# 2019MCM/ICM

# Problem B

# Send in the Drones: Developing an Aerial Disaster Relief Response System

**Xunzi Liu Wanteng Ma Junjie Song**

## Summary

With the frequent occurrence of natural disasters, how to design an efficient disaster relief system has become increasingly important. In this paper, we focus on designing a disaster response system by using drones. Our target includes selecting a proper location of container, choosing the best type of drone to deliver the medical package, maximizing the space utilization of drones and containers. Since this problem contains a subproblem of Bin Packing Problem, so it's NP-hard, We can't find a polynomial-time algorithm to solve it explicitly. Therefore, we will separate this question into three parts, and use three models, including selection model, packing model, and route model to solve it.

First, we abstract the actual map into Graph for analysis. We treat the road intersection as a node in the graph and approximate the road on the map as a straight line, so that the original actual map is abstracted into a simple Graph structure consisting of 25 points and several lines. Based on the assumptions, we can conclude that the drone must fly along the straight line on the graph. Then we use the Dijkstra's Algorithm to calculate the shortest flying distance from one node to all other nodes.

Second, in order to find the viable container security location, we traversed all ternary groups consisting of any three of the 25 nodes, excluding those that could not deliver the goods to the hospital even using the drones with the longest flying distance.

Third, we need to choose the proper drones to deliver the medical package and propose a plan to put the package into the plane to maximize space utilization. On the issue of drone selection, we design a scoring function to evaluate the drone and get the optimal drone selection. When it comes to the packing problem,

Fourth, we need to select the best schedule for the drone's route. We let a series of planes form a small team, so that we could use them to meet the cargo demand first, and then explore the road as far as possible before the demand for the goods is insufficient. This is the best flying schedule we conclude.

Last but not least, we did some sensitive test and analyzed the strength and weakness of our model, which make our model well adapted to different situation.

# Content

# 1. Introduction

## 1.1 Problem Background：

Due to the impact of human activities on the environment, extreme climate disasters have become more frequent in recent years, causing enormous damage to human beings. In 2017, a hurricane hit Puerto Rico, causing nearly 2,900 casualties and huge property damage. The smallness of human beings is undoubtedly revealed compared to the terrible disaster.

However, people are often struggled to provide adequate and timely response during or after natural disasters. Therefore, one non-governmental organization -HELP, Inc. – attempts to provide a disaster response system called “DroneGo.” It will use rotor wing drones to transport pre-packaged medical supplies and provide high-resolution aerial video reconnaissance at the same time. In order to increase the efficiency of the system, we need to choose proper drones based on the medical packages demand of different hospital while probe as many roads as possible.

## 1.2 Problem restatement

The problem can be analyzed into five parts

• Choose the best location to place standard dry cargo containers

• Choose the proper drones to deliver medical packages as much as possible and maximize aerial video reconnaissance.

• Find the optimal packing plans to maximize the space utilization for the given plane and the cargo containers.

• Find the best shipping path for the drones.

• Clearly understand the tradeoffs if the demand requirements exceed the capabilities of the drone fleet

## 1.3 General assumptions:

In order to simplify the analysis, we make the following assumptions.

**Assumption I**: Treat the intersection of the road as a node

**Reason:** Because most of the intersections of the roads are cities, and the flow between people is mainly between the city and the city, in order to simplify the analysis, when we choose where to place standard dry cargo containers, we only consider the intersection of the road.

**Assumption II**: The drone flies in a straight line between two points where the city is directly connected. Since the space for exploration of the aircraft has a certain range, it can still explore the whole road when flying in a straight line.

**Reason:** Take a specific road as an example, we need to explain that if we flied in a straight line, the distance from the farthest point on the road to the flight path is less than the search radius of the drones. As we can see, in the picture, the farthest distance is 1.11km, while the search radius of the drone (we assume that the flight altitude is 500m and the maximum angle at which the lens is placed is 70°) is 1.33km, which is approximately enough for the searching.

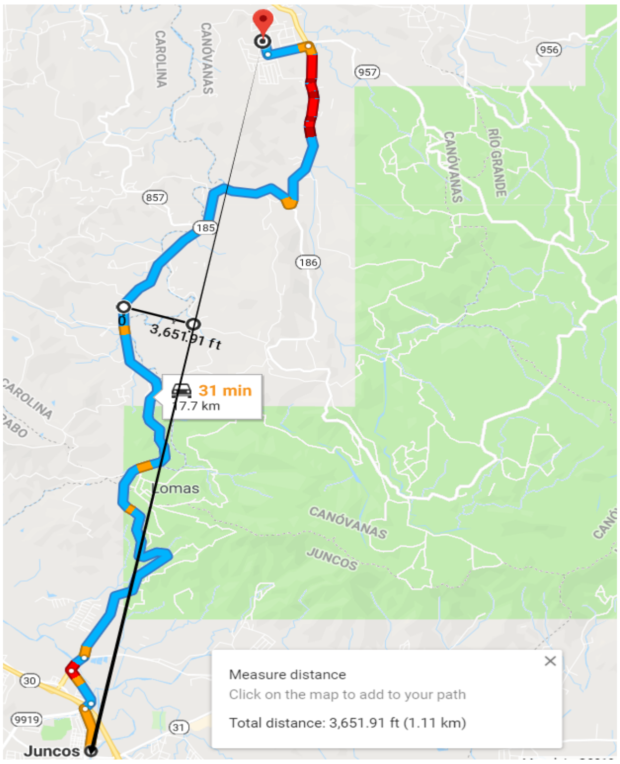



Figure 1 Google Map evidence

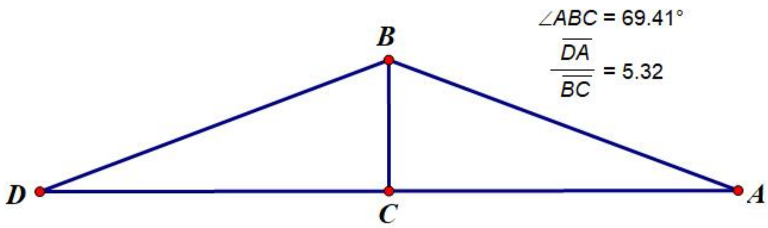



Figure 2 A transversal graph of detecting radius

**Assumption III:** If the two cities are not directly connected with the road, we still have to fly along the road, which means our flight route will be a fold line. Based on this, we can calculate the shortest flying distance between the two cities using Dijkstra's Algorithm.

**Reason**: In order to explore as much distance as possible, we need to fly along the road. Therefore, if the two cities are not directly connected, we couldn't regard the straight distance between two cities as our flying distance. After using Dijkstra's Algorithm, we can conclude that the 'real' distance between them, which can be shown as follows:

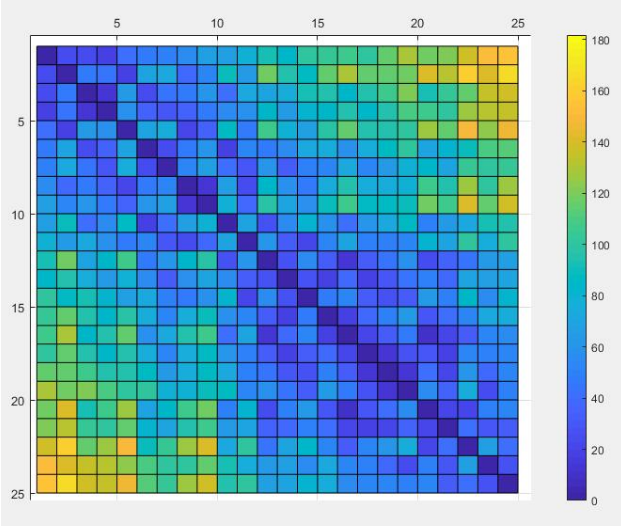



Figure 3 'Real' Distance

**Assumption IV**: All drones must return to the vicinity of the container for recharging after completing one mission.
**Reason**: Since It is known from reality that the distance of a drone flight is limited, and in order to continue to use the drone, we must make the drone land near the container so that it can be recharged and reused.

**Assumption V**: For a certain region, we can directly subtract the latitude and longitude of the point for distance calculation.
**Reason**: Considering that the change in longitude and latitude in this small area is small, we can omit the cross term and calculate directly by latitude and longitude subtraction. The formulation is actually an approximation of the reality. Assuming that $1°$ on the equator symbolize 111 km, so we have:

$$\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} * 111 * \cos\frac{x_1 + x_2}{2}$$

**Assumption VI**: It is assumed that both the cargo and the drone can be rotated in any direction when packing.
**Reason**: This assumption guarantees that the shape of the medical package and the drone will not influence our analysis.

**Assumption VII** : The flight distance of the drone at full load is approximately equal to the flight distance of the drone at no load.
**Reason**: Because the impact of the load on the flight distance of the drone is difficult to estimate and the impact is relatively small, in order to simplify the analysis, we do not consider the impact of the load on the flight distance of the drone.

# 2. Notation

| Symbols | Meanings |
|---|---|
| $(x_1, x_2, x_3)$ | All the cases of the cargo containers' position |
| $v_i$ | Node in the graph |
| $n$ | The number of intersections of roads |
| $S$ | The weighted graph after Dijkstra Algorithm |
| $\boldsymbol{s_{ij}}$ | The shorted length of path from $\boldsymbol{v_i}$ to $\boldsymbol{v_j}$ |
| $H$ , $W$, $D$ | The length, width, height of a thing |
| $P$ | Max payload capability |
| $L$ | Flying range of a drone, which is $\boldsymbol{Speed \cdot FlightTime}$ |
| $\boldsymbol{L_{max}}$ | The maximum flying range of all the drones |
| $\gamma$ | The number of boxes using |

# 3. Packing model

## 3.1 Model description

The Packing model is a kind of model that placing a series of items with different sizes into a given space (such as drone interior, container interior) to maximize space utilization. Since we only have two kinds of cargo container and a kind of container, we just need to solve the problem with different containers for three times.

## 3.2 Packing given set of medicines with least cargos

According to the daily needs of the hospital, we increase the demand for medicines in all hospitals. We obtain the proportion of medicines contained in each container. Our packing scheme consists of grouping drugs according to the proportion of drugs and loading each group of drugs into two cargo compartments to maximize volume utilization and minimize number of cargos.

Now this problem is as a three-dimensional bin packing problem (known as 3D-BPP). Specifically, we can compare the two kinds of cargos with the dimensions of three medicines.

First, the table below shows the Cargo volume times by medicine volume. From this we can assume that to carry as many medicines as possible, we may consider planes with cargo 2 in priority.

**Table 1 Volume Comparison**

| | Cargo 1 | Cargo2 |
|---|---|---|
| **Medicine 1** | **2.28** | **19.59** |
| **Medicine 2** | **5.6** | **48** |
| **Medicine 3** | **3.33** | **28.57** |

In order to make our program more practical and more feasible in reality, we are applying the following model.

First let we consider a single-cargo problem, we assume that there are rectangular cargo conai *C*, $\gamma$ cargos in total; a set of medicines B=*{b1,b2,,,bn}*, n medicines in total. The cargo container *C* has a dimension of H in length, W in width, D in height, similarly every medicines have a dimension of hi in length, wi in width, di in height. So every medicine has a volume of *vi=hi*wi*di*, So our problem can be denote as the following linear programming problem:

$$Cargo\ function: \qquad min\,\gamma$$

$$s.t.\ l_{ij} + l_{ji} + u_{ij} + u_{ji} + b_{ij} + b_{ji} + c_{ij} + c_{ji} = 1 \quad i < j$$
$$x_i - x_j + W(l_{ij} - c_{ij} - c_{ji}) \le W - w_i \qquad i \ne j$$
$$y_i - y_j + H(u_{ij} - c_{ij} - c_{ji}) \le H - h_i \qquad i \ne j$$
$$z_i - z_j + D(b_{ij} - c_{ij} - c_{ji}) \le D - d_i \qquad i \ne j$$
$$(\bar{\gamma} - 1)(\,l_{ij} + l_{ji} + u_{ij} + u_{ji} + b_{ij} + b_{ji}) + \gamma_i - \gamma_j + \bar{\gamma} + c_{ij} \le \bar{\gamma} - 1$$
$$l_{ij}, u_{ij}, b_{ij}, c_{ij} \in \{0, 1\}$$

In this system, l, u, b, c are all binary variables, $l_{ij} = 1$ if item i is in the left of item j, $u_{ij} = 1$ if item i is under item j, $b_{ij} = 1$ if item i is in the back item j and $c_{ij} = 1$ if $\gamma_i < \gamma_j$. And we can also list the practical dimension constraints as below:

$$0 \le x_i \le W - wi, \qquad i = 1, \cdots, n,$$
$$0 \le y_i \le H - hi, \qquad i = 1, \cdots, n,$$
$$0 \le z_i \le D - di, \qquad i = 1, \cdots, n,$$
$$0 \le \gamma_i \le \gamma \le \bar{\gamma}, \qquad i = 1, \cdots, n.$$

With all the constraints we have, we need to construct a method to fit all the medicines into the cargos in a given sequence and a given rotation. This can be denoted in a matrix $M_1$, for example, a container location $x_1$ has a medicine proportion as

$$m_1 = [2,1,3]$$

which means the hospital matching to $x_1$ has a demand proportion of 5 Medicine 1, 1

Medicine 2, 3 Medicine 3. As we already have the medicine dimension matrix $M$,

$$M = \begin{bmatrix} 14 & 7 & 5 \\ 5 & 8 & 5 \\ 12 & 7 & 4 \end{bmatrix}$$

Mix the $M$ and $m_1$, we get $M_1$ as

$$M_1 = \begin{bmatrix} 14 & 7 & 5 \\ 14 & 7 & 5 \\ 5 & 8 & 5 \\ 12 & 7 & 4 \\ 12 & 7 & 4 \\ 12 & 7 & 4 \end{bmatrix}$$

The $M_1$ has denoted two essential kinds of information, which can describe a method of fitting:

- The sequence of medicines put into the cargo, which is denoted with the sequence of rows in matrix $M_1$.
- The rotation of each medicines put into the cargo, which is denoted with the sequence of column of each rows in matrix $M_1$.

With the information above, our current goal is to put the given drug combinations into the cargo hold in the given order and direction. We are going to claim for some special definitions.

### 3.2.1 Potential Placing Point

We embed the cargo as a *bottom-left-back* coordinate system, so that the bottom left corner of the bottom is at the origin of the coordinates, and the length, width, and height are on the x, y, and z axes, respectively, and the medicines are considered in order. According to experience, the currently placed medicine must be placed against the container or the medicine placed in front, so we consider the record potential placing point $p_i \in P$ in this method:

- Record the medicine *i bottom-left-back* position in the coordinate, with a vector
$$p_i = [x_i, y_i, z_i]$$
- Refer to $M_1$ for the direction of medicine i, in which the row i denotes the medicine i's direction.

As our assumptions, the initial potential placing point will be [0,0,0] in every cargo, this is the baseline. When add every new medicine into the cargo, we have to:

- Delete the used potential placing point
- Add three new potential placing points into the aggregation P. Which are
$$[x_i + h_i, y_i, z_i], [x_i, y_i + w_i, z_i], [x_i, y_i, z_i + d_i]$$

That is, for the *n*th medicine, we have up to *2n+1* potential placing points in P.

### 3.2.2 Reference Line

We are considering to set two reference lines, *Lz* on the *z* axes and *Lx* on the *x* axes. Our placement strategy is to place the medicine in given order, when considering whether the medicine i can be placed into the position [*x, y, z*], we use the reference line as a boundary. So the sufficient condition will be as follows:

- $x + h_i \leq L_x$, $z + d_i \leq L_z$, that's the new medicine will not be across the reference line.
- $y + w_i \leq W$ , that's the new medicine will not exceed the width of that cargo.

  But the conditions above are not sufficient, because in the condition I the reference line can be moved upward as long as it's below the *H* or *D,* according to the rules below,

  If $H > L_x$, move $L_x$ upward to $x + h_i$

  If $\mathrm{H} \leq \mathrm{Lz}$, move $L_z$ upward to $z + d_i$

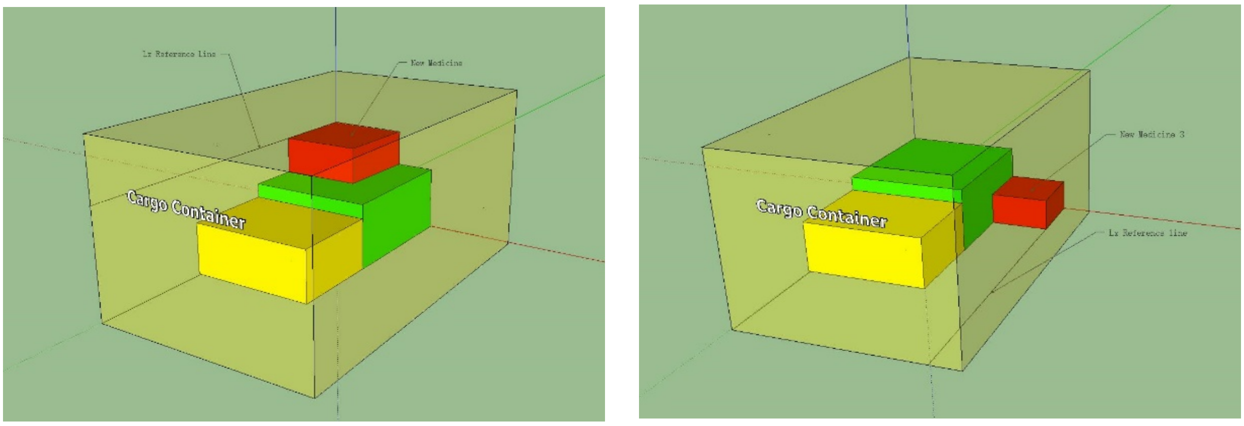



Figure 4 Reference Lines

In the worst case, we will fail to place some of the medicines due to the size constrains, then we may open a new box after ergodic the $M_1$.

### 3.2.3 Translation

After we successfully place the box, we may have another step to move the smallest *bottom-left-back* coordinate [*x, y, z*], which can be done when we reset the new potential placing point, according to the following rules,

- Assume the new potential placing point is *p'*= [*x', y', z'*], then compare it with all the potential placing points in the aggregation, denoted as $[x_j, y_j, z_j]$.

- Define a relationship $\gtreqqless$ which symbolize a dominant. This can be viewed as if $\exists\ \mathrm{p_j} = [\mathrm{x_j}, \mathrm{y_j}, z_j]$ , that [*x', y', z'*] $\gtreqqless$ $[x_j, y_j, z_j]$, then we can assert that the [*x', y', z'*] cannot be add into P.

3.3 Searching the $M_{best}$ with Simulate Anneal

As we've analyzed before, our task is to find the best $\mathrm{M}_1$ as $\mathrm{M}_{\mathrm{best}}$ which can reach the smallest γ, and up to now we've reached a method of adding all the medicines with given sequence and direction.

So now we are applying the Simulate Anneal method to search for a global optimal $M_1$.

As generally we set the initial temperature *Et* and end temperature *St*, with a decreasing parameter *dt*, every times we change the sequence or the rotation of the $M_1$, then compare the $M_1$ with $M_{best}$ in terms of the volume utilization and cargos.

After the simulation, setting parameter initial temperature *Et*, end temperature *St* with a decreasing parameter *dt*, we can reach a comparatively good choice.

# 4. Selection Model

## 4.1 Model description:

The Selection model is composed of two parts, the first part selects the possible location of the containers, the second part selects the proper drones to deliver the cargos.

## 4.2 The First selection:

We can note all the strategy of selection as a ternary tuple, which means we choose point $x_1$,$x_2, x_3$ as the locations to position cargo containers, note as $(x_1, x_2, x_3)$

In the first selection, we will delete all the points that are too far from the medical demand areas. More specifically, when the ternary tuple$(x_1$,$x_2, x_3)$ satisfy:

$$\exists x_i, i \in 1,2,3$$
$$s.t.\ \forall v_j, s_{ij} > L_{max}$$

We delete it.

All the tuples left are the cases that if we have enough drones, we can satisfy all the demand of medicines.

## 4.3 The second selection:

Our main idea is to find an evaluation function to calculate the scores of all the different ternary tuples after the first selection. To do this, we design an approximation polynomial-time algorithm to find the strategy of drones selection and medical packages.

Then we will use this 3-D Bin Packing Problem algorithm to pack drones into the cargo containers. After packing, we evaluate the ability of transportation and the ability of reconnaissance, then we use a ratio to balance them.

For each possible solution after the first selection, we can match each medical demand area with exactly one cargo container position which is the nearest node in the weighted graph *S*.

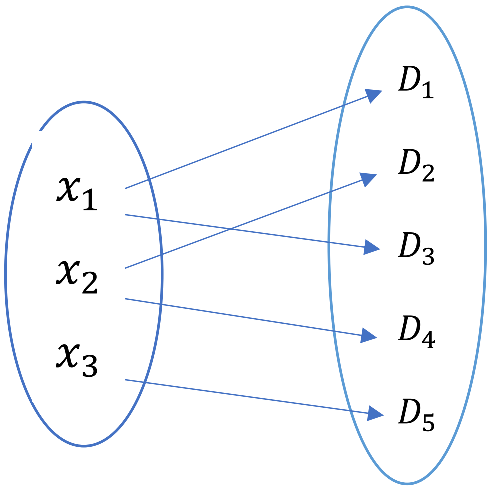



Figure 5 Matching

As is shown in the graph, why we match them like this? Because in the real case, the number of medicines a cargo containers carry is much larger than the demand of many delivery locations, so we can do like this.

After the first selection, we can ensure that every cargo container has at least ones delivery location, so that the supply will not be isolated from demand, and medicine will not be wasted.

This can also help us simplify the problem

After match, we can get 5 paths from cargo containers to different delivery locations

Then for each path we choose a specific kind of drones to transport the medicine. For every determined path, we can find the best fit of drone that can cover the path and transport most medicine or take up least space at the same time.

Actually we construct an evaluation function to achieve the quantization of fitness of different drones.

$$max\ E = \left(\frac{f \cdot P}{W \cdot D \cdot H}\right)^{\alpha} \cdot L^{\beta}$$

$$f = \begin{cases} k & \text{(when drone carry cargo2)} \\ 1 & \text{(when drone carry cargo1)} \end{cases}$$

$$\alpha + \beta = 1$$

We use $f, P$ to describe the ability of transportation, and because cargo2 is much larger than cargo1, we can choose a reasonable $k$ in $f$ to make the evaluation more reliable.

And the drone's volume of the shipping container is divided because with less volume, we can pack more drones in the cargo containers, which means more medical packages available.

$L$ represent the ability of reconnaissance. And we fix both of them with power $\alpha$ and $\beta$, so that they can be more

Searching $E_{max}$ we can find the best drone for the path. Multiply it so that they can support a day's demand of the delivery location. and then for each cargo container, pack all the chosen drones of the connected path in to a group, then put as many groups of drones as possible into the cargo container

For example, there are two path connected with $x_1$. the best drone of two path are Drone1 and Drone2. We can use at least *a* Drone1s to and *b* Drone2s Then the group is ($a \cdot Drone1,\ b \cdot Drone2$), and we will pack cargo container1 with many groups of ($a \cdot Drone1,\ b \cdot Drone2$).

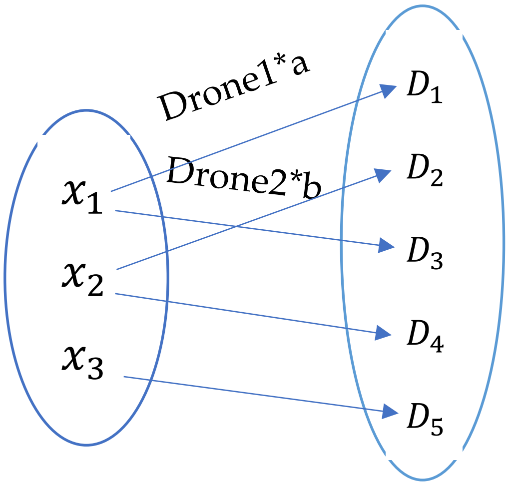



Figure 6 Drones choice on Matching

Then we can order the plane which fly to carry the reconnaissance problem.

After using SA to pack medical packages into drone cargo bays and put groups of drones into cargo containers, we will do another evaluations for them and use them to choose the final $(x_1, x_2, x_3)$

Firstly, Note $Med(x_i)$ as the number of medical packages in the cargo $x_i$. We can use

$$Trans = \sum_{i=1}^{3} Med(x_i)$$

To represent the ability of medical package supply.

Then, to evaluate the reconnaissance ability of the choice, we will calculate the number of the nodes which the chosen drones of $(x_1, x_2, x_3)$ can get to them from the cargo containers. Note them as $recon$

But specially, if there are different $x_i$ that a drone can fly between them, which means

$$\exists i, j (i \neq j)\ s_{ij} \leq L_{max}$$

The strategy is better because we can use at least one kind of drone to dispatch some medicines. This will make the system handle more serious emergencies, we will give it an extra point $bonus$

then, the evaluation function can be written as:

$$g = Trans^{\gamma} \cdot recon^{\lambda} + bonus$$

$$\gamma + \lambda = 1$$

Searching $g_{max}$ and we can find the best $(x_1, x_2, x_3)$

# 5. Route Model

Since we have decided the schedule based on our design. we pack many groups of drones. Correspondingly we can put a group into a fleet, and use them to transport the packages every day. But this establishment has fault, because actually there are many cases which a fleet can carry packages far more than a day's demand when top loaded.

we can choose:

- Every day select new drones to form fleets for every delivery location.
- If a delivery location has enough packages, then dismiss the fleet.
- Separate the other drones or the drones that don't carry packages to other node that the drones can get without repetition.
- If any two $x_i$can be detected each other, then use a drone which fly the longest to transport packages from one to another.

# 6. Case study

## 6.1 possible location chain

We find 39 set of the possible locations, and then we can find the optimal choice through the evaluating functions. In the function we set $\alpha = 0.4$ and $\beta = 0.6$, since the function of sending medicine is prior to the function of video taking, we make the weight $\beta > \alpha$.

Then we get the optimal choice as is shown in the chart below:

**Table 2 Container Locations in Exceed Demand Situation**

| Container Locations | Medicine Buddle | Pairing Hospitals |
|---|---|---|
| **Utuado** | **[1, 0, 0]** | **Acrecibo** |
| **Bayamon** | **[5, 2, 3]** | **San Panlo, SanJuan, Bayamond** |
| **Loiza Aldea** | **[1, 0, 1]** | **Jajardo** |

Depending on the distance between the locations, we can denote the connection matrix as

$$\chi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

1 symbolize that the distance between two areas below 52.4. 0 vice versa.

If we have the algebraic cofactor as $\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$, we assert that the locations embedded can be treated as an aggregation 'chains'. And when we are planning for the drones and medicines in the aggregation, we can solve them as a whole.

## 6.2 bin-packing-problem

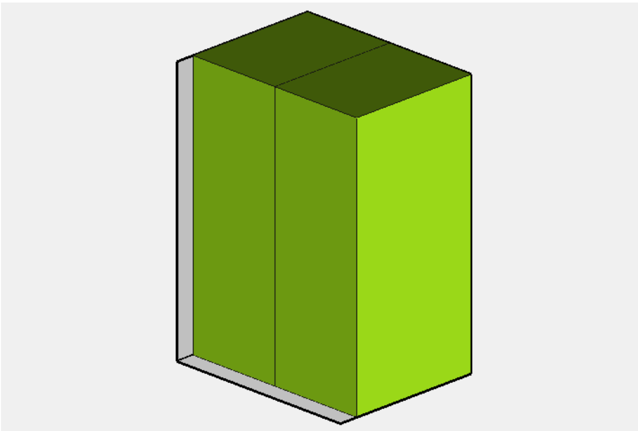
After finding the optimal choice, using our model we can conclude that we need the drone B F G to be packed in different cargo containers.

**Table 3 Drones selection for specific location**

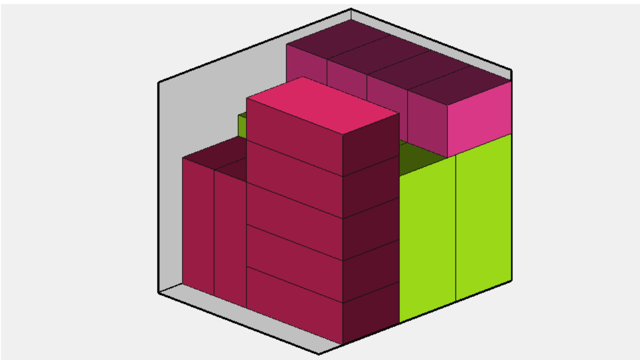
| Drone Type | Fitting Locations |
|---|---|
| B | Acrecibo, San Panlo, Fajardo |
| F | San Juan |
| G | Bayamond |

Then we can pack medical packages into cargo bays of different drones.

1) **Acrecibo** We can carry 2 med1 in the CargoBay1 on drone B, and its for Acrecibo. They are packed like this:

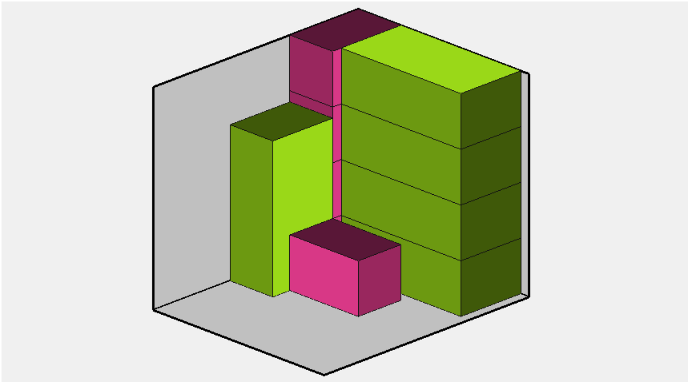


2) **Bayamond** We can carry 8 med1, 4 med2, and 7 med3 in the CargoBay2 on drone G, and it 's for Bayamond. Notice that although it is overweight, we can place

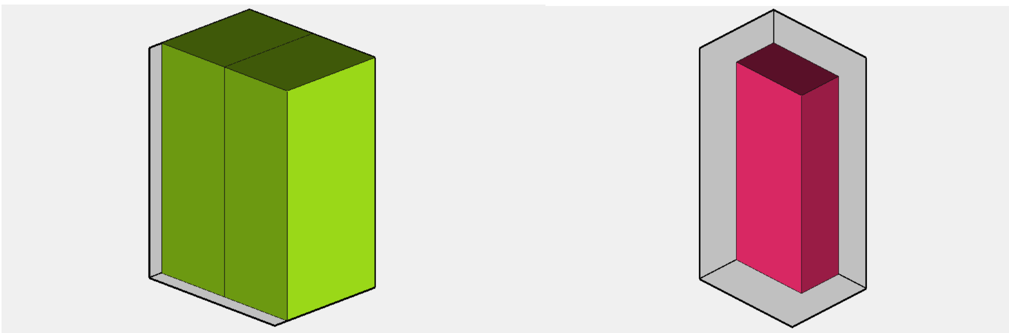


the cargo containers in the delivery location directly, so that we don't need to move it. They are packed like the graph above.

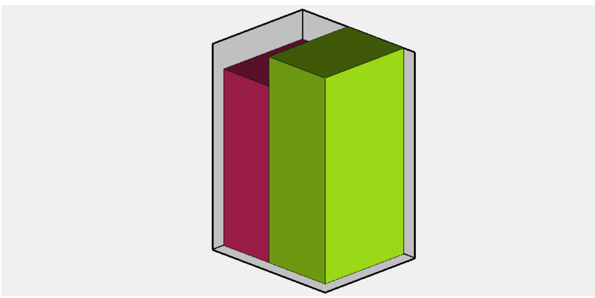
3) **San Juan** We can carry 5 med1, 5 med2 in the CargoBay2 on drone F, and its for San Juan.



4) **San Panlo** We can carry 2 med1 in the CargoBay1 on drone B, and another 1 med3 in another drone B. And its for San Panlo.



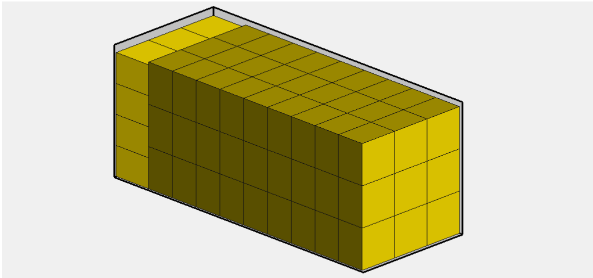
5) **Fajardo** We can carry a med1 and a med3 in the CargoBay1 on drone B, and its for Fajardo.



All the strategy above can ensure that we can use drones to satisfy all the demand of packages

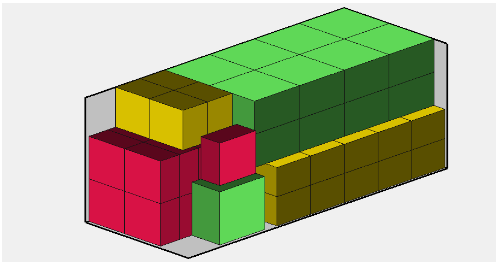
Then with the respect of the matching pairs, we pack many drone B into 2 cargo containers, and pack drone B F G into the last one. Then put them to Utuado, Loiza Aldea and Bayamon.

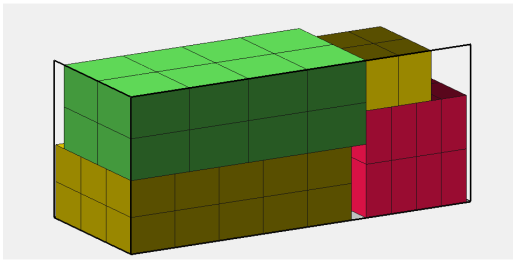
**1)** For Utuado and Loiza Aldea, we can pack Drone B like this:



We can pack 93 Drone B totally.

**2)** For Bayamon, we can pack drone B F G like this





We can pack 34 drone B, 17 drone F, 17 drone G totally.

### 6.3 delivery schedule

Now we describe a schedule for the drones:

1) In Utuado, we fly drone B which carry packages to Acrecibo and retuen once every 2 days,
2) In Bayamon, we release all the cargo bay in drone G, and fly 2 drone B with different packages to San Panlo every day. then fly drone F to San Juan once every 5 days. And every day we fly drone G to each of Farjardo, Comerio, Donado, Vega Baja and retuen.
3) In Loiza Aldea we fly drone B which carry packages to Fajardo and retuen once a day.
4) Then we let 2 drone B fly between Bayamon and Loiza Aldea every day to dispatch medical packages.

# 7. Sensitivity Analysis

As is required by the HELP.Inc, we may consider some conditions beyond the given situation, for the possibility that the disaster area will spread over the whole island and bring more damage to the cities. So we have to take this case into account. There are 2 potential exceeds for our systems capability.

## 7.1 New excess demand from some of the hospitals

As designed in our plans, we have the three locations of the containers in a ‘chain’, which means they are a least within 52.4km (which is the farthest distance covered by drone B) from another location. So with the available medicines in the original problem, we can cover the excess demand in some situations.

We can assume that the Hospital Pavia Saturce in San Juan is claiming for excess demand of a buddle [2, 1, 1], that is one more demand for every medicine. Then we can add it into the models and recalculate the suitable planes for each containers location, the calculation results are as below:

**Table 3 Container Locations in Exceed Demand Situation**

| Container Locations | Medicine Buddle | Max Cover Days |
|---|---|---|
| Utuado | [1, 0, 0] | 20 |
| Comerio | [2, 0, 1] | 20 |
| Loiza Aldea | [5, 2, 4] | 20 |

***The Max Cover Days are all the same because drones can transport the medicine between three containers, since they are in a ‘chain’.**

We can see that the tradeoff can be easily made between the containers, if one of the containers is facing more demand, the gap can be covered by a transportation between containers.

## 7.2 New hospital in demand of medicine

This situation will be harder than the first one to overcome, when we add a quantity of cities into the demanding list, we may find that 3 containers are definitely not sufficient due to the distance that a single drone can cover is strictly limited. These are some of our simulation test result:

**Table 4 Available chains in Different Simulations**

| Hospitals(Cities) add in | Numbers of Possible Container Location Chains |
|---|---|
| Mercdita | 8 |
| Parguera | 0 |
| Quebradillas, San Sebastin | 0 |

The result is obtained by simply eliminating the locations beyond 26.2 km from the hospitals, which is necessary but not sufficient.

After we add the number of demanding hospitals to 7, there rarely exist a proper location chain that can cater all the demand.

So when the hospitals in demand increase, the HELP.inc have to consider sending more containers to the island.

# 8.Strength and Weakness

## 8.1 Strength

1. Reasonable assumption

We made a lot of reasonable assumption to simplify the question, so that we can only concentrate on the core problem of this question.

2. Applying the idea of analogy

In the process of analysis, we widely applied the idea of analogy, such as making an analogy between the real map and the graph with only dots and lines, which makes us abstract the vague concept into something concrete thus makes the model universally applicable.

3. Good flexibility

According to the sensitivity analysis, our model can be well adapted to different conditions, which means it has Good flexibility.

## 8.2 Weakness

1. Lack of algorithmic efficiency

Because we only need to process less data, we relax the need for algorithmic efficiency, which makes our model inefficient when dealing with large amounts of data.

2. Approximation error

When dealing with the distance between two points, we use the method of estimating. If the distance between the two places is much bigger, the approximation algorithm will bring a large error.

3. Calibrating error

When calibrating the parameters of the model, a high accuracy will be difficult to achieve due to lack of data.

# Reference

[1] Maarouf, Wissam & Barbar, Aziz & Owayjan, Michel. (2008). A New Heuristic Algorithm for the 3D Bin Packing Problem. 10.1007/978-1-4020-8735-6_64.

[2]Crespo Cuaresma, Jesús, Wolfgang Lutz, and Warren Sanderson. 2014. “Is the Demographic Dividend an Education Dividend?” Demography 51 (1): 299–315. doi:10.1007/s13524-013-0245-x.

[3] Shao Yanhua, Zhang Depei, Chu Hongyu, Chang Zhiyuan, Zhan Huayi, Rao Yunbo.Design and Implementation of Field Search and Rescue UAV Based on Infrared Vision[J].Sensors and Microsystems,2019,38(01):104-106 +109.

[4] Fu Jiangyue,Chen Gang.Evolution Model and Algorithm of Axis-spray Urban Logistics Network[J].Application Research of Computers, 2015, 32(12): 3605-3608+3617.

# Appendix

## function BPP( ) to solve the packing problem

```
function BPPresult=BPP(items,box)%items is the sequence of sorti
H=box(1);
W=box(2);
D=box(3);
boxnum=0;
n=length(items(:,1));
fang={};
Vitems=0;
while isempty(items)==0
    I={[0,0,0]};
    Lx=0;
    Lz=0;
    i=1;
    n=length(items(:,1));
    already=[];
while i<=n
    flag=0;
    for X=I
        X=X{1};
        x=X(1);
        y=X(2);
        z=X(3);
        if items(i,1)+x<=Lx  && items(i,3)+z<=Lz && items(i,2)+y
            flag=1;
            break
        end
    end
    if flag==0
        if Lx==0
            if items(i,3)+Lz<=D
                x=0;
                y=0;
                z=Lz;
                flag=1;
                Lz=Lz+items(i,3);
                Lx=items(i,1);
                for j=1:length(I)
                    X=I(j);
                    X=X{1};
                    if isequal(X,[x,y,z])
                        fang{length(fang)+1}=[x,y,z];
                        if x+items(i,1)<=H
                            I{j}=[x+items(i,1),y,z];
                        end
                        if y+items(i,2)<=W
                           I{length(I)+1}=[x,y+items(i,2),z];
                        end
                        if z+items(i,3)<=D
                           I{length(I)+1}=[x,y,z+items(i,3)];
                        end
                        Inew={};
                        for k=1:length(I)
                            da=0;
                            for t=1:length(I)
                                if k==t
                                    continue
                                elseif ismember(0,I{k}>=I{t})==0
                                   da=1;
                                   break
                                end
                            end
                            if da==0
                                Inew{length(Inew)+1}=I{k};
                            end
                        end
                        I=Inew;
                        Vitems=Vitems+items(i,1)*items(i,2)*items(i,3);
                        already=[already,i];
                        break
                    end
                end
            elseif Lz<D
                Lz=D;
                Lx=H;
                i=i-1;
            end
        elseif Lx~=H
```

```
for X=I
    X=X{1};
    x=X(1);
    y=X(2);
    z=X(3);
    if z+items(i,3)<=Lz && Lx+items(i,1)<=H && x==Lx && y==0
        flag=1;
        Lx=Lx+items(i,1);
        for j=1:length(I)
            X=I(j);
            X=X{1};
            if isequal(X,[x,y,z])
                fang{length(fang)+1}=[x,y,z];
                if x+items(i,1)<=H
                    I{j}=[x+items(i,1),y,z];
                end
                if y+items(i,2)<=W
                   I{length(I)+1}=[x,y+items(i,2),z];
                end
                if z+items(i,3)<=D
                   I{length(I)+1}=[x,y,z+items(i,3)];
                end
                Inew={};
                for k=1:length(I)
                    da=0;
                    for t=1:length(I)
                        if k==t
                            continue
                        elseif ismember(0,I{k}>=I{t})==0
                          da=1;
                          break
                        end
                    end
                    if da==0
                        Inew{length(Inew)+1}=I{k};
                    end
                end
                I=Inew;
                Vitems=Vitems+items(i,1)*items(i,2)*items(i,3);
                alreadv=[alreadv,i];
                                break
                            end
                        end
                        break
                    end
                end
                if flag==0
                   Lx=H;
                   i=i-1;
                end
            elseif Lx==H && Lz+items(i,3)<=D
                Lz=Lz+items(i,3);
                i=i-1;
            elseif Lx==H && Lz==D
                break %jump out of the while
            end
        elseif flag==1 %if flag=true we can fit the item in at the first time
            for j=1:length(I)
                X=I(j);
                X=X{1};
                if isequal(X,[x,y,z])
                    fang{length(fang)+1}=[x,y,z];
                     if x+items(i,1)<=H
                         I{j}=[x+items(i,1),y,z];
                     end
                     if y+items(i,2)<=W
                         I{length(I)+1}=[x,y+items(i,2),z];
                     end
                     if z+items(i,3)<=D
                         I{length(I)+1}=[x,y,z+items(i,3)];
                     end
                    Inew={};
                    for k=1:length(I)
                        da=0;
                        for t=1:length(I)
                            if k==t
                                continue
                            elseif ismember(0,I{k}>=I{t})==0
                              da=1;
                              break

                          end
                     end
                      if da==0
                         Inew{length(Inew)+1}=I{k};
                      end
                 end
                 I=Inew;
                 Vitems=Vitems+items(i,1)*items(i,2)*items(i,3);
                 already=[already,i];
                 break
             end
         end
     end
i=i+1;
end
itemsnew=[];
for k=1:n
    if ismember(k,already)
        continue
    else
        itemsnew=[itemsnew;items(k,:)];
    end
end
items=itemsnew;
boxnum=boxnum+1;
end
B=Vitems/(H*W*D*boxnum);
BPPresult={B,boxnum,fang};
end
```

# Memo:

To: CEO of HELP, Inc
From: A group of modelers who are enthusiastic about mathematical modeling.
Date: 28 January 2019
Subject: Summary of our modelling results

We are a group of students having conducted a mathematical modeling about the designing of your company's disaster response system. And it will be an great honor for us to summarize our research to you so that you could share these with your board of directors.

In order to solve this problem, we separate this question into three parts. And for each part, we introduce a kind of model to illustrate it, which are named packing model, selection model, and route model, to find the best solution.

For packing model, When fitting the medicine into the drone cargo containers, we can treat it as a 3D-BPP（Bin-Packing-Problem）. approximate the optimal solution through the following two steps. First, we fit all the medicine into the least Cargo containers in given directions and sequence. Secondly, we apply the SA algorithm to find the best sequence and directions. The process to fit the drones into containers is very similar.

For selection model, on the issue of container's location selection, we traversed all ternary groups consisting of any three of the 25 nodes, excluding those that could not deliver the goods to the hospital even using the drones with the longest flying distance. On the issue of drone selection, we design a scoring function to evaluate the drone and get the optimal drone selection.

For route model, We let a series of planes form a small team, so that we could use them to meet the cargo demand first, and then explore the road as far as possible before the demand for the goods is insufficient.

Based on the analysis above, our disaster response system can be presented as following:

**(1) Container's location**

| Container Locations | Medicine Buddle | Pairing Hospitals |
|---|---|---|
| Utuado | [1, 0, 0] | Acrecibo |
| Bayamon | [5, 2, 3] | San Panlo, SanJuan, Bayamond |
| Loiza Aldea | [1, 0, 1] | Jajardo |

**(2) Drone choice**

| Drone's type | Fit Hospital |
|---|---|
| Drone B | Acrecibo, San Panlo, and Fajardo |
| Drone F | San Juan |
| Drone G | Bayamond |

**(3) Packing plan for the drone**

| Drone's type and its corresponding hospital | cargos |
|---|---|
| Drone B for Acrecibo. | 2 med-1 |
| Drone G for Bayamond | 8 med-1, 4 med-2, 7 med-3 |
| Drone F for San Juan | 5 med-1 , 5 med-2 |
| Two Drone B for San Panlo. | One for 2 med-1 , another for 1 med-3 |
| Drone B for Fajardo | 1 med-1 , 1 med-3 |

**(4) Packing plan for the container**

| Container location | drones |
|---|---|
| Utuado and Loiza Aldea | 93 Drone B |
| Bayamon | 34 drone B, 17 drone F, 17 drone G |

**(5) Flying schedule**

- Every day select new drones to form fleets for every delivery locations.
- If a delivery location has enough packages ,then dismiss the fleet.
- Separate the other drones or the drones that don't carry packages to other node that the drones can get without repetition.
- If any two $x_i$ can be detected each other, then use a drone which fly the longest to transport packages from one to another.

I hope that our disaster response system will be helpful to your company.

A Group of Enthusiastic Modelers
Jan 28th 2019