Delft University of Technology
Faculty of Aerospace Engineering

Optimization of visiting landmarks using rechargeable drones and charging nodes

Operations Optimization
AE4441 - 16

N. Pauly - 4550447
M. Sarkar - 5219507
M. D'heer - 4557476

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Rainforests cover less than 2% of the earth's surface, but contain over 50% of all the species.[8] It is very important to conserve and protect these areas. Due to the extreme environment, monitoring areas in the rainforest is very challenging. Besides the humid and warm environment, the dense forest makes it impossible to use any sort of road vehicle. Helicopters are very expensive and not efficient. This makes Unmanned Aerial Vehicles (UAV's) the only viable option. However, the dense forest allows for only a few landing spots and battery charge is not infinite. It is therefore incredibly important to optimize the flight path and the location of the charging stations. The aim of this report is to provide an overview of a model which optimizes the flight path and location of charging points to visit various landmarks which represent the habitat of endangered species.

This model is an expansion of the classic Vehicle Routing Problem (VRP). The problem analyses here is the scanning of an area as cost and time effective as possible. To achieve this, the objective is set to visit different landmarks as fast as possible while also taking the shortest path and thus lowering power consumption. The model along with the constraints & parameters are thoroughly described in chapter 2. In chapter 3 the model is implemented to find out the possible results and the objective. Later on a sensitivity analysis is done in chapter 4 by altering the different parameters. Also this serves as the final analysis on how each parameter effects the whole model and how they can effect the outcome. The main sensitivity parameters are chosen to be the time and charge it takes to scan a landmark, the minimum charge required at all times and the amount of drones used. The final conclusion and recommendations can be found in chapter 5.

Chapter 2

Model Description

The model is be described as a VRP model where the objective is to scan landmarks with the as power and time efficient as possible. The objective here is to find the most optimum path in order to reduce the amount of charge consumed. To analyse the model and get an output the six basic V.R.P constraints are used [10], adapting them to the problem statement:

1. Each node has to be visited at least once.
2. The drone must leave the depot.
3. The drone must return to the depot.
4. If a drone arrives at a node, it must also leave.
5. Time at a node has to be equal or greater than the time in the previous node.
6. The payload(charge in this case) demand should be met.

The model description mainly focuses on the assumptions that are made while constructing the model, along with determination of the objection functions and the constraints that are used in the model. It also gives a brief overview of the model architecture alongside the determination the model parameters and the drones and related costs to the model.

2.1 Assumptions

To make the model simple and linear in nature some basic assumptions are considered. These assumptions are only made to make calculations easier. The various parameters of the drone are made linear in nature with either time or distance. The geometry of the path and turns are also fixed to reduce the complexity of the model. All the assumptions that are taken into account are described below:

1. The drone flies in a straight line.
2. The velocity and the altitude remain constant during the whole flight.
3. The battery power consumption is linear with time.
4. The drone has no inertial while turning & thus can make 90° turns.
5. Drones will not fail during the flight.
6. Drones know what path to fly.

2.2 Model Outline

The costs associated for scanning are as follows:

- C_1 cost per unit distance of scanning.
- C_2 cost per unit percentage of charge for scanning.
- C_3 cost per unit time for scanning.

The variables used to make are model are:

- x_{ij} resources used from node i to j while scanning.
- d_{ij} distance between node i to j.
- t_{ij} time taken to travel from node i to j while scanning a unit distance.
- t_i time taken to reach location i.
- t_j time taken to reach location j.
- q_{ij} charge used by the drone while scanning form node i to j in percent.
- q_i charge remaining of the drone at node i in percent.
- q_j charge remaining of the drone at node j in percent.

The parameters that are used to run the model area as follows:

- Q total charge capacity of the drone. Value set at 100%.
- p_d charge consumed per unit distance.
- p_s charge consumed for scanning per location.
- t_d time needed for travelling per unit distance.
- t_s time needed for scanning per location.
- t_c time for charging per unit percentage of battery.
- K number of drones.
- S_i location of the charging station where i is the node at which the charging station is located.

The objective is to find the optimal path to scan so that every coordinate is covered. The objective function is:

Minimize:

$$Z = C_1 \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=1, j\neq i}^{j=n+1} x_{ij}^k d_{ij}^k + C_2 \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=1, j\neq i}^{j=n+1} x_{ij}^k q_{ij}^k + C_3 \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=1, j\neq i}^{j=n+1} x_{ij}^k t_{ij}^k d_{ij}^k \tag{2.1}$$

Subject to :

$$\sum_{i=0}^{n} \sum_{j=1, j\neq i}^{n+1} \sum_{k=1}^{K} x_{ij}^k = 1 \quad \forall i, \quad \forall j, j \neq i, \quad \forall k. \tag{2.2}$$

$$\sum_{j=1}^{n+1} x_{0j}^k = 1 \quad \forall j \quad \forall k \tag{2.3}$$

$$\sum_{j=0}^{n} x_{j,n+1}^k = 1 \quad k = 1, ..., K \quad and\ \forall j \tag{2.4}$$

$$\sum_{k=1}^{K} \sum_{j=0, j\neq i}^{n} x_{ji}^k = 1 \quad i = 1, ..., n\ and\ \ \forall j\ and\ \forall k \tag{2.5}$$

$$\sum_{j=0, j\neq i}^{n} x_{ji}^k = \sum_{j=1, j\neq i}^{n+1} x_{ij}^k \quad i = 1, ..., n\ \ and\ k = 1, ..., K \tag{2.6}$$

$$q_i \geqslant 0 \qquad \forall i \tag{2.7}$$

$$q_i > q_j \quad \forall i, \quad \forall j \tag{2.8}$$

$$q_i^k - q_j^k - p_d d_{ij} x_{ij}^k - p_s + (1 - x_{ij}^k)M \geqslant 0 \quad \forall i\ \forall k\ \forall j, j \neq S_i \tag{2.9}$$

$$q_i^k - q_j^k - p_d d_{ij} x_{ij}^k + (1 - x_{ij}^k)M \geqslant 0 \quad \forall k \quad \forall i \quad \forall j \tag{2.10}$$

$$t_i^k - t_j^k - t_d d_{ij} x_{ij}^k + t_s + (1 - x_{ij}^k)M \leqslant 0 \quad \forall i \quad \forall j \quad \forall k \tag{2.11}$$

$$t_i^k - t_j^k - t_d d_{ij} x_{ij}^k + t_c(100 - q_{i,j}) + (1 - x_{ij}^k)M \leqslant 0 \quad \forall i \quad \forall j \quad \forall k \tag{2.12}$$

The equation 2.2 says that each node has to be visited by at least once while scanning. Equation 2.3 makes sure that the drones leave the depot and equation 2.4 that the drones return back to the depot after the scanning is done. Equation 2.5 ensures that only one drone visits a node. Equation 2.6 is to make sure that if a drone enters a node it must also leave. Equation 2.7 makes sure that charge at each node should always be greater than 0. Equation 2.8 is an extension of equation 2.7 which says the percentage of charge left at the location i has to be greater than the percentage of charge left at location j. Equation 2.9 and equation 2.10 are extensions of equation 2.7 which ensures that the optimal path is taken by the drones to reach each scanning location. On top of that equation 2.9 also considers the charge needed for scanning. Equation 2.11 considers the scanning time for each location and equation 2.12 takes account the charging time of the drones. Equations 2.9 - 2.12 are used for the sub-tour elimination.

2.3 Model Architecture

In Figure 2.1 can the flowchart of the model be observed. There is chosen for object oriented programming in python as it provides an excellent framework to efficiently create multiple objects using different variables. This greatly simplifies the additional code needed to perform a sensitivity analysis. The commercial solver Gurobi is selected because of its easy implementation in python. In this way, everything is automated and there is no need to use any external programs.

The model takes two inputs: the arguments provided to the object and an excel sheet generated by google earth containing the coordinates of the landmarks. If provided by these inputs, the only thing left to do is run the model and two outputs are generated: a plot of the path and an excel sheet containing the output values. The abstraction of the object oriented programming style allows the model to be easily used by non-technical people while still being understandable and expandable to engineers.

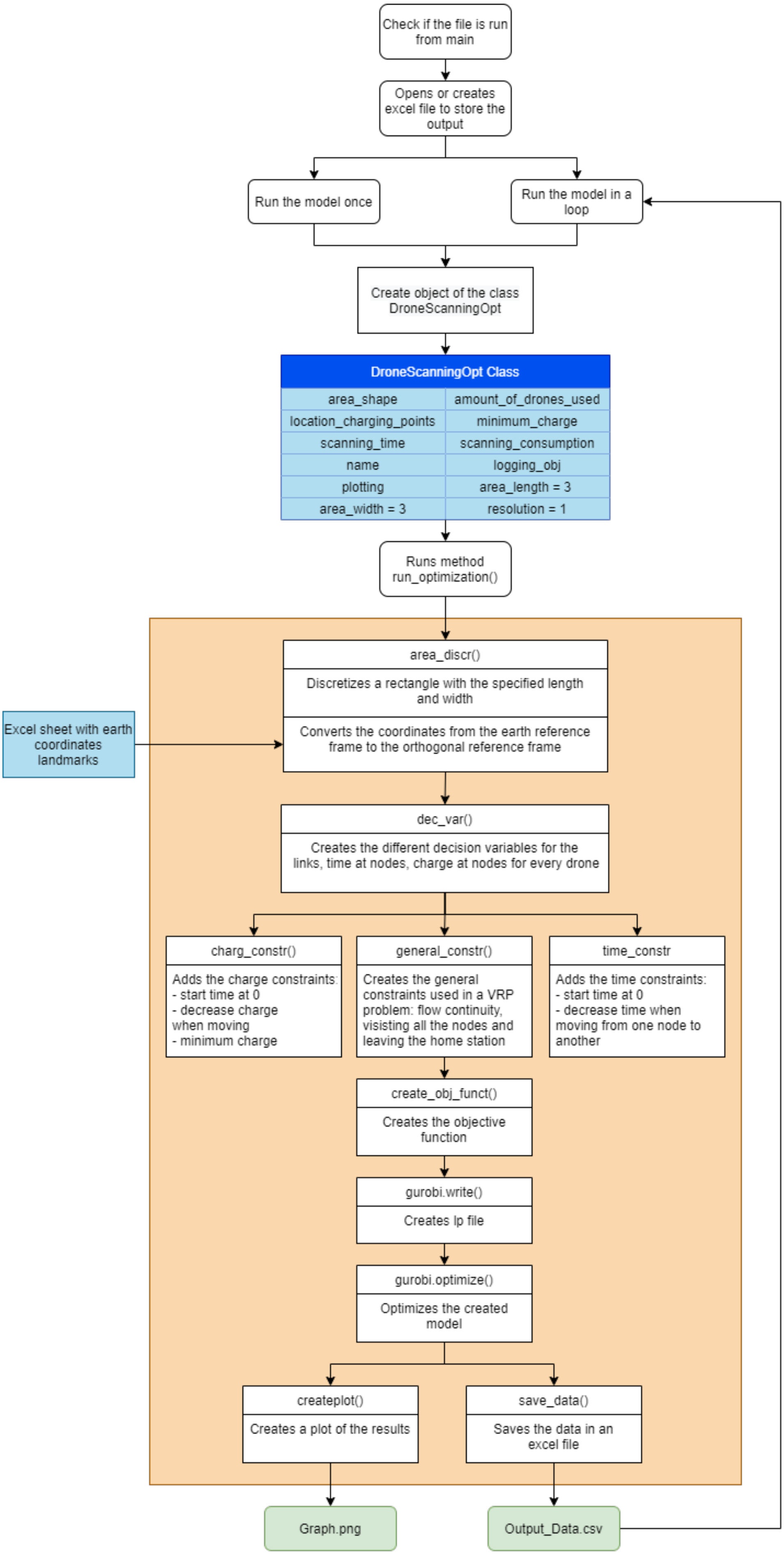

Figure 2.1: Model Architecture

2.4 Parameter Determination

To make the outcome of the optimization as realistic as possible, the aim of this section is to provide a motivation for the parameters used in the optimization. First are the objective function coefficients determined. The second step is determining which drone will be used in the optimization. Subsequently the parameters of that drone are used to calculate realistic values for the charge time, battery consumption, cruise speed and scanning speed.

2.4.1 Cost

In the objective function are three constants C_1, C_2 and C_3 present. These constants give a weight to the three different parts of the objective function: distance, charge level and time respectively. The aim is to minimize the distance flown to achieve the most efficient flight path. The second objective is to finish the scanning operation as fast as possible.

The value of C_1 is the operation cost of the drone per unit distance. C_3 is the operational cost incurred for per unit time of operation. The values of for both the cost constants can be referred from the data in figure 2.2.

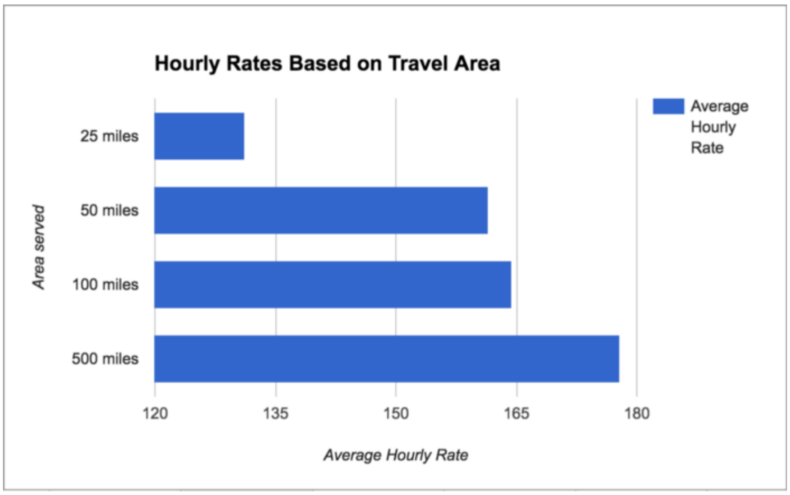

Figure 2.2: Hourly rates based on travel area [6]

Time min	Distance km	Hourly rate $
60	160.934	≈ 160

Table 2.1: Cost of operations

Since the model used here has a rough distance overage of 150 km, the data for the hourly rate of area served within 100 $miles$ can be used from figure 2.2. The approximate values are shown in the table 2.1 and the vales of C_1 and C_3 can be calculated from them easily.

$$\begin{aligned} C_1 &= \frac{160}{160.934} \approx 1 \ \$/km \\ C_3 &= \frac{160}{60} \approx 2.67 \ \$/min \end{aligned} \tag{2.13}$$

The charge is implemented in the objective function in order to achieve the correct values at the output. In Table 2.2 an optimization result can be observed if the charge is not included in the objective function. It is clear these values do not represent reality knowing each drone start with a battery level of 100%. The model only needs to meet the requirement of having a charge which is at least a certain amount lower than the charge at the previous node. It does not matter if this amount is 10% or 80%, both will meet the requirement. In order to get the 'real' values the charge is added to the objective function with a negative coefficient. This causes the solver to maximize the level of charge at each node and providing the correct values. An example of this can be found in Table 2.2. The value of C_2 must be negative, but also small. A value too large causes the model to choose less efficient paths in terms of distance and time. C_2 is therefore chosen to be -0.001.

C_1	C_2	C_3	**Obj. Value**	Computation time
1	0	2.67	2687.7	1164.04 s
1	-0.001	2.67	2689.9	946.07 s

Table 2.2: Variation of C_2 for 1 drone

2.4.2 Drone Selection

In order to achieve realistic results, selecting the appropriate drone for this mission is very important. To provide an overview of the different available drones, Table 2.3 is created. The trade-off criteria are chosen to be range, vertical take-off, efficiency and cost. Range is very important as there is aimed to cover at least an area of 100km. Charging points will be located at open area's in the forest, this means that only a small area is available and vertical take-off is required. To decrease cost and environmental impact is a high efficiency desired. The operational and unit cost is also important as there is aimed to lower the costs as much as possible. [9]

Types of UAV's	***Range [km]***	***Vertical Take-off***	***Efficiency***	***Cost***
Multi Rotor Drones	50	Yes	Very Low	Low
Fixed Wing Drones	161	No	High	Low
Single Rotor Helicopter	180	Yes	Low	High
Fixed Wing Hybrid VTOL	150	Yes	High	Medium

Table 2.3: Trade-off table [9] [5] [1] [2] [7] [3]

From the trade-off table can clearly be seen that the multi rotor and the fixed wing drone do not meet the requirements of range and vertical take-off respectively. There can be observed that the fixed wing hybrid VTOL is better in terms of efficiency and cost. There is chosen for this concept. Designing a custom UAV for this project would be beyond the scope. Therefore is chosen to use an existing drone. The Avy Aera drone meets all the requirements and is thus chosen as reference drone throughout the rest of the report. [4] The technical specifications can be found in Table 2.4.

Avy Aera	**Symbol**	**Values**	**Units**
Flight time	t_tot	60	min
Cruise speed	v_cr	70	km/h
Range	R	100	km
Battery Capacity	Q_max	22.000	mAh

Table 2.4: Technical specifications of the Avy Aera

2.4.3 Charge time

Using the battery capacity from Table 2.4 and the average value of 10A, using Equation 2.14, there can be found that the time to charge from 0% to 100% will then take 132 minutes. This means it takes 1.32 min to charge 1%. However, it takes time to land and take-off from the charging point. The UAV needs to land very precisely in a harsh and humid environment. There is assumed that before the UAV starts charging, an automatic system check is done and all the data is transferred to the charging station. The results in an extra delay of approximately 8 min which is added to the model.

$$t_{0-100} = \frac{Q_{max}}{q_{const}} = 2.2h = 132min \tag{2.14}$$

2.4.4 Battery and time consumption

The battery consumption during cruise is calculated using the flight time and the cruise speed. Using Equation 2.15, it is found that the range at cruise velocity is equal to 70km, resulting in a battery consumption of 1.43% per kilometer during cruise.

$$R_{cruise} = \frac{v_{cr}}{t_{tot}} = 70km \tag{2.15}$$

To find the time it takes to cover 1km during cruise, Equation 2.16 is used.

$$t_{const} = \frac{60}{70} = 0.857min/km \tag{2.16}$$

2.4.5 Scanning Specifications

There is assumed that scanning a landmark will take approximately 15 min. Taking into account that a landmark can span a few square kilometers. The velocity is also decreased to 30 km/h to allow for more accurate data and allow measurements from different sensors. During the scanning operation, the low velocity demands assistance from to top propellers to generate enough upward force to maintain altitude. This extra battery consumption combined with the battery consumption of all the sensors and cameras results in an average battery consumption of 8% each visited landmark.

2.4.6 Overview

To run the optimisation model certain parameters are used which have been calculated in the preceding sections. In this section all the parameters are summarized below in the table 2.5 :

Parameters	Values	Units
Q	100	%
p_d	1.43	%
p_s	8	%
t_d	0.857	min/km
t_s	15	min
t_c	1.32	min/%
C_1	3.12	$/km
C_2	-0.001	$
C_3	2.67	$/min

Table 2.5: Values of the parameters used in the model

Chapter 3

Results

This chapter will discuss the nature and expected results. Followed by the verification and validation of the model and finished by the results of the optimization. The output of the model is provided in two different forms. A .csv file with the results of all calculated values such as: objective value, total distance, calculation time, etc and a plot showing the most optimized path.

3.1 Discussion

As discussed in section 2.3, the model requires two inputs. The input arguments for the object and the .csv file with the coordinates of the landmarks. This file is constructed using google earth, the coordinates can be found in Figure 3.1. The output of the model is provided in two different forms. A .csv file with the results of all calculated values such as: objective value, total distance, calculation time, etc and a plot showing the flight path.

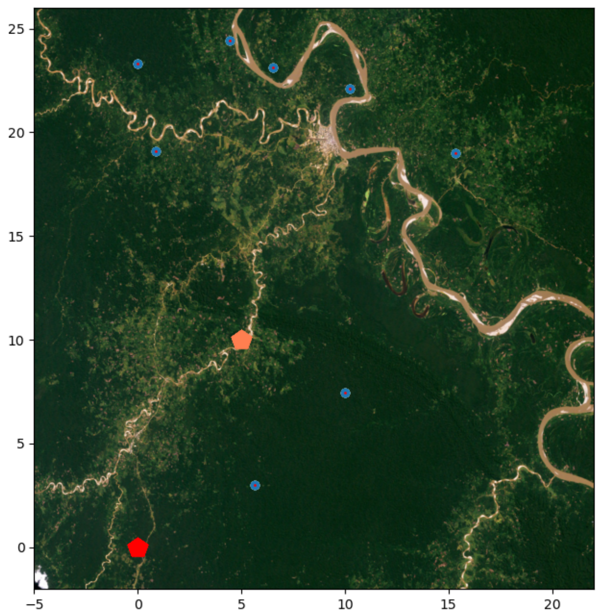

Figure 3.1: Google earth nodes

There is chosen to run the model using 8 nodes which can be seen Figure 3.1. This is a trade-off between size of the model and runtime. Increasing the amount of nodes results in an exponential increase in runtime. The coordinates are chosen arbitrary in the amazon forest. The blue dots represent the chosen areas to scan. The red dot is the start point (home base) of the drones. Assumed to represent a small village. The orange dot is a charging point placed in an open area in the forest.

3.2 Verification of the model

To verify the simulation. Two verification procedures have been used throughout the development of the simulation. Unit test are performed to test the mathematical calculations of the program and a reality check was done after the run of each program. In addition to this, the strategy of developing the program. That is, to build it up and check along the way if it is correct. The second verification procedure is optimizing a simplified the model by hand and comparing the results to the output of the optimization.

3.2.1 Unit test

To perform unit tests, the nodes were positioned in a grid for easier calculation. After each run of the program, the cost value of the drone paths was compared to the cost function calculated by hand. Before adding another complication to the program, i.e. adding drones or charging stations. These tests had to performed successfully.

Link function

To accurately determine the distance between each node the following definition was used which is just the pythagorean theorem:

```
    # Calculates the distance between two nodes
    def link(self, node_i, node_j):
        dist = np.sqrt((self.area_coord[node_j][1] -
        self.area_coord[node_i][1]**2 + ((self.area_coord[node_j][2])
            - self.area_coord[node_i][2]) ** 2)
        return dist
```

To check if this is correct, the total distance of the result was calculated by hand.

Objective function

The following code snippet shows the definition of the objective function with the definition of the constants. This can also be verified by hand

```
    # Creates the objection function and adds it to the gurobi model.
    def create_obj_funct(self):
        c1 = 1
        c2 = -0.001
        c3 = 2.67
        #  Each part in the objective function has a different weight
        self.obj_function = c1 * self.obj_dist + c2 * self.obj_charge +
            c3 * self.obj_time
```

```
        self.gm.setObjective(self.obj_function, gp.GRB.MINIMIZE)
```

Plotting

Another result that needs to be verified is the plotting, this can easily be checked by seeing which links have been activated.

```
        Variable             X
    -------------------------
    Link0-9_Drone 0          1
    Link1-0_Drone 0          1
    Link2-1_Drone 0          1
    Link3-2_Drone 0          1
    Link4-3_Drone 0          1
    Link5-4_Drone 0          1
    Link6-5_Drone 0          1
    Link7-6_Drone 0          1
    Link8-7_Drone 0          1
    Link9-8_Drone 0          1
```

For each X=1 the link between nodes has been activated. This can then be checked with the graph whether it is correct or not.

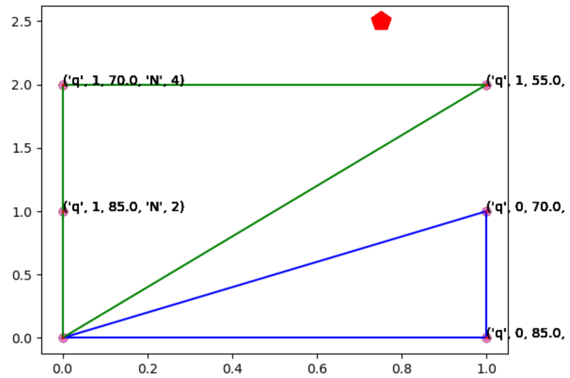

Figure 3.2: Verification graph

3.2.2 Reality check

Early on in the development of the simulation. Plotting the result has been given more attention. This is so that after each run, the path of the drone was plotted and it could clearly be seen whether it was taking the most optimized path.

Before the results of the simulation are determined. It is helpful to predict the outcome of the more simpler simulations.

Considering the biggest contribution to the objective function is the time. The time and thus distance for each drone shall be minimised where possible. Therefore the following things can be expected:

- Drones shall not visit nodes more then once.

- Drones will either visit the most nearby node or a charging point.
- Charging points will only be visited when charge is nearly depleted.
- Multiple drones will likely cover an almost equal distance to minimise total time.

Should any of these statements not comply with the provided result. There would be an error within the program.

3.2.3 Verification of cost

To verify that the model gives the correct value of cost function, a simplified rectangular area is chosen to get a test output. The value of the objective function is then compared to the theoretically calculated value of the cost function using the formula 2.1. The parameters used in the model remain the same. The output of time and charge are taken from the model itself and the parameters of the drone in 2.4.6 are used to calculated the final cost. If the error is very minimal or zero, then it can be verified that the model works as per the definition.

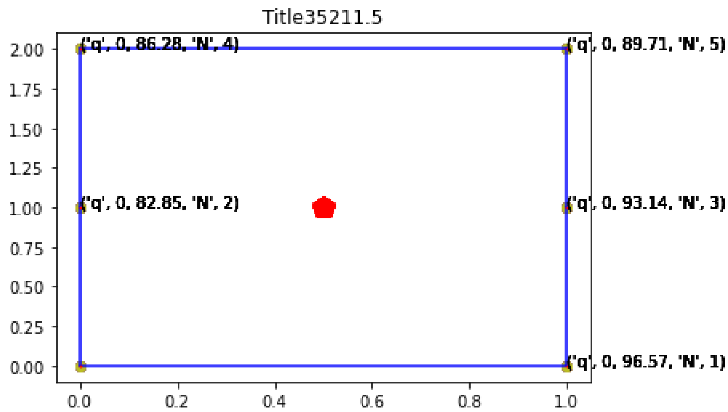

Figure 3.3: Test result of a 1x2 area

3.3 Final Result

Below are three figures of solutions from the model. They are for one, two and three drones. The small blue points are the areas which need to be scanned while the red shapes are the charging points for the drones to charge. The two axis show the horizontal and vertical distance in kilometers. The coloured lines are the path which the drones take, each colour refers to a different drone. Along with the lines, each point shows the charge the drone has at that point after scanning that point and which drone visits it.

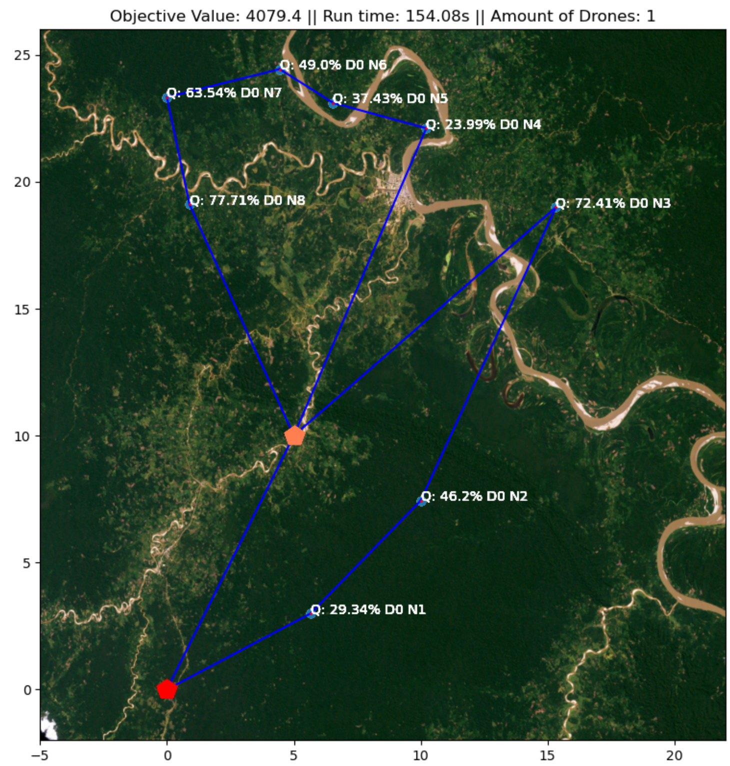

Figure 3.4: Path optimisation for a single drone operation

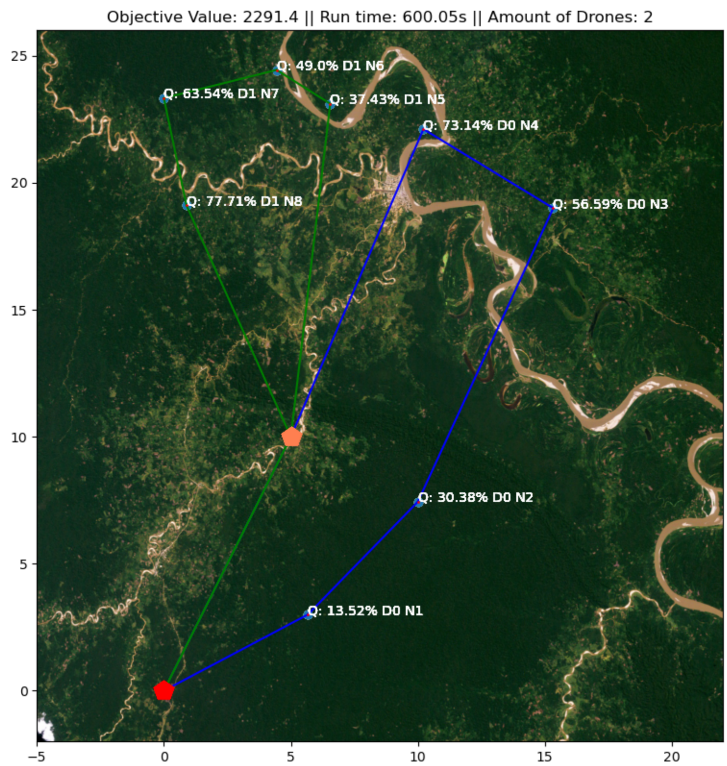

Figure 3.5: Path optimisation for two drones operation

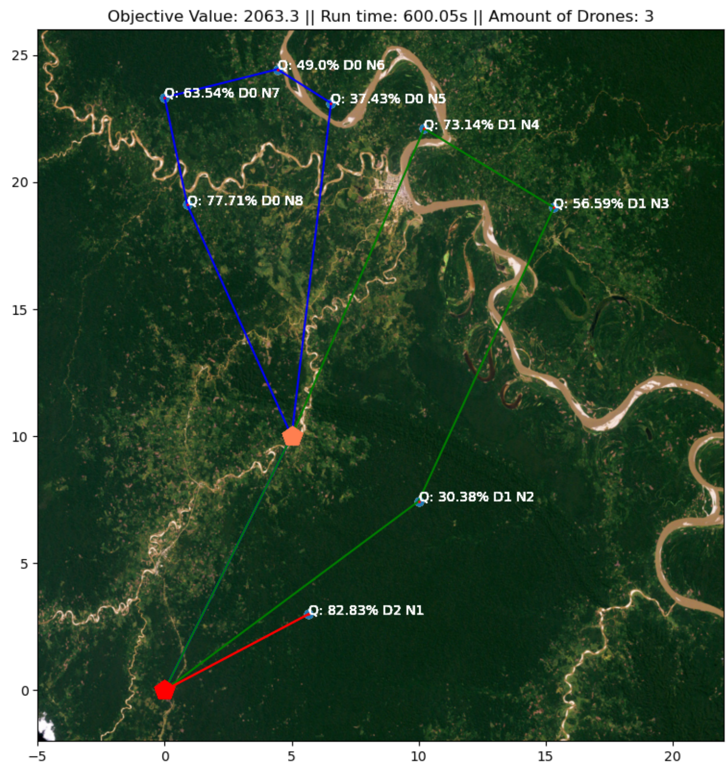

Figure 3.6: Path optimisation for three drones operation

Chapter 4

Sensitivity Analysis

Sensitivity analysis helps to understand the output of the model. The cost is effected when the input variables and the parameters are altered. In this scenario of drone scanning, the goal is to follow the most optimal path which in turn conserves charge and also takes less time. The model uses different parameters which all impact the outcome. Most of these parameters are calculated and assumed to be close to reality, these calculations can be found in chapter 2. However, not all the parameters could be calculated and some are estimated. The goal of this chapter is to investigate the change in the outcome if these estimated parameters are altered. The following parameters are chosen to be changed:

- Minimum charge required at all time.
- Scanning charge consumption.
- Position of the charging station.

4.1 Model Capacity

In the ideal world would every simulation be run until a gap of zero is reached. Of course this is not a feasible approach for every problem. It could be that the optimization needs be available within a specific time frame such as for airline planning. It could also be that the runtime simply becomes too large. VRP problems are notorious for their long runtime and exponential increase in the runtime with the linear addition of variables. It is therefore identified that the scaling behaviour of this model is important to be discussed. The following section will describe the relation between the amount of links (size) of the problem with relation to the gap and the variation in computation time with different runtimes.

4.1.1 Amount of links vs Gap

There is chosen to consider 8 different nodes for this report. Including the charge and home base point, the results in a total of 10 nodes. The variables of time and charge for every node are fixed and cannot be changed (without changing the amount of nodes). The only thing which can be done to decrease the size of the model is reducing the amount of available links. In order to allow for reusing links, each link has two different decision variables. The problem can be reduced by not facilitating reusing links. Another way to reduce the problem is eliminating links. This is implemented in the creation of the model. A function calculates the length of the link and compares it to a set threshold. The higher the threshold, the more links are added which allows for more paths but also greatly increases the problem. However, some paths could easily be removed which would clearly never be taken in an

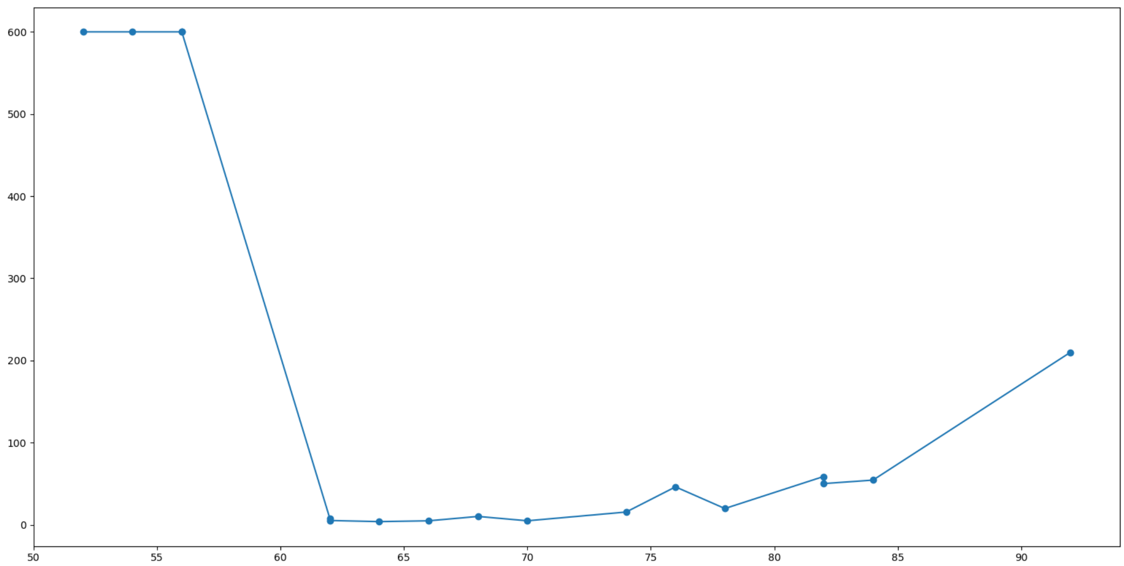

Figure 4.1: Graph of amount of links (x-axis) vs runtime (y-axis)

optimal solution, such as the longest links.

Figure 4.1 shows the amount of links versus the runtime. Note that a time of 600 seconds corresponds to an unfeasible model. It is very dependent on the application of the model if removing links is feasible.

4.1.2 Computation time vs Objective value

In Figure 4.2 can the relation be found between the runtime and the objective value. There can be observed that the graph is not as smooth as one would expect. The results show more of a step behaviour due to the nature of how the graph is created. The graph is created by letting the optimization run for a certain amount of time and saving the reached objective value. This means in order to create Figure 4.2, 100 optimizations were performed with a duration ranging from 0 to a 100 seconds. At 8 seconds, 42 seconds and 46 seconds can a jump be observed in runtime. This is due to the fact the simulation is ran in the background of a computer where there are also other programs that require inconsistent computational power. A computer is not an embedded system and can therefore not guarantee consistent results within a certain time frame.

Meaningful insight can be gained from Figure 4.2. In the case of time constraints, it could be sufficient to run the optimization for 16 seconds to already be close to the optimal solution.

There is an analysis done on how the gap changes with respect to runtime. This can be found in Figure 4.3. Note that the function is smooth and some outliers are present. This is due to the nature of how the graph is created, also as mentioned before, a laptop is not an embedded device and is sensitive to other processes as well.

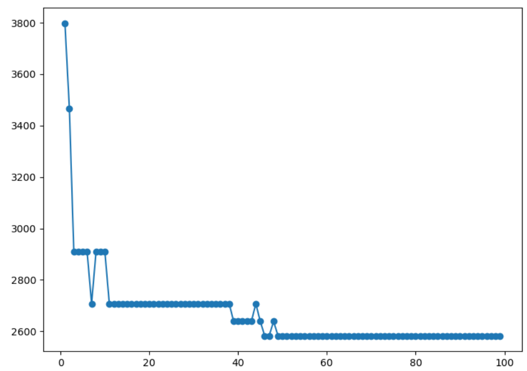

Figure 4.2: Graph runtime in seconds (x-axis) vs objective value (y-axis)

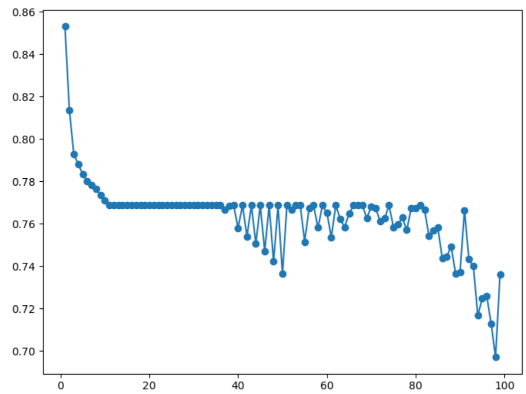

Figure 4.3: Graph runtime in seconds (x-axis) vs gap in percentage (y-axis)

4.2 Parameter

4.2.1 Minimum Charge

The minimum charge parameter is a requirement that states how much charge percentage a drone must have before arriving back at the origin or at a charging point. The reason for this is that there needs to be enough contingency in case the efficiency of a drone drops (due to high wind speeds or other inconveniences). This percentage needs to be as high as possible whilst also limiting the negative effect it has on results of the simulation.

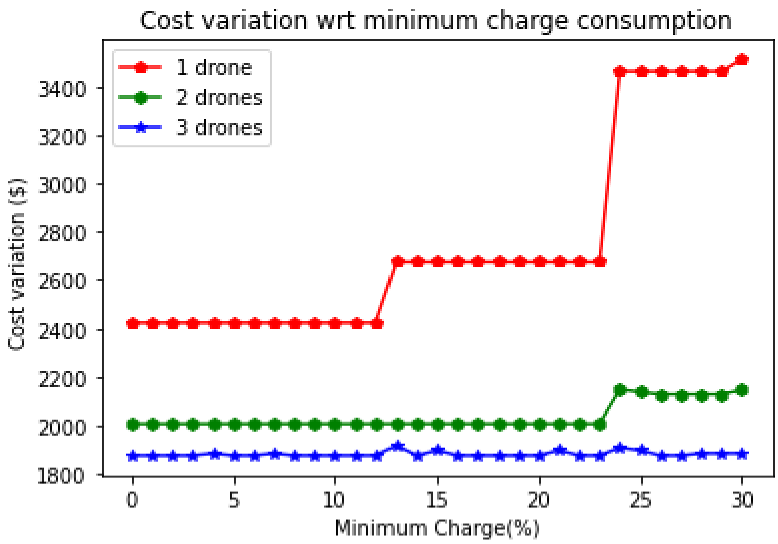

Figure 4.4: Cost variation wrt minimum charge level

An optimisation model is run to check the changes in the cost function when the minimum required charge is varied for each set of drones.The area used can be referred from Figure 3.1. The result of the simulation is shown in Figure 4.4. Since the area is too vast for a single drone to cover, it is expected that the operation charge of a single drone will be more than using 2 or 3 drones together. Again, a single drone has to make multiple trips back to the charging point to recharge its battery, thus also including the overall cost. For 2 drones or 3 drones operating together, the whole area is divided among the drones more or less equally, thus each drone has to cover a significant less area than if operating alone. This also enables them to make less trips to the charging point since a small area means that charge will be depleted slowly. This justifies the results from Figure 4.4. Also it has to noted that its generally not feasible to use multiple drones for small areas.

4.2.2 Scanning charge consumption

When scanning each location the drones use a certain percentage of charge which is same for every location. This charge takes up a significant portion in the cost value and also determines how fast the charge of the battery is depleted. Analytically, more the charge consumed for scanning, faster the charge will deplete and hence the drone has to come back to recharge its battery again, thus increasing the overall cost. To visualise how this parameter effects the overall cost, a simulation is run, setting the minimum charge required in the drone to 8 %. The results are shown in Figure 4.5.

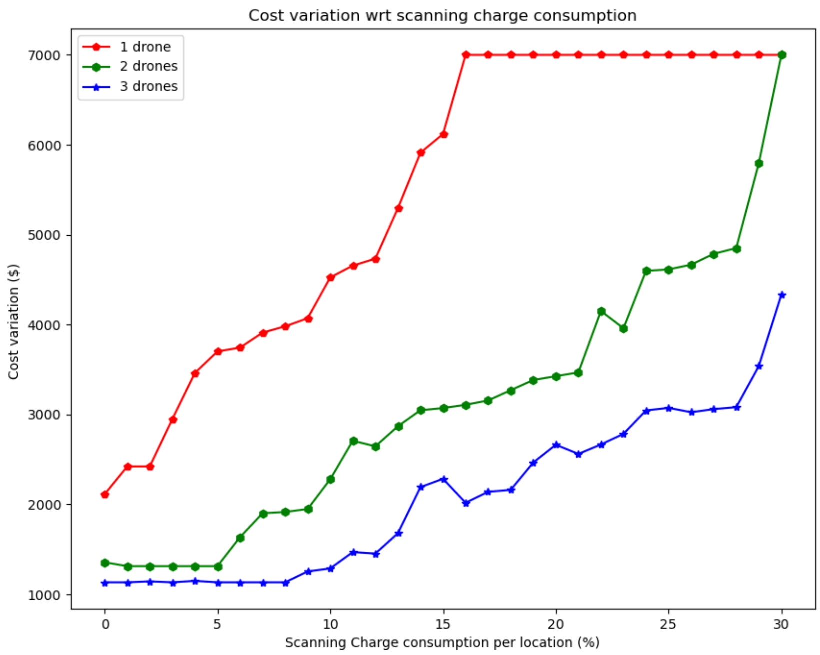

Figure 4.5: Cost variation wrt scanning charge consumption

It can see be seen that when 3 drones are used, the cost is the minimum, the analogy behind this is that each drone has to cover a less area thus reducing the overall cost. At some level of the scanning charge consumption, a drastic jump is the cost can be noticed, this is because of the fact that at those levels, the drone battery gets near to the minimum charge and the drones have to make multiple trips to the charging point to recharge their batteries. Overall an increasing trend is noticed. All from Figure 4.5 when the cost touches the mark of *7000 $*, it implies the model has failed. The model takes into account the 3 drones fails at higher values of charge consumption only. There can be observed that for higher scan times of the landmarks, it becomes more efficient to use more drones.

4.2.3 Charging Point Position

To estimate the best position of for the charging point, the simulation was iterated for every unit coordinate, for a total of $26 \cdot 26 = 676$ simulations. In order to keep the runtime within reasonable limits, the simulation is run for 100 seconds each, total of 19 hours, resulting in an average gap of 50.8 %. The results are shown in Figure 4.6

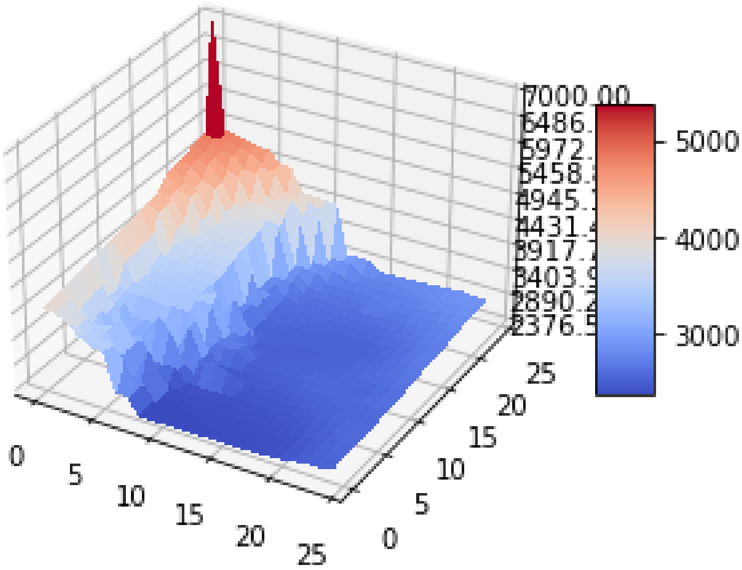

Figure 4.6: Cost variation with respect to a charging point location for a single drone

Where the vertical axis is the cost function. The red peak is where no solution is found. The most optimal position of the charging point is where the cost function is lowest, this lies at the point (15,17).

Chapter 5

Conclusions and recommendations

This report analyzes the option of monitoring rain forests. However, it can be applied to any similar problem related to trajectory optimization for a number of Unmanned Aerial Vehicles (UAVs or drones). The goal was to monitor different landmarks (habitat of endangered species) as efficiently as possible by optimizing the flight path and other parameters such as number of drones and position of recharging station.

The model comes from the classic Vehicle Routing Problem (VRP) already discussed in the course. To prepare the construction of the model, first the assumptions and outline of the model is discussed. From this, the model parameters are determined. The objective function of the model consists of the addition of 3 parts; 1 the total distance, 2. The charge at the end of flight and 3. The total time it takes to visit each node once. Each part has a constant (C1,C2 and C3 respectively) that were determined. After that a reference drone was chosen to be a Fixed Wing Hybrid VTOL for its better characteristics. The AVY drone is selected due to its high performance and connection to Delft University of Technology.

With the parameters decided, the simulation was run and a verification procedure consisting of unit test and reality checks was applied to verify the results. The model was also run for a simple grid which could be optimized by hand and verified with the outcome of the model.

To finally decide on the final parameters, such as number of drones and location of charging point and gain more inside in the characteristics of the model, a sensitivity analysis was performed. From this analysis can be seen that 2 drones was the optimal amount for the given area and drone selection. The best position for to place a charging point was demonstrated to be at (15,17).

This model shows very nice results and is created in a robust way. Improvement could be made to the code and the model could be extended to eliminate some of the assumptions. Also other solvers could be explored such as CPLEX to decrease the runtime.

Bibliography

[1] Affordable fixed-wing drone for commercial applications. *Unmanned System Technology*. Accessed: 2020-12-15.

[2] Airboxer unmanned helicopter system. https://www.higheye.com/product/. Accessed: 2020-12-15.

[3] Avy aera specsheet. :https://www.avy.eu/bvlos-vtol-drone-aircraftspecs.

[4] Drones for a better world. :https://mobility-logistics.messefrankfurt.com/worldwide/en/list/mobility-logistics-news/start-up-avy.html.

[5] Drones with long flight times. https://hse-uav.com/uav-industries-and-applications/drones-with-long-flight-times/. Accessed: 2020-12-15.

[6] Hourly rates for drone operations. https://medium.com/aerial-acuity/how-to-price-your-drone-mapping-services-b5fc31c86934.

[7] Hybrid vtol fixed wing uav for commercial applications. *Unmanned System Technology*. Accessed: 2020-12-15.

[8] Bradford, A. Facts about rainforests. *LIVESCIENCE* (7 2018).

[9] JOJO. Types of drones – explore the different models of uav's. *CIRCUITS TODAY* (2020).

[10] Rolling, P. Ae4441 operations optimization slides.