Optimal Scheduling with a Reliable Data Transfer Framework for Drone Inspections of Infrastructures

Golizheh Mehrooz and Peter Schneider-Kamp

Dept. of Mathematics & Computer Science
University of Southern Denmark
Odense, Denmark
email: mehrooz@imada.sdu.dk email: petersk@imada.sdu.dk

***Abstract*—Unmanned Aerial Vehicles (UAVs) or drones have gained a lot of interest due to their advantages for inspecting infrastructures. However, they have a limited flight time. In order to solve this problem, we designed a cloud server and analyzed a reliable communication link between the cloud server and the drones. Furthermore, we have proposed an optimal scheduling algorithm to assign an energy-efficient trajectory for the Internet of Drones applications of infrastructure inspection. An optimal scheduling algorithm based on extended OR-Tools as a travelling salesman problem solver is hosted as a Docker container in the cloud server. We implemented a framework and validated the quality aspect of the optimal scheduling algorithm along with the communication link between the cloud and the drones. The overall architecture of the designed platform is illustrated along with the static analysis of the communication link and scheduling algorithm.**

***Keywords*—*UAV*, *Cloud server*, *ROS*, *Scheduling*, *Rosbridge*.**

I. INTRODUCTION

Drones, also called Unmanned Aerial Vehicles (UAVs), are defined as small aircraft, which are operated without a human pilot [1]. With the development of new technologies, drones have received an increasing amount of attention in various areas for automatizing labor-intensive tasks [2]. Likewise, new European Union regulations for drone inspections have eased the process of obtaining permission for inspecting the special case of linear infrastructures such as power pylons. These regulations require the drone to stay within close range to the linear infrastructure [3] [4]. Therefore, finding optimal routing and scheduling algorithms that minimize the flight time drones spend away from the immediate vicinity of the linear infrastructure [4] represents a highly relevant task. In this regards, designing a cloud-based platform, which includes a Python package for the optimal routing and scheduling solution for inspection drones, will improve the inspection speed, cost, accuracy, and safety.

For this reason, a considerable amount of research has been conducted in order to transfer data between the drone and the cloud server. In these scenarios, the data collected should be transferred to the cloud server, where they can later be aggregated and analyzed using specialized data processing algorithms [5]. In the work presented in this paper, we propose a novel Internet of Drones (IoD) data transfer framework for cloud-based applications. Data includes either navigational data for controlling drones or the images to be analyzed in the cloud by using customized machine learning algorithms to detect and explain outliers [6]. To transfer these data between the cloud and the drones, we utilize a reliable communication link named *Rosbridge* [7]. We analyze the communication delay by considering various data sizes. The main contributions of the work presented in this paper are as follows:

1) An optimal scheduling algorithm based on extended OR-Tools [8] as a Traveling Salesman Problem (TSP) solver for inspection drones along the linear infrastructure.
2) A system design and architecture framework for transferring data such as navigational data or the images from the drone to the cloud server and vice versa.
3) An analysis of *Rosbridge* for transferring various data sizes between the cloud and the drones.

Figure 1 illustrates the overall system architecture and design corresponding to contribution (2). As can be seen in the figure, the proposed framework has two layers. The first layer is the cloud server. The cloud server is based on containerizing applications by using Docker. The design further is divided into a frontend (i.e., OpenLayers) and a backend (i.e., Linear infrastructure Mission Control (LiMiC)).

LiMiC is implemented as a python package for solving the routing and scheduling problem. OpenLayers is used as an open-source frontend technology for designing a web interface to monitor and control drones. The Docker platform is used to accelerate the development process and scale applications with ease. The image processing service, which is also designed as a Docker container, hosts customized machine learning algorithms for analyzing the images along the infrastructure. Kubernetes is designed for managing the container applications. In addition, we use databases for storing data in the cloud server.

The second layer in Figure 1 (2) shows the Robot Operating System (ROS)/ROS2 as a drone control unit. This layer hosts ROS/ROS2 as a high-level software for the drone system. The communication link between the cloud server and the drone system is *Rosbridge* while the communication link between the users and the cloud server is Hyper Text Transfer Protocol (HTTP).

The paper unfolds as follows. In Section II, we summarize the background of the scheduling and IoD applications for

 ISBN: 978-1-61208-893-8

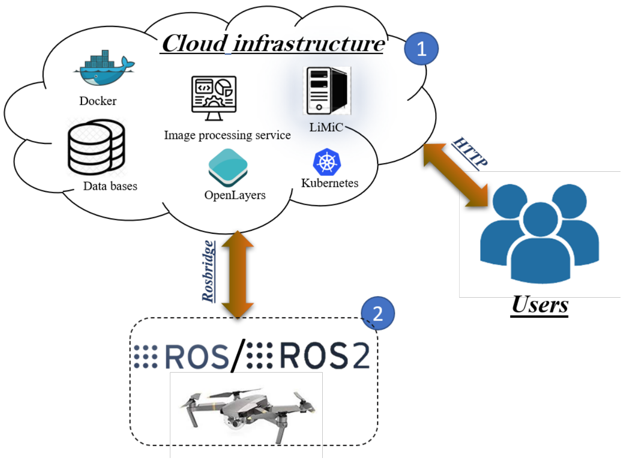


Figure 1. Overall system structure

drones inspection of linear infrastructure. In Section III, we explain the optimal multi-drone scheduling problem by using the extended OR-Tool. We measured the time of multi-drones scheduling in Fyn, Denmark. In Section IV, we analyze *Rosbridge* as a communication protocol for IoD applications. Sections V concludes this paper and presents directions for future research.

II. Related work

In this section, we explore the background of the communication link between the cloud server and the drone. We also investigate the optimal routing and scheduling algorithms for IoD applications.

A. Data Transfer framework for IoD applications

A variety of research and publications have been undertaken on different IoD applications for establishing a communication link between drones and the cloud server. In this regard, the widely used IoD protocols are *Rosbridge* [4], and HTTP [9].

The HTTP protocol is a popular communication protocol for Internet of Things (IoT) applications. It provides a request/reply mechanism for transferring data. HTTP implements four methods: GET, POST, PUT, and DELETE. In [10], the authors integrated drone resources as a web service into the cloud. In this scenario, the drone becomes part of the cloud server and can be accessed ubiquitously [10]. The authors have used RESTful HTTP components in their system architecture. They have used the HTTP GET method with the resource URI to retrieve the current energy level. In the work presented in this paper, we have also used HTTP protocol for communicating between the backend, e.g., LiMiC [4], and the web interface which is hosted in the cloud. The HTTP GET method in this paper is used to update the drone telemetry data on the web interface.

Rosbridge is another increasingly popular communication protocol. It provides a communication interface to application programs without venturing into the specialized world of robotics engineers [11]. *Rosbridge* acts as a middleware abstraction layer to access the applications programs that are not themselves robotics. A cloud-based web application that uses *Rosbridge* has gained a lot of interest for providing real-time flight data monitoring and management for Drones [12].

In [7], the authors designed and implemented a web interface for controlling and monitoring drone flight. They have used *Rosbridge* as a communication link between the cloud and the drone. They have analyzed the communication delay in their framework. However, they have not measured the communication delay for transferring large data such as images. Additionally, they have not considered the scheduling problem for multi-drones. In the work presented in this paper, we extended the previous work [4] [7] by considering the scheduling problem (extending LiMiC) and analyzing the delay time for transferring large data such as images.

B. Optimal Routing and Scheduling algorithm

The state of art of drone routing and scheduling mainly involves the TSP [13] and the Vehicle Routing Problem (VRP) [14]. TSP is the classic routing problem, in which there is just one vehicle, while the VRP is the generalization of TSP with multiple vehicles [8]. In VRP, each customer is served by exactly one vehicle [15]. Each vehicle starts the path from the start position, performs the task, and returns to the start position.

In [15], the authors considered the occurrences of failures that make drones unable to continue the flight. They have found that the amount of lost demand depends on the location where drones fail, such as failure at the beginning of the path or on the way back to the start position. In the work presented in this paper, inspection drones should not return to the start position. Furthermore, we have not considered the drone failures scenario because in the case of infrastructure inspection, if one drone fails it does not cause a significant disturbance of the overall inspection mission. The main reason is that, in real drone flight for inspections, they usually fly within a small distance from each other. Therefore, in the case of failure, another drone close to that location can cover the inspection task.

The Drone Scheduling Problem (DSP) aims to design a group of flight tours for the drones. In [16], the authors explain DSP to solve the problem of inspecting as many vessels as possible in a short time. In this regard, they have prioritized highly weighted vessels to be inspected first [16]. In the case of power pylons inspection, the main important parameter is the time due to the battery constraint on the drone. Therefore, our main goal in this work is to find the optimal scheduling solution in a short time for a group of drones.

III. Optimal Scheduling with Extended TSP solver

There are well-known developed solutions for the TSP that can solve a multitude of path planning problems. The goal of these algorithms is to find the shortest route (less costly path) for a salesman who needs to visit customers at different locations and return to the starting location.

OR-Tools [8] is an open-source software for solving the optimization problem for vehicle scheduling. In order to use

 ISBN: 978-1-61208-893-8

OR-Tools, we need to create data. The data include the number of drones and the distance matrix detailing the distance between any two locations under consideration, as well as the start and the end location for the route. In this paper, as we are not interested in the drone returning to the start location, we modify the distance matrix to 'trick' the solver. This comprises adding an extra row to the matrix with 0 values. This assumption does not contribute any additional cost to the path. For the case of power line inspections, the distance matrix is an array, in which i, j entry is the distance from pylon A to pylon B. The distance from pylon A to pylon B is calculated based on an extended A* algorithm as explained in [4]. In order to create a distance matrix without undue waste of computational time, we use the fact that the distance from pylon A to pylon B is the same as the distance from pylon B to pylon A. Therefore, we have a symmetric distance matrix as it is shown in Figure 2. Here, the pairs marked in red have a reverse-ordered counterpart, which implies that we can skip the calculation of these pairs.

	A	B	C	D
A	AA	AB	AC	AD
B	BA	BB	BC	BD
C	CA	CB	CC	CD
D	DA	DB	DC	DD

Figure 2. Symmetric matrix with repeated pairs marked

OR-Tools offers two general approaches for scheduling problems such as the *first solution strategies* and the *meta-heuristic strategies*. The *first solution strategies* are designed to find a single path between all points. This approach is a fast method for finding the optimal route. The *meta-heuristic strategies* are slower than the *first solution strategies* in general but more reliable at finding the optimal route. The advantage of using the *meta-heuristic strategies* is that, its potential to avoid local minima, which the *first strategies* often end up in.

We have measured the scheduling time for 10 power pylons in Denmark by using the above approaches for a single drone. The scheduling using the *first solution strategies* takes **0.45** seconds while the scheduling time using the *meta-heuristic strategies* takes **2.49** seconds. We have also performed multi-drone scheduling by using the *meta-heuristic strategies*. We considered 5-10 power pylons on Fyn (Denmark). Figure 3 illustrates the results.

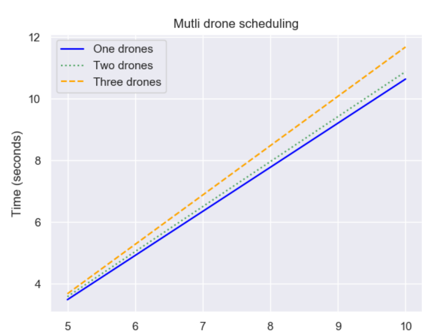


Figure 3. Multi-drone scheduling in Fyn (Denmark)

As can be observed in this figure, there is a linear relationship between the time and number of the power pylons. By increasing the number of power pylons calculation, time is also increased to a similar degree. However, there is a slight difference of scheduling times to be noted for different numbers of drones. We can, thus, conclude that the calculation time increases linearly with the number of power pylons. The reason is, unsurprisingly, that data matrix for solving the scheduling problem for 10 power pylons has more rows and columns compared with the data matrix with 5 power pylons.

IV. Data Transfer Framework

In this section, we demonstrate a framework for transferring large data such as images from the drone to the cloud and vice versa. We have also analyzed the communication delay for transferring data of various sizes between the cloud and the drones. Figure 4 illustrates the overall drone-cloud based framework corresponding to Figure 1.

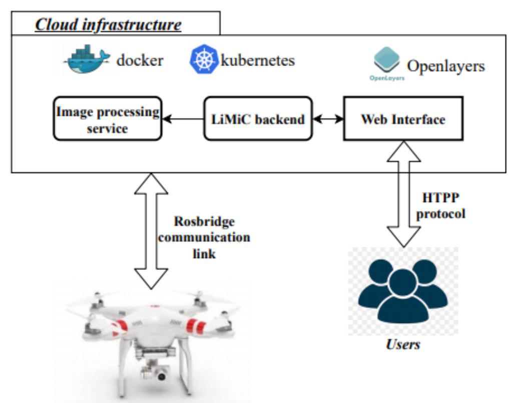
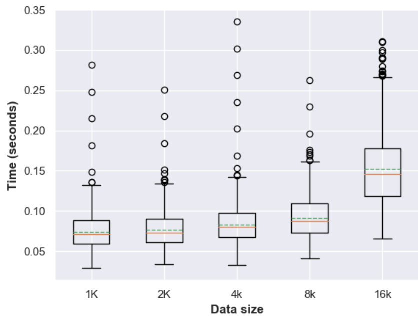


Figure 4. Overall system communication framework

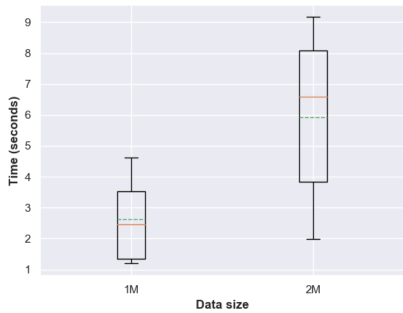
As can be seen from this figure, we have used various technologies such as Docker [7] for containerizing applications, Kubernetes [7] for scaling and managing container applications, and OpenLayers [4] for designing the web interface. We have also implemented different services such as an image processing service, LiMiC, and the web interface. The image processing service is designed for applying machine learning algorithms on the images. It is implemented as a Docker container. LiMiC is designed as a backend service for solving optimal routing and scheduling problems. The web interface is implemented as a frontend service for controlling and monitoring drone flight.

The communication link between the cloud server and the drones is provided by *Rosbridge*, while the communication link between the cloud server and users uses HTTP. In this regard, as an example, the user sends the HTTP GET request to the server to get the drone's battery status. In this paper, we have measured the *Rosbridge* communication delay for various data sizes such as images. Figure 5 demonstrates the results.

 ISBN: 978-1-61208-893-8



(a) Small data size



(b) Large data size

Figure 5. Data transferring delay time

As clearly can be seen in Figure 5(a), the average time for transferring data in the range of 1KByte to 8Kbyte range from 0.07 to 0.09 seconds. Therefore, there is only a slightly and statistically likely insignificant difference for transferring data in this ranges. However, the time for transferring 16KByte data is close to double the time for transferring 8KByte.

Figure 5(b) illustrates the data transfer delay for large data sizes, particularly 1MByte and 2MByte. As can be seen in this figure, there is a significant increase in the delay for transferring 1MByte data compared 2Mbyte data from **2.5** seconds to **6** seconds on average. Generally, we can conclude that the communication delay time over the *Rosbridge* protocol for small data sizes is within the range of **0.05** seconds to **0.10** seconds, however, for large data size, it is increased to more than double from **2.5** seconds to **6** seconds on average. We have not been able to fully explain this statistically significant non-linear increase in delay time. However, wireless network communication performance can also be considered as an important factor for this non-linear significant increase in this data range.

V. Conclusion

This paper proposed an algorithm for optimal scheduling and a communication link between the drones and the cloud server for IoD applications of linear infrastructure inspection. *Rosbridge* is designed as a communication link between the cloud server and the ROS. We analyzed the *Rosbridge* communication delay time by measuring the transfer time and delays for different data sizes.

The cloud server has been designed based on containerizing applications by using Docker and includes LiMiC. The LiMiC has been extended for solving the optimal routing and scheduling problem. We have implemented an optimal scheduling algorithm with the extended OR-Tools algorithm as a TSP solver for inspection drones along linear infrastructures. We have investigated two general approaches for scheduling such as the *first strategies* and the *meta-heuristic strategies*. We have also analyzed multi-drones scheduling times with *meta-heuristic strategies*.

Future work could be to implement Data Distribution Service (DDS) in ROS2 for providing the communication link between the cloud server and the drones. Furthermore, we could consider using secure channel such as HTTPS instead of simple HTTP.

Acknowledgement

The research leading to these results has received funding from the Innovation Fund Denmark Grand Solutions grant 8057-00038A Drones4Energy project.

References

[1] F. Li, S. Zlatanova, M. Koopman, X. Bai, and A. Diakité, “Universal path planning for an indoor drone,” *Automation in Construction*, vol. 95, pp. 275 – 283, 2018.

[2] E. Es Yurek and H. C. Ozmutlu, “A decomposition-based iterative optimization algorithm for traveling salesman problem with drone,” *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 249 – 262, 2018.

[3] Energy world, “Utilities in europe to use long-distance drones to inspect transmission liness,” Available: https://energy.economictimes.indiatimes.com/news/power/utilities-in-europe-to-use-long-distance-drones-to-inspect-transmission-lines/65007676?redirect=1, 10 2021.

[4] G. Mehrooz and P. Schneider-Kamp, “Optimal path planning for drone inspections of linear infrastructures.” in *GISTAM*, 2020, pp. 326–336.

[5] R. Montella, M. Ruggieri, and S. Kosta, “A fast, secure, reliable, and resilient data transfer framework for pervasive iot applications,” in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 710–715.

[6] J. H. Sejr and A. Schneider-Kamp, “Explainable outlier detection: What, for whom and why?” *Machine Learning with Applications*, 2021.

[7] G. Mehrooz and P. Schneider-Kamp, “Web application for planning, monitoring, and controlling autonomous inspection drones,” in *Mediterranean Conference on Embedded Computing (MECO)*, 2021, pp. 1–6.

[8] Google OR-Tools, “Traveling sale man problem,” Available: https:/developers.google.com/optimization/routing/tsp.

[9] B. Mah, “An empirical model of http network traffic,” in *Proceedings of INFOCOM ’97*, 1997, pp. 592–600.

[10] S. Mahmoud, N. Mohamed, and J. Al-Jaroodi, “Integrating uavs into the cloud using the concept of the web of things,” *Journal of Robotics*, vol. 2015, June 2015.

[11] C. Crick, G. Jay, S. Osentoski, and O. C. Jenkins, “Ros and rosbridge: Roboticists out of the loop,” in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012, pp. 493–494.

[12] S. Sarkar, M. W. Totaro, and K. Elgazzar, “Leveraging the cloud to achieve near real-time processing for drone-generated data,” in *2019 IEEE Women in Engineering (WIE) Forum USA East*, 2019, pp. 1–6.

[13] R. Rasmussen, “Tsp in spreadsheets—a fast and flexible tool,” *Omega*, no. 1, pp. 51 – 63, 2011.

[14] C. Prins, “A simple and effective evolutionary algorithm for the vehicle routing problem,” *Computers and Operations Research*, vol. 31, no. 12, pp. 1985–2002, 2004.

[15] M. Torabbeigi, G. Lim, and S. J. Kim, “Drone delivery schedule optimization considering the reliability of drones,” in *International Conference on Unmanned Aircraft Systems*, 06 2018, pp. 1048–1053.

[16] J. Xia, K. Wang, and S. Wang, “Drone scheduling to monitor vessels in emission control areas,” *Transportation Research Part B: Methodological*, vol. 119, pp. 174–196, 2019.