| For office use only | Team Control Number | For office use only |
|---|---|---|
| T1 ________________ | **1919009** | F1 ________________ |
| T2 ________________ | | F2 ________________ |
| T3 ________________ | Problem Chosen | F3 ________________ |
| T4 ________________ | **B** | F4 ________________ |

**2019**
**MCM/ICM**
**Summary Sheet**

# Modeling DroneGo System for Disaster


**Summary**

In 2017, Hurricane Maria has wreaked havoc in the Puerto Rico. People hope the drones, capable of medical delivery and video reconnaissance, contribute to the emergency relief.

To address this situation, our paper provides a detailed analysis of an aerial response system named "DroneGo." We design the DroneGo system to fully analyze different aspects, such as the number of the drones, the best location(s) to place the container(s), and the flight plan.

The DroneGo system consists of three main submodels, Container Loading Optimization Model, Drone Selection Model, and Drone Routing Selection Model.

We devise Container Loading Optimization Model to find optimal packing configuration for the cargo container and the cargo bay. We transform three-dimensional loading into two-dimensional loading through **space segmentation and mergence**. We get that placing 63 drones B, 2 drones H, 60 cargo bays 1, and several medical packages is the optimal configuration for the cargo container, reaching 97.00% fill rate.

We devise Drone Selection Model to determine the best type drone(s). We introduce **Energy Consumption Equation** to estimate the each drone's flight time with certain payload. According to the flight capacity we formulate to evaluate the performance of each drone, we only choose drone B for our drone fleet.

We devise Drone Routing Selection Model to acquire efficient working routes. We define **alternated road graph (ARG)** to discretize road network. We utilize the **revised K-means algorithm** and apply **Simulated Annealing algorithm** to every cluster. Results show that our drone fleet cover all its working range in three hours, even less.

Another section of our paper further discusses the tradeoffs to assess more road information. If we assign total 180 drones evenly to twelve airports, the coverage rate will rise to 90% from 41%.

In the end, we make sensitivity analysis, dissect pros and cons of our model, and present a memo of our work to the HELP, Inc. CEO.



**Keywords**: Space Segmentation and Mergence; Energy Consumption Equation; Alternated Road Graph (ARG); revised K-means algorithm; Simulated Annealing algorithm

# Modeling DroneGo System for Disaster

April 3, 2019

## Contents

# 1 Introduction

## 1.1 Problem Background

Puerto Rico, meaning "rich port" in Spanish, consists of the main island of Puerto Rico and various smaller islands [5]. However, with a long coastline and exposed to the Atlantic Ocean, Puerto Rico is vulnerable to natural disasters, especially hurricanes. In September of 2017, Hurricane Maria, which is regarded as the worst natural disaster on record, smashed directly into Puerto Rico. On arrival, the hurricane wreaked havoc on the island, damaging transportation road networks seriously and destroying most of the cellular communication network. A number of people lost their lives. Besides, hundreds of residents demanded badly for medical supplies and stayed isolated for several weeks because of the electrical power and cell services outages. What they only could do was to wait for rescue.

Apart from the official rescue teams, *Non-governmental organizations (NGOs)* volunteer to give their hands when natural disasters happen. Their work begins by assessing the extent of the damage and providing immediate emergency relief. For the first few weeks, NGOs continuously distribute emergency aid and relief to residents affected by the disaster. Over the following months, they help families stabilize by providing non-food items and livelihood support [9].

One of NGOs - HELP, Inc. - is trying to devise a disaster response system with the help of the drone fleet named "DroneGo." The overwhelming advantages of drones (Figure 1) to view the bigger picture of an issue from above and approach where ground traffic cannot reach dramatically cut down on the time, and this gives people more time to save lives. Therefore, application of drones could function well in disasters such as the hurricane in Puerto Rico.


Figure 1: Three examples of drones: a — fixed wing; b — helicopter; c — quadrotor [6]

## 1.2 Task at Hand

Our team is going to help HELP, Inc. design a DroneGo disaster response system to support the Puerto Rico hurricane disaster scenario. We plan to give the recommendation as follows.

- A packing model to calculate the exact number of the drone with certain types, the cargo bay, and each medical packages.
- A judging model to determine the best location(s) to place the container(s).
- A routing model to plan the drone's path that maximizes the work efficiency.

### 1.3 Our Work

To address this situation, we construct Cargo Loading Optimization Model employing recursive analysis to achieve optimal packing configuration for each cargo container and cargo bay. We define Flight Capacity for drones to quantitatively describe the performance of diverse types of drones. We build Drone Routing Selection Model based on revised K-Means Clustering and Simulated Annealing algorithm, paired with real-world data of road networks in Puerto Rico, to find optimal planning routes for medical package delivery and road reconnaissance.

In the paper, we state our basic assumptions and provide nomenclature of basic parameters in Section 2 and Section 3. Section 4, 5, and 6 contains detailed analysis of DroneGo System Model. In Section 7, we conduct sensitivity analysis, discuss strengths and weaknesses of our model, and provide the tradeoffs to better fit the general disaster conditions. At last, we draw conclusions in Section 8 and write a memo to provide the recommendations to HELP, Inc.

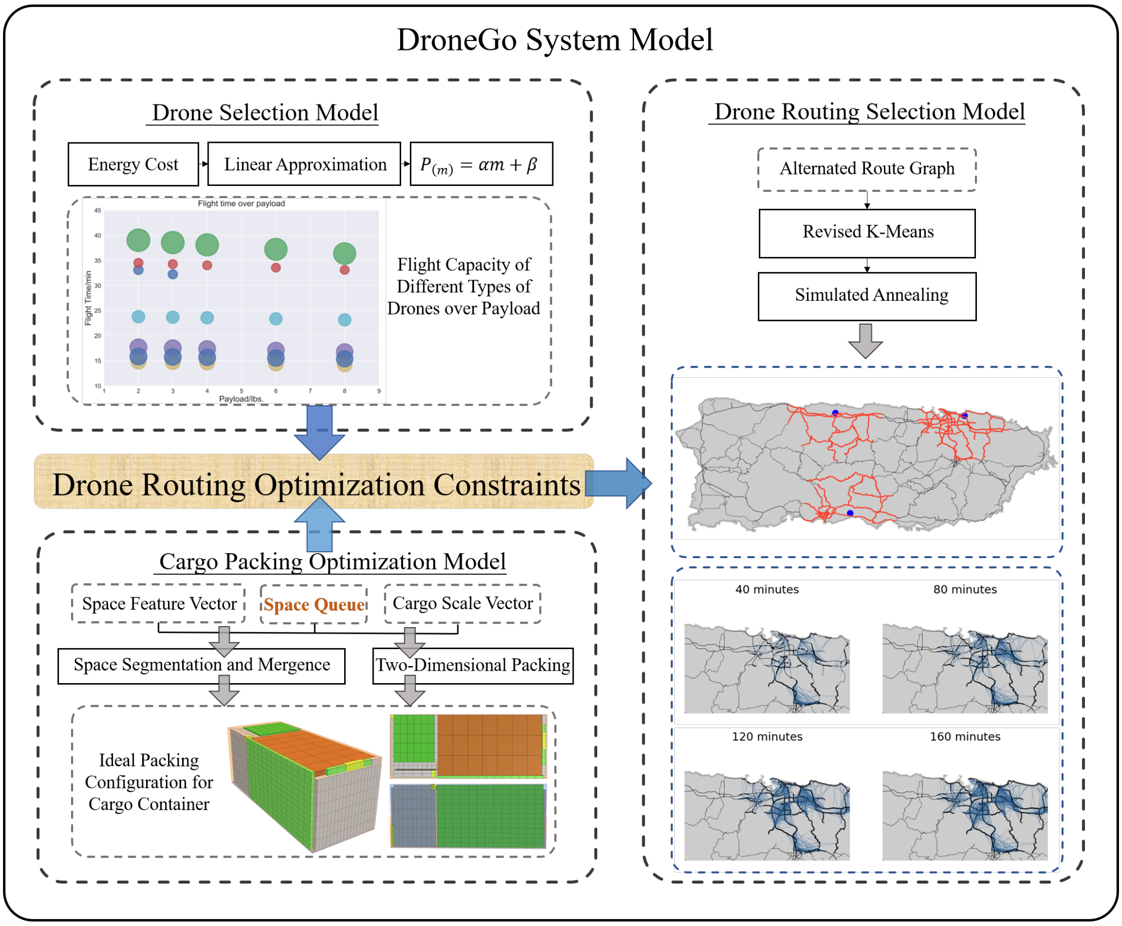



Figure 2: Overview of our model

## 2 Assumptions and Justification

We make some general assumption to simplify our model. These assumptions with corresponding justification are listed below.

- **Airports in Puerto Rico are the only candidates of location to position containers.**

Containers are delivered by cargo ship or airplane and unloaded in the dock or at the airport respectively. Unfortunately, tides even tsunamis are frequently followed by hurricanes. These natural calamities could easily make docks unable to work. In comparison, it is more possible to unload containers at the airport, which may be less attacked by the hurricane. Moreover, we could hardly shift containers far from the airport since most of road network has already collapsed. Therefore, we regard airports as feasible locations.

- **All types of drones get power from the universal batteries and the number of charged batteries satisfies drone energy requirements.**

  About 50 Km is the longest one-way flight distance by "drone B" from Attachment 2. This kind of limited reconnaissance cannot help emergency relief effectively. More assumption of the batteries' details will be discussed in Section 5.

- **The roads we investigate are only limited to motorway, trunk, primary, and secondary highways.**

  The value of these kinds of roads which helps indicate the importance of the highway is greatest among all roads. Motorway, trunk, primary, and secondary highways can be regarded as the major highways and roads. According to the related classification standard [3], reasons for granting a road a higher classification includes the volume of traffic that road should take. Therefore, if we prioritize those roads, the delivery of relief supplies can be accelerated significantly.

# 3 Nomenclature

| Notation | Meaning |
|---|---|
| $c$ | Flight capacity |
| $\alpha$ | The power consumed per kilogram of the payload |
| $\beta$ | The power to keep the drone's frame in the air |
| $g$ | A function to measure the distance between two vertices in a graph |
| $l$ | The length of a road |
| $d$ | A function to measure the distance between two points on the earth |
| $\Gamma$ | Temperature variables in simulated annealing algorithm |
| $\psi$ | The travelling time between two vertices |
| $a$ | The time at which a drone comes back to its cargo container |
| $h$ | The time at which a drone finishes scanning its last target |

# 4 Container Loading Optimization Model

Considering the limited ISO cargo containers and the expected disaster relief supplies, we aim to derive a model to optimize the packing configuration, based on the specific need of each cargo container.

In this section, we decompose container loading model into two submodels − Space Selection, Segmentation and Mergence Model as well as Cargo Selection and Packing Model. We will discuss the details about two parts. This optimization model is capable of managing the ideal packing of cargos, and maximizing fragmented space utilization.

## 4.1 Space Selection, Segmentation and Mergence Model

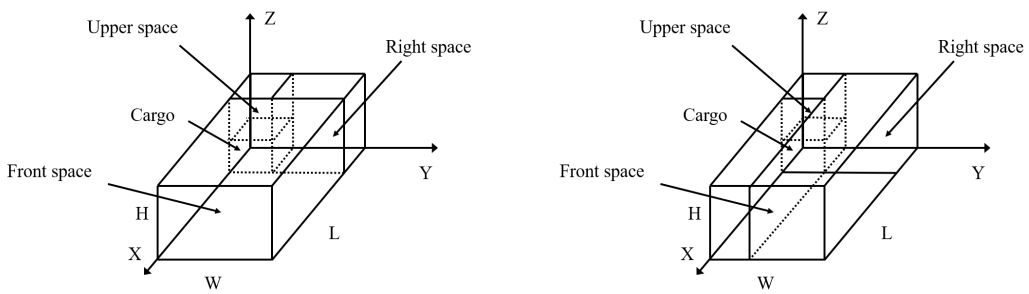
In this part, to quantitatively describe the space, we first establish a Cartesian coordinate system, and employ a six-dimensional vector $V = [x, y, z, L, W, H]$. The first three-element subvector $[x, y, z]$ represents the coordinates of the point at the bottom left corner of the space, serving as its position. $[L, W, H]$ represents its extension in three dimensions, which means length, width, and height respectively. For the cargo $i$, we develop a three dimensional vector $E_i = [l_i, w_i, h_i]$, indicating its length, width, and height. We use a queue Q to store the information of the space within the ISO cargo container, which initially contains one element, the whole space of the container.

- **Space selection**

  To achieve more reasonable space allocation, we sort the available space according to the coordinates of each one. We aim to place the cargos on the $YZ$-plane in priority, and then extend along the $X$-axis. Therefore, we choose space with smaller $x$-value, $y$-value and $z$-value in order to place the cargo. In this way, we traverse the entire space of the ISO cargo container.

- **Space segmentation**

  After loading cargos, the initial space of the container is divided into several parts. As is shown in Figure 3, there are two segmentation methods available. Through these two methods, we can split the original space into three subspaces, namely upper space, front space and right space, each with its own feature vector $V$. The two different ways only lead to the variation in the spatial scales of each subspace, but share the same position coordinates. It is hard to find which one is better. We need to evaluate the performance of each method based on the specific requirements and determine which to adopt. After implementing space segmentation, we remove the original space form $Q$ and add three new subspaces into the queue.




(a) Space segmentation method 1 (b) Space segmentation method 2

Figure 3: Two methods of space segmentation. The subspaces of the two methods share the same position coordinates, but have different spatial scales. Method 1 (a): Upper space, $V_u = [x_0, y_0, z_0+h_i, l_i, w_i, H_0-h_i]$, Right space, $V_r = [x_0, y_0+w_i, z_0, l_i, W_0-w_i, H_0]$, Front space, $V_f = [x_0 + l_i, y_0, z_0, L_0 - l_i, W_0, H_0]$. Method 2 (b): Upper space, $V_u = [x_0, y_0, z_0 + h_i, l_i, w_i, H_0 - h_i]$, Right space, $V_r = [x_0, y_0+w_i, z_0, L_0, W_0-w_i, H_0]$, Front space, $V_f = [x_0+l_i, y_0, z_0, L_0-l_i, w_i, H_0]$

- **Space mergence**

  Space segmentation results in a host of fragmented spaces, which separately cannot be placed into any cargo, but can be merged to contain small goods. Therefore, we should consider space consolidation after each segmentation. We define each time of loading cargo as a layer, and space mergence aims to integrate the fragment space from the previous layer into the available space of the current layer. To acquire the reasonable mergence and maximize space utilization, we take adjacent and spatial ductility in $X, Y, Z$ three dimensions into consideration. Space mergence contributes to the extension of the space in a specific dimension, and we replace the original space in $Q$ with the merged one.

## 4.2 Cargo Selection and Packing Model

In this section, we transform the three-dimensional integer programming problem into a two-dimensional problem. And based on the cargo packing, we employ a recursive model to find the best solution.

When implementing multilayer placement, we focus on the volume, and there are six directions of rotation resulting in six different $E_i$ to fill the space. While in a single-layer problem, the cuboid is converted into rectangle, and we care about the covered area. $E_i$ is reduced to two dimensions, with only two directions of rotation in consideration. Thus, it is feasible to reduce the dimensions of the optimization problem.

In cargo selection and packing model, we combine the same type of cargos, which means we pack those of the same size as a block. In this way, we get some blocks of cargos which can be regarded as the entire layers in placement and some remainders of each type. The development of each block turns into a two-dimensional packing problem, to cover the largest area on the plane with given rectangles. Therefore, we divide a complex multilayer problem into several single-layer problems and a simplified multilayer problem.

## 4.3 Implementation of Cargo Loading

Based on the above two submodels, we employ a recursive model to perform operations on space and cargo. We deal with the single-layer problems, cargo packing, in priority, and then turn to the multilayer placement of remainders. In each layer of placement, we in turn implement space segmentation and mergence until there is no more space to contain any cargo.

Our recursive model can be easily adopted to different conditions. When implementing the placement under actual circumstances, we only need to modify the initial spatial scale parameters of the container and cargos, which are $[L, W, H]$ and $[l_i, h_i, w_i]$ respectively.

For the cargo containers, there is a slight difference between the interior and the door opening scale. The door opening scale limit is only related to the single package scale limit and the last layer placement scale limit. Therefore, we consider the interior scale limit for the inner layers, and modify one constraint for the outmost layer of placement.

When we arrange the disaster response system in the ISO container, Figure 4 shows the ideal packing configuration. In the same way, we also get the best packing configuration of medical packages in the cargo bay.

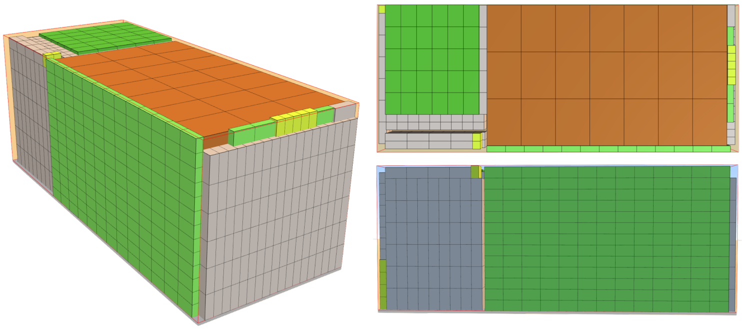


Figure 4: Implementation sample of cargo loading

We list the specific plan to support the Puerto Rico hurricane disaster scenario in advance and it will be explained in later sections.

Table 1: Cargo container packing configuration and position

| Container | Drone B | Drone H | Bay 1 | MED1 | MED2 | MED3 | Fill rate | City |
|---|---|---|---|---|---|---|---|---|
| Con A | 63 | 2 | 60 | 301 | 86 | 172 | 97.00% | San Juan |
| Con B | | | | | | | | Garrochales |
| Con C | | | 0 | 0 | 0 | 0 | 82.50% | Ponce |

# 5 Drone Selection Model

Considering the actual flight time is closely related to its load, we derive a submodel to figure out the relation between the flight time and the load. Through the given data in Attachment 2, it is difficult to directly give an explicit equation for the flight time as a function of the load. Moreover, the drone's weight without cargo which is highly correlated to the flight time is unknown. To address the problems above, we collect the typical proportion of the payload capacity and the battery weight to the drone's weight (20%, 20%) [6]. After deriving the total weight and the battery weight from the proportion, we apply the energy consumption equation for rotor wing drones [2]:

$$P = \frac{T^{3/2}}{\sqrt{2\rho\varsigma}} \tag{1}$$

the where $T \approx (W + m)g$ represents the thrust, given the frame weight $W$ in $Kg$, the payload and the battery weight $m$ in $Kg$ as well as the gravitational acceleration $g$, $\rho$ represents fluid density of air in $Kg/m^3$, and $\varsigma$ is the spinning blade disc in $m^2$.

Assuming the flight speed remains constant, we simplify the equation (1) and obtain the linear approximation:

$$P(m) = \alpha m + \beta \tag{2}$$

where $\alpha$ means the power consumed per kilogram of the payload and the battery weight $m$, and $\beta = \alpha W = const$ is the power to keep the drone's frame in the air.

To determine the coefficients $\alpha$ and $\beta$, we estimate the power consumption of the flight without cargo. Typically, the battery on a drone is the rechargeable lithium battery, whose specific energy is in the range of $170 - 200\ Wh/Kg$ [4]. Therefore, we suppose the specific energy of the battery in our model is $150\ Wh/Kg$ to guarantee power surplus. We replace the power $P$ in equation (1) with the average power $\overline{P}$. Then, we get $\alpha$ and $\beta$ by:

$$\alpha = \frac{\epsilon m_0}{(W + m_0)t_0}, \qquad \beta = \alpha W \tag{3}$$

where $\epsilon = 150\ Wh/Kg$ represents specific energy, $W$ means the frame weight, and $m_0, t_0$ mean the battery weight and flight time without cargo.

To select the best drone for delivery and reconnaissance, we devise the **flight capacity** to describe the performance of the drone. $c = L/V$, where $c$ is the flight capacity, $L$ is the flight distance, and $V$ is the volume of the drone occupied in the cargo container. Finally, we get the relation between flight time over payload as shown in Figure 5. We choose several possible sets of medical package as samples to investigate all kinds of drones' delivery performance. The vertical height means the actual flight time and the bubble radius means the flight capacity.

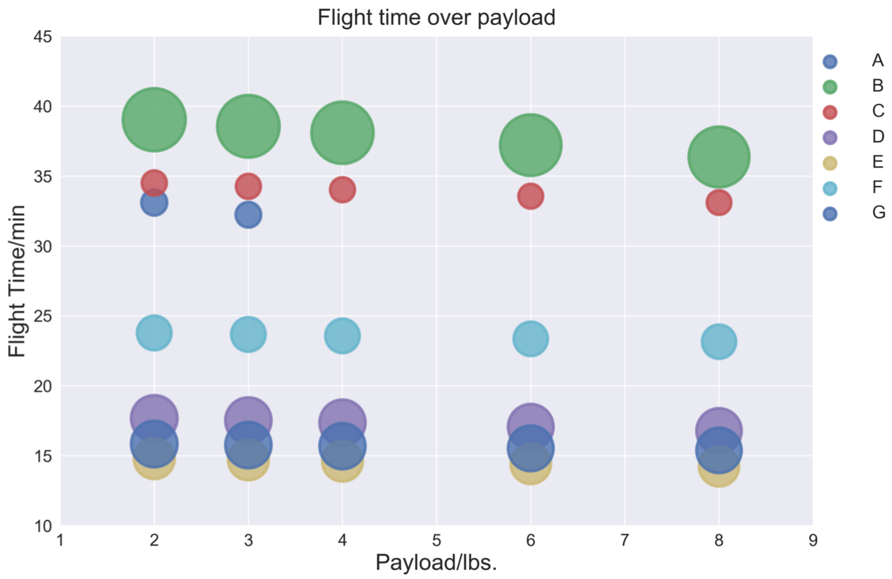



Figure 5: The bubble diagram of flight time and capacity over payload

Figure 5 shows that drone B has the overwhelming advantages of flight performance over other drones when medical package weight is less than 9 pounds. Moreover, since 9 pounds per drone can basically meet the daily need of every delivery location, we decide to only choose **drone B along with cargo bay 1** to complete the mission.

**Modifications of Flight Time** Considering a drone should take off and land, wasting the actual flight time, we introduce the **up&down time** to describe the time a drone spends on the take-off and landing. According to the reference [1][10], in general, the professional-class drones' up&down speed is $6\ Km/h$ and the flight height is around $100\ m$. To avoid the impact caused by contingency, we suppose the up&down time is $1$ minute. Therefore, the flight time mentioned in following sections is the flight time minus the up&down time.

# 6 Drone Routing Selection Model

Our system needs to direct our drones to certain places so that they can accomplish the tasks. In order to provide practical routes in Puerto Rico, real world geographic data should be the basis of our model. Therefore, we gather the geographic information from OpenStreetMap, an open source GIS database, which contains the information of the roads in Puerto Rico. By filtering the roads using their tags, we get the major road network's information. Similarly, we get other useful geographic information that is used during modeling, like the location of Airports.

Our routing model needs to provide routes for two different tasks: video reconnaissance and medical supply delivery. We notice that generally video reconnaissance can be only done during daytime when light is sufficient. Under this assumption, the drones had better do medical supply during the night. As a result, the original routing model can be divided into two smaller and isolated submodels to accomplish the tasks. One for reconnaissance and the other for delivery. Since they are isolated, we build them separately. First, we would introduce our reconnaissance routing model.

## 6.1 Reconnaissance routing model

Road networks can be regarded as a graph, denoted as $G = (V, E)$, whose edges are roads and vertices are cross. We use function $g$ to represent the distance between two vertice. However, to simplify this routing problem, we apply transformation, $L$, on the original road graph. $L(G) = (E, F)$ is defined as $F = \{\{e_1, e_2\} \in P_2(E) | e_1 \cap e_2 \neq \emptyset\}$, which means that we turn the edges, $e_1$ and $e_2$, into vertices, $v_1$ and $v_2$, and vertices into edges. The vertices in this graph represent the centroids of roads denoted as $c$ while edges have no physical meaning. We define function $d$ as the distance between two points on the earth and $l$ as the length of a road, then function $g$ is defined as following for this graph:

$$g(v_1, v_2) = \begin{cases} \dfrac{l_1 + l_2}{2} & e_1 \cap e_2 \neq \emptyset \\ \dfrac{l_1 + l_2}{2} + d(c_1, c_2) & e_1 \cap e_2 = \emptyset \end{cases} \tag{4}$$

We name this graph **alternated road graph** (ARG). For assisting HELP, Inc. to know the situations of the ground of Puerto Rico, our drones need to pass every road at least once. In the ARG, this means that we need to go through every vertex. Our reconnaissance model uses real geographic information, so $l$ and function $d$ are just the physical length and distance. Therefore, a route and its cost on our ARG can be mapped to physical world route and its required time.

### 6.1.1 Representation of a Route in ARG – a Vector Approach

In our ARG, we use natural numbers to represent a vertex. We use 0 to represent the vertex of cargo container. A drone's route can be described as a vector whose elements are, in sequence, the vertices that it goes through. A drone needs to depart from the cargo container and finally comes back. Once it comes back, it will change a full-charged battery and continue to move. So, a route can be described as

$$\mathbf{r} = \langle\ 0\ \mathbf{r'}\ 0\ \rangle$$

$$\mathbf{r'} = \langle\ v_1\ ...\ v_i\ \rangle$$

All the routes can be combined in a big vector

$$\mathbf{r} = \langle\ 0\ \mathbf{r'}\ 0\ \mathbf{r''}\ 0\ ...\ 0\ \rangle$$

This is a vector whose first and last elements are all 0 and $v_i \in \mathbb{N}$.

### 6.1.2 Constraints and concepts for routes in ARG

Our ARG representation only contains the connection information, while our drones are supposed to consider other constraints like battery, time, and so on. Referring to the algorithm for vehicle routing problems for drone delivery [2], we apply following constraints on the routing model to get reasonable and physical routes. All these constraints and newly defined variables will be used as conditions during programming the model.

**Completeness constraint** To go through all roads quickly, our routes should not pass a road twice. Therefore, one vertex can be only reached once in our graph, which implies following constraints.

$$\begin{aligned} &\sum_{i\neq j} x_{ij} = 1 && \forall i \in \mathbb{N}_0, \forall j \in \mathbb{N} \\ &\sum_{i\neq j} x_{ij} - \sum_{i\neq j} x_{ji} = 1 && \forall i \in \mathbb{N}, \forall j \in \mathbb{N} \end{aligned} \tag{5}$$

In this constraint, $x \in \mathbb{N}$, and $x_{ij} = 1$ when a drone moves from location i to j (else, $x_{ij} = 0$). This constraint makes sure that every point is detected and is only detected once.

**Reusability constraint** In our model, we suppose the time to change a battery is so short that it can be ignored. As a result, all drones are immediately usable when comeing back. We formulate this constraint as:

$$\begin{aligned} \sum \sigma_{ij} \leq x_{i0} \\ \sum \sigma_{ji} \leq x_{0i} \end{aligned} \qquad \forall i \in \mathbb{N}_0, \forall j \in \mathbb{N}_0 \tag{6}$$

We define $\sigma \in \mathbb{N}$, and let $\sigma = 1$ when a drone comes back from $i$ to cargo container and then leaves for $j$. Otherwise it would be 0. This constraint ensures that our drones keep working without resting.

**Number constraint** The total number of drones is restricted by

$$\sum x_{0i} - \sum_{i\neq j} \sigma_{ij} \leq M \qquad \forall i \in \mathbb{N}_0, \forall j \in \mathbb{N}_0 \tag{7}$$

where $M$ is a constant. The total number of drones is less than or equal to $M$.

**Time constraint**

$$\begin{aligned} &t_i - t_j + g(i,j)/V \leq K(1 - x_{ij}) \\ &t_i - a_i + g(i,0)/V \leq K(1 - x_{i0}) \\ &a_i - t_j + g(0,j)/V \leq K(1 - \sigma_{i0}) \\ &t_i \leq l \leq T \end{aligned} \qquad \forall i \in \mathbb{N}_0, \forall j \in \mathbb{N}_0 \tag{8}$$

where $d$ is the distance function, which measures the distance between $i$ and $j$. We define $t_i$ as the time $i$ visited by a drone. $V$ is the drone's velocity. $a$ is defined as the time that a drone goes to the cargo container directly after it leaves $i$. $K$ is a large constant which forms an upper limit. $l$ is the total time cost and $T$ is a constant, which represents time limit. First three constraints are brought by the definition of $t$ and $a$, which hold directly. The last constraint ensures that the delivery time of our route is bounded.

**Energy constraint**

$$f_i - f_j + p(g(v_i, v_j)/V) \le K(1 - x_{ij}) \qquad \forall i \in \mathbb{N}, \forall j \in \mathbb{N} \tag{9}$$

Here $f_i$ represents the energy in kJ consumed by a drone when it reaches location $i$. $p$ is the power of a drone. $V$ is the drone's velocity. This constraint ensures that energy holds in the whole route.

## 6.2 ARG routing algorithm

Inspired by the algorithm from vehicle routing problems for drone delivery [2], instead of finding a optimal solution we consider to use simulated annealing (SA) algorithm to find a sub-optimal solution. However, we cannot directly use simulated annealing algorithm, since our ARG is strongly restricted by real world situation. The main possible problems are:

1. The number of roads is great according to our gathered geographic information, but the running time of a typical vehicle routing algorithm based on SA will increase exponentially as vertices increase [2]. Directly using SA may result in an unacceptable time requirement.
2. Length of the road and distance between roads are challenges for drone due to their short battery life. Most of the solutions are unreasonable. Therefore, even if we manage to use SA, the algorithm may not steadily provide correct solution.

To prevent these possible problems from occurring, we revise K-means method and combine it with simulated annealing (SA) algorithm to build our routing model.

### 6.2.1 Cargo containers' positions

To reduce the number of vertices for the SA algorithm, we first need to drop those roads that can never be reached by our drones from a certain cargo container, which means that we need first choose the best airport(s). Since we want to see as many roads' situations as possible, the best airport(s) can cover more roads and deliver those medical supply. We achieve this by observing their equivalent vertices in ARG: for a road's corresponding vertex $v_r$ and the airport's vertex $v_a$, if

$$g(v_a, v_r) = d(c_a, d_r) + l_r < R$$

Where $g$, $d$, and $l$ are the same concepts in Equation 4. We choose twelve major airports and ranks their coverage in total length of reachable roads, which is shown in Figure 6

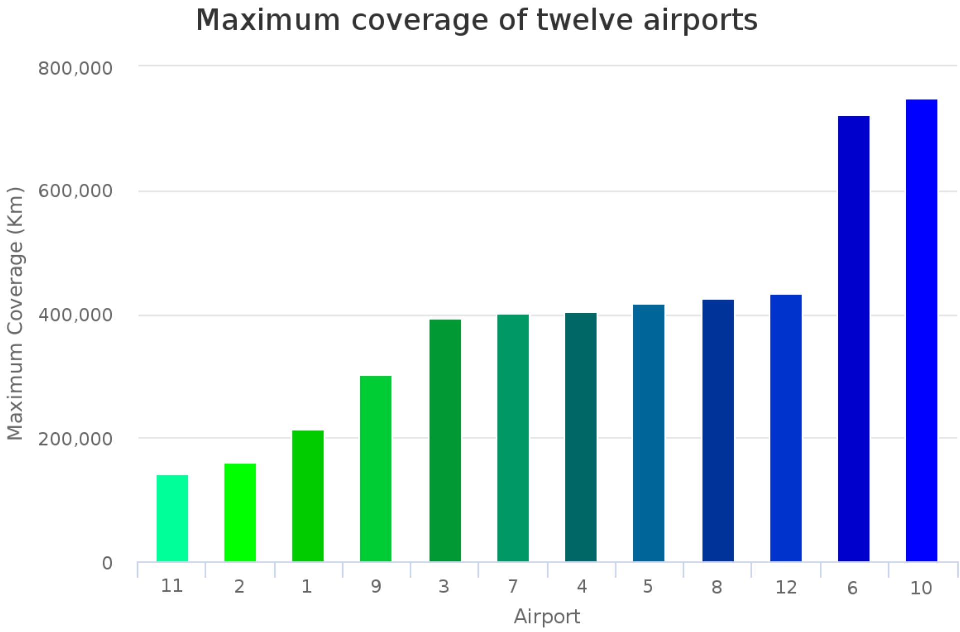



Figure 6: The rank of maximum coverage of twelve airports

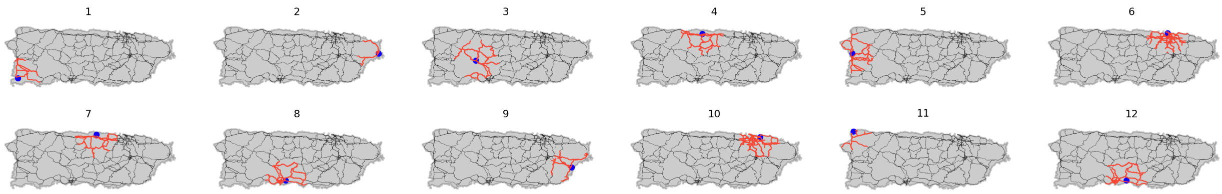



Figure 7: Each airport's maximum covering region

The airport *10* has a very special location: **not only because it can cover most, but also it reaches 4 of 5 hospital that needs medical supply.** The details for the routing model for medical delivery are covered in later Section 6.3.3. Now, we would choose the first three airports to discuss further: drone fleets in airport 10 and airport 4 will be able to delivery medical supply. Since airport 6 is close to airport 10, we choose the third location as airport 12 to maximize coverage.

### 6.2.2 Revised K-means method

Then for the remaining roads, we take their equivalents in and try to cluster them. We hope that by using a clustering algorithm, we can group vertices in ARG.

First, we directly use K-means, but the output groups may cover a large area and roads are separated. We suspect that the center of the K-means method may fall in the middle of two far away vertex, which results in undesired result: the ARG representation does not contain length information, but two far away long roads are hard to handle during routing.

Therefore, we revised K-means as following: force the center of K-means to be located on a road. The revised K-means method, with $k$ equals to 6, has a satisfying result. Take the clustering result of airport 10 (Figure 8) as an example, the clustering results are: Then the

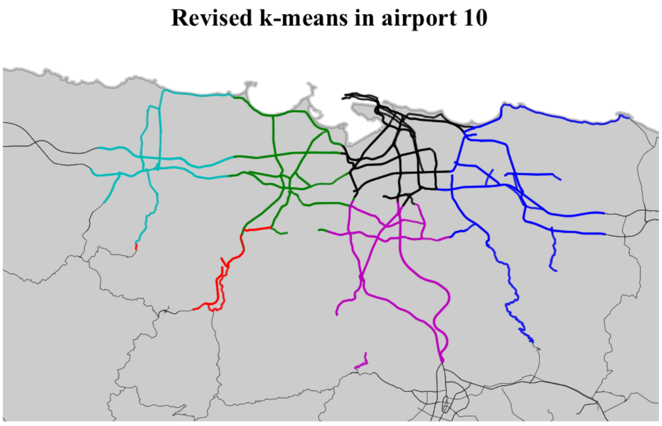



Figure 8: Clustering result of revised K-means method

whole region is divided into 6 smaller sub-regions. Now we are ready to apply SA to each sub-region.

### 6.2.3 The cost function for routes

We first define our cost function. In our case, the cost of the solution vector is influenced by both delivery time and energy consumed.

**Energy consumed** The energy consumed is only related to the solution vector – the solution vector is a sequence of routes, and these routes would be assign to several drones. Since the power of all drones are the same, the energy consumed is only related to solution vector. Obviously, the energy consumed by a drone cannot exceed its battery capacity (Equation (9)). We treat the violation as a penalizing term. Algorithm 1 is the persudo code for this function.

**Algorithm 1:** ENERGYPENALTY

**Input:** s - The solution vector
**Output:** $\lambda$ - possible penalty for violating energy constraint

**for** $k \leftarrow Len(s) - 2$ ***to*** $0$ **do**
  $i \leftarrow s_{k+1}$
  $j \leftarrow s_k$
  **if** $i \neq 0 \vee j \neq 0$ **then**
    $\psi_{ij} \leftarrow g(i,j)/v$
    $t \leftarrow t + \psi_{ij}$
    **if** $j == 0 \bigwedge i \neq 0$ **then**
      $E \leftarrow \frac{\beta t}{1-(\alpha/\epsilon)t}$
      **if** $E < 0$ **then**
        $\lambda \leftarrow \lambda - K \cdot E$
      **if** $E/\epsilon > Q$ **then**
        $\lambda \leftarrow \lambda + K \cdot (E/\epsilon - Q)$
      $t \leftarrow 0$
**return** $\lambda$

In this algorithm, $\psi_{ij}$ means the traveling time between $i$ and $j$; $\alpha$, $\beta$, and $\epsilon$ are the same variables in Section 5; the relationship between time consumed, $t$, and energy consumed, $E$, is deduced through

$$E = \sum_{a}^{b-1} p(E/\xi)\psi k(k+1) = \frac{\beta \sum_{k=a}^{b-1} \psi_{k(k+1)}}{1-(\alpha/\xi)\sum_{k=a}^{b-1} \psi_{k(k+1)}} = \frac{\beta t}{1-(\alpha/\xi)t}$$

**Time consumed** To calculate the time consumed by using this route, we first need to figure out the time used by each sub-route. We define $a$ as the time a drone returns to the cargo containers, and the $h$ as the total time consumed. With these two time variables, we can do list scheduling Algorithm 2 shows the persudo-code of finding arrival and delivery times. Algorithm 3 is the persudo-code of list scheduling for these sub-routes. Note that this scheduling method only returns the total consumed time. However, after we get a good solution vector, we need to implement a another version of list scheduling to get the complete route schedule.

**Algorithm 2:** ARRIVAL AND DELIVERY TIMES

**Input:** s - The solution vector
**Output:** **u** - a vector of $a$ and $h$ of sub-routes

$h \leftarrow 0$
$a \leftarrow 0$
$\mathbf{u} \leftarrow \emptyset$ **for** $k \leftarrow 1$ ***to*** $Len(s) - 1$ **do**
  $j \leftarrow s_{k+1}$
  $i \leftarrow s_k$
  **if** $i \neq 0 \vee j \neq 0$ **then**
    $h \leftarrow a$
    $\psi_{ij} \leftarrow g(i,j)/V$
    $a \leftarrow a + \psi_{ij}$
    **if** $i == 0 \bigwedge j \neq 0$ **then**
      $\mathbf{u} \leftarrow \mathbf{u} \cup (h, a)$ $a \leftarrow 0$
**return** ***u***

**Algorithm 3:** LISTSCHEDULE

**Input:** **u** - $(a, h)$ pair vector
**Output:** The overall consumed time

$\mathbf{k} \leftarrow \emptyset$
**for** $k \leftarrow 0$ ***to*** $M - 1$ **do**
**for** $k \leftarrow 0$ ***to*** $Len(u) - 1$ **do**
  $j \leftarrow s_{k+1}$
  $i \leftarrow s_k$
  **if** $i \neq 0 \vee j \neq 0$ **then**
    $(h', a') \leftarrow POPMINA(\mathbf{k})$
    $(h, a) \leftarrow u_i$
    $INSERTELEMENT(\mathbf{k}, (a' + a, a' + h))$
**return** $MAXCONSUMEDTIME(\boldsymbol{k})$

**Cost function** Combine the original time cost and the penalty function, we can get our cost function. The Algorithm 4 is the persudo code of our cost function. Note that if the total time consumed violates time constraint, we also give it a great penalty.

**Algorithm 4:** COST

**Input:** **s** - The solution vector vector
**Output:** c - The overall cost

$l \leftarrow LISTSCHEDULE(s)$ **if** $l > T$ **then**
  $c \leftarrow c + K(l - T) + ENERGYPENALTY(s)$
**else**
  $c \leftarrow c + ENERGYPENALTY(s)$
**return** $c$

#### 6.2.4 Simulated Annealing algorithm

**Exchange rules** In SA algorithm, we need to find the neighboring solution of current solution. Here we used the rule from [2]. Figure 9 is the description of the exchange rules.

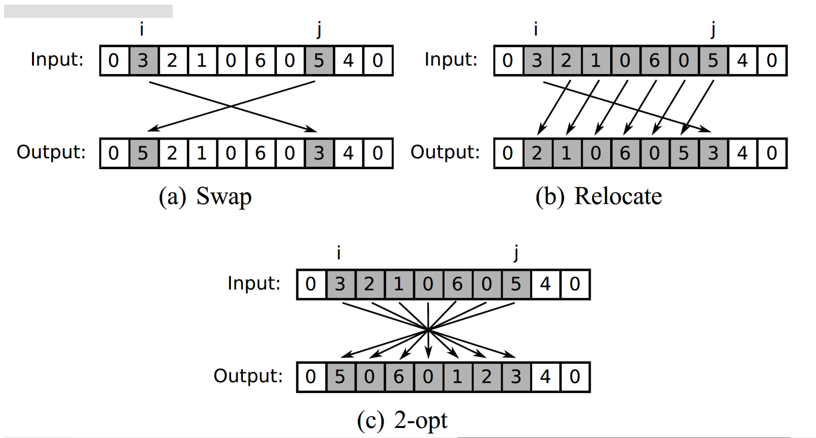



Figure 9: Exchange rules

**Random solution** To start a SA algorithm, we first need a random solution. The method we used is: first generate a list contains all vertices' indices. The indices will appear only once. Then we apply exchange rule to them, together with same number of zero. Repeat the process for 1000 times. The result solution will be a random solution.

**Simulated annealing algorithm** The Algorithm 5 provides the presudo code of SA algorithm[]. In this algorithm, $\Gamma_0$ is the initial temperature, $\Gamma'$ is the final temperature, $\mu$ is the cooling factor, and $\eta$ is the number of exchanges in each cooling cycle. Our implementation use $\Gamma_0 = 1$, $\Gamma' = 0.7$, $\mu = 0.99$, and $\eta = 1000$. Generally, the cost of solution will become stable after several cooling cycles. We use Metroplis algorithm to determin whether the neighboring solution should replace current one.

#### 6.2.5 Distributed scheduling

We now need to apply the SA algorithm to the sub-regions clustered by our revised k-mean method. Instead of letting all drones go to a single sub region, then the second ..., we choose to divide our drone fleet into smaller groups, according to the coverage of each sub-region. The size of group is determined through following formula.

$$N = round(L_i/L' \cdot M)$$

**Algorithm 5:** SIMULATEDALGORITHM

**Input:**
$\Gamma_0$ - The initial temperature
$\Gamma'$ - The final temperature
$\mu$ - The cooling factor
$\xi$ - The number of exchange per round of cooling
**Output:** $s'$ - The sub-optimal solution

$s \leftarrow RANDOMSOLUTION()$
$\Gamma \leftarrow \Gamma_0$
**while** $\Gamma > \Gamma'$ **do**
  $\Gamma \leftarrow \mu\Gamma$ **for** $k \leftarrow 1$ ***to*** $\eta$ **do**
    $i \leftarrow RANDINT(1, LENGTH(s) - 1)$
    $j \leftarrow RANDINT(1, LENGTH(s) - 1)$
    $R \leftarrow RANDINT(1, 3)$
    $s' \leftarrow EXCHANGE(s, R, i, j)$
    $X \leftarrow RAND()$
    **if** $exp(-\frac{COST(s')-COST(s)}{\Gamma}) \geq X$ **then**
      $s \leftarrow s'$
**return** $s$

where $M$ is the total number of our drone fleet, $L_i$ is the coverage of $i$-th sub-region, i.e. the total length of roads in $i$-th sub-region, and $L'$ is the coverage of the whole region.

To show the advantage of distributed scheduling, we define **idle ratio** as

$$r = \frac{\sum n_i}{\sum n}$$

where $\sum n_i$ is the total number of drones that is idle in all plans, and $\sum n$ is the total number of drones that is able to be used in all plans. Suppose that there are 60 drones can be used in each airport, we can compare their idle ratio. The lower the idle ratio, the higher the efficiency. The result is shown in Figure 10.

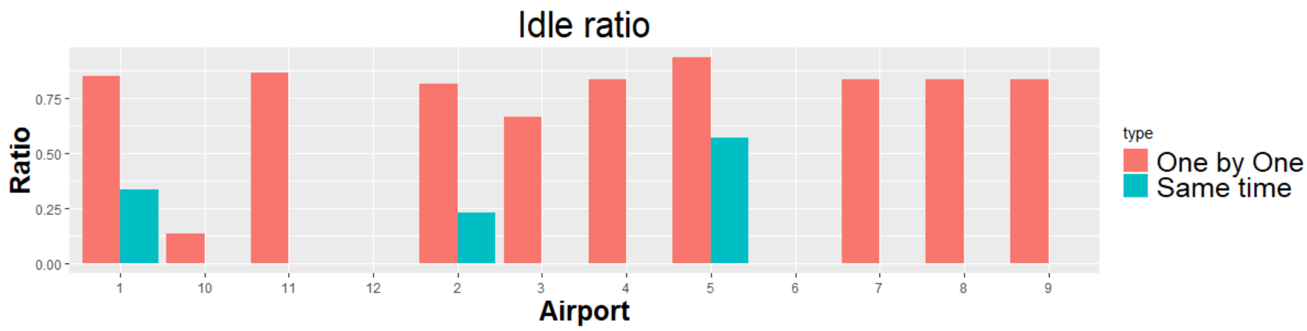



Figure 10: Idle ratio

In most airports, distributed scheduling shows higher efficiency. Therefore, we would apply SA algorithm in the way of distributed scheduling.

## 6.3 Routing result

### 6.3.1 Number of drones

We want to decide the proper number of drones through this routing model. The task of deliver medical supply is light, but vedio reconnaissance needs large number of drone, since there are many roads. We hope that HELP, Inc. get sufficient information as quick as possible so that they can take action promptly. We take the cargo container at No. 10 airport as an example. When we use 30 drones, ARG routing algorithm gives us a plan takes 19251.2 seconds to complete the task, while when we use 60 drones, ARG's solution only needs 19521.2 seconds. However, when we use 70 drones, ARG's solution consumes 17746.3 seconds, which, due to the error of SA algorithm, takes even longer time than previous case. We think that using more than 60 drones will not shorten the consumed time. Considering the configuration of the container, there is no influence in other cargo if we increase the number of drones B to 63 ($7*3*3$). Therefore, we prepare 63 drones B in each cargo container.

Besides, due to the destruction of communication networks, we need tethered drone H to serve as communication base stations. Considering each drone H can provide network of coverage with a 50 $Km$ radius [7], we take two drones H for each area.

### 6.3.2 Example of flight plan

Since our routing model would behave similarly under different conditions, we will use the flight plan for the drone fleet located in No. 10 air port as an example. The whole plan is long and tedious, so we would use graph to illustrate our plan.

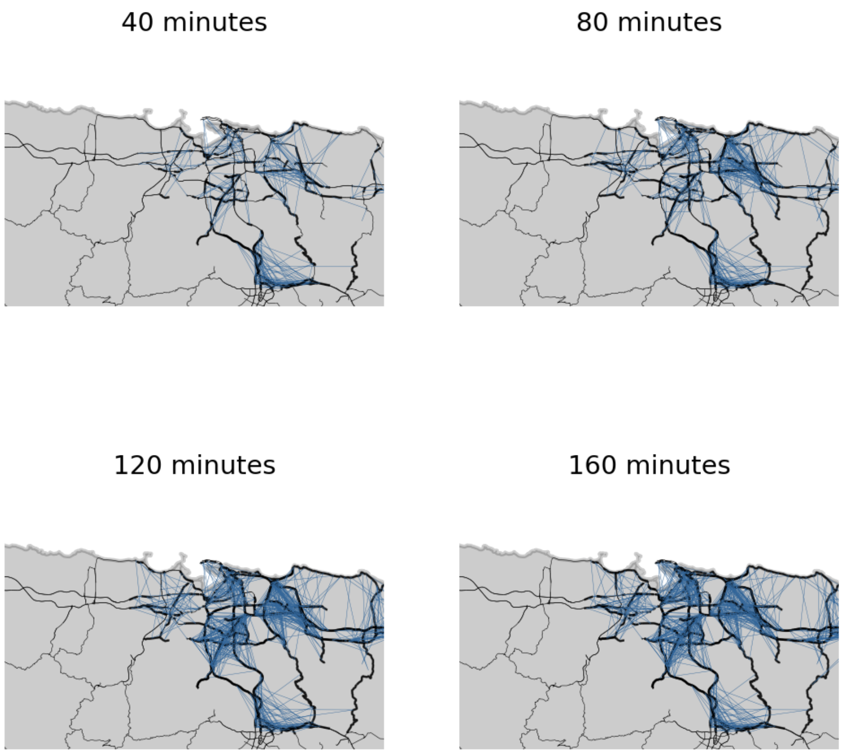



Figure 11: Drone reconnaissance process using ARG's route

In Figure 11, the light blue lines represent drones' movement, also the edges in ARG.

Following the direction of the blue lines, a drone can scan the roads one by one. Note that, these blue lines are not real scanning routes, they just connected all roads that is scanned in sequence.

The black curves represent the road that has been scanned or is being scanned. We notice that in the first 80 minutes, many of the roads have been scanned.


Figure 12: Covered distance over reconnaissance time

Figure 12 shows how the drone fleet uses the result of ARG routing algorithm by presenting the increase length of scanned roads. The $y$ axis shows the total length of roads that is being scanned or has been scanned. We find that, with each 60 drones in three airports, ARG's solution can let drone fleet finish most part of its job within 3 hours.

### 6.3.3 Medical package delivery routing

We assume that delivering medical packages is implemented in night, separated from road network reconnaissance, so we only need to consider the straight line distance between delivery locations and their nearest container positions. Detialed information and the range of corresponding drone for delivery is listed in Table 2, and the routes between cargo containers and delivery locations are visulized in Figure 13.

Table 2: Distance between delivery location and container location

| Delivery Location | Nearest Position | Distance $[Km]$ | Range $[Km]$ |
|---|---|---|---|
| Puerto Rico Children's Hospital | San Juan | 17.60 | 43.22 |
| Hospital Pavia Santurce | San Juan | 7.38 | 46.56 |
| Hospital HIMA | San Juan | 24.54 | 43.22 |
| Caribbean Medical Center | San Juan | 38.91 | 46.56 |
| Hospital Pavia Arecibo | Garrochales | 17.23 | 48.84 |

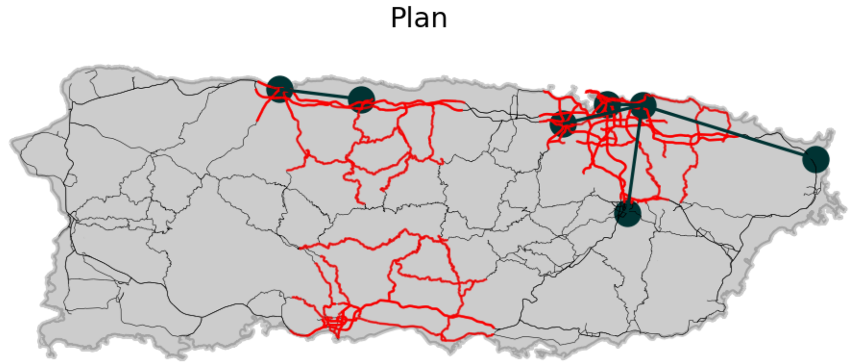



Figure 13: The black straight lines indicate the flight routes between positions of cargo containers and delivery locations. The red curves represent coverage of road network reconnaissance.

We see that all of five delivery locations are within the range of drone B with appropriate payload for each. More specifically, for Hospital HIMA and Caribbean Medical Center, we dispatch drone B for one-way flight to deliver medical packages. For the other three, drone B is capable of travelling between the two places. Besides, there is sufficient time, 12 hours per day [8], for medical package delivery, so daily medical requirements are easily fulfilled.

# 7 Model Analysis

## 7.1 Sensitivity Analysis

In the Drone Selection Model, the up&down time will have an impact on the results. Therefore, it is necessary to analyze the influence of this factor on the ultimate results.

Up&down time influences the effective flight time for the drone, which results in different flight range. We increase the up&down time of all drones, and get the optimal position(s) for cargo container(s). The changes are listed below in Table 3.

As can be seen from Table 3, the flight range of each type of drone decreases no more than 1.6% with the increase rate of up&down time ranging from 10% to 50%, and the optimal positions for cargo containers remain the same. From this perspective, the model can be considered as a robust one.

Table 3: Flight range and position selection over up&down time

| Change of time | +10% | +20% | +30% | +40% | +50% |
|---|---|---|---|---|---|
| Drone A | -0.30928% | -0.61856% | -0.92784% | -1.23711 | -1.54639% |
| Drone B | -0.15424% | -0.30848% | -0.46272% | -0.61697% | -0.77121% |
| Drone C | -0.19108% | -0.38217% | -0.57325% | -0.76433% | -0.95541% |
| Drone D | -0.20408% | -0.40816% | -0.61224% | -0.81633% | -1.02041% |
| Drone E | -0.20408% | -0.40816% | -0.61224% | -0.81633% | -1.02041% |
| Drone F | -0.15424% | -0.30848% | -0.46272% | -0.61697% | -0.77121% |
| Drone G | -0.19108% | -0.38217% | -0.57325% | -0.76433% | -0.95541% |
| Position 1 | San Juan | San Juan | San Juan | San Juan | San Juan |
| Position 2 | Garrochales | Garrochales | Garrochales | Garrochales | Garrochales |
| Position 3 | Ponce | Ponce | Ponce | Ponce | Ponce |

## 7.2 Strengths and Weaknesses

### 7.2.1 Strengths

- **The Container Loading Optimization Model is highly flexible and adaptable.** When we change the number of each cargo, new optimal loading strategy can be quickly derived without revising the algorithm.
- **Mining accurate data from prestigious websites and database.** We extract the road network data from OpenStreetMap. Only with these reliable data, can we conclude that the model we construct is appropriate.
- **Design a novel and comprehensive model to achieve our objective.** For example, we apply revised K-means Clustering to optimize Simulated Annealing algorithm, which decreases greatly the complexity of algorithm.

### 7.2.2 Weaknesses

- **Drone selection standard is not multiple.** We only investigate the flight distance and the volume of the drone. In fact, we need take bay volume into consideration.
- **Convert 3D routing to 2D routing.** We ignore the possibility of the tall objects like skyscrapers blocking the drone's path.
- **The road data we acquire are kind of fractional.** In our model, the branch road number effects the routing selection. We had better find other data source.

## 7.3 Further Discussion

It is impossible to know the whole island's road network's situation by only three cargo containers and the drone fleets. However, we notice that even if we use small number of drones, we can still quickly know the surrounding area's situation. Therefore, if we can turn 3 large drone fleets to 12 small drone groups, we may know more about the condition of Puerto Rico. In that case, most of the roads can be covered, like what Figure 14 shows.

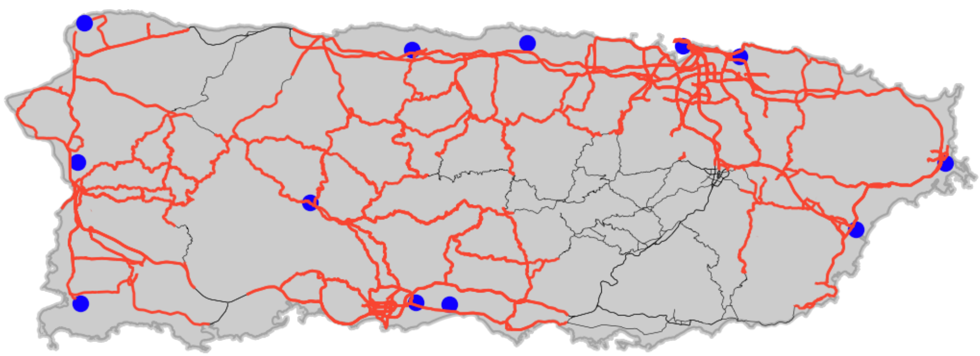


Figure 14: Twelve drone groups' maximum covered region

# 8 Conclusion

Our paper manages to model DroneGo system to comprehensively take drone selection, packing configuration, container position selection, and drone route selection into consideration, and offer the optimum management of disaster response system. We make reasonable assumptions. We define flight capacity to measure the performance of different type of drones for ideal selection. We construct Cargo Loading Optimization Model to reach the best packing configuration of DroneGo disaster system in ISO cargo container with recursive analysis. We derive optimal routing planning for medical package delivery and road reconnaissance by Drone Routing Selection Model through Simulated Annealing algorithm and revised K-means Clustering.

We gather accurate data about disasters to analyze the results through the DroneGo system model. We further find the tradeoffs for addressing the shortcomings of current disaster response system. We also conduct sensitivity analysis to test our model, and strengths and weaknesses are discussed. One memo is written to provide our results and recommendations to Help, Inc.

# Memo

**To:** Chief Operating Officer

**From:** Team 1919009

**Data:** January 28, 2018

## Dear Chief Operating Officer,

We are glad to hear the plan about designing a DroneGo disaster response system, which will definitely be an important milestone for the HELP, Inc. to become more efficient and intelligent. For such a system, it is necessary to think carefully before making final decision. To help you determine the location to position cargo containers and figure out the package configuration and the flight route, we give our recommendation through mathematical modelling.

Firstly, we build up a Container Loading Optimization Model to find the best packing configuration for each of the cargo based on specific cargo requirements. Then we define flight capacity to describe the capacity of the drone and decide on drone B for delivery and reconnaissance, and drone H to provide communication network. Considering the traffic and climate conditions in the disaster area, we choose the airports to position cargo containers. Combined with the Route Relection Model for medical package delivery and video reconnaissance, we get a comprehensive result.

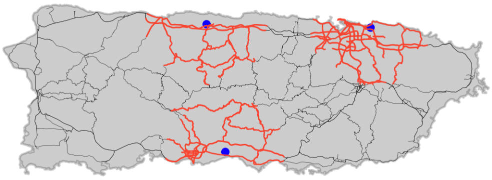
Our model indicates that it is better to use all three cargo containers, numbered Con A, Con B, and Con C, to transport DroneGo disaster response system, which is respectively positioned at three airports, Luis Munoz Marin International Airport, Antonio Nery Juarbe Pol Regional Airport, and Fort Allen Airport, serving as supply stations. Through Container Loading Optimization Model, we have confirmed that cargo demand for each container is met. The detailed packing configuration and city of each container is shown in Table $4$, and the details about the accurate position is shown in Figure $15(a)$.

Table 4: Cargo container packing configuration and position

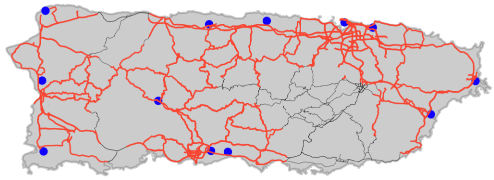
<table>
<tr><th>Container</th><th>Drone B</th><th>Drone H</th><th>Bay 1</th><th>MED1</th><th>MED2</th><th>MED3</th><th>Fill rate</th><th>City</th></tr>
<tr><td>Con A</td><td rowspan="3">63</td><td rowspan="3">2</td><td rowspan="2">60</td><td rowspan="2">301</td><td rowspan="2">86</td><td rowspan="2">172</td><td rowspan="2">97.00%</td><td>San Juan</td></tr>
<tr><td>Con B</td><td>Garrochales</td></tr>
<tr><td>Con C</td><td>0</td><td>0</td><td>0</td><td>0</td><td>82.50%</td><td>Ponce</td></tr>
</table>

From Table $4$, three containers in all load three months medical packages for the whole diaster zone. Con A is responsible for medical package delivery at 4 locations, Caribbean Medical Center, Hospital HIMA, Hospital Pavia Santurce, Puerto Rico Children's Hospital. Con B takes medical packages for Hospital Pavia Arecibo, with Con C no task for medical package transportation. When considering futher usage for future potential disasters, you can fill Con C with the same amount of medical packages as Con A and Con B to better fit general conditions.

Drones in all three containers reconnaissance road conditions within their range during the daytime, and deliver medical packages if needed in night. We confirm timely medical



(a) Current container positions



(b) Better container positions

Figure 15: The blue points in the figure represents the airport chosen to position cargo containers, and the red line means roads scanned or being scanned. (a): Current optimal positions at three airports and its maximum reconnaissance coverage. (b): Recommanded better positions for cargo containers, at 12 airports, and its maximum reconnaissance coverage with the total number of drones remain same.

package supply for five delivery locations, and maximize the reconnaissance scope. The detailed drone routings and covered distances are shown in Figure 13.

From the result of our model, we reach the conclusion that the cargo containers are large enough to load adequate drones and medical packages to disaster area, and the medical package demand is easily fulfilled. One limitation of the current management is that the maximum coverage of the drones is limited. Drones in the same container can only cover limited area of a circle centered at their supply station. Therefore, it is not the increase of the number of drones in one station but the increase of the number of supply stations that expand the reconnaissance regions. Assuming the number of drones is constant, we select more locations scattered on the island to serve as supply stations. The decline in the number of drones allocated to each station only leads to longer time to reach maximum detection coverage, but ends up with the same reconnaissance area.

Therefore, we recommend maintaining current medical packages and drones supplies, but adopting more smaller cargo containers for DroneGo disaster response system transportation, and choose 12 scattered airports, presented in Figure $15(b)$ to serve as supply stations. In this way, drone fleets are capable of covering around 95% area of the island. Additionally, we place tethered drone H at 6 locations to serve as a communication base station to provide instant mobile network coverage with a 50 $Km$ radius. The communication network covers the entire disaster area, enabling drone B for medical packing delivery and road reconnaissance.

Thanks again for trusting us. We hope that our suggestions are useful for your company.

Sincerely,

**Team 1919009**

# References


[1] DJI. Technical parameters for dji. `https://www.dji.com/cn/products/professional?site=brandsite&from=footer`.

[2] Kevin Dorling, Jordan Heinrichs, Geoffrey G Messier, and Sebastian Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, 2017.

[3] Department for Transport. Guidance on road classification and the primary route network, 2012.

[4] R Gitzendanner, F Puglia, C Martin, D Carmen, E Jones, and S Eaves. High power and high energy lithium-ion batteries for under-water applications. *Journal of power sources*, 136(2):416–418, 2004.

[5] Oscar Lewis et al. *La vida: A Puerto Rican family in the culture of poverty-San Juan and New York*, volume 13. Random House New York, 1966.

[6] Francesco Marinello, Andrea Pezzuolo, Alessandro Chiumenti, and Luigi Sartori. Technical analysis of unmanned aerial vehicles (drones) for agricultural applications. *Engineering for Rural Development*, 15, 2016.

[7] Mary-Ann Russon. Drones to the rescue![online]. `https://www.bbc.com/news/business-43906846`, May 2018.

[8] Time and Date. Sunrise and sunset. `https://www.timeanddate.com/sun/@4566966`.

[9] World Vision. Rush your support[online]. `https://www.worldvision.in/WaysToGive/emergency-relief.aspx`.

[10] LIN Zongjian. Uav for mapping—low altitude photogrammetric survey. *International Archives of Photogrammetry and Remote Sensing, Beijing, China*, 37:1183–1186, 2008.