and ships but the monitoring requests are generated dynamically. Thus, although the aim remains maximizing the number of ships being monitored, the approach differs to cope with the dynamic nature of the problem.

In this algorithm (see Algorithm 1), the way in which the set of requests is generated has a peculiarity. In particular, each request is characterised by three variables, the request time variable, which refers to the time at which the program will accept the given request, the node variable, which refers to the node where the ship requesting the inspection will be located, and the timeSlot variable, which refers to the point time at which the inspection is required. This algorithm is based on four checks shall be carried out at each time point of the given time period:

(1) The first check concerns the creation of the set of requests. For each time point, it is verified whether one or more requests have been generated. If so, the set of requests is updated accordingly.
(2) The second check involves supervising the drones when they need to return to the nearest station. This could be due to a low battery requiring recharging, or to reach a station before the end of the time period.
(3) The third control concerns the assignment of a request to a drone. It involves checking and finding the first available drone (a drone is considered available when no request is assigned to it) and assigning it the closest request in time under the following three conditions:
   (a) The drone has enough battery to fly to the request node, inspect the ship, and return to the closest station from the node where the inspection was made.
   (b) There must be enough time for the drone to fly to the demand node, inspect the ship, and return to the nearest station from the node where the inspection was made before the time expires.
   (c) There must be enough time for the drone to reach the demand node before the time period in which the inspection must be made.
(4) The fourth and final check concerns the renewal of each drone's battery. Specifically, for each time point a drone is outside a node station, the battery will be reduced by one unit. While a drone is in a station node, the battery variable will be set to the maximum value (i.e., the battery is assumed to be replaced with a fully charged one).

Algorithm 1 Greedy online algorithm

```
for Each time slot do
    for Each vessel do
        if Demand time of the vessel matches the current time slot then
            Assign the demand to the table with pre-supervised demands
    for Each drone do
        if Remaining battery + remaining time equals duration then
            Send the drone to the nearest station
    for Each drone d do
        if d is not flying then
            for Each node n do
                for Each time t do
                    if There is demand at node n at time t and there is enough time for d to satisfy it then
                        if Remaining battery and time are sufficient for d to reach and return from node n then
                            Send d to node n to satisfy the demand
                            Update the table with satisfied demands
    for Each drone d do
        if d's current position is not a station or d is flying then
            Reduce d's battery by one unit
        else
            Recharge d's battery to maximum capacity
```

		Execution time (Secs)	
		Optimal	Greedy
Number of drones	1	2.189	0.00739896
	2	92.818	0.01819772
	3	2412.301	0.02489308

Table 2: Execution time Optimal vs Greedy.

		Efficiency (% completed requests)	
		Optimal	Greedy
Number of drones	1	21%	8%
	2	28%	17%
	3	29%	19%

Table 3: Percentage of completed requests Optimal vs Greedy.

6 EXPERIMENTAL EVALUATION

Due to the difficulty in finding real data, synthetic realistic data is used in the examples. It would also be good to clarify that, for the sake of this experimental evaluation, we will assume that all drones have the same characteristics .In particular, the sets of nodes used consisted of nodes of the form (1,1), (2,2), etc. The requests were the result of a random generator, assuming that every ship has a request. In order to ensure the comparability of the solutions, code was also developed to store and read the demands from files. In this section we will examine the effect that the number of drones has on the set of requests that are satisfied. To do this, 10 different sets of requests were created and fed to each solution. So each solution was run for 10 different sets of requests, with each set being tested to see if it could be met for 1, 2 and 3 drones. The reason for using 10 different sets is to have transparency in the results of this test, and so by averaging the results for each value of the number of drones, the desired transparency is achieved. Apart from the number of drones that was variable, 5 nodes, 15 time points, 2 stations, 10 time points autonomy and 20 vessels existed.

It is easy to see (Table 3 and Figure 2) that the optimal solution gives better results on the same data with the same parameters. However, the heuristic is considerably faster (Table 2 and Figure 1) than the optimal, especially when using 2 and 3 drones. Note that the heuristic has the disadvantage that, although it could satisfy some requirements if they were known in advance, this is very unlikely to happen as there is a possibility that these requirements will be generated and become known to the heuristic algorithm at a time when it has just enough time to start a trip towards them.

Overall, the optimal solution calculated an optimal solution for the problem in hand, but its highly combinatorial nature makes this solution hard to find and, thus, time consuming. Therefore, this solution is usable for small to medium sized problems and for benchmarking purposes. At the same time, although the greedy online algorithm has a performance deficite compared to the optimal, it scales very well and can be considered usable for a wider range if problem sizes.

It is also interesting to examine the influence that the number of drones, combined with the total number of nodes, has on the total number of requests served by the heuristic algorithm. To achieve this, 10 different sets of requests were created and given to the algorithm. So the algorithm was run for 10 sets of requirements, where each set was tested for each of the combinations concerning the number of drones, with values 2, 4, 6, 8, and the total number of nodes, with values 5, 10, 15, 20. The reason for using 10 different sets is to have transparency in the results of this test, and so by averaging the results for each combination of the values of the number of drones and the values of the number of nodes, the desired transparency is achieved. To get better results, the number of stations is also varied according to the number of nodes. Apart from the number of drones and nodes that were variable, 40 time points, 2 stations, 10 time points autonomy and 20 vessels existed.

Figure 1: Execution time (Optimal vs Greedy) in seconds.

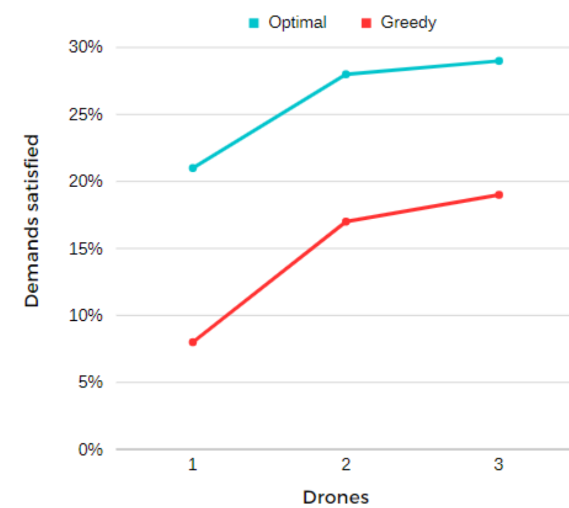

Figure 2: Percentage of completed requests (Optimal vs Greedy).

		Efficiency (% completed requests)			
		Number of nodes			
		5	10	15	20
Number of drones	2	50.5%	31%	25.5%	21.5%
	4	63%	43%	32%	28.5%
	6	64.5%	46%	33.5%	28.9%
	8	64.5%	46.5%	33.5%	28.9%

Table 4: Percentage of executed tasks Greedy - number of nodes and number of drones.

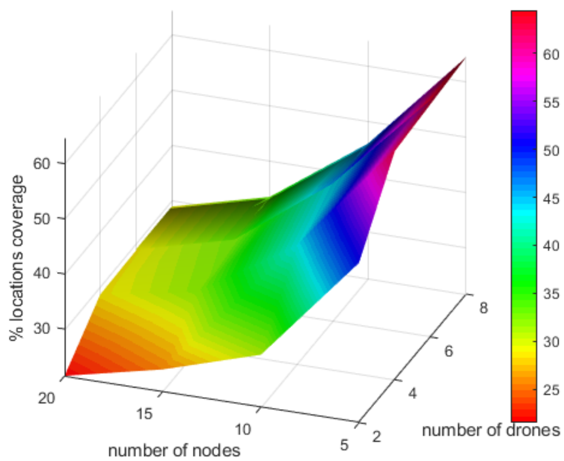

Figure 3: Percentage of completed requests related to the number of nodes and the number of drones - Greedy

From Table 4 and Figure 3, we can see that the fewer nodes the algorithm deals with, the better the results it returns. This is perfectly normal, as too many nodes mean long distances and this results in many requests not being satisfied because there is no time to travel to the nodes requesting surveillance. On the contrary, as the number of drones increases, regardless of the number of nodes, there is a proportional change in the satisfied set of requests.

7 CONCLUSIONS AND FUTURE WORK

The aim of this work was to schedule a number of drones to monitor as many ships as possible in order to reduce air pollutant emissions to the desired level and to find the optimal routes for the drones, taking into account the limited autonomy of the drone battery and the limited time frame. To achieve these objectives, two different approaches were implemented. The first approach concerns the scenario in which the demand set (i.e., the set of nodes and times at which each ship requires surveillance by a drone) is known in advance. This approach uses the CPLEX library to solve the linear model representing the problem and can provide an optimal solution. The second approach deals with the scenario where certain data, such as the battery autonomy of the drones, the time period, the set of nodes through which the ships pass, and the set of drones, are known, but the set of requirements is not known in advance. In this case, each demand becomes known dynamically as it arrives. In conclusion, judging from the information derived from the previous paragraph, the best option to achieve the objectives set at the beginning is to know the demands in advance and to use an optimization algorithm, as in the first approach. However, this scenario may not always be realistic since it is quite normal for the set of requirements to be unknown in advance and for other requests to arrive dynamically.

In terms of future work, we are interested in evaluating the proposed solution in a set of real data both in terms of monitoring requests and locations and in terms of using different types of drones with different characteristics in terms of speed, altitude, range and capacity. This would clarify whether the proposed approaches are still capable in calculating correct and high quality solutions, or whether adaptations would be needed. Also, currently we assume that the battery of each drone is replaced with a fully charged one when it is at a node station, which may not always be feasible. Thus, battery charging should also be studied. Finally, regarding the online algorithm, the selection of the demand to be satisfied at each point in time is based purely on the order in which the demands arrive. Instead of this approach, other criteria could be developed to decide which demand to satisfy at a given time.

REFERENCES

[1] Tolga Bektas. 2006. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega* 34, 3 (2006), 209–219.
[2] Siyi Liao, Jun Wu, Jianhua Li, Ali Kashif Bashir, and Wu Yang. 2021. Securing Collaborative Environment Monitoring in Smart Cities Using Blockchain Enabled Software-Defined Internet of Drones. *IEEE Internet of Things Magazine* 4, 1 (2021), 12–18.
[3] Xiaosong Luo, Zhao-Hui Sun, and Siqi Qiu. 2021. Ant colony system based drone scheduling for ship emission monitoring. In *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 241–247.
[4] Michalis Mavrovouniotis, Changhe Li, Georgios Ellinas, and Marios Polycarpou. 2019. Parallel ant colony optimization for the electric vehicle routing problem. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1660–1667.
[5] José RT Neto, Azzedine Boukerche, Roberto S Yokoyama, Daniel L Guidoni, Rodolfo I Meneguette, Jo Ueyama, and Leandro A Villas. 2017. Performance evaluation of unmanned aerial vehicles in automatic power meter readings. *Ad Hoc Networks* 60 (2017), 11–25.
[6] Junayed Pasha, Zeinab Elmi, Sumit Purkayastha, Amir M. Fathollahi-Fard, Ying-En Ge, Yui-Yip Lau, and Maxim A. Dulebenets. 2022. The Drone Scheduling Problem: A Systematic State-of-the-Art Review. *IEEE Transactions on Intelligent Transportation Systems* 23, 9 (2022), 14224–14247.
[7] Emmanouil S. Rigas, Panayiotis Kolios, and Georgios Ellinas. 2020. Extending the Multiple Traveling Salesman Problem for Scheduling a Fleet of Drones Performing Monitoring Missions. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 1–6.
[8] Godall Rohi, O'tega Ejofodomi, and Godswill Ofualagba. 2020. Autonomous monitoring, analysis, and countering of air pollution using environmental drones. *Heliyon* 6, 1 (2020), e03252.
[9] Lixin Shen, Yaodong Wang, Kunpeng Liu, Zaili Yang, Xiaowen Shi, Xu Yang, and Ke Jing. 2020. Synergistic path planning of multi-UAVs for air pollution detection of ships in ports. *Transportation Research Part E: Logistics and Transportation Review* 144 (2020), 102128.
[10] Kurt W Smith. 2015. The use of drones in environmental management. In *World Environmental and Water Resources Congress 2015*. 1352–1361.
[11] Poly ZH Sun, Xiaosong Luo, Tienyu Zuo, Yuguang Bao, Yan-Ning Sun, Rob Law, and Edmond Q Wu. 2022. Emission monitoring dispatching of drones under vessel speed fluctuation. *IEEE Transactions on Intelligent Transportation Systems* 23, 11 (2022), 21833–21847.
[12] Zhao-Hui Sun, Xiaosong Luo, Edmond Q Wu, Tian-Yu Zuo, Zhi-Ri Tang, and Zilong Zhuang. 2021. Monitoring scheduling of drones for emission control areas: An ant colony-based approach. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2021), 11699–11709.
[13] Chengliang Wang, Fei Ma, Junhui Yan, Debraj De, and Sajal K Das. 2015. Efficient aerial data collection with UAV in large-scale wireless sensor networks. *International Journal of Distributed Sensor Networks* 11, 11 (2015), 286080.
[14] Jun Xia, Kai Wang, and Shuaian Wang. 2019. Drone scheduling to monitor vessels in emission control areas. *Transportation Research Part B: Methodological* 119 (2019), 174–196.