Hurricane Maria struck in 2017 becoming the worst hurricane to ever hit the island of Puerto Rico. It cost damage to infrastructure, had severe inclement weather, disrupted communication networks, and hindered emergency efforts on roads and highways. Using this disastrous scenario as a model, HELP, Inc. seeks to be better prepared to deliver medical supplies and obtain video surveillance of the affected areas to provide efficient relief. In this paper, we explore this problem using prepackaged medical supplies and drones provided by the organization.

To investigate the most effective way of packing the materials into an International Standards Organization (ISO) cargo container, the Less Flexibility Principle was used. The principle allowed the problem to be approached with logic despite being heuristics. In this way we were able to determine an optimal way of packing the ISO container with the necessary supplies. Although HELP, Inc. gave the option of 2 cargo bays and 8 drones, our findings concluded that using only cargo bay type 1, a H drone, and B drones for deliveries and surveillance was the optimal solution for accomplishing the appointed objectives. The ISO containers are dropped at Latitude and Longitude coordinates determined using spherical geometry and a maximum sum decision model. This model is meant to optimize surveillance radius while ensuring that all deliveries will be made.

In determining a way to provide surveillance of the area, our flight plans were modelled after N-leaf rose pattern in a polar graph. By allowing the “petals” to start and end at our designated ISO drop location, the pattern was able to be mimicked by using way points on the longitude-latitude coordinate system. The allows the drones to not only survey the roads and highways, as desired by the organization, but to also survey much of the surrounding area which also allows emergency efforts to help victims who might not have receive aid.

In addition to the models used, we explore the different potential information technology issues and solutions that would help better prepare for an efficient and rapid response to future incidents. Considerations regarding the amount of data collected and how it will be processed will be made utilizing deep learning methodologies.

Table of Contents

1. Restatement and clarification of the problem

The problem has our team tasked with assisting HELP, Inc. in designing a drone based disaster response system using the 2017 hurricane situation in Puerto Rico. This response system, called “DroneGo”, will use drones suggested from the company to provide both pre-packaged medical assistance to hospitals and video reconnaissance to help emergency services on the ground in planning their routes. HELP, Inc. has planned 3 different medical packages referred to as MED 1, MED 2, and MED 3 that will be delivered to 5 selected hospitals/medical centers using these drones. Considering each locations anticipated demand we will determine what package(s) are to be loaded on a cargo bay, of which are 2 types, that will be delivered to the hospitals. Along with these deliveries, the drones will be providing video of major roads and highways to assess the conditions. This high-resolution video will go to HELP, Inc.’s command and control center where it will then be used for planning ground based operational routes.

With these facts in mind our team is to determine which drones are to be selected to use in the DroneGo fleet and also provide a schedule of when they will make deliveries and take video of the area. Once we decide on the cargo bays with the medial packages and the drone fleet, we then need to figure out the packing configuration of the DroneGo system into International Standards Organization (ISO) cargo containers. Up to 3 containers can be dropped in up to 3 locations in Puerto Rico. In doing so we want to maximize the storage space so that minimal buffer material is needed. If it’s found that the demand exceeds the drone’s capabilities then we must inform HELP, Inc. of ways to address any shortcomings.

2. Explain assumptions and rationale/justification

One of the first assumptions that we need is that for every pound of cargo the drone would lose 0.5 minutes of flight time. This arbitrary yet reasonable assumption was used to determine how far our drone’s would fly. Additionally, we assume the speed will remain constant due the modern drone’s ability to stabilize on it’s own in inclement weather. We also assume that the drones can avoid objects because the standard drones for commercial use are able to.

Essential to the scheduling of the drones is having knowledge of the available daylight throughout the day and how much time is needed to recharge the drone. This is where we assume 12 hours of daylight in Puerto Rico[10] and an approximate charge time of 2 hours.

2.1 Individual Drone Survey techniques

Using the camera from the DJI Mavic Pro Platinum for reference, we can have an idea of how we can effectively collect data on the status of the roads. This camera’s specifications are standard within the UAS industry and can provide high quality imagery for analytics. To get optimal ground coverage, the drones will be surveying at an altitude of 200 meters AGL. With the camera and altitude we calculate that each frame captured by the drone will cover 96 meters by 144 meters of the ground. As the drone travels forward at an assumed maximum speed for 79 kph, we can calculate the optimal image capture rate. With a 50% image overlap to minimize error rate, each drone will be programmed to collect 1 image every 2.18 seconds. With the reference camera, the image produced will be high quality with 12.1 megapixel resolution. This image along with flight telemetry such as GPS location would be transmitted back to the ISO containers for immediate analysis.

The Photographic Footprint of a Camera on a Drone

variables		
xsensor	36	width of sensor in mm
ysensor	24	height of sensor in mm
focallen	50	focal length of lens in mm
altitude	200	height in m
xgimbal	0	x-axis gimbal angle
ygimbal	0	y-axis gimbal angle

Field of view wide: $2\tan^{-1}\left(\frac{36}{2\times 50}\right) = 39.6°$

Field of view tall: $2\tan^{-1}\left(\frac{24}{2\times 50}\right) = 26.99°$

From drone to bottom of picture: $200 \times \tan\left(0 - \frac{1}{2} \times 39.6\right) = -72.0m$

From drone to top of picture: $200 \times \tan\left(0 + \frac{1}{2} \times 39.6\right) = 72.0m$

From drone to left of picture: $200 \times \tan\left(0 - \frac{1}{2} \times 26.99\right) = -48.0m$

From drone to right of picture: $200 \times \tan\left(0 + \frac{1}{2} \times 26.99\right) = 48.0m$

Height of photo footprint: $48.0 - -48.0 = 96.0m$

Width of photo footprint: $72.0 - -72.0 = 144.0m$

[7] Photographic footprint of a camera on a drone, Stack Exchange.

2.2 Data Gathering

The drones being deployed will all be gathering raw image data and flight telemetry during their flights on both delivery and survey missions. To maximize are use of drones we are assuming 12 hours of daylight in Puerto Rico. This is typical in the area except during the winter months, only outside of the hurricane season. Our flight plans for the surveying will utilize the maximum flight time possible. We also assume that there will be a 2 hour period to recharge the battery on the drone. With this information we determined that each drone is capable of completing 5 missions each day.

Our drone fleet is comprised of 78 B-type drones and will be conducting continuous missions throughout daytime hours. The fleet is capable of completing 390 missions per day. Each drone will produce approximately 24,750 Mb of raw image data each day based on the 12.1 megapixel resolution and 2 second image capture frequency. Assuming a daily fleet accumulation of 500,000 Mb of flight telemetry, we will be producing 2.42 Tb of raw data per day of operation. This data will be transmitted back to to our ISO cargo containers with the aid of the tethered H-drone with the mesh network at these locations.

An important consideration to make would be coordination with the FAA as well as any ongoing rescue operations as these drones would be operating in class E airspace and would be operating outside of typical legal limits [9]. Steps would need to be taken to ensure that drone presence is communicated and properly avoided.

3. Model design and justification

The problem calls for drones delivering medical packages and providing video surveillance and then packing everything into a ISO container. Our solution involves 2 models to properly represent a solution for each objective. To pack materials as efficiently as possible, we have provided a physical model. And to determine the flight plans for the drones a mathematical model was used.

In our search to optimize filling a rectangular prism with smaller rectangular prisms we found an article explaining a way in which you might do this. "An effective quasi-human based heuristic for solving the rectangle packing problem" by Yu-Liang Wu et al. explains the process that was accepted for thousands of years from use however there wasn't a mathematical way to show the optimization of a packing space. This article however only showed how to use what they called the Less Flexibility Principle to pack a 2 dimensional rectangle instead of a prism which is what we need. Since this packing problem is a heuristic problem in 3 dimensional space, we felt that a physical model would be more efficient.

Conversely, a mathematical model was used for the drop locations, flight plans, and scheduling so the drones could be as autonomous as possible. We also chose this type of model in order to consider the longitudes and latitudes of locations as accurately as possible. To use a physical method would involve a lot of tedious work by hand and would have left a lot of room for error. Using a mathematical model would allow for a more precise longitude and latitude position that is essential for efficient operation planning.

3.1 Describe model and sensitivity analysis

As stated before, we used a physical model to optimize the packing space of the cargo bay and ISO container. We first analyzed Cargo Bay Type II due to its larger dimensions. This was quickly determined to be inefficient since at maximum capacity it was too heavy to be carried by any of the drones provided in the list. By choosing to use only Cargo Bay Type 1 we found the possible packing configurations of MED 1, 2, and 3. Our result was the following table:

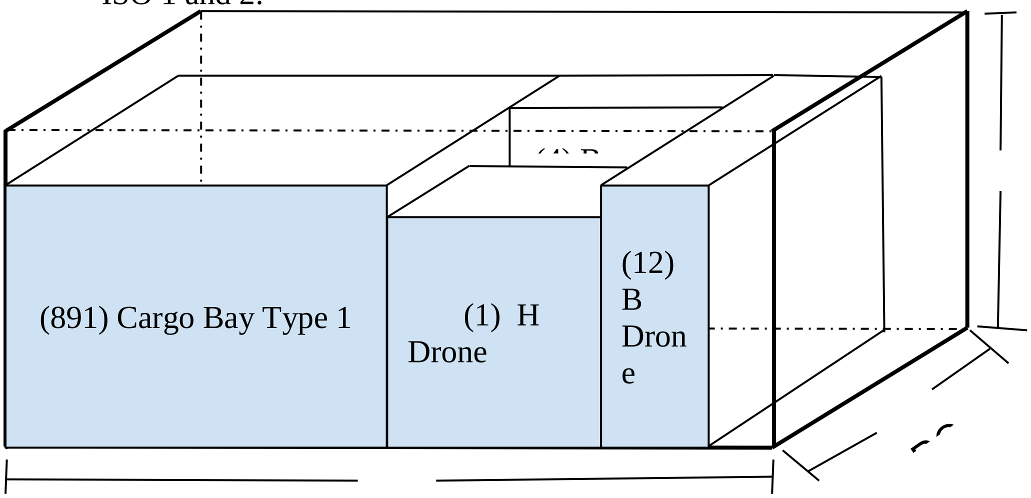
Cargo 1 Configurations	
Medical Packs	Weight (lbs)
1,2,2	6
1,2	4
1,1	4
2,2,2,2	8
3,3	6

With this data, only drones B and D could be used to make these deliveries; the A drone would not be able to carry the weight of any fully packed cargo bay. The B-Drone was selected as the ideal drone for delivery and surveillance. While it has a modest payload capacity of 8lbs, it is more than sufficient to deliver the required daily medical supplies. It has the highest speed of 79 kph, longest flight time of 40 minutes, and can cover a distance of 52.6 km in a single battery life. B-Drone will be able to cover enough ground to collect data effectively and efficiently.

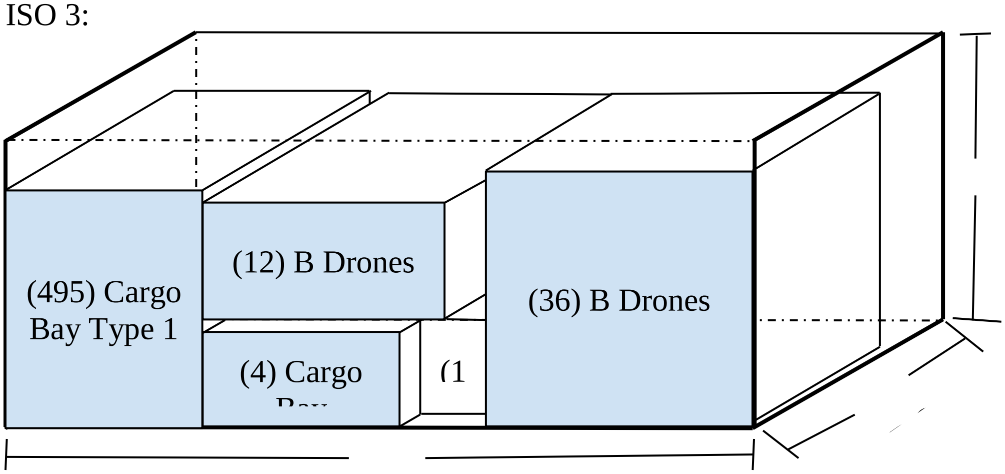
3.2 ISO Packing Configurations

The knowledge of what to pack inside of the ISO allowed us to easily configure the packing of the container efficiently. ISOs 1 and 2 are packed exactly the same while ISO 3 is different although the same logic was used. It was done by creating the largest rectangular prisms of cargo bay type 1 and B drones that would fit with a H drone container in between them. Any empty space was filled with additional cargo bays and drones until the space was optimized. The logic for our packing configurations was modeled following the basics of the Less Flexibility Principle by filling the corners then sides and finally using trial and error to minimize the remaining emptiness. Our ISO containers would look like the following:

ISO 1 and 2:



ISO 3:



The cargo bays would be packed in a way that corresponds to the following tables. ISO 1 will contain 356 of cargo bays packed with the (3,3) configuration and 535 packed with the (1,1) configuration. ISO 2 will contain 89 cargo bays with the (2,2,2,2) configuration, 356 cargo bays packed with the (1,1) configuration, 178 cargo bays packed with configuration (3,3) and 268 cargo bays with configuration (1,2,2). Finally, ISO 3 will contain 499 cargo bays with the configuration (1,1).

ISO 1		
	Fajardo / San Pablo	
	Day 1	Day 2
Trip 1	3,3	1,1
Trip 2	1,1	3,3
Trip 3	1,1	Survey
Trip 4	Survey	Survey
Trip 5	Survey	Survey

ISO 3		
	Arecibo	
	Day 1	Day 2
Trip 1	1,1	Survey
Trip 2	1,1	Survey
Trip 3	Survey	Survey
Trip 4	Survey	Survey
Trip 5	Survey	Survey

ISO 2					
	Bayamon			San Juan	
	Day 1	Day 2		Day 1	Day 2
Trip 1	1,1	1,1		1,2,2	Survey
Trip 2	3,3	3,3		1,1	Survey
Trip 3	2,2,2,2	1,1		1,2,2	Survey
Trip 4	Survey	3,3		Survey	Survey
Trip 5	Survey	Survey		Survey	Survey

3.3

Determining ISO Drop Locations

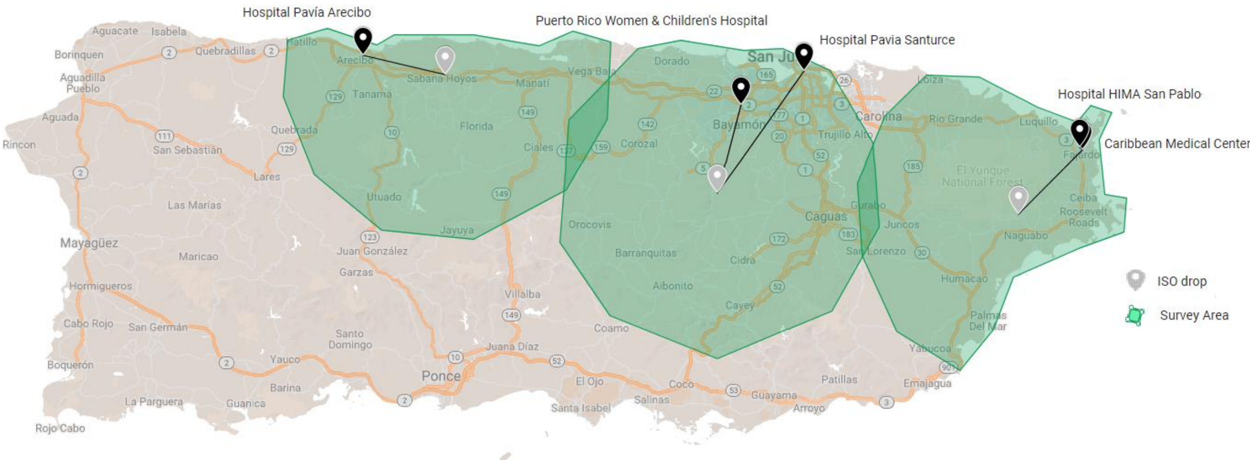
The drop locations are dependent upon the range of B drone. Utilizing:

$$Range_{\ Under\ Load}\,(km) = Speed\ (km/h) * ((Flight\ Time(min) - Payload(lbs) * 0.5(min))\ /\ 60.0(min))$$

the drone could make a trip of 23.7 km under its maximum load of 8 pounds. ISO cargo container locations were selected using a python program that would take the locations of the medical centers and the max drone range under load to find points on the outer boundaries. The goal was to find a point towards the center of the island that would be just close enough to be in range of the medical centers. This would allow for a larger search area for the B drones and give optimal communications coverage for the H drones. Since this needs to be purposeful on a latitude-longitude coordinate system, the Haversine Formula

$$hav(\theta) = hav(\varphi_2 - \varphi_1) + cos(\varphi_1)cos(\varphi_2)hav(\lambda_2 - \lambda_1)$$

was used to measure the distances between two points where φ_1, φ_2 is the latitude of point 1 and latitude of point 2 and λ_1, λ_2longitude of point 1 and longitude of point 2. These methods resulted in ideal locations to drop the ISO containers and the area in which the flight plans could be constructed. The following map is a visual representation of this using Google Maps software:



The black points are the selected medical locations. From right to left: Caribbean Medical Center, Hospital HIMA, Hospital Pavia, Puerto Rico's Children's Hospital, Hospital Pavia Arecibo. The gray points are the determined ISO drop locations. From right to left: ISO 1, ISO 2, ISO 3.

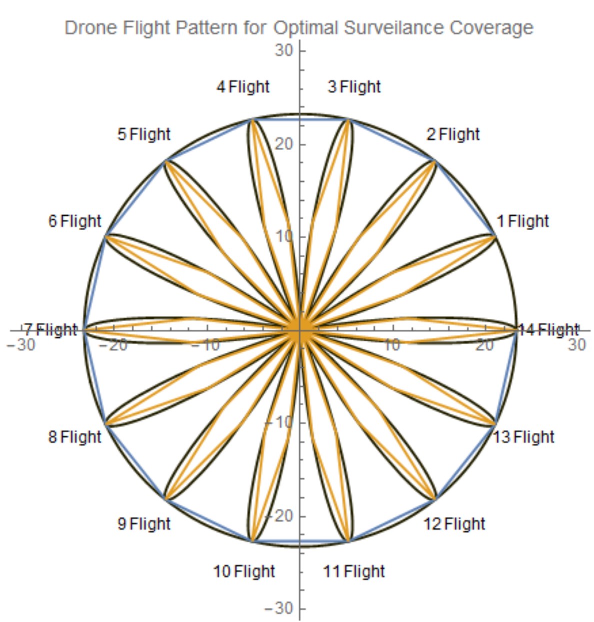
3.4 Video Reconnaissance

For medical deliveries, the flight plan follows the schedule of the previous tables for each ISO respectively. This schedule will be followed on a Day 1, Day 2 rotation schedule to supply the medical centers with their anticipated demand then surveying designated areas to provide video reconnaissance. Drones conducting surveys will be queued to follow waypoints approximating the pattern of a 14-petaled polar rose with a period 25.714° (see figure). This pattern was chosen due to the urgency of disaster area surveillance, and the availability of capable drones in the fleet, to maximize the area covered, particularly in urban regions.

Polar roses can be plotted analytically as $r(\theta) = m(1 + cos(k\theta)$,where $m = r_{max}/2$, k is the number of petals and θ, in radians, is the angle swept by the curve. The drone fleet is able to cover an arc length of 52.6 km, calculated via the methods above, during its battery life. In order to compensate for events negatively affecting battery life, such as path deviations and strong winds, the survey path arc length was 90% of drone capability. The value of m was determined by utilizing the arc length formula for polar functions,

$$L = \int_{\frac{\pi}{14}}^{\frac{3\pi}{14}} \sqrt{r(\theta)^2 + (\frac{dr}{d\theta})^2}\, d\theta.$$

The bounds for this arc length were selected based on the the period of the curve as well as the chosen value of k. This integral and pattern was



evaluated by utilizing the computing environment *Mathematica (*See Appendix B), since the solution had to be found numerically, yielding a value of $m = 11.6367.$. Error in this numerical approximation was ignored due to the 10% difference in arc length from the maximum. Each drone flight will consist of four waypoints which lie on the polar curve separated by an angle of 6.428°. Upon completion of all 14 flights in the job queue for the drone fleet the pattern will be rotated counterclockwise by 6.428° and the job queue for the new adjusted waypoints will open.

The waypoint Latitude/Longitude coordinates were calculated using a Python code that utilizes the Haversine formula and spherical trigonometry (see Appendix C). The formulas used were

$$\varphi_2 = asin(sin\ \varphi_1 cos\ \delta\ +\ cos\ \varphi_1 sin\ \delta cos\theta)\ ,$$

and

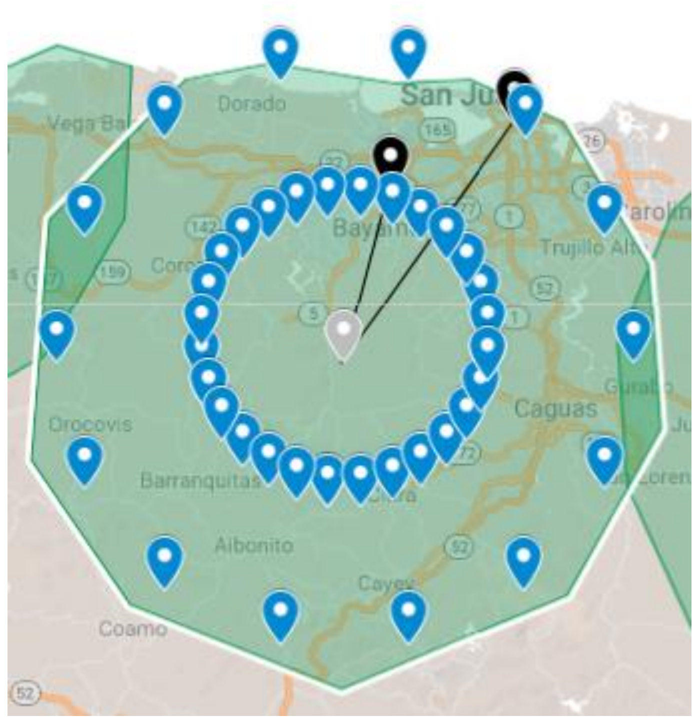
$$\lambda_2 = \lambda_1 + atan2(sin\ \theta sin\ \delta cos\ \varphi_1, cos\ \delta - sin\ \varphi_1 sin\ \varphi_2.$$

Here, $\varphi$is latitude, λ is longitude, and $\theta$is the bearing of drone from the ISO drop location. Furthermore, $\delta$is the ratio of the distance across the earth's surface to the radius of the earth [15]. The results of our code can be shown seen on the map below where each blue marker represents a waypoint spaced at 6.428°.

3.5 Job Queue:

Each day 1 drone will be assigned to a medical center and will deliver supplies, following the medical supplies delivery schedule, until completed. If it completed less than 5 trips, the drone can join the job queue to begin surveying. Each survey mission will be listed in a queue. This will determine which jobs the drones will do. This will be generated based on the algorithms used above. Once a drone is charged and idle, it will pull the next job from the list until it is night time.

Code was created in python to take the mathematical formula from the previous sections and converts points in the pedals to latitude / longitude waypoints that the drones can be programed to follow. Each mission will consist of 4 waypoints after a successful launch. See Appendix C and D for details.



4. Model strengths and weaknesses

Image of the first wave of drone missions.

4.1 Networking

Using the 2017 disaster as a basis, the island was without power and cell communication for months and this definitely hinders emergency efforts. To prevent this in future situations such as this one, an H drone is also included inside of each ISO container. The tethered drone would

provide a way for HELP, Inc. to communicate with ground services about optimal routes to conduct rescue missions and supply aide to areas of need. This drone has the opportunity to work in a multitude of roles. Primarily they will be used to stay in communication with our drone fleet. This drone will mitigate any interference from ground obstructions and allow for omnidirectional drone missions. Transmitters and receivers would need to be able to handle up to 36 connections at a given time. This would be possible utilizing a wireless mesh network [11]. This would be designed using a special routing technique to funnel data gathered by B-drones back towards the tethered H-drone and back down to the ISO container. The combination of the mesh network with the H-drone is ideal because it is a very energy efficient method with multiple opportunities for data error checking and has sufficient range for our application [11].
H-Drones would be equipped with LTE radios and antennas. This device could provide temporary voice, data, and internet service around the ISO containers. The system deployed in Puerto Rico in 2017 by AT&T included a single tethered drone was able to provide coverage to up to 8,000 people simultaneously over a 40 square mile area [12]. With three of these drones, the system would support 24,000 devices in the most densely populated areas in the region. Getting this communication network up and running is critical in getting emergency services to be aware of needs on the ground around the island.

4.2 Information Management and Analytics:
Due to the enormous amount of data being generated it will be important to take into consideration how much data can be transmitted for analysis. Since communication networks on the island are expected to be down during the emergency, It would be possible to upload the information over a satellite connection. However, uplink speeds to this network cap out at around 3 Mbps [13]. Even if this speed was increased to 10Mbps, it would take nearly 600 hours to upload one day's worth of data gathered in this model. This solution is not viable as it does not meet our needs to quickly take this data and turn it into meaningful information.

A possible solution would be including a information system within a single ISO cargo container. All of the data generated by the drones would be sent to a single location and analysis would happen though a deep learning model that has already been trained to detect anomalies in the road system. This model would follow the architecture of a convolutional neural network (CNN) [4]. Unlike a simple neural network, this type of architecture is able to take the images collected from the drone as input. The high-speed connectivity between the drones, facilitated by the mesh network, would be able to constantly supply the CNN with data.

With this solution it would be possible for system monitoring from a remote location. When the CNN detects an anomaly it can then transmit the image, as well as GPS location data. This system would also eliminate the need to parse image data collected from the drone since they will be following non-conventional flight waypoints. This would result in the limited bandwidth available to only transmit critical information as it is discovered. This would also leave satellite bandwidth open to other responders in the area.

4.3 Power consumption

An important consideration to make in this model would be how power is provided to enable the functionality of this system. Due to the time limitations of this project, we did not consider including batteries, generators, or other methods of producing power inside of the ISO containers. It would be possible to include these items into the containers, but we would need more information regarding the specific consumption of the drones in order to efficiently prepare to meet those needs.

Memo **HELP, Inc.**

To: Chief Operating Officer (CEO) of HELP, Inc.
From: DroneGo project workers
Subject: Result of the DroneGo project
Date: January 29, 2019

Message

The assigned workers of the DroneGo project have assessed the situation and can say that some results have been found and conclusions have been made. Based on our models, we have determined that dropping 1 container in 3 locations would sufficiently fulfill the requirements for an automated disaster relief drone fleet. Further considerations are necessary pending project approval. By dropping ISO 1 at [18.24055, -65.74596], ISO 2 at [18.271, -66.19904], and ISO 3 at [18.43874, -66.60805], HELP, Inc. will be able to deliver the medical packages while getting maximum coverage for video surveillance, of which we suggest using only the B drone.

It is recommended that for this to be effective personnel must be stationed on the ground to unload the containers and attach and detach the cargo bays to and from the drones. However while these actions require personnel, the drones flight plans and delivery schedule are autonomous. With the implementation of our recommended flight schedule program, the drones will be able to survey 75% of the eastern half of the island in 5 days time utilizing a polar 14-petaled rose pattern which is able to sweep a radius of 23.2735 km with maximum efficiency. In addition, all hospital deliveries needs will be met well in advance of the requested daily deliveries. Particularly, the hospitals in Bayamon, San Juan and Arecibo will receive surplus supplies which could provide a fail safe for any technical difficulties or time frame constraints. This could prove to be an essential aspect of our system, especially in urban environments.

We have also concluded that using only Cargo Bay Type 1 should be used since the burden of a fully packed Cargo Bay Type II could not be lifted by any of the drones we had considered. In total, 891 Cargo Bay Type I can be shipped in ISO’s 1 and 2, while ISO 3 will carry 499 cargo bays due to its reduced need at the drop location. Each ISO container will be equipped 1 H drone to allow the implementation of wireless mesh network which will enhance ground communication and increase drone fleet data transfer and synchronization capabilities.

Reference List

[1] "Topographic Map Puerto Rico". *Topographic-Map.Com*,http://en-us.topographic-map.com/places/Puerto-Rico-797932/.

[2] Allain, Rhett et al. "Physics Of The Amazon Octocopter Drone". *WIRED*, 2013, https://www.wired.com/2013/12/physics-of-the-amazon-prime-air-drone/.

[3] "Humanitarian In The Sky: Drones For Disaster Response". *Virgin*, 2019, https://www.virgin.com/virgin-unite/business-innovation/humanitarian-sky-drones-disaster-response.

[4] Karunakaren, Dhanoop. "Simple Image Classification Using Deep Learning — Deep Learning Series 2". *Medium*, 2018, https://medium.com/intro-to-artificial-intelligence/simple-image-classification-using-deep-learning-deep-learning-series-2-5e5b89e97926.

[5] Wu, Yu-Liang et al. "An Effective Quasi-Human Based Heuristic For Solving The Rectangle Packing Problem". *European Journal Of Operational Research*, vol 141, no. 2, 2002, pp. 341-358. *Elsevier BV*, doi:10.1016/s0377-2217(02)00129-7.

[6] "Purchasing A Sensor – What You Need To Know | Crop Copter". *Crop Copter.Co*, http://www.cropcopter.co/purchasing-a-sensor-what-you-need-to-know/.

[7] Zimmerman, Ryan. “The Photographic Footprint of a Camera on a Drone”. *StackExchange*, November 12, 2014. https://photo.stackexchange.com/questions/56596/how-do-i-calculate-the-ground-footprint-of-an-aerial-camera

[8] Zimmerman, Ryan. “dronecamerafov”. January, 31, 2015. https://github.com/mountainunicycler/dronecamerafov/tree/master

[9] "DJI Mavic Pro Platinum – Specs, Tutorials & Guides – DJI". *DJI Official*, 2018, https://www.dji.com/mavic-pro-platinum/info.

[10] "Sunrise And Sunset Times In Puerto Rico". *Timeanddate.Com*, 2019, https://www.timeanddate.com/sun/@4566966.

[11] Sempf, Aaron. “Swarm communication”. April 20, 2015. medium.com/cælus-concept/swarm-communication-33cffc47db6d

[12] Brodkin, Jon. “An AT&T drone is now providing cellular service to people in Puerto Rico”. November 11, 2017. https://arstechnica.com/information-technology/2017/11/att-drone-brings-lte-access-to-hurricane-damaged-puerto-rico/

[13] ”Questions and Answers.” *HughesNet*, 2019.

[14] Wikipedia contributors. "Haversine formula." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 6 Jan. 2019. Web. 28 Jan. 2019.

[15] Veness, Chris. “Calculate distance, bearing and more between Latitude/Longitude points.” 2017. http://www.movable-type.co.uk/scripts/latlong.html

Appendix A

This code was used to find the ideal locations of the ISO cargo containers. The Haversine formula calculates the distance between two latitude / longitude pairs. The medical centers and a search area is defined. All possible landing locations are gathered in a list. This list is then conditionally filtered to find the max sum of distances from the ISO container to 2 medical center locations. The output was then tweaked manually to satisfy the need for a flat and open area for a drone fleet to launch from.

```python
import math
from math import *
from numpy import *
import pandas as pd
import numpy as np

# Haversine Formula:
def haversine(lon1, lat1, lon2, lat2):
    try:
        # convert decimal degrees to radians
        lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])

        # haversine formula
        dlon = lon2 - lon1
        dlat = lat2 - lat1
        a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
        c = 2 * atan2(sqrt(a), sqrt(1-a))
        r = 6371 # 6371 Radius of earth in kilometers.
        return c * r
    except:
        return -1

# Hospitals: long / Lat:
Fajardo = [-65.65012, 18.33211]     # These have been updated from what was given
San_Pablo = [-65.65485, 18.33673]
San_Juan = [-66.06898, 18.44424]
Bayamon =  [-66.16327, 18.39657]
Arecibo = [-66.73172, 18.46705]
```

```
# Drones V/MPC/Speed (km/h)/FT no cargo/Range no cargo/ Range max cargo
DroneB = [19800.0, 8.0, 79.0, 40.0, 52.667,47.4]

# Area Search for San Juan and Bayamon: Long / Lat
NW1 = [-66.210, 18.455]
SW1 = [-66.210, 18.267]
NE1 = [-65.960, 18.455]
SE1 = [-65.690, 18.267]
Point1 = [-66.210, 18.267]

#Loop for San Juan and Bayamon
i = Point1[0]
j = Point1[1]

Haver_out1 = pd.DataFrame(columns = ['Lats','Lons','San Juan','Bayamon','Valid Points'])
san_juan = []
bayamon = []
lats1 = []
lons1 = []
for i in np.arange(NW1[0], NE1[0],0.001):
    for j in np.arange(SE1[1], NE1[1],0.001):
        r_1 = haversine(float(San_Juan[0]), float(San_Juan[1]), i, j)
        r_2 = haversine(float(Bayamon[0]), float(Bayamon[1]), i, j)
        san_juan.append(r_1)
        bayamon.append(r_2)
        lats1.append(j)
        lons1.append(i)
Haver_out1['Lats'] = lats1
Haver_out1['Lons'] = lons1
Haver_out1['San Juan'] = san_juan
Haver_out1['Bayamon'] = bayamon
```

```
#loop to get valid points
k = 0
Vpoints =[]

for k in range(len(Haver_out1)):
    if DroneB[5]/2 - Haver_out1.iloc[k]['San Juan'] >= 0 and DroneB[5]/2 - Haver_out1.iloc[k]['Bayamon'] >=0:
        Vpoints.append([Haver_out1.iloc[k]['Lats'], Haver_out1.iloc[k]['Lons']])
    else:
        Vpoints.append(np.nan)

Haver_out1['Valid Points'] = Vpoints
Haver_out1 = Haver_out1.dropna()

# loop to get max sum of distances to get best drop locations
Distance = 0
best_index = []

for k in range(len(Haver_out1)):
    Distance_calc = Haver_out1.iloc[k]['San Juan'] + Haver_out1.iloc[k]['Bayamon']
    if Distance_calc > Distance and Distance_calc >= 38: #and Haver_out1.iloc[0]['San Juan']:
        Distance = Distance_calc
        best_index = Haver_out1.iloc[k]
        print best_index
```

Appendix B

Mathematica Code for Optimal Drone Surveillance Coverage

```
In[177]:= Clear["`*"]
```

Launching synchronized drone flights that have a 14-petaled pattern (one for each of our available surveillance drones) the polar function must be:

```
r[θ_, m_] := m + m Cos[14 (θ)];
```

Since we know the maximum flight distance for our drone, 52.6, and and the number of petals we can find the maximum arc length of the total 3-day flight plan as 14*52.6 = 736.4. Using the Arclength formula for polar coordinates and the fact that the period of this will be $\pi/7$ we have 52.6 = $\int_{\pi/14}^{3\pi/14} \sqrt{r(\theta)^2 + \left(\frac{dr}{d\theta}\right)^2}\, d\theta$. So we need the derivative, $r'(\theta)$.

```
rp[θ_, m_] := D[r[θ, m], θ]
```

Now we can solve for the value of m with a 10% flight adjustments and error allowance

```
In[183]:= SetPrecision[Solve[Integrate[√((r[θ, m])^2 + (rp[θ, m])^2), {θ, π/14, 3π/14}] == .9*52.6, m, Reals], 10]
```

```
Solve: Solve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.
```

```
Out[183]= {{m → -11.63673284}, {m → 11.63673284}}

In[184]:= m = 11.63673284

Out[184]= 11.6367
```

Check the maximum radius covered and values of waypoints.

```
In[324]:= r[π/7, m]
          r[π/28, m]
          π/28 180/π // N

Out[324]= 23.2735

Out[325]= 11.6367

Out[326]= 6.42857
```

Knowing the value of m we can plot the function to get a visual

```
p1 = PolarPlot[{r[θ, m], r[π/7, m]}, {θ, 0, 2π}, PlotStyle → ■, PlotLabel → "Drone Flight Pattern for Optimal Surveilance Coverage",
   PlotRange → 30];

p2 = ListPolarPlot[{Table[{θ, r[θ, m]}, {θ, π/7, 14π/7, π/7}], Table[{θ, r[θ, m]}, {θ, 0, 56π/28, π/28}]}, Joined → True];


Do[g[i] = Graphics[Text[ToString[i] Flight, {(r[π/7, m] + 3.5) * Cos[i*π/7], (r[π/7, m] + 3.5) * Sin[i*π/7]}]], {i, 1, 14}];

Show[{p1, p2, g[1], g[2], g[3], g[4], g[5], g[6], g[7], g[8], g[9], g[10], g[11], g[12], g[13], g[14]}]
```

Waypoint Lat/Lon coordinates determined using python code.

Appendix C

```
import math
from math import *
from numpy import *
import pandas as pd
import numpy as np

R = 6371       # km Radius, do not change!!
d = 23.2735    # km Distance to find waypoint 2
d1 = 11.637    # km distance to find waypoint 2 and 3
b1 = pi/2 + pi/28 #initial bearing

#Creating waypoints and lists for looping
waypoint_1 = [0,0]
waypoint_2 = [0,0]
waypoint_3 = [0,0]
waypoint_4 = [0,0]

list_waypoint_1 =[]
list_waypoint_2 =[]
list_waypoint_3 =[]
list_waypoint_4 =[]

#ISO Drop locations:
ISO2 = [18.271 , -66.1990]
# ISO1 = [18.24055, -65.74596]
# ISO3 = [18.43874, -66.60805]

# converting ISO degrees to radians:
ISO2_lat_rad = ISO2[0] * pi / 180
ISO2_lon_rad = ISO2[1] * pi / 180
ISO2_rad = [ISO2_lat_rad, ISO2_lon_rad]
```

```
# loop to get all waypoints on flight path
i = 0
while i <= 14:    # loop to iterate 14 times
    i = i + 1

    #WAYPOINT #1
    d1 = 11.637
    waypoint_1[0] = np.arcsin(np.sin(ISO2_rad[0]) * np.cos(d1/R) + np.cos(ISO2_rad[0]) * np.sin(d1/R) * np.cos(b1))
    waypoint_1[1] = ISO2_rad[1] + np.arctan2(sin(b1) *  np.sin(d1/R) *  np.cos(ISO2_rad[0]),  np.cos(d1/R) - np.sin(ISO2_rad[0]) *  np.sin(waypoint_1[0]))
    #convert back to decimal
    waypoint_1[0] = waypoint_1[0] * 180 / pi
    waypoint_1[1] = waypoint_1[1] * 180 / pi
    # Append to list
    list_waypoint_1.append([waypoint_1[0],waypoint_1[1]])
    #Update Bearing
    b1 = b1 - (pi/28)

    # WAYPOINT #2
    waypoint_2[0] = np.arcsin(np.sin(ISO2_rad[0]) * np.cos(d/R) + np.cos(ISO2_rad[0]) * np.sin(d/R) * np.cos(b1))
    waypoint_2[1] = ISO2_rad[1] + np.arctan2(sin(b1) *  np.sin(d/R) *  np.cos(ISO2_rad[0]),  np.cos(d/R) - np.sin(ISO2_rad[0]) *  np.sin(waypoint_2[0]))
    # convert back to decimal
    waypoint_2[0] = waypoint_2[0] * 180 / pi
    waypoint_2[1] = waypoint_2[1] * 180 / pi
    # Append to list
    list_waypoint_2.append([waypoint_2[0],waypoint_2[1]])
    #Update Bearing
    b1 = b1 - (pi/28)
```

```
    #WAYPOINT #3
    waypoint_3[0] = np.arcsin(np.sin(ISO2_rad[0]) * np.cos(d1/R) + np.cos(ISO2_rad[0]) * np.sin(d1/R) * np.cos(b1))
    waypoint_3[1] = ISO2_rad[1] + np.arctan2(sin(b1) *  np.sin(d1/R) *  np.cos(ISO2_rad[0]),  np.cos(d1/R) - np.sin(ISO2_rad[0]) *  np.sin(waypoint_3[0]))
    # convert back to decimal
    waypoint_3[0] = waypoint_3[0] * 180 / pi
    waypoint_3[1] = waypoint_3[1] * 180 / pi
    #Append to list
    list_waypoint_3.append([waypoint_3[0],waypoint_3[1]])
    #Update Bearing
    b1 = b1 - (pi/28)

    #WAYPOINT #4
    waypoint_4[0] = ISO2[0]
    waypoint_4[1] = ISO2[1]
    list_waypoint_4.append([waypoint_4[0],waypoint_4[1]])

    #Update Bearing
    b1 = b1 - (pi/28)
#end loop

#Add lists to dataframe
mission_list = pd.DataFrame(columns =['waypoint 1', 'waypoint 2', 'waypoint 3', 'waypoint 4'])
mission_list['waypoint 1'] = list_waypoint_1
mission_list['waypoint 2'] = list_waypoint_2
mission_list['waypoint 3'] = list_waypoint_3
mission_list['waypoint 4'] = list_waypoint_4

print mission_list
```

Appendix D

This is an example of mission waypoints for drones launched from ISO container 2 generated by the Python code in Appendix C. These waypoints would be grouped into missions based on the index of the lists. During deployment, more waypoints would be generated based on the mathematical model from Appendix B. Waypoint 4 is the return-to-home location of the ISO container.

```
                                waypoint 1
0   [18.259251311806274, -66.08948999422394]
1     [18.3055369068776, -66.09495352399853]
2   [18.344985786919807, -66.12103620092186]
3    [18.36977776802263, -66.16257904858679]
4   [18.374995615998742, -66.21134706135042]
5    [18.359604170488176, -66.2576654637138]
6    [18.32665666116626, -66.29234777648921]
7    [18.28268636767381, -66.30852479504469]
8    [18.236406865907522, -66.3030050373753]
9   [18.196982657189288, -66.27689733657648]
10    [18.17221534746747, -66.2353795126554]
11  [18.167003592828046, -66.18666772797681]
12  [18.182377965197393, -66.14039441698382]
13  [18.215298091383918, -66.10571210410285]
14  [18.259251311806274, -66.08948999422394]
```

```
                                waypoint 2
0     [18.27087378218598, -65.97858405961318]
1   [18.361710760166982, -66.00030794078307]
2   [18.434590900823654, -66.06144278523006]
3      [18.4750496557258, -66.14989486926014]
4      [18.4750496557258, -66.24810513073984]
5   [18.434590900823654, -66.33655721476993]
6   [18.361710760166982, -66.39769205921692]
7     [18.27087378218598, -66.41941594038681]
8   [18.180084326042916, -66.39748421894912]
9   [18.107310966758764, -66.33629804041321]
10    [18.06693784464067, -66.24798978666405]
11    [18.06693784464067, -66.15001021333595]
12  [18.107310966758764, -66.06170195958677]
13  [18.180084326042916, -66.00051578105086]
14    [18.27087378218598, -65.97858405961318]
```

```
                                waypoint 3
0     [18.28268636767381, -66.0894752049553]
1    [18.32665666116626, -66.10565222351079]
2   [18.359604170488176, -66.14033453628618]
3   [18.374995615998742, -66.18665293864957]
4    [18.36977776802263, -66.23542095141319]
5   [18.344985786919807, -66.27696379907812]
6     [18.3055369068776, -66.30304647600147]
7   [18.259251311806274, -66.30851000577606]
8   [18.215298091383918, -66.29228789589713]
9   [18.182377965197393, -66.25760558301617]
10  [18.167003592828046, -66.21133227202317]
11    [18.17221534746747, -66.1626204873446]
12  [18.196982657189288, -66.12110266342351]
13  [18.236406865907522, -66.09499496262468]
14    [18.28268636767381, -66.0894752049553]
```

```
                              waypoint 4
0   [18.2710000000000, -66.1990000000000]
1   [18.2710000000000, -66.1990000000000]
2   [18.2710000000000, -66.1990000000000]
3   [18.2710000000000, -66.1990000000000]
4   [18.2710000000000, -66.1990000000000]
5   [18.2710000000000, -66.1990000000000]
6   [18.2710000000000, -66.1990000000000]
7   [18.2710000000000, -66.1990000000000]
8   [18.2710000000000, -66.1990000000000]
9   [18.2710000000000, -66.1990000000000]
10  [18.2710000000000, -66.1990000000000]
11  [18.2710000000000, -66.1990000000000]
12  [18.2710000000000, -66.1990000000000]
13  [18.2710000000000, -66.1990000000000]
14  [18.2710000000000, -66.1990000000000]
```