| For office use only | Team Control Number | For office use only |
|---|---|---|
| T1 ________ | **1915522** | F1 ________ |
| T2 ________ | | F2 ________ |
| T3 ________ | Problem Chosen | F3 ________ |
| T4 ________ | **B** | F4 ________ |

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

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


**Summary**

This paper mainly analyzes and discusses the various issues of UAVs in the rescue operation in Puerto Rico. At the same time, there are many parts that can be adjusted in the model, which is also instructive for other rescue problems.

First of all, the first question is about the packing of medical package, drones, and drone cargo bay in cargo containers. The basic guiding principle of this part is to find a specific assembly plan to minimize the remaining space in the container. At the same time, we considered the impact of various types of drones on transport drugs and exploration routes, and the impact of the number of medical packages on rescue operations. For the overall consideration of the above factors, we give a high quality factor for evaluating the quality of the package. Therefore, based on the idea of optimization, a nonlinear programming model is established, and the genetic algorithm is used to select the model. Finally, we came up with a plan for container configuration.

Secondly, in the second question, we need to consider the location of each container. First, in the search for a port, it was discovered that in the northeastern part of Puerto Rico, only San Juan Bay has ports that can provide containers to shore. Thus, the idea of transporting containers from ports is impractical. So we abandoned this limitation and constructed our model from the perspective of the ability to achieve transport functions and optimize the flight paths that may be covered. We searched for the best container location by adaptive image processing.

Next is the thinking about the third question. The first part is to determine the packaging configuration of drone cargo bays and shipping route of the drone that delivers the drug. We give the optimal configuration by considering the type of each type of drone. The second part is the problem of the route allocation of each type of drone. We have established indicators for evaluating the quality of drones based on different flight routes, speed of drones, and maximum flight distance. Based on the above indicators, the optimal formation scheme is given by the method of constraining extremum.

Finally, we wrote a memo for the Chief Operating Officer (CEO) of HELP, Inc. In the memo, we give the basic ideas and guidelines of the model. At the same time, we gave the best solution we simulated, hoping to help the rescue in Puerto Rico.



**Keywords**: Packing Configuratio;Drone Fleet; Genetic Algorithm

# Contents


**0 Introduction** **3**

0.1 Background . . . 3

0.2 Restatement of the Problem . . . 3

0.3 Assumptions . . . 3

0.4 Notations . . . 4

**1 Analysis of the Problem** **5**

**2 Locations Identification [Mission B]** **6**

2.1 Plane of disaster area and coordinates of delivery locations . . . 6

2.2 Medical Demands and Drones' Domain . . . 6

2.3 Reconnaissance of Road Networks . . . 7

**3 Packing Configuration for Cargo Containers [Mission A]** **9**

3.1 Evaluation-$Q$ factor . . . 9

3.2 Algorithm . . . 10

3.3 Analysis and model simplification . . . 10

3.4 Implementation details of
Container Fulfilling strategy using Genetic algorithm . . . 11

3.5 Algorithm analysis . . . 13

3.6 Experiments and results . . . 13

3.7 Another method with reinforcement learning . . . 14

**4 Drone Fleet Design [Mission C]** **15**

4.1 Medical drones [Mission C.(a)] . . . 15

4.2 Reconnaissance drones [Mission C.(b)] . . . 16

**5 Sensitivity Analysis** **18**

**6 Modified Model** **18**

**7 Strengths and Weaknesses** **20**

**8 Memo** **20**

**Appendices** **22**

**Appendix A Matlab Codes** **22**

A.1 drone.m . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
A.2 evolve_global.m . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
A.3 evolve_scope.m . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
A.4 evolve.m . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
A.5 main.m . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

# 0 Introduction

## 0.1 Background

On September 20, 2017, Super Hurricane "Maria" attacked Puerto Rico. The strongest hurricane in the past 90 years has caused serious damage to local infrastructure. [1]

The hurricane destroyed most of Puerto Ricos buildings and roads. In particular, the east coast of the hurricane transit has been hit hard. The hurricane destroyed almost all of the poles and transmission routes, causing the island's power supply to be nearly completely interrupted. At the same time, the mobile communication system has suffered a devastating blow, and normal communication has been impossible. To make matters worse, the floods caused by the hurricane drowned many highways and roads in Puerto Rico, so the ground traffic on the island was basically paralyzed.

After the hurricane transited, there was a shortage of medical supplies due to the widespread increase in demand for medical services on the island. [2] Accelerating the delivery of medical supplies to designated hospitals is a matter of urgency. At the same time, dredging traffic is also a top priority. In order to accomplish the above two goals, one non-governmental organization (NGO) decided to send drones to support.

## 0.2 Restatement of the Problem

HELP, Inc. is going to design a transportable disaster response system called DroneGo. DroneGo will use rotor wing drones to deliver pre-packaged medical supplies and provide high-resolution aerial video reconnaissance. There are 7 candidate rotor wing drones choose from to build its DroneGo fleet. Drones will carry three different medical packages referred to as MED1, MED2, and MED3 within drone cargo bays for delivery to selected locations. HELP, Inc. will use International Standards Organization (ISO) standard to pack less than 3 individual shipping containers with dronescargo bays and medical packages. These three containers can be transported to up to three locations with the least amount of space left.

## 0.3 Assumptions

One of the key problems is the placement of objects. If we place medical supplies arbitrary, it may cause issues such as leakage of drugs and the damage of the wings of drones. What's more, the placement of the warehouse may affect its load-bearing capacity, and the door of the container also should be fixed. During delivery, the warehouse requires top loading, thus cannot be placed sideways or upside down. As result, we do not allow the invert or side of all items (including Medcial Package, Drone/UAV, Drone Cargo Bay(DCB), and Cargo Container/Shipping Container), but rotations about the vertical direction are allowed.

Moreover, we do not allow pack medical package into Drones Cargo Bay then put them into the containers, though it can increase the number of medical package. If so, we have to fill buffer materials into the DCB, which will make the problem more complicated. On the other hand, this operation will This operation will increase the weight of the cargo and may exceed the carrying capacity.

Another important problem is the flight of drones. Although the hurricane will cause power outages in most areas, we assume that drones can and only can be charged at the locations of cargo containers and at the delivery locations. In addition, we suppose that the relationship between charge and charge time is linear.

## 0.4 Notations

To aviod ambiguity, we list the sets of various objects and positions at Table 1

| | |
|---|---|
| $\mathbb{I} = \{\text{I,II,III}\}$ | Cargo Containers. By default, $i$ is used to represent elements |
| $\mathbb{J} = \{\text{A,B,C,D,E,F,G,H}\}$ | Drones/UAV. By default, $j$ is used to represent elements |
| $\mathbb{K} = \{1, 2, 3\}$ | Medical Package. By default, $k$ is used to represent elements |
| $\mathbb{L} = \{\text{i,ii}\}$ | Drone Cargo Bay. By default, $l$ is used to represent elements |
| $\mathbb{P} = \{a, b, c, d, e\}$ | Delivery Location. By default, $p$ is used to represent elements |

Table 1: Sets for objects and positions

Table 2 list the symbols for some important quantities

| Symbol | Quantity (unit) |
|---|---|
| $L$ | Length (in) |
| $W$ | Width (in) |
| $H$ | Height (in) |
| $V$ | Volume (in$^3$) |
| $G$ | Weight or Max Payload Capability (lbs) |
| $N$ | Number |
| $n$ | Number |
| $v$ | Velocity (km/h) |
| $t$ | Flight Time No Cargo (min) |
| $r$ | Flight radius (km) |
| $\phi$ | Latitude (°) |
| $\lambda$ | Longitude (°) |
| $x$ | Abscissa (km) |
| $y$ | Ordinate (km) |
| $S$ | Distance (km) |

Table 2: Symbols for quantities

For instance, $V_i$ ($i \in \mathbb{I}$) presents the volume of cargo containers, $v_{\text{A}}$ is the velocity of Drone A, which is $40\,\text{km/h}$, and $n_1^a$ means the number of MED 1 sent to Deliver Location $a$ (i.e. Caribbean Medical Center)

# 1 Analysis of the Problem

There are three main missions:

A. Container problem: Recommend a drone fleet and set of medical packages, and design the associated packing configuration for each cargo containers.

B. Placement problem: Identify the best locations on Puerto Rico

C. Drone problem: (a)Provide the drone payload packing configurations delivery routes and schedule for medical UAVs; (b) Provide a drone flight plan for reconnaissance UAVs.

Here medical UAVs are the drones whose mission is to deliver medical packages to hospitals, and reconnaissance UAVs are drones whose mission is to assess the highways and roads. The reason why we made this classification is the daily medical requirements is not too many, so a few drones can satisfy those requirement.

Therefore, we will determine the type(s) of the medical drones from the demands of the hospitals, and then determine the location of cargo containers based on the distance between the delivery locations, the flight range of the drones and the reconnaissance requirements. [Mission B]

Then, we will evaluate the packing configuration for containers from three aspect: how much the medical supplies are adequate, how much the reconnaissance missions are carried out, and the cost control of buffer materials. Combine other constraint, the remaining problem can be turned into a multi-objective multi-constrained optimization problem [Mission A]

After that, we design packing configurations, delivery routes and schedule for medical UAVs [Mission C.(a)], and provide a plan for reconnaissance UAVs after assigning weight to every road[Mission C.(b)].

# 2 Locations Identification [Mission B]

## 2.1 Plane of disaster area and coordinates of delivery locations

Althougy the surface of the earth is approximately spherical, the latitude differences between the five delivery locations are not larger than $0.25°$, and their longitude differences are not larger than $1.08°$, hence we can regard this aera as a plane, and we can derive the Cartesian coordinates from the latitudes and longitudes. We will check this error later.
Take $\phi = \phi_b = 18.22°$ as $x$-axis, and $\lambda = \lambda_e = -66.73°$ as $y$-axis. Strictly speaking, the transformation formula for $x$ of each position should be related to its latitude. However, due to the small latitude differences, we can take a median latitude $\bar{\phi} = 18.34°$ as a fixed value for all positions. After that, the transformation formulas for $x$ and $y$ from $\phi$ and $\lambda$ are

$$\begin{cases} x = R_{\oplus} \cos \bar{\phi} (\lambda - \lambda_e) \\ y = R_{\oplus} (\phi - \phi_b) \end{cases} \tag{1}$$

where $R_{\oplus} \approx 6371\,\mathrm{km}$ is the (average) radius of the Earth. With the formula (1), we can compute the Cartesian coordinates for 5 delivery locations (and any other position in this aero), which are listed in table 3.

| Location Name | $x$(km) | y(km) |
|---|---|---|
| Caribbean Medical Center (a) | 113.99 | 12.23 |
| Hospital HIMA (b) | 73.88 | 0 |
| Hospital Pavia Santurce (c) | 69.66 | 24.46 |
| Puerto Rico Children's Hospital (d) | 60.16 | 20.01 |
| Hospital Pavia Arecibo (e) | 0 | 27.80 |

Table 3: Cartesian coordinates for delivery locations

As a test, we take the two delivery locations farthest apart, point $a$ and point $e$. With the spherical distance formula

$$\widehat{ae} = R \arccos \left[ \cos \phi_a \cos \phi_e \cos(\lambda_a - \lambda_e) + \sin \phi_a \sin \phi_b \right] \tag{2}$$

the spherical distance is $\widehat{ae} = 115.14\,\mathrm{km}$. Their distance based on Cartesian coordinates is $|ae| = 115.05\,\mathrm{km}$, the relative error is $-0.078\%$, which supports our approximation.

## 2.2 Medical Demands and Drones' Domain

Every delivery location require MED 1 and/or MED 3. Unfortunately, they cannot be packed into DCB 1. Assume we only use Drones C, the Drones which can fly farthest with DCB 2, to deliver medical packages. The maximum no cargo flight distance of Drone C is $r_{\mathrm{C}} = v_{\mathrm{C}} t_{\mathrm{C}} \approx 37.33\,\mathrm{km}$. The load may decrease the flight distance, so we suppose the coefficient is $85\%$, so the Drones C can fly at most $r'_{\mathrm{C}} \approx 31.0(73)\,\mathrm{km}$ with load. Then set five delivery locations as centers of circles, and $r'_{\mathrm{C}} \approx 31\,\mathrm{km}$ as the radius, (the circle of center $p$ is denoted as $\bigcirc_p$, $p \in \mathbb{P}$) as shown in figure 1

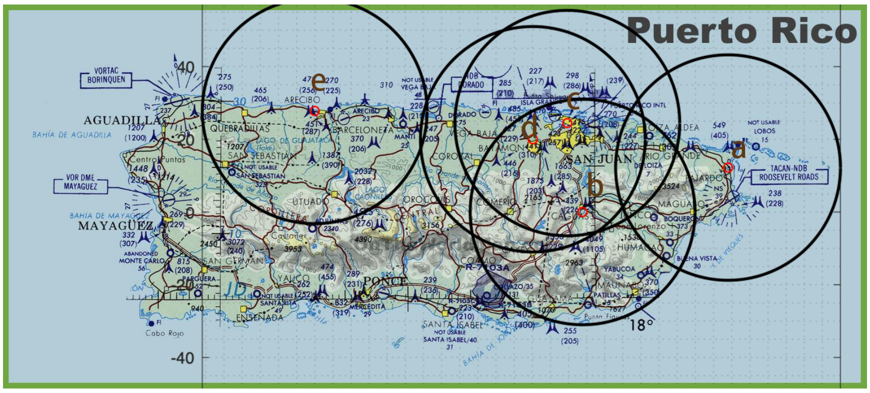


Figure 1: Circles around delivery locations

### 2.3 Reconnaissance of Road Networks

We hope all domains are unicom, so that the drones can move between all locations, which can not only enhance the medical circulation, but also simplify the problem. Based on this purpose, the first cargo container should be located at $\bigcirc_d \cap \bigcirc_e$. Since this intersection is not very big, the coordinates of container I can be set as

$$\begin{cases} x_{\mathrm{I}} = (x_d + x_e)/2 = 30.08\,\mathrm{km} \\ y_{\mathrm{I}} = (y_d + y_e)/2 = 23.90\,\mathrm{km} \end{cases} \tag{3}$$

For the second container, we need to consider that we can transport medicines for hospital $a$. So we need to pick a location in the area $\bigcirc_a \cap (\bigcirc_b \cup \bigcirc_c \cup \bigcirc_d)$. At the same time, we can't ignore the fact that the selected location is better on the main route, which has great benefits for the transportation of our medicines and the exploration of routes. Taking into account the above, we have chosen the most suitable location:

$$\begin{cases} x_{\mathrm{II}} = 91.08\,\mathrm{km} \\ y_{\mathrm{II}} = 17.74\,\mathrm{km} \end{cases} \tag{4}$$

After determining these two positions, the domain is shown in figure 2, where two bright dots are the position of the first and second cargo containers

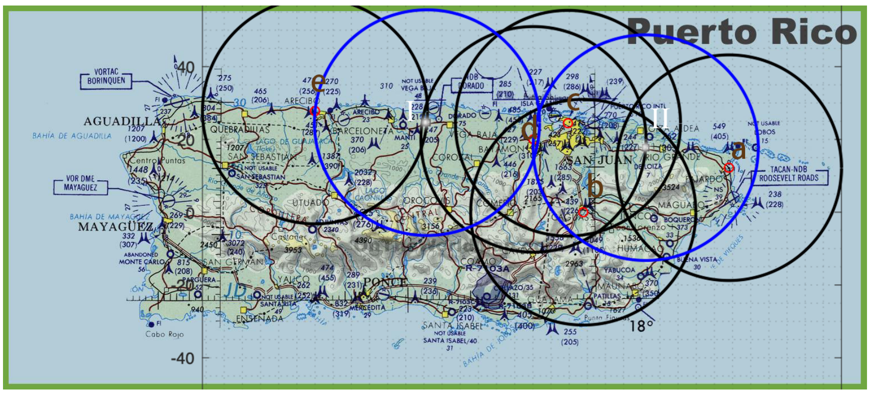


Figure 2: Circles around locations of hospitals and containers

At last, we need to determine the position for the third drones. The algorithm is quite simple but effective consisting with 3 crucial part shown in A.1. First, we define the weights for every roads according to its geological properties like position, population and topographical information. Basically, the roads in the north is more important since most populations are located around San Juan port and the terrain is relatively flat, with more important functionality.Second, we obtain the potential domain of the third point according to the former 2 decided position. Third part, we traversal all points in the domain and calculate value of the point with the largest value. The value is defined to be the sum of the weights of points in roads that drones from the third point can reach. And the third point is shown in 3, whose coordinates is shown in (5)

$$\begin{cases} x_{\mathrm{III}} = 62.02\,\mathrm{km} \\ y_{\mathrm{III}} = 6.886\,\mathrm{km} \end{cases} \tag{5}$$

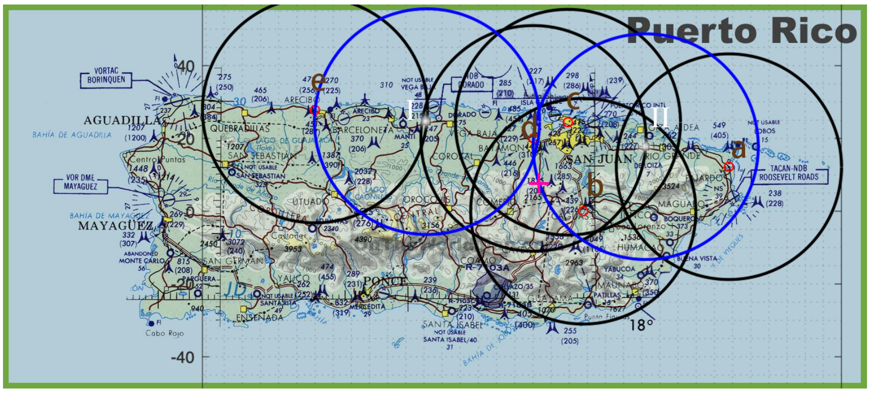


Figure 3: The final containers

Through the selection of the above three points, we finally determined the location of the three containers. It is worth mentioning here that the reason we choose three different placement points is that we hope that our drones can cover a larger range. We have also implemented this principle when we choose to drop. First, we tend to choose locations near the road for greater mobility. Second, we hope to cover more roads, in order to improve the scope of the search.

At the same time, our models and the principles we think about are easily applied to other rescue operations. This reflects the flexibility and adaptability of our model.

# 3 Packing Configuration for Cargo Containers [Mission A]

## 3.1 Evaluation-$Q$ factor

We have three main purpose: medcial supply, video reconnaissance of road networks, and minimize the use of buffer materials.

For the last term, we evaluate it with the volume of unused space. The less the remaining space, the lower the cost of the cushioning material. We define

$$Q_{\mathrm{V}} := \frac{1}{V}\sum_{i=\mathrm{I}}^{\mathrm{III}} V_i \tag{6}$$

For the medcial supply, the proper evaluation is how long the storage of medicine can fulfill all the hospitals. Note that if the duration is long enough, its further improve is not so effective. For example, suppose our medicine can support all hospitals for one year. It is not much better than the case of three months, because the road may be rebuild in three months. Hence we construct

$$Q_{\mathrm{M}} := 1 - \exp\left[-\kappa_{\mathrm{M}} \min\left(\frac{N_1}{n_1}, \frac{N_2}{n_2}, \frac{N_3}{n_3}\right)\right] \tag{7}$$

Here $n_k$'s are the total daily demands of MED $k$, whose values are listed at table 6

| Type of medical packages | Total daily demands $n_k$ |
|---|---|
| MED 1 | 7 |
| MED 2 | 2 |
| MED 3 | 4 |

Table 4: Values of $n_k$

and $N_k$'s are total number of MED $k$ packed in all containers

$$N_k = \sum_{i=\mathrm{I}}^{\mathrm{III}} n_k^i$$

The dimension of parameter is $\dim \kappa_{\mathrm{M}} = [T]^{-1}$, which presents the expected time that hospitals require our drone fleet to support.

The video reconnaissance of road networks is hard to evaluate. It should be related with the number of drones. More specificly, it depends on the reconnaissance frequency. The function should be non-linear, because if the frequency is high enough, its further improve is not very effective. Based on this characteristic, we assume the function form is also $1 - \mathrm{e}^{-\kappa_f f}$. Further more, the flight distance $r_j = v_j t_j (j \in \mathbb{J})$ is a decisive factor. The flight range for all UAVs are listed in table 5 (Note that UAV F is vedio capable)
We have determined the position of cargo containers. For every UAV, if it can fly from one base location (include delivery locations and containers locations) to another through a road, the road will be called bridge road. Otherwise, the road will be called branch road. The total flight distance for drone type $j$ during a loop (which may depends on the location of its container) is

$$S_j^i = s_j^i + n_j^i r_j$$

| Type of UAV | Flight range $r_j$ (km) |
|---|---|
| A | 23.33 |
| B | 52.67 |
| C | 37.33 |
| D | 18 |
| E | 15 |
| F | 31.6 |
| G | 17.07 |

Table 5: List of $r_j$

The first term $s_j^i$ is the total length of bridge roads (depands on the type of UAV and its container's location), and $n_j^i$ is the number of branch roads.

Note that the work of UAV with less flight range is repetitive, when we count the number of drones, we should also take those drones with larger flight range. In summary, the $Q$ factor is

$$Q_{\mathrm{D}} := \frac{1}{3\times 7}\sum_{i=\mathrm{I}}^{\mathrm{III}}\sum_{j\in\mathbb{J}\setminus\{\mathrm{F}\}}\left[1-\exp\left(-\kappa_{\mathrm{D}}\frac{v_j}{S_j^i}\sum_{j'|r_{j'}\geq r_j} N_j^i\right)\right] \tag{8}$$

The dimension of $\kappa_{\mathrm{D}}$ is $\dim\kappa_{\mathrm{D}} = [T]$, which presents the inverse of expected reconnaissance frequency.

The range for three $Q$-factor defined by (6),(7) and (8) are $[0,1)$. We define the comprehensive $Q$-factor as

$$Q := Q_V \cdot Q_M \cdot Q_D \tag{9}$$

To simplify the problem, we consider every container seperately, so that we can ignore $\sum_i$. We will take the first container as an example.

## 3.2 Algorithm

To begin with, it does not hurt to recall the problem we are facing here. We have 8 kinds of drones, 2 types of cargo bays for the drones to carry medicines and 3 kinds of medicine in total. Our mission is to use ISO sized container to give a plan to organize a certain of number of each kind of those objects above, and we have 3 criteria to evaluate our model, which are to evaluate the space left in the container, the performance of road exploring and the performance of medical assistance.

## 3.3 Analysis and model simplification

We need to provide an algorithm to give the best list of objects with its best strategy to pack them in the container. To simplify the problem, we can divide the problem into two parts, one to provide the name list and the other is to provide the best strategy to pack them into the container. To be more specifically, we can use extra objects beyond the name list to try to fulfill the container assuring all the objects with at least the number stated on the name list contained to the container. We can consider this problem as a decision problem with a very large state scale.

Our basic idea is to give a strategy to fulfill the container. The strategy provide a name list of objects and the direction and position where they are placed in the container. So after execute the complete strategy, we would get a filled container and its objects. Then we can evaluate the strategy separately. We can iterate the strategies using the evaluation of previous trial of strategy to amend the strategy or the algorithm to generate the strategy and in the end, we would get a good enough solution to the problem. Although we would like the empty space after fulfilling the container to be as little as possible, this is after all merely one of three criteria and we consider it a less important one.

Besides all the objects are normally needed in a relatively large number, therefore, to reduce the time complexity of the problem, we would like to simplify the first sub problem, fulfilling the container to leave as little empty space as possible, to a lower dimension which becomes fulfilling a rectangular area with little rectangles leaving as little empty area as possible. In the setting of the problem, we think that all of the objects are not capable to rotate when placed in the container. The reason is that many medicines, especially injection does and other medicine in glass containers are not capable to rotate the box in the process of transportation [3] and for the drones, most brands of those requires to be placed with one direction pointing up [4]. As for the cargo bays, in the problem statement, there are statements about the maximum weight it can carry and therefore we consider its strength is limited and we decide to keep its upper direction fixed. Therefore, the fulfilling question finally becomes that using 13 different types of little rectangles, which can rotate in the 2-dimension plane, to fulfill the big rectangular area.

According to the idea, we come up with two different algorithm, one combined with Genetic Algorithm and another inspired by reinforcement learning. [5] The algorithm with the idea of genetics would be stated with more details beneath.

### 3.4 Implementation details of Container Fulfilling strategy using Genetic algorithm

The algorithm is implemented with MATLAB with source codes attached A.2∼A.5

We use a matrix sized (2, 200) to denote a strategy. Each gene in the first row of DNA represent the kind of object and the second row represent the pose of the corresponding object. The line number of a gene in the DNA represent the sequence of placing the object. After initialize the DNA base, we put all the DNA into evolution part, in which, we use corner priority strategy to place the objects sequentially until the end of the DNA. After that we obtain a Qvalue (means Quality Value of a DNA) with the three Q functions stated in (6),(7) and (8). At the end of one epoch, we would have a Qvalue for each DNA sequence respectively. Sort the DNA with key as Qvalue in descending order, weed out half of the DNAs with smaller Qvalues and update them by combining two random DNAs in the half of DNAs with larger Qvalues. In the process of updating DNAs, there would be possibility for a certain gene or genes in DNA mutating. Repeat this process until convergence or a number large enough.

Pseudocode:

```
// initialize DNA base
DNAs = initDNAs;
// initialize list of all Qvalue
Qdnas = zeros(DNAn,1);
repeat epoch {
    for dna in DNAs {
        // put DNA into evolution
        Qdnas(i) = evolve(dna);
    }
    // Sort the DNA with key as Qvalue
    [Qdnas,Ps] = sort(Qdnas, 'descend');
    for dna in {half Of DNAn with smaller Qvalue} {
        // Update DNA with small Qvalue
          DNAs(dna) = genDNA(DNAs);
    }
}
```

In the process of evolution, we use a matrix m to represent the rectangular area, whose size is (231, 92). Each element in m represents its corresponding area unit whether is occupied or occupied by which kind of object and we keeps a matrix bitm to trace the number of objects placed into the container. The flot chart is shown in figure 4

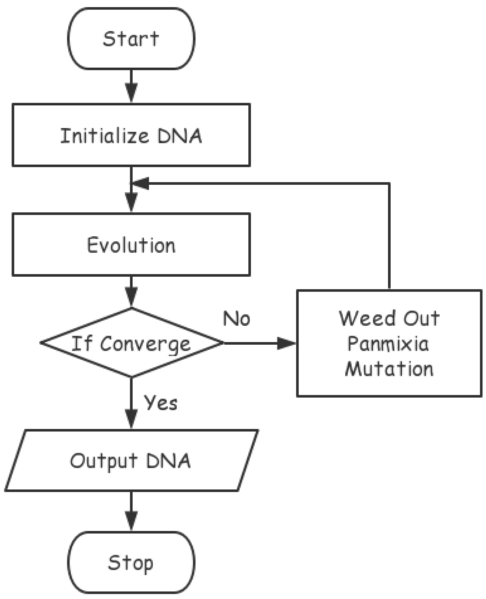


Figure 4: Flot chart of algorithm

### 3.5 Algorithm analysis

The algorithm is apparently quite efficient because only the evolution part is a little time consuming and the rest part is basically $O(n)$, with n being the scale of the DNA base. The time complexity of the evolution part is basically $O(nm)$ with n being the scale of the DNA base and m being the length of a DNA sequence. Therefore the efficiency of calculating one epoch is acceptable. As for the speed of convergence, as we observed in experiments, the speed is also quite acceptable as shown in figure 5

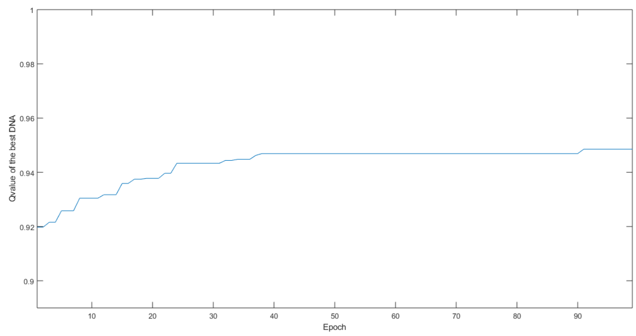


Figure 5: Convergence speed

### 3.6 Experiments and results

Here is a result of an experiment of 100 epoch as shown in figure 6 [**??** and we think this result is quite good and psychological plausible. First the number of drone B is quite large and there are no drone D, drone E and drone F which is a really good decision because drone B has a relatively small volume and long distance capability and on the contrary, drone D, drone E and drone F would seem like a burden with large volumes and really short fight range. Besides, among them drone F even does not have video capability. However, the number of cargo bay type1 is extraordinarily large, but we think it is also a good choice because of its small size, using it for try to take as much space as possible without taking the position of other objects. And the amount of the medicine, constrained by $Q_{\mathrm{M}}$, can assure to provide medicine for this area for at least half a year.

| $n_{\mathrm{A}}$ | $n_{\mathrm{B}}$ | $n_{\mathrm{C}}$ | $n_{\mathrm{D}}$ | $n_{\mathrm{E}}$ | $n_{\mathrm{F}}$ |
|---|---|---|---|---|---|
| 3 | 20 | 6 | 0 | 0 | 0 |
| $n_{\mathrm{G}}$ | $n_{\mathrm{i}}$ | $n_{\mathrm{ii}}$ | $n_1$ | $n_2$ | $n_3$ |
| 15 | 42 | 4 | 396 | 270 | 230 |

Table 6: Values of $n_k$

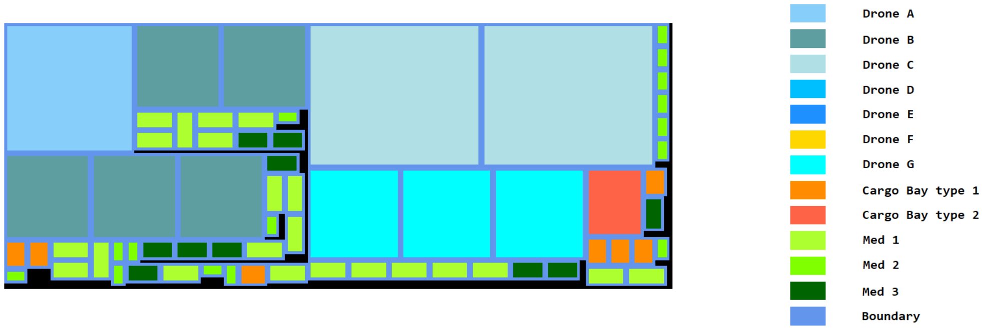


Figure 6: Configuration of containers

### 3.7 Another method with reinforcement learning

Idea of this algorithm comes from our basic idea that we do not plan for the total strategy at the beginning but give a policy to take action to a particular state. Therefore, we come up with the idea of using reinforcement learning, with the reward of closure state calculated by the same means in genetic algorithm using the three criteria $Q_\text{V}$, $Q_\text{M}$ and $Q_\text{D}$ and other $Q$ value initialized as 0. When converging, we would obtain the best policy for placing the objects into the container.

# 4 Drone Fleet Design [Mission C]

## 4.1 Medical drones [Mission C.(a)]

At section determining the locations of Cargo containers and deliver routes, let's estimate how many medical packages a drone bay can transport. There are two constraint, space of cargo by and payload capability of drone. The volume of DCB 2 is $V_{\text{ii}} = 9600 \text{ in}^3$, and the volumes for three types medical packages are $V_1 = 490 \text{ in}^3, V_2 = 200 \text{ in}^3, V_3 = 336 \text{ in}^3$, so roughly speaking, one DCB 2 can be packed into 19 MED 1, or 48 MED 2, or 28 MED 3. On the other hand, the maximum payload capability of Drone 3 is $W_{\text{C}} = 14\text{lbs}$, thus one Drones can only deliver 7 MED 1, or 7 MED 2, or 4 MED 3. Therefore, the payload capability is the real constraint

$$\sum_{k=1}^{3} n_k^p W_k \leq W_{\text{C}} \tag{10}$$

Next, we will discuss the possible packing configurations for each deliver location.

**Caribbean Medical Center (MED 1 + MED 3):** In order to make $n_1/n_3$ as close to 1 as possible, we can pack 2 MED 1 and 3 MED 3, or 4 MED 1 and 2 MED 3. That is

$$\begin{cases} n_1^a = 2 \\ n_3^a = 3 \end{cases} \text{or} \begin{cases} n_1^a = 4 \\ n_3^a = 2 \end{cases} \tag{11}$$

For 2 MED 1 and 3 MED 3 combination, we can pack two MED 1 at the first layer, Let their $L$-$H$ planes contact to each other. Then pack three MED 3 at the second layer.
For 4 MED 1 and 2 MED 3 combination, at the first and second layer, we can pack 2 MED 1. Then pack 2 MED 3 at the third layer.
At the first two days, we can deliver 2 MED 1 and 3 MED 3, then deliver 4 MED 1 and 2 MED 3 at the next day, 3 days as a period, which can meet the medical center's eight-day needs.
In fact, there is some other plans, e.g. 1 MED 1 and 4 MED 3, and adjust the schedule to make the average ratio close to 1.

**Hospital HIMA (2×MED 1 + MED 3):** To this location, 4 MED 1 and 2 MED 3 is the best conbination

$$\begin{cases} n_1^b = 4 \\ n_3^b = 2 \end{cases} \tag{12}$$

The configuration has beed discussed before.
The schedule is to deliver once every two days.

**Hospital Pavia Santurce (MED 1 + MED 2):** To this location, 4 MED 1 and 3 MED 2, or 3 MED 1 and 4 MED 2 are the best configurations

$$\begin{cases} n_1^c = 4 \\ n_2^c = 3 \end{cases} \text{or} \begin{cases} n_1^c = 3 \\ n_2^c = 4 \end{cases} \tag{13}$$

For 4 MED 1 and 3 MED 2 combination, we can put 2 MED 1 at the first and second layer. Then put 3 MED 2 at the third layer.
For 3 MED 1 and 4 MED 2 combination, we can put 2 MED 1 at the first layer, 4 MED 2 at the second layer, and the last MED 1 at the top.
We can alternately deliver two combinations. Every two-day delivery can meet the hospital's sever-day needs.

**Puerto Rico Children's Hospital (2×MED 1 + MED 2 + 2×MED 3)** For this location, the conbination is just 2 MED 1, MED 2 and 2 MED 3

$$\begin{cases} n_1^d = 2 \\ n_2^d = 1 \\ n_3^d = 2 \end{cases} \tag{14}$$

All medical package can be put in one layer.
It should be delivered daily.

**Hospital Pavia Arecibo (MED 1)** We can pack 7 MED 1 to this location at a time

$$n_1^e = 7 \tag{15}$$

We put 2 MED 1 at the first three layer, and the last MED 1 is put at the top.
Each delivery can meet the needs of the hospital for a week.

The schedule can be summaried as table 7 (the format is $n_1 + n_2 + n_3$)

## 4.2 Reconnaissance drones [Mission C.(b)]

**Considerations for the detection of road conditions**
First we identify the main routes of each route and assign them the importance factor $\omega_q$ for the pair. This importance factor is set in consideration of the following points:

- Whether the route is the main road.
- Whether the route is in a populated area of the population.

The specific roads map can be given by the figure 7, and the importance factors are listed in table 8

| | Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|---|
| a | | 2+0+3 | | 2+0+3 | | 4+0+2 | |
| b | 4+0+2 | | 4+0+2 | | 4+0+2 | | 4+0+2 |
| c | | 3+4+0 | | | | 4+3+0 | |
| d | 2+1+2 | 2+1+2 | 2+1+2 | 2+1+2 | 2+1+2 | 2+1+2 | 2+1+2 |
| e | | | | 7+0+0 | | | |

Table 7: Schedule for medical UAV


Figure 7: Specific roads map

| Roads | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Weight | 2 | 1 | 1 | 1 | 3 | 1 |
| Roads | 7 | 8 | 9 | 10 | 11 | 12 |
| Weight | 1 | 4 | 1 | 2.5 | 2.5 | 3 |
| Roads | 13 | 14 | 15 | 16 | 17 | |
| Weight | 1 | 4 | 2 | 1 | 2 | |

Table 8: List of $\omega_q$

**Preliminary consideration** Then we decided to allocate the route problem in the last model of the last consideration. Think of it this way: we assume the number of various aircraft $n_q$ on each route (the idea here is very clever, we calculate on average what kind of aircraft there are on the road, so there is no need to exchange the aircraft everywhere), then Multiply the route importance factor $\omega_q$. Also need to consider that the efficiency degradation factor caused by too many drones on one route can be expressed as $1/\sqrt{n_q}$

$$\Omega = \sum_q \sqrt{n_q}\omega_q \tag{16}$$

**Drone flight plan design** The above is our preliminary consideration. However, we have noticed several facts:

1 In our solution to the A problem, we have found that the flight distance of the aircraft B/C is long enough to see it can travel between hospitals and container points.

2 Second, other short-range drones are not able to make round trips between stops. So they can only explore near the medical point, so the path that can be explored will be halved.

3 The number of drones that carry the demand for drugs is extremely small and can be ignored. In fact, the result shows that most of drodes are type B and C.

Thus our route plan is:

- For drone B/C: We assign all drones B in all three containers to each route according to the proportion of all route weights $\omega_q$.
- For drones A, D, E, G: We assign them by weight (by the location of the container).

# 5 Sensitivity Analysis

In these three problems, the placement problem and drone problem are not parameter sensitive, so we try to alaysis the sensitivity for container problem. The main parameters in this problem are $\kappa_{\mathrm{M}}$ and $\kappa_{\mathrm{D}}$. We have set $\kappa_{\mathrm{M}} = (10\,\mathrm{day}^{-1})$ and $\kappa_{\mathrm{D}} = 0.5\,\mathrm{h}$. We adjust $\kappa_{\mathrm{M}}$ from $(8\,\mathrm{day}^{-1})$ to $(12\,\mathrm{day}^{-1})$, the results are shown in table 9

| $\kappa_{\mathrm{M}}$ | $n_{\mathrm{A}}$ | $n_{\mathrm{B}}$ | $n_{\mathrm{C}}$ | $n_{\mathrm{D}}$ | $n_{\mathrm{E}}$ | $n_{\mathrm{F}}$ | $n_{\mathrm{G}}$ | $n_{\mathrm{i}}$ | $n_{\mathrm{ii}}$ | $n_1$ | $n_2$ | $n_3$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1/8 | 6 | 12 | 6 | 0 | 0 | 0 | 15 | 30 | 8 | 306 | 270 | 207 |
| 1/9 | 3 | 20 | 6 | 6 | 0 | 3 | 5 | 84 | 4 | 234 | 234 | 138 |
| 1/11 | 6 | 12 | 6 | 6 | 3 | 0 | 5 | 54 | 12 | 288 | 180 | 207 |
| 1/12 | 9 | 12 | 3 | 3 | 0 | 0 | 10 | 54 | 12 | 378 | 252 | 230 |

Table 9: Results for different $\kappa_{\mathrm{M}}$

It can be found that our model is not sensitive.

# 6 Modified Model

### Part A

1 The first point is the loss problem. Because the weather conditions and road conditions after the disaster in Puerto Rico are very complicated, we need to consider that we will have losses when preparing materials for them. This suggests that we need to provide redundancy for the drones, cargo bays and medical packages. This means that in practice the medicines may be held for a shorter period of time and the drone will take longer to explore.

2 The second point is the impact of an uncertain increase in demand that may exist. For example, there may be an increase in the demand for certain drugs due to the prevalence of certain infectious diseases, or in order to accurately locate a certain route, we need more drones to shoot. The second point of consideration is similar to the first one. It also reminds us to provide some elements that can be adjusted.

3 By consulting the relevant website, we have the following information about the tethered drone. Tethered drones are powered from the ground while sensors and surveillance in the air leads to a wide variety of different payload options. The tethered drone can perform many tasks including disaster recovery, tornadoes, earthquakes, floods, etc. Tethered drones could greatly assist Us through improved surveillance capabilities, persistent surveillance and communication. [7] It is necessary to keep a tethered drone in our three container locations.

### Part B

We spent a lot of our time thinking about the choice of locations cargo containers.

1 First, we carefully searched and searched the port area along the northeast coast of Puerto Rico and found that there is only two portsSan Juan Port and Port of

Mayagüez in San Juan Bay that can provide container transportation. However, this is clearly not enough to meet the demand for transporting our goods to two hospitals farther away. [6]

2 Due to the difficulty of shipping in other regions, we made the assumption that the container was dropped by airdrop. Therefore, we have a greater degree of freedom in choosing the location of the container.

3 We can get one container from San Juan Bay into Puerto Rico by sea. And the remaining two are by airdrop to their locations.

4 If in other cases we need to ship more containers, then the consideration of transportation costs is critical.


Figure 8: Position of San Juan Bay

## Part C

1 When considering the flight path of a drone, a very important consideration is the charging time of the drone. We worked hard to find the relevant information, but we unable to match the drone given in the data to the charging time. Therefore, consideration of this point in our model is not enough.

2 Terrain is also an important factor limiting the performance of drones. We only roughly considered this part of the model. However, if more precise routes are required, we need to take into account the details of the terrain. Of course, it will bring a huge amount of work.

3 The last point is that we cannot predict the future weather conditions in the area. However, we can give different adjustments for different weather.

# 7 Strengths and Weaknesses

## Strengths

- Match the true situation: In the model, we take into account the realities encountered in the real world and the additional terms that need to be provided in order to make the entire system work well.
- High-quality factors: When we built the model, we took into account the impact of our choices on various aspects and selected a high-quality evaluation factor.
- Flexible: The entire model is not completely closed, and we have enough room for adjustment. There are a lot of parameters that can be adjusted, which is convenient for us to apply to other types of rescue operations.
- Algorithm: The various algorithms used in the model (such as genetic algorithms) and the algorithms that can be further used (such as enhanced learning algorithms) are all very effective algorithms that have been tested.
- Scientific: We creatively use the method of transformation to build the whole model, referring to the way physicists think about new things: using appropriate simplification to simplify the model, using perturbation to make the model closer to the real situation.

## Weaknesses

- Due to the limited information provided by the topic, some factual assumptions have to be added.
- Because the globally considered model has too much demand for computer capabilities and a huge increase in overall complexity, we have chosen to simplify the model to weaken the correlation of various factors.
- A certain adjustment range is given to make the model more applicable, but at the same time it may not be conducive to its performance in specific problems.

# 8 Memo

Dear CEO,
In accordance to your requests, our team use the situation in Puerto Rico to design a DroneGo disaster response system that will fit within the containers noted while meeting the anticipated medical supply demands during a potential similar future disaster scenario.

Taking into account the complexities we may face and the limited information we know, we have made reasonable assumptions after taking into account the actual situation (of course, these assumptions can be adjusted with different realities):123

There are many trade-offs among these models, the most important of which are three.

- Trade-offs between drugs and drones. We have carried the number of drones as much as possible while keeping the drugs we carry for more than 4 months.
- Trade-offs between filling containers and carrying better materials. We hope to bring more excellent drones, but it will also damage the remaining space of our containers.
- Trade-offs between different roads. Since different roads are of different importance in rescue, we must make a choice.

Based on the model we built, here are the results we gave:123

Finally, in the design of the model we are very concerned about the universality of the model. Therefore, when it comes to different needs of different rescue operations, it is easy to give reliable conclusions by changing our parameters. At the same time, we also provide a lot of potential model corrections, which increase the flexibility of the model.

Sincerely,
Team# 1915522

# References

[1] Zorrilla, C. D. (2017). The view from puerto rico hurricane maria and its aftermath. The New England Journal of Medicine, 377(19), 1801-1803. doi:http://dx.doi.org/10.1056/NEJMp1713196

[2] Kishore, N., Marqués, D., Mahmud, A., Kiang, M. V., Rodriguez, I., Fuller, A., . . . Buckee, C. O. (2018). Mortality in puerto rico after hurricane maria. The New England Journal of Medicine, 379(2), 162-170. doi:http://dx.doi.org/10.1056/NEJMsa1803972

[3] Scheler, S., Saupe, S., Herre, A., & Fahr, A. (2010). Preservation of liquid drug preparations for oral administration. Journal of Pharmaceutical Sciences, 99(1), 357-367.

[4] https://www.dji.com/cn

[5] Bertsekas, Dimitri P.; Tsitsiklis, John (1996). Neuro-Dynamic Programming. Nashua, NH: Athena Scientific.

[6] https://en.wikipedia.org/wiki/Transportation_in_Puerto_Rico#Seaports_&_harbors

[7] https://taskandpurpose.com/military-tethered-drones-applications

# Appendices

## Appendix A Matlab Codes

Here are simulation programmes we used in our model as follow.

**Input matlab source:**

### A.1 drone.m

```
%% Init
% clc;clear;close;
imgNm = 'map_in.png';
img = imread(imgNm);

% figure(1);
% imshow(img);

hsv = rgb2hsv(img);
[x,y,~] = size(img);

%% Retrieve the region
imgin = zeros(size(img));
for i=1:x
    for j=1:y
        t = hsv(i,j,:);
        if t(2) < 0.01 && t(3) > 0.9
            imgin(i,j,1) = 1;
        end
    end
end

% figure(2);
% imshow(imgin);

%% Load the road map
rdmpNm = 'rdmp.png';
rdmp = imread(rdmpNm);

% figure(3);
% imshow(rdmp);

% Fix the blure and normalize
[x,y,~] = size(rdmp);
```

```
hsv = rgb2hsv(rdmp);
for i=1:x
    for j=1:y
        t = hsv(i,j,:);
        if t(2) > 0.5 && t(3) > 0.5
            rdmp(i,j,1) = 255;
            rdmp(i,j,2) = 0;
            rdmp(i,j,3) = 0;
        else
            rdmp(i,j,1) = 0;
            rdmp(i,j,2) = 0;
            rdmp(i,j,3) = 0;
        end

%           if rdmp(i,j,2) > 50 && rdmp(i,j,2) > 50
%           if rdmp(i,j,1) < 40
%               rdmp(i,j,1) = 0;
%           else
%               rdmp(i,j,1) = 255;
%           end
    end
end

% figure(4);
% imshow(rdmp);
rdmp = im2single(rdmp);

%% Calculate the value of several points
vm = 0;
xi = 0;yi = 0;
for i=1:x
    if mod(i,50)==0
        disp(['calculating...',num2str(i),'/679']);
    end
    for j=1:y
        if imgin(i,j,1) == 1
            vans = calQ(i,j,rdmp);
            if vm <= vans
                vm = vans;
                xi = i;yi = j;
            end
        end
    end
end

%% Visualization
% vm
% xi
% yi
wid = 2;
lth = 17;
% imgout = imread('map.jpg');
% imgout = imread('map.png');
imgout = imread('combine.png');
tr = 255;
tg = 0;
tb = 170;
for i = -lth:1:lth
    imgout(xi+i,yi-wid:yi+wid,1) = tr;
    imgout(xi+i,yi-wid:yi+wid,2) = tg;
    imgout(xi+i,yi-wid:yi+wid,3) = tb;
```

```
end
for i = -lth:1:lth
    imgout(xi-wid:xi+wid,yi+i,1) = tr;
    imgout(xi-wid:xi+wid,yi+i,2) = tg;
    imgout(xi-wid:xi+wid,yi+i,3) = tb;
end
figure(16);
imshow(imgout);
imwrite(imgout,'imgout.png');

%% Utility function
function vans = calQ(cx,cy,rdmp)
    R = 100;
    R2 = R^2;
    vans = 0;
    il = max(1,cx-R);
    ir = min(679, cx+R);
    jl = max(1,cy-R);
    jr = min(1531,cy+R);
    for i=il:ir
        for j=jl:jr
            if (i-cx)^2 + (j-cy)^2 < R2
                vans = vans + double(rdmp(i,j,1));
            end
        end
    end
end
```

## A.2 evolve_global.m

```
function Qvalue = evolve(dna)
%% init
    global m
    m = zeros(231, 92);

    global bitm
    bitm = zeros(1, 12);

    global sR
    sR = zeros(12, 3);
    sR(1,:) = [45,45,24];  %A
    sR(2,:) = [30,30,22];
    sR(3,:) = [60,50,30];
    sR(4,:) = [25,20,25];
    sR(5,:) = [25,20,27];
    sR(6,:) = [40,40,25];
    sR(7,:) = [32,32,17];  %G
    sR(8,:) = [8,10,14];  %type1
    sR(9,:) = [24,20,20];  %type2
    sR(10,:) = [14,7,5];  %med1
    sR(11,:) = [5,8,5];
    sR(12,:) = [12,7,4];

    global xylist
    xylist = [1;1];

%% body
    flag = 0;
```

```
    for i = dna
%         pos = decidePos(x, y, i);
        boxtype = i(1);
        pos = i(2);
        [x,y] = decidexy(boxtype);
        switch boxtype
            case 1
                D_A( x, y, pos);
            case 2
                D_B( x, y, pos);
            case 3
                D_C( x, y, pos);
            case 4
                D_D( x, y, pos);
            case 5
                D_E( x, y, pos);
            case 6
                D_F( x, y, pos);
            case 7
                D_G( x, y, pos);
            case 8
                Bay1(x, y, pos);
            case 9
                Bay2(x, y, pos);
            case 10
                Med1( x, y, pos)
            case 11
                Med2( x, y, pos)
            case 12
                Med3( x, y, pos)
            otherwise
                disp('ERROR: WRONG gene');
        end
    end

    if flag == 0
        disp('ERROR: NOT FILLED !!!');
    end

    Qvalue = getQ(bitm);
end


%% Utility functions
function  D_A( x, y, pos)
    Al = 45;
    Aw = 45;
    Ah = 25;
    setM( 1, x, y, pos, Al, Aw, Ah);
end
function  D_B( x, y, pos)
    Bl = 30;
    Bw = 30;
    Bh = 22;
    setM( 2, x, y, pos, Bl, Bw, Bh);
end
function  D_C( x, y, pos)
    Al = 60;
    Aw = 50;
    Ah = 30;
    setM( 3, x, y, pos, Al, Aw, Ah);
```

```
end
function  D_D( x, y, pos)
    Al = 25;
    Aw = 20;
    Ah = 25;
    setM( 4, x, y, pos, Al, Aw, Ah);
end
function  D_E( x, y, pos)
    Al = 25;
    Aw = 20;
    Ah = 27;
    setM( 5, x, y, pos, Al, Aw, Ah);
end
function  D_F( x, y, pos)
    Al = 40;
    Aw = 40;
    Ah = 25;
    setM( 6, x, y, pos, Al, Aw, Ah);
end
function  D_G( x, y, pos)
    Al = 32;
    Aw = 32;
    Ah = 17;
    setM( 7, x, y, pos, Al, Aw, Ah);
end
function  Bay1( x, y, pos)
    l = 8;
    w = 10;
    h = 14;
    setM( 8, x, y, pos, l, w, h);
end
function  Bay2( x, y, pos)
    l = 24;
    w = 20;
    h = 20;
    setM( 9, x, y, pos, l, w, h);
end
function  Med1( x, y, pos)
    l = 14;
    w = 7;
    h = 5;
    setM(10, x, y, pos, l, w, h);
end
function  Med2( x, y, pos)
    l = 5;
    w = 8;
    h = 5;
    setM(11, x, y, pos, l, w, h);
end
function  Med3( x, y, pos)
    l = 12;
    w = 7;
    h = 4;
    setM(12, x, y, pos, l, w, h);
end
function D_H(x, y, pos)
    l = 65;
    w = 75;
    h = 41;
    setM(13, x, y, pos, l, w, h);
end
```

```
function  setM( key, x, y, pos, dl, dw, dh)
    global m
    if pos ~= 0
        pos = 1;
    end
    if pos == 0
        m(x:x+dl-1,y:y+dw-1) = key;
        newco = [x,x+dl;y+dw,y];
    else
        m(x:x+dw-1,y:y+dl-1) = key;
        newco = [x,x+dw;y+dl,y];
    end

    global bitm
    bitm(1, key) = bitm(1, key) + floor(94/dh);

    global xylist
    xylist = [xylist,newco];
    disp('in setM')
    xylist
    xylist = sort(xylist, 2);
end

function [x, y] = decidexy(boxtype)
    global sR
    size = sR(boxtype,:);
    bl = size(1);
    bw = size(2);

    global xylist
    xylist
    disp('size(xylist)')
%     size(xylist)
    size(xylist)
    for i = 1:size(xylist,2)
        xi = xylist(1, i);
        yi = xylist(2, i);
        bans = checkBox(xi, yi, bl, bw);
        if bans == 1
            x = xi;
            y = yi;
            xylist(:,i) = [];
            break;
        end
    end
end

function pos = decidePos(x, y, box)
    global sR
    size = sR(box,:);
    bl = size(1);
    bw = size(2);
    pos = 0;
    % check whether can be placed regularly
    if x+bl > 231 || y+bw > 92
        pos = 1;
    end
    % check whether can be placed rotated
    if x+bw > 231 || y+bl > 92
        pos = 2;
```

```
    end
    if pos == 2

    end
    switch pos
        case 0

        case 1
        case 2
    end
end

function bans = checkBox(x, y, l, w)
    global m
    boxm = m(x:x+l,y:y+w);
    sans = sum(sum(boxm));
    if sans == 0
        bans = 1;
    else
        bans = 0;
    end
end
```

## A.3 evolve_scope.m

```
disp('test');
function Qvalue = evolve(dna)
%% init
    m = zeros(231, 92);
    flag = 0;
    global bitm
    bitm = zeros(1, 12);

%% body
    for i = dna
        pos = i(2);
        switch i(1)
            case 1
                D_A(m, x, y, pos);
            case 2
                D_B(m, x, y, pos);
            case 3
                D_C(m, x, y, pos);
            case 4
                D_D(m, x, y, pos);
            case 5
                D_E(m, x, y, pos);
            case 6
                D_F(m, x, y, pos);
            case 7
                D_G(m, x, y, pos);
            case 8
            case 9
            case 10
            case 11
            case 12
            otherwise
                disp('ERROR: WRONG gene');
```

```
        end

    end

    if flag == 0
        disp('ERROR: NOT FILLED !!!');
        break;
    end
end


%% Utility functions
function mans = D_A(m, x, y, pos)
    Al = 45;
    Aw = 45;
    Ah = 25;
    setM(m, 1, x, y, pos, Al, Aw, Ah);
end
function mans = D_B(m, x, y, pos)
    Bl = 30;
    Bw = 30;
    Bh = 22;
    setM(m, 2, x, y, pos, Bl, Bw, Bh);
end
function mans = D_C(m, x, y, pos)
    Al = 60;
    Aw = 50;
    Ah = 30;
    setM(m, 3, x, y, pos, Al, Aw, Ah);
end
function mans = D_D(m, x, y, pos)
    Al = 25;
    Aw = 20;
    Ah = 25;
    setM(m, 4, x, y, pos, Al, Aw, Ah);
end
function mans = D_E(m, x, y, pos)
    Al = 25;
    Aw = 20;
    Ah = 27;
    setM(m, 5, x, y, pos, Al, Aw, Ah);
end
function mans = D_F(m, x, y, pos)
    Al = 40;
    Aw = 40;
    Ah = 25;
    setM(m, 6, x, y, pos, Al, Aw, Ah);
end
function mans = D_G(m, x, y, pos)
    Al = 32;
    Aw = 32;
    Ah = 17;
    setM(m, 7, x, y, pos, Al, Aw, Ah);
end
function mans = B_1(m, x, y, pos)
    l = 8;
    w = 10;
    h = 14;
    setM(m, 8, x, y, pos, l, w, h);
end
function mans = B_2(m, x, y, pos)
```

```
    l = 24;
    w = 20;
    h = 20;
    setM(m, 9, x, y, pos, l, w, h);
end


function mans = setM(m, key, x, y, pos, dl, dw, dh)
    mans = m;
    if pos ~= 0
        pos = 1;
    end
    if pos == 0
        mans(x:x+dl-1,y:y+dw-1) = 1;
    else
        mans(x:x+dw-1,y:y+dl-1) = 1;
    end

    global bitm
    bitm(1, key) = bitm(1, key) + 1;
end
```

### A.4 evolve.m

```
function Qvalue = evolve(dna, mode)
%     disp('in evolve()');
%% init
    m = zeros(231, 92);
    bitm = zeros(1, 12);

    global sR
    sR = zeros(12, 3);
    sR(1,:) = [45,45,24];  %A
    sR(2,:) = [30,30,22];
    sR(3,:) = [60,50,30];
    sR(4,:) = [25,20,25];
    sR(5,:) = [25,20,27];
    sR(6,:) = [40,40,25];
    sR(7,:) = [32,32,17];  %G
    sR(8,:) = [8,10,14];  %type1
    sR(9,:) = [24,20,20];  %type2
    sR(10,:) = [14,7,5];  %med1
    sR(11,:) = [5,8,5];
    sR(12,:) = [12,7,4];

    xylist = [1;1];

%% body
    dnast = 0;
    for i = dna

        dnast = dnast + 1;
%         disp(['============== geneNum: ', num2str(dnast),' ================']);

%         pos = decidePos(x, y, i);
        boxtype = i(1);
        pos = i(2);
```

```
        [xylist, x,y] = decidexy(m, xylist, boxtype, pos);
        if x == -1
            continue;
        end
%          disp('out decidexy()');

        sizeBox = sR(boxtype,:);
        bl = sizeBox(1);
        bw = sizeBox(2);
        bh = sizeBox(3);

        [m, bitm, xylist] = setM(m,bitm,xylist, boxtype, x, y, pos, bl, bw, bh);
%          disp('out setM()');
    end

    V = calV(bitm);
    Qvalue = getQ(bitm, V);

    %% Visualization
    if mode ~= 0
        mode = 1;
    end
    if mode == 1
        disp(['V: ',num2str(V)]);
        disp(['Qvalue: ',num2str(Qvalue)]);
        disp('bitm');bitm
        % visualize m
        figure(1);
        imshow(m);
    end
end


%% Utility functions
function [m, bitm, xylist] = setM(m,bitm,xylist, key, x, y, pos, dl, dw, dh)
%      disp('in setM');
    if pos ~= 0
        pos = 1;
    end
    if pos == 0
        m(x:x+dl-1,y:y+dw-1) = key;
        newco = [x,x+dl;y+dw,y];
    else
        m(x:x+dw-1,y:y+dl-1) = key;
        newco = [x,x+dw;y+dl,y];
    end

    bitm(1, key) = bitm(1, key) + floor(94/dh);

    xylist = [xylist,newco];
    xylist = xylist';
    xylist = sortrows(xylist, 1);
    xylist = xylist';
end

function [xylist, x, y] = decidexy(m, xylist, boxtype, pos)
    x = -1;y = -1;
    global sR
    sizeBox = sR(boxtype,:);
    if pos == 0
        bl = sizeBox(1);
```

```
        bw = sizeBox(2);
    else
        bw = sizeBox(1);
        bl = sizeBox(2);
    end

%     disp('In decidexy()');

    for i = 1:size(xylist,2)
        xi = xylist(1, i);
        yi = xylist(2, i);
        bans = checkBox(m, xi, yi, bl, bw);
        if bans == 1
            x = xi;
            y = yi;
            xylist(:,i) = [];
            break;
        end
    end
end

function bans = checkBox(m, x, y, l, w)
    if x+l > 231 || y+w > 92
        bans = 0;
        return;
    end
    boxm = m(x:x+l,y:y+w);
    sans = sum(sum(boxm));
    if sans == 0
        bans = 1;
    else
        bans = 0;
    end
end

function Vans = calV(bitm)
    dntr = 231 * 92;
    global sR
    vsep = zeros(12, 1);
    for i=1:12
        vsep(i) = sR(i,1) * sR(i,2);
    end

    vsum = bitm * vsep;
    Vans = (dntr - vsum) / dntr;
end

function Qans = getQ(bitm, V)
    Qans = 0;
    if bitm(3) == 0 || bitm(9) == 0
        return;
    end
    Qans = QV(V) + QM(bitm) + QD(bitm);
end

%% Seperate Q's
function qans = QV(V)
    qm = V;
    Wv = 1;
    qans = 1/qm * Wv;
end
```

```
function qans = QM(bitm)
    med1 = bitm(1, 10);
    med2 = bitm(1, 11);
    med3 = bitm(1, 12);
    ndMed1 = 7;
    ndMed2 = 2;
    ndMed3 = 4;
    qm = min([floor(med1/ndMed1), floor(med2/ndMed2), floor(med3/ndMed3)]);
    Wm = 1;
    qans = qm * Wm;
end

function qans = QD(bitm)
    nA = bitm(1);
    nB = bitm(2);
    nC = bitm(3);
    nD = bitm(4);
    nE = bitm(5);
    nF = bitm(6);
    nG = bitm(7);
    kt = 0.5;
    qans = 1 - exp(-kt*nB*0.155*21);
    qans = qans + 1 - exp(-kt*(nB+nC)*0.156*21);
    qans = qans + 1 - exp(-kt*(nB+nC+nF)*0.211*21);
    qans = qans + 1 - exp(-kt*(nB+nC+nF+nA)*0.857*4);
    qans = qans + 1 - exp(-kt*(nB+nC+nF+nA+nD)*1.667*4);
    qans = qans + 1 - exp(-kt*(nB+nC+nF+nA+nD+nG)*1.882*4);
    qans = qans + 1 - exp(-kt*(nB+nC+nF+nA+nD+nG+nE)*2*4);
end
```

## A.5 main.m

```
%% Generate initial DNA's
global DNAl
DNAl = 200;

global nGen
nGen = 12;

global DNAn
DNAn = 500;

DNAs = initDNAs;
disp('end of initDNAs');

Qdnas = zeros(DNAn,1);

%% Body
nEpoch = 100;
Vmx = 0;
for epoch = 1:nEpoch
    disp(['=========== epoch:', num2str(epoch),' ===========']);
    %% Evolve
    for i=1:DNAn
%         if mod(i, 100) == 0
%             disp(['DNA num: ',num2str(i)]);
%         end
```

```
        p = (i-1) * 2 + 1;
        dna = DNAs(p:p+1,:);
        Qdnas(i) = evolve(dna, 0);
    end

%     disp(['end of evolution of epoch:', num2str(epoch)]);

    %% Update
    if epoch ~= nEpoch
        [Qdnas,Ps] = sort(Qdnas, 'descend');
        for i = 250:DNAn
            p = Ps(i);
            p = (p-1) * 2 + 1;
            DNAs(p:p+1,:) = genDNA(DNAs, Ps);
        end
    end

    %% Exit on convergence
    if Vmx < Qdnas(1)
        vmx = Qdnas(1);
    else
        break;
    end

end

%% Result visualization
i = 1;
p = Ps(i);
p = (p-1) * 2 + 1;
dna = DNAs(p:p+1,:);
evolve(dna, 1);

% %% FOR TEST
% m = zeros(231, 92);
% for i = 50:100
%     for j = 10:20
%         m(i,j) = 1;
%     end
%     for j = 21:40
%         m(i,j) = 2;
%     end
% end
% figure(1);
% imshow(m);

%% Utility functions
function DNAs = initDNAs
    global DNAn
    global DNAl
    DNAs = ones(DNAn * 2, DNAl);

    % obtain object category
    for i=1:DNAn
        p = 2 * (i-1) + 1;
        for j=1:DNAl
            obj = floor(rand() * 12) + 1;
            DNAs(p,j) = obj;
        end
    end
```

```
    % obtain pos's
    for i=1:DNAn
        p = i*2;
        for j=1:DNAl
            posRRate = 0.1;
            pos = 0;
            if rand() < posRRate
                pos = 1;
            end
            DNAs(p,j) = pos;
        end
    end

    % Validation check for DNAs
    for i=1:DNAn
        p = 2 * (i-1) + 1;
        dna = DNAs(p:p+1, :);
        checkDNA(dna);
    end
end

function dna = genDNA(dnas, Ps)
    global DNAl
    sppt = floor(rand() * (DNAl-1)) + 1; % split point
    if sppt <= 1
        sppt = sppt + 1;
    elseif sppt > DNAl-2
        sppt = sppt - 1;
    end

    dp1 = floor(rand() * 250)+1;
    dp2 = floor(rand() * 250)+1;
    dp1 = Ps(dp1);
    dp1 = 2*(dp1-1)+1;
    dp2 = Ps(dp2);
    dp2 = 2*(dp2-1)+1;
    d1 = dnas(dp1:dp1+1, :);
    d2 = dnas(dp2:dp2+1, :);
    dna = [d1(:,1:sppt), d2(:,sppt+1:DNAl)];
    dna = mutate(dna);

    checkDNA(dna);
end

function dna = mutate(dnaIn)
    global DNAl
    mutationRate = 0.1;
    dna = dnaIn;
    if rand() < mutationRate
        sppt = floor(rand() * (DNAl-1)) + 1;
        obj = floor(rand() * 12) + 1;

        dna(1, sppt) = obj;
    end
end


function bans = checkDNA(dna)
    bans = 1;
    for i = dna
        if i(1) == 0
```

```
            bans = 0;
            break;
        end
    end
    if bans == 0
        disp('ERROR: DNA invalid !!!');
    end
end
```