For office use only
T1 ________________
T2 ________________
T3 ________________
T4 ________________

Team Control Number

36178

Problem Chosen

B

For office use only
F1 ________________
F2 ________________
F3 ________________
F4 ________________

2015 Mathematical Contest in Modeling (MCM) Summary Sheet
(Attach a copy of this page to your solution paper.)

Type a summary of your results on this page. Do not include the name of your school, advisor, or team members on this page.

In this paper, we develop a model for finding a plane that was lost over a large body of water. Our model is very robust and can be used by searchers throughout the entire search process.

Using only basic information about the flight and its last known location and heading, we are able to carefully calculate the area where the plane might have gone down. This area is not just a simple circle - the area is limited by the turn radius of the plane and by the maximum distance the plane could have traveled after losing communications.

Once our initial search area is determined, our model calculates how ocean currents would cause the search area would move from day to day, as well as change shape. And when debris has been found, the same ocean current calculations can be used in reverse to trace debris back to the original location of the crash.

Our model also implements three different algorithms for searching the area determined by our previous calculations. Our algorithms are very flexible, and adapt to changing weather conditions, number of search planes available, and the size of the area to search. Based on detailed analysis of these three algorithms in over 900 unique search scenarios, our model was able to recommend the ideal algorithm for finding any lost plane. Given any combination of factors for a search, our model can predict the number of days it will take to find debris in that region.

So You Lost a Plane?

February 10, 2015

Contents

1 Introduction to Problem

The problem we chose to solve was how to most effectively search for a lost airplane that is expected to have gone down over a large body of water. We developed a method to find lost planes in the least amount of time, which entails obtaining an accurate estimation of where the plane may have gone down, developing a fast and efficient search algorithm that adapts to new information, and calculating the location of the lost plane based on ocean currents.

2 Assumptions and Parameters

2.1 Assumptions and Justifications

Our model makes the following assumptions:

1. Generally speaking, the plane crash will occur in the way that will make the search the most difficult. This is justified because assuming the most difficult scenario in modeling makes the model compatible with the most plane crash scenarios.

2. We assume that air drag will not affect the distance a plane travels in a crash. The first reason we do not consider air drag is because it greatly complicates the model. Second, data necessary for calculating air drag such as the cross-sectional width of the planes and the coefficient of air drag of the planes was not readily available. Finally, in determining the distance that a plane will travel while crashing, air drag will both limit the distance and increase the distance. This is because air drag opposes the forward motion of the plane but also increases the time that the plane remains in forward motion by limiting the speed with which the plane falls.

3. Due to the small size of airplane debris, search planes will not be able to use infrared or radar and will have to rely on visual search alone, even in very calm seas. Our source indicated that even in rather calm seas, forward-looking airborne radar and infrared technology would be completely unable to detect small debris.[2]

4. The field of debris will not scatter so far from the site of the crash that it will change our overall search area. Justification for this assumption can be found in Section 4.2.

5. The area in which the plane crashes is completely water and is flat enough to be modeled as a geometric plane. The curvature of the ocean is extremely small and would be insignificant for all calculations.

6. Fatigue of searchers will not impact their effectiveness. Fatigue is too difficult to model because it has different effects on different individuals, and we can assume further that individuals impaired by fatigue would not be allowed to participate in the search.

7. All search planes are assumed to take off 200 nautical miles from the search location. We consider this a reasonable distance, since our model does not know where the nearest land mass or aircraft carrier may be. (A discussion of this assumption can be found in Section 7.)

8. There will be sufficient daylight for searching for only 10 hours each day. In most locations, this will let searchers start at least one hour after sunrise and end at least one hour before sunset.

9. Floating debris will not sink until after the search has been completed. Because it's impossible to accurately model when the debris will sink, we must assume that it has not sunk yet.

2.2 Parameters

Our model uses several parameters to help determine where a plane was likely to crash. We use the altitude, velocity, and mass of the downed plane as parameters to help determine how far the plane likely traveled while crashing. The wind speed at the time of the crash is also considered when determining this distance.

Another main focus of our model was developing the most effective search pattern. We took many parameters into consideration while determining this search method including the altitude, velocity, and range of search planes, the number of search planes that are available, the time since the crash, and the time allotted for the search each day (assumed to be 10 hours), and what the weather conditions of current search day.

Our model also describes how to effectively use found debris to locate the site of the crash. The parameters we used to determine this method are the speed and direction of the ocean currents and the time since the crash, as mentioned earlier.

3 Analysis of the Problem

3.1 Hypotheses

Helping plan a useful search for a generic downed plane is a problem which can be split into two parts: predicting where it went down, and searching that area. After some discussion, we concluded that we should be able to figure out the area where the plane went down by modeling how far a plane can glide without power, while accounting for the possibility of turning while gliding. We also realized that the currents of the ocean would be moving the debris away from the plane crash as time went on, so the area to search would move each day. We then hypothesized that there are several ways to search the area, such as the three illustrated in Figure 1.

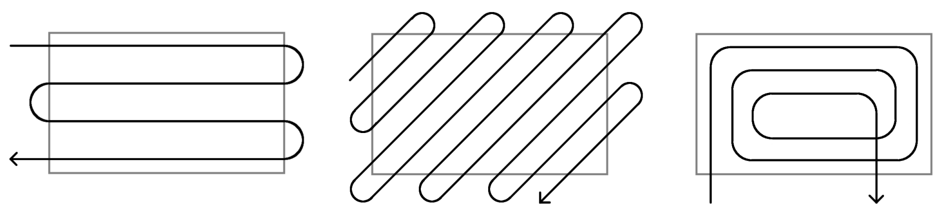


Figure 1: Some of our initial ideas for how to search a rectangular area.

Some search methods were discounted immediately due to obvious inefficiency and inability to be adapted to multiple searchers. We hypothesized that going in straight lines as much as possible when searching an area would maximize the efficiency of the search, an assumption that let us rule out searches like the spiral search in Figure 1. That same assumption also brought to the forefront searches that use parallel sweeps, such as the other two illustrated in Figure 1.

4 Statement of Model

Our model provides suggestions for searchers throughout the entire search process. Based on the parameters stated above, our model calculates the area in which the plane could have crashed, calculates the motion of that area due to ocean currents over time, provides estimates for the time required to find crash debris, and uses the speed and direction of local ocean currents to trace debris back to its original location, near the site of the plane crash.

4.1 Components of the Model

Based on our analysis of the problem, we split the model into the following components:

1. Modeling a generic downed plane using parameters
2. Calculating the size and shape of the area where the plane likely went down

3. Calculating the probable location of the debris field over time using ocean current data
4. Modeling a generic search plane using parameters
5. Searching the probable debris field effectively using search planes
6. Using reverse-drift calculation to approximate the origin of the debris

4.1.1 Modeling a generic "downed plane" using parameters

We model a downed plane using the following parameters:

1. Its last known longitude and latitude as established by an automatic handshake or an Air Traffic Control scheduled check-in
2. Its direction of travel, in terms of degrees. (0 degrees is East, 90 North, 180 West, and 270 South)
3. The family that the plane belongs to:
   (a) Small bush plane
   (b) Military fighter
   (c) Private jet
   (d) Commercial airliner
   (e) Cargo plane
4. The parameters determined by the family of the plane:
   (a) Cruising speed
   (b) Cruising altitude
   (c) Typical takeoff mass

The five main families of planes each have different speeds, altitudes, and masses, so we decided to pick a representative from each family and model it. That representitive would have to be commonly flown and thus be more likely to crash, making it a good choice to model. The representatives chosen are as follows:

1. Small bush plane: Cessna 172 Skyhawk[11]
2. Military fighter: Lockheed Martin F-22A Raptor[12]
3. Private jet: Bombardier Challenger 300[13]
4. Commercial airliner: Boeing 737-300 Classic[14][15][16]
5. Cargo plane: Lockheed C-5B Galaxy[17]

The height and speed of the plane were used to calculate its maximum glide distance without power, which is then used to determine the shape of the probable "splash-down" location. The mass is used to calculate its inertia, which makes it more resistant to the effects of crosswinds as it glides.

4.1.2 Calculating the present location of the debris field using ocean current data

The likely outcome of a plane crash in open water is that, upon impact, the plane is destroyed and debris from the plane will be propelled a relatively short distance. Some of this debris will sink, but some of it will float and be carried by the currents in that area. After determining an approximate location for the site of the plane crash, we can calculate where the ocean current will carry the debris. This process is outlined in section 4.3.

The parameters that we used to model the ocean currents were the speed and direction of the currents. The debris resulting from a plane crash will continue to be carried by the currents until it is found by searchers. When a piece of debris is found in a search, the time and location of that debris provides important information that can be analyzed mathematically to obtain a better estimation of the location of the crash.

4.1.3 Modeling a generic search plane using parameters

The primary parameters that we use for modeling a generic search plane are:

1. Flight speed while searching
2. Altitude while searching
3. Range of the search plane determined by fuel capacity

The speed is used to calculate the search plane's turn radius, as well as to determine how much area the plane can search before the sun goes down. Range is also used as a limiting factor when calculating the area the plane should search.

Speed and altitude are also used to calculate the aircraft's effective sweep width. (A description of effective sweep width is presented in section 4.4.) Due to the many varying factors that may affect an airplane's sweep width, there is no one formula to calculate this. The United States National Search and Rescue Committee maintains tables of estimated effective sweep widths based on altitude, weather conditions, and size of the lost object.[2] We used *Mathematica* to interpolate this data and create a global sweep width function that accepts any inputs, even those not on the table.

4.1.4 Searching the probable debris field effectively using search planes

Once the model has generated a search area, it can calculate the best way to search it for debris. There are many parameters used in this calculation: the number of search planes available, the types of planes available and their search effectiveness, prevailing weather conditions, visibility (how far one can see in terms of nautical miles), and the size of the area to search. How the search is actually performed is described below in section 4.5

4.1.5 Using reverse-drift calculation to approximate the origin of the debris

The exact time and location should be noted when a piece of debris is found. With accurate data that gives the speed and direction of the ocean currents, the calculation used to predict the future location of the debris can be used in reverse to trace the debris back to its origin. This gives us a more accurate estimation of the location of the crash. This process is described mathematically in section 4.3.

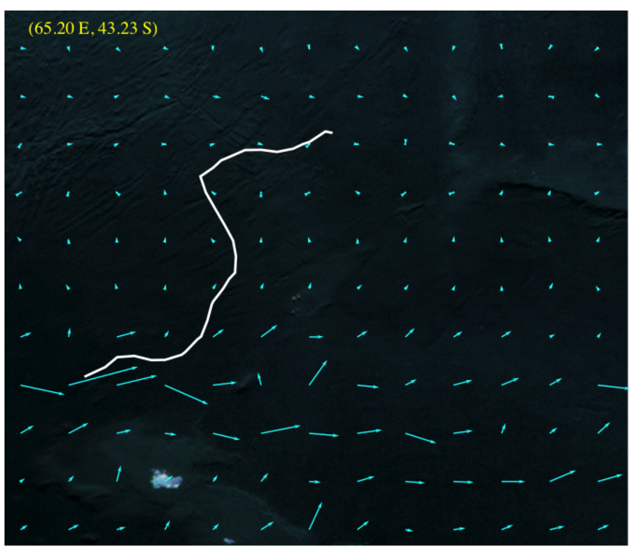


Figure 2: A vector field representing the ocean currents moving an object[21]

4.2 Generating the Search Area

To estimate the area where the plane could have gone down, our model takes into account various attributes of the plane. We calculate the maximum unpowered glide distance (d) using this formula:[19]

$$d = \frac{altitude}{\tan(\theta)}$$

where θ is the angle between the incoming plane's trajectory and the ocean surface at which the plane descends. We treat d as the maximum distance the plane could have traveled since we last had contact with it.

A simpler model would use this maximum travel distance as the radius of a circle. But by using the turn radius of the airplane, we can refine our estimate and reduce the search area. We consider that the plane could have changed direction after we lost contact with it, but that the radius of its turn will affect how far it can travel after changing direction. To calculate the turn radius, we start by assuming completely horizontal motion initially, then integrating horizontal velocity (in nautical miles per second) as the aircraft turns 90 degrees. At a rate of 1.5 degrees per second, this takes 60 seconds.

$$\begin{aligned} V_x(t) &= speed * \cos(1.5t) \\ r &= \int_0^{60} V_x(t)dt \end{aligned} \tag{1}$$

Consider an aircraft travelling at 500 knots, with a maximum glide distance of 40 nautical miles. Its turn radius is then 5.31 nautical miles. If the airplane does not turn, it can travel 40 nautical miles from the last point of contact. But if it makes a 90°turn, it can only travel another 32 nautical miles because of the distance spent turning. If it turns 180°, it can only travel another 23 nautical miles. Figure 3 demonstrates this fact visually.

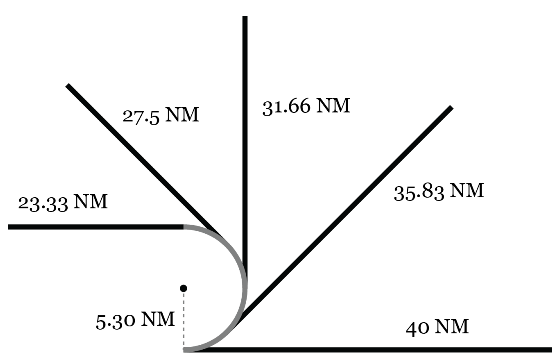


Figure 3: The more our example airplane turns, the less it can fly after the turn ends.

By generalizing this notion, we arrive at the following equations for the final horizontal and vertical positions of the plane, given the angle turned:

$$\begin{aligned} x(\theta) &= r * \sin(\theta) + \cos(\theta) * \big(d - arclength(\theta, r)\big) \\ y(\theta) &= r * \big(-\cos(\theta) + 1)\big) + \sin(\theta) * \big(d - arclength(\theta, r)\big) \end{aligned} \tag{2}$$

Using these formulas for our example plane gives us the result in Figure 4. When we mirror this result vertically to allow for the plane turning to the right, we get the result in Figure 5.

After generating these points, the model applies a rotation matrix to orient the shape to the heading of the airplane at the last point of contact. Because we take the speed of the wind at the time of the crash as a parameter, we know how fast it was blowing, but not which direction it was blowing in. Because of that uncertainty, we decided to increase the size of the search area dependent on that wind. To that end, we

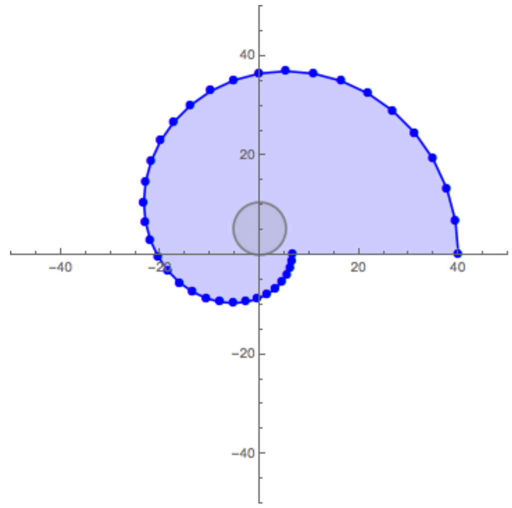


Figure 4: The possible end locations for our example plane, if it turns only to the left. The smaller circle represents the airplane's turn radius.

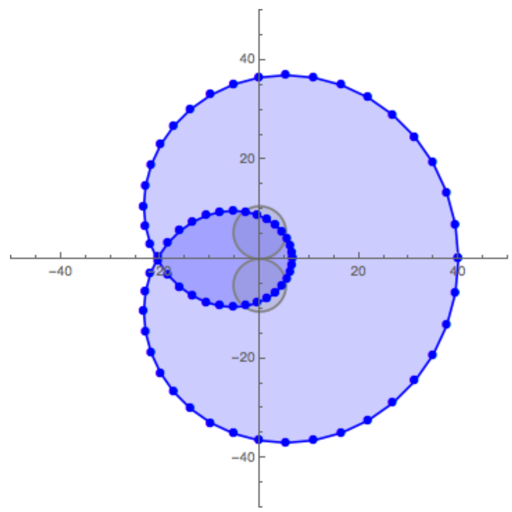


Figure 5: The full range of end locations for our example plane.

had the model iterate through the originally generated edge points, calculate their normal vectors from the center, and add the normal vector scaled by the wind speed to the original point. Thus, the entire perimeter was expanded by the speed of the wind.

The model then applies a convex hull algorithm and saves the result as the initial search area. For those unfamiliar with the concept of a convex hull algorithm, it is a method that produces a polygon that "shrinkwraps" a set of points in the (x,y) plane. It does this by iterating through the points to construct a polygon obeying the following rules:

1. All points in the set are either inside the polygon or are vertices
2. The chord drawn between any two points is either an interior chord or an edge; it can never leave the boundary of the polygon

By using a convex hull algorithm, we were able to simplify the complex search shape into an almost elliptical shape, and as such we arrived at our initial area in which the plane could have gone down.

We had originally considered adding an additional expansion of perimeter based on the debris flying away from the initial point of impact, creating a "splash" effect. However, we found that due to the terminal velocities of typical pieces of debris, such an expansion would only have a marginal impact on the original shape. As such, we chose not to model this aspect of the crash, and we believe that it does not affect our calculations significantly.

4.3 Adjusting the search area over time

Now that we have an n-gon that bounds where the plane could have landed in the water at the time of impact, we needed to account for the drifting of debris due to ocean currents. We were able to obtain ocean current data for the vicinity of the Malaysia Airlines MH370 crash, so we decided to model the planes as if they crashed near there. To calculate how the search area would move and reshape, we took the original search area and placed it over a grid. Each square in the grid was given a "search marker", and that search marker's longitude and latitude were recorded. To simplify the model, we made it such that the search area only moves discretely overnight, as opposed to the real-life continuous movement of currents. Each cycle took the current search marker locations, drifted that marker by 24 hours, then stored that new location back.

We used Euler's method to approximate the motion of the debris over time. To do that, we scaled the velocity vector of the current at the debris' location by the time interval, and then added that vector to the position. This process is demonstrated below:[21]

$$\begin{aligned}(longitude_n, latitude_n) \approx (longitude_0, latitude_0) + \Delta t\left(\frac{\partial(longitude)_0}{\partial(hours)}, \frac{\partial(latitude)_0}{\partial(hours)}\right) + ... + \\ \Delta t\left(\frac{\partial(longitude)_{n-1}}{\partial(hours)}, \frac{\partial(latitude)_{n-1}}{\partial(hours)}\right)\end{aligned} \tag{3}$$

Once all the original search markers were drifted by a whole day's displacement, we called the convex hull algorithm again to create the new containing n-gon, and the subsequent search area.

4.4 Search Theory

In the field known as Search Theory, the underlying goal with each search is to maximize the Probability of Detection (POD). The POD is generally defined using the following parameters and equation:[3]

1. t = time allotted to perform the search, or the time that the search has been going on
2. n = number of searchers
3. r = rate of search, or speed of the searchers
4. s = effective search width
5. a = total area to be searched

$$\mathrm{POD} = 1 - e\hat{}\left(-\frac{\mathrm{t} \cdot \mathrm{r} \cdot \mathrm{n} \cdot \mathrm{s}}{\mathrm{a}}\right) \tag{4}$$

The effective search width is not an easily definable number, but more of a concept. One of the simpler ways to think about it is as such: suppose there is a rectangular search area, a, that has n objects to find in it. Now suppose that you had just one pass through a and you wanted to find at least $\frac{n}{2}$ objects. Then the effective search width is the the minimum distance around you that would allow you to find those $\frac{n}{2}$ objects by walking in a straight line through a.

4.5 Conducting an Effective Search

Now that we have a search area that changes location and shape with time, the second part of our model compares three search methods and decides which of them gives the best Probability of Detection. These were the methods that we tested:

1. A search that has the searchers moving parallel to the longer edge of the bounding rectangle in parallel segments
2. A search that has the searchers moving diagonally at an angle of 45°to the longer edge of the bounding rectangle in parallel segments
3. An optimized version of the first search method.

All three search algorithms take the following parameters: a list of search planes, the size of the bounding rectangle, visibility, and wind conditions. All three algorithms return the total area swept during that day's search. This will be represented by A in the sections below.

4.5.1 Parallel search

The parallel search method is the simplest search algorithm we tested. In this algorithm, search planes simply fly in straight paths parallel to the long edge of the search rectangle. This minimizes time spent turning around for another sweep and maximizes time spent searching.

The *United States National Search and Rescue Supplement to the International Aeronautical and Maritime Search and Rescue Manual* defines some helpful terminology for describing this type of search. The long tracks that the search planes fly along the search area are called searchleg straightaways. The paths that the planes fly while going from one searchleg straightaway to another are called crosslegs. (See Figure 6 for a visual representation of this terminology.)

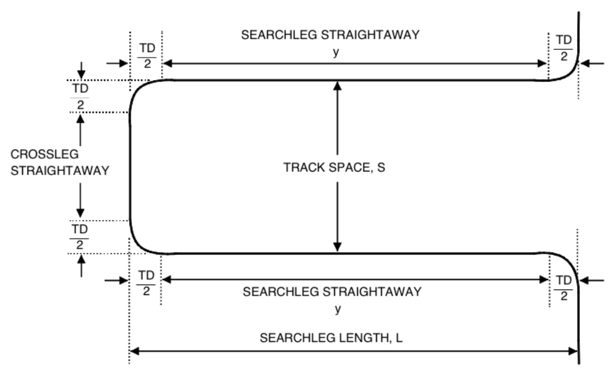


Figure 6: A visual representation of one leg of a search plane's search path. (Source: *United States National Search and Rescue Supplement to the International Aeronautical and Maritime Search and Rescue Manual*)

In our algorithm, each leg of the plane's search after the first one consists of two searchleg straightaways and two cross segments.[2] The first leg of the search only contains one cross segment. Because the searchleg straightaway is the part of the sweep actually spent searching, we attempt to maximize the time spent on these segments. The crosslegs are made up of two quarter-turns and a crossleg straightaway to bring the plane to the next sweep location.

Based on the range of each plane, the speed at which they can search, and the total area left to search, the algorithm computes how many full search legs each plane can complete that day. The width of that search area is then assigned to that plane, and the algorithm moves on to the next plane. This process repeats until either there are no planes left with which to search or the entire area has been searched.

The area swept by this algorithm can be calculated by Equation 5, where n is the number of planes used in the search and L is the length of one full sweep across the search area:

$$A = \sum_{i=1}^{n} \Big(\text{sweepwidth}(plane_i) \cdot \text{numlegs}(plane_i) \cdot 2\Big) \cdot L \tag{5}$$

4.5.2 Diagonal search

The diagonal search method is based on the same principles as the parallel search method, but in this method the search planes fly at a 45°angle to the long side of the rectangle. This changes the search in a few key ways.

Firstly, this search method allows search planes to make much longer sweeps across the search area. It thus maximizes time spent searching near the middle of the search area. However, near the corners of the search area, the searchlegs can be extremely short, and many more turns must be made. In addition, the angle of the paths causes the crossleg straightaway distance to increase for all crosslegs.

The area swept by this algorithm is more difficult to calculate. Equation 6 provides a formula for the calculation, where w and h are the search area's longer and shorter dimensions, respectively.

$$\begin{aligned} SW &= \sum_{i=1}^{n} \Big(\text{sweepwidth}(plane_i) \cdot \text{numlegs}(plane_i) \cdot 2\Big) \\ c_1 &= h \cdot \sin\left(\frac{\pi}{4}\right) \\ c_2 &= w \cdot \sin\left(\frac{\pi}{4}\right) \\ A &= \begin{cases} SW^2 & \text{if } SW < c_1 \\ c_1^2 + 2c_1 \cdot (SW - c_1) & \text{if } c_1 \le SW < c_2 \\ wh - (c_2 + c_1 - SW)^2 & \text{if } c_2 \le SW \le c_2 + c_1 \end{cases} \end{aligned} \tag{6}$$

4.5.3 Optimized parallel search

A major drawback of both the first two methods is that they cannot use any more search planes once 100% of the search area has been swept. We can improve on this behavior by assigning extra search planes to sweep the area again, increasing the search coverage and probability of detection.

This algorithm begins by running the original parallel search algorithm, and saving each of the plane's search paths as a "search block" for that type of plane. Each search block stores the type of plane it contains, the number of search legs it contains, and the number of search planes assigned to the block. The algorithm assumes that a search block can contain at most as many planes as it has search legs; for example, a search block with five search legs can hold at most five search planes.

The planes in a search block cannot all begin their search at the same time; instead, each must "wait in line" for the plane in front of them to complete a leg. Because all aircraft must finish searching in the allotted time, aircraft that begin their search later cannot search as great an area. This algorithm has each search plane complete one less search leg than the plane before it; for example, in a block with three search legs and three planes, the first plane would complete all three possible legs, the second plane would complete two, and the third plane would complete only one. All planes in the block finish at the same time, before their allotted search time expires.

For every consecutive plane, the algorithm finds the search block of that plane's type that has the most empty space for another plane. It repeats this process until all the blocks are filled or there are no more planes left with which to search.

To compute the area effectively swept in each block, the algorithm calculates the number of total search legs swept by planes in the block. If L_i is the number of legs in block i and P_i is the number of planes in block i, then the total legs swept in block i is:

$$\begin{aligned} TL_i &= L_i + (L_i - 1) + (L_i - 2) + \ldots + \big(L_i - (P_i - 1)\big) \\ &= \sum_{k=1}^{P_i} L_i - \big(1 + 2 + 3 + \ldots + (P_i - 1)\big) \\ &= L_i \cdot P_i - \frac{P_i \cdot (P_i - 1)}{2} \end{aligned} \tag{7}$$

The total area swept can then be calculated using Equation 8 below, where n is the number of search blocks and L is the length of one full sweep across the search area.

$$A_{swept} = \sum_{i=1}^{n} \Big(\text{sweepwidth}(block_i) \cdot TL_i \cdot 2\Big) \cdot L \tag{8}$$

4.6 Full Statement of the Model's Main Equation

Using the equation above and the equations below, our model can approximate how many days it will take to complete the search.

$$\begin{aligned} A_{search} &= (\text{max width of n} - \text{gon}) \cdot (\text{max height of n} - \text{gon}) \\ \text{POD} &= 1 - e^{\wedge}\left(-\frac{A_{swept}}{A_{search}}\right) \\ \sum_{i=1}^{days} &\text{POD} > 0.95 \end{aligned} \tag{9}$$

4.7 Tracing the Findings to the Crash

Given the location of an object that has been moved by the current for t hours, we use Euler's method once again to trace the object back to its original location. This is done by taking the difference of the position of the debris and the velocity vector of the current scaled by a time interval, Δt. This process is demonstrated below and is repeated $\frac{t}{\Delta t}$ times to approximate the original location of the debris.

$$\begin{aligned} (longitude_0, latitude_0) \approx (longitude_f, latitude_f) - \Delta t\left(\frac{\partial(longitude)_f}{\partial(hours)}, \frac{\partial(latitude)_f}{\partial(hours)}\right) - \ldots - \\ \Delta t\left(\frac{\partial(longitude)_1}{\partial(hours)}, \frac{\partial(latitude)_1}{\partial(hours)}\right) \end{aligned} \tag{10}$$

5 Results

5.1 Comparisons to real-world searches

In order to determine whether our search area and current drift calculations are accurate, it is helpful to draw comparisons to searches for lost planes in the real world. The most famous airline crash of recent memory was the March 2014 crash of Malaysia Airlines flight MH370. However, the crash of MH370 involved several other unforeseen variables and uncertainties that make finding an accurate search area difficult, if not impossible. Instead, we will compare our model's result to the 2009 crash of Air France flight 447, which is a good example of a standard commercial airliner crash that was found using standard search methods. The search area for AF447 was determined to be a circle of radius 40 nautical miles.[1] However, when we plugged the data from that type of plane into our model (an Airbus A330[18]) we calculated a search radius of about 125 nautical miles. The reason ours is so much larger is because our model assumes we know very little about the crash. In the case of AF447, flight control expected to receive two messages only 60 seconds apart, but it only received one. That gave them a very precise estimation of the time and manner in which the plane went down.[1]. Our model does not expect this level of precision, and thus generates an outer bound of where the plane could have gone down.

5.2 Optimal search methods

Our model ran detailed analysis on our three different search methods with varying numbers of search planes, weather conditions, visibility distances, and sizes of search areas. For each combination of variables, we calculated the number of days each algorithm would require in order to raise the probability of detection above 95%. Each search method showed strengths and weaknesses depending on the unique combination of these variables. Figure 7 gives a visual representation of the results from the latter two search methods.

5.2.1 Behaviors of different search algorithms

The simple parallel search worked reliably in all of our different test cases. While it was usually not as efficient as the diagonal search, it did not produce any unexpected or impractical results. Its worst-case time was 3581 days, with only 1 plane searching in 30+ knot wind and 1 nautical mile of visibility over 112,640 square nautical miles. However, the best-case time for this algorithm was three days, even with dozens of extra planes, because it did not have the means to cover more than 100% of a search area in a given day.

The diagonal search generally performed better than the simple parallel search, usually achieving 95% POD in a median of two fewer days. However, if run with only a few planes under adverse conditions, the algorithm produced extremely inefficient searches. With only one plane and limited visibility, this algorithm sometimes took upward of 35,000 days to reach 95% POD—nearly 100 years of searching! The diagonal search algorithm also has the same best-case time of three days, since it likewise cannot put extra planes to use.

The diagonal search was our best-performing algorithm overall. Its performance was better than the simple parallel search in most cases. Even though under adverse conditions this search does extremely poor, most of the time, given typical search conditions, it will be faster than the optimized parallel. Under intense time pressure, when the lives of the plane's passengers may be at stake, this factor is highly desirable.

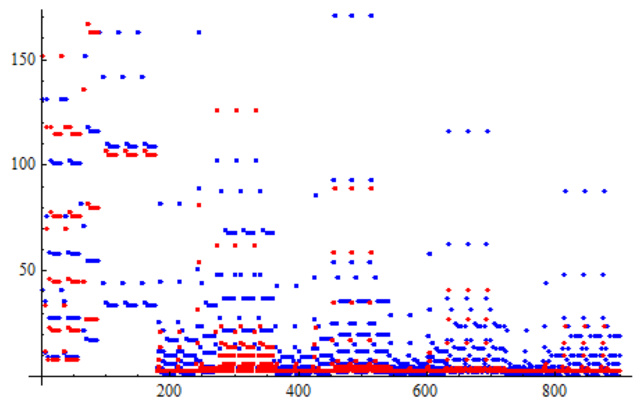


Figure 7: A visual representation of the results from our analysis of different search methods. Red dots represent results from the diagonal search algorithm; blue dots represent the optimized parallel search algorithm. The horizontal axis is the unique index of the search in our data. The vertical axis is the number of days in which the search was completed. Values toward the right of the plot represent searches with more planes and higher wind speeds.

5.2.2 Choosing the right algorithm for each search

It is clear that none of our search methods are perfect for every search, because each method has a best and worst case depending on the parameters of the search. When planning a real-life search for a missing aircraft, every single day spent searching is another day that lives may be lost, so choosing the optimal search method for each day is critical. We will compare and contrast our latter two search methods: the diagonal search and the optimized parallel search.

The diagonal search algorithm usually exhibits better performance than the optimized parallel search. We suggest that this is because search legs near the middle of the search area can be much longer than those in the parallel method. However, the diagonal search exhibits worse performance than the optimized parallel search under a few specific conditions—sometimes worse than the optimized parallel search by several orders of magnitude. The worst performance of the diagonal search algorithm occurs when searching a large area with very few search planes and bad weather conditions. As visibility decreases and wind speed increases, this algorithm exhibits poorer and poorer performance until it reports the incredible 35,000 days mentioned above.

The diagonal search also exhibits worse performance when searching a small area with many search planes in good weather conditions. In these situations, the optimized parallel search is able to sweep sections of the search area more than once each day, which can dramatically increase coverage and probability of detection. Under these conditions, the optimized parallel search algorithm can sometimes complete its search in just one day.

Based on these results, we can recommend the following courses of action:

- If using few search planes, the search area is large, and weather conditions are bad: Use the optimized parallel search.
- If using many search planes, the search area is small, and weather conditions are good: Use the optimized parallel search.
- Otherwise: Use the diagonal search.

6 Error/Sensitivity Analysis

Because our model does not find an optimal solution, but instead predicts the number of days it will take to search any area, our sensitivity analysis will be based on a median problem, and we will then see how changing the input parameters affects the number of days the search will take. The parameters we analyzed are:

1. Height of the crashed plane
2. Speed of the crashed plane
3. Speed of the wind at the time of the crash
4. The number of search planes in the air searching
5. The speed of the wind while searching
6. The visibility range during the search

For our sensitivity analysis, we had each of those parameters set to the following as the "baseline" we would compare against:

1. 45000 feet cruising altitude
2. 459 knot cruising speed
3. 10 knot wind at time of crash
4. 21 search planes (AP-3C Orion)[9]
5. 20 knot wind while searching
6. 14.6 nautical miles of visibility

These parameters generate a search area of 85,957 square nautical miles to search. This baseline generates an estimated search time of 11 days.

To perform our analysis, we changed each parameter individually and compared the new resulting estimated days to our baseline of 11 days. We kept altering the deviation until we had a change on both sides, or until we were convinced that changing in one direction would be ineffective.

6.1 Sensitivity with respect to the height of the crashed plane

We found that changing the altitude that the crashed plane was flying at by a deviation of 4000 feet created a change in both directions. Decreasing the height by 4000 feet led to 10 days of searching, and increasing by 4000 feet led to 12 days of searching. As such, it appears that a linear change of the height results in a linear change in the number of days to search at a rate of 1/4000.

6.2 Sensitivity with respect to the speed of the crashed plane

When we changed the speed of the crashed plane, it only altered the turn radius which in turn altered the size of the search area. However, we found that any reasonable change in the speed from our baseline (459 knots) only had a marginal effect on the estimated days, and as such this does not appear to affect our sensitivity analysis.

6.3 Sensitivity with respect to the speed of the wind at time of crash

When we altered the speed that the wind was blowing at to be anything less than 10 knots, we found that it did not affect the estimated number of days. However, increasing the speed by 5 knots gave us 1 more day that we had to search. Thus, this appears to affect the answer in a linear fashion above 10 knots at a rate of 1/5.

6.4 Sensitivity with respect to the number of search planes

As we changed the number of planes, we found an interesting result: decreasing the number of search planes by 2 planes increased the search time by 2 days, but adding 2 search planes only decreased the search length by 1 day. As such, it appears that the search time has an inverse exponential relation to the number of search planes.

6.5 Sensitivity with respect to the speed of the wind while searching

Because of the piecewise function our model has to use to calculate sweep width based on the wind speed at the time of the search, we had to make large changes to the wind speed in order to get a change. When we increased wind speed by 6 knots, it doubled the search time, and when we decreased by 6 knots it halved the search time. In real life, this is most likely a logistic relation, as the time will eventually cap regardless of wind speed, and will bottom out when wind speed gets below a certain level.

6.6 Sensitivity with respect to the visibility while searching

When we performed our sensitivity analysis on the visibility, we found that any visibility above our original 14.6 nautical miles had no effect on the time it would take to search the area, but going down by 1 nautical mile increased the time it would take by 1 day. From this behavior, it appears that this is another inverse exponential relation.

7 Analysis of our Model

7.1 Strengths

Our model can calculate the possible "splash-down" area using only the altitude, speed, and last known coordinates of the plane, which is much less data than most searches typically require. Using those three parameters, the model is able to generate a more realistic search shape than just a circle or rectangle. It can also use real-life ocean current data to have the debris search area drift accordingly, allowing it to change shape and size. Other models might just have a static search area, and debris might drift out of that. A third strength of the model is that, regardless of the shape of the area, we are able to model and predict how long it will take to search that area sufficiently to have a 95% chance of finding a piece of debris, if there is any debris to be found.

7.2 Weaknesses

While our model is quite robust, it has a few important weaknesses. One of them is that, in order to run our search patterns, the model has to search a rectangle that bounds the search area, rather than only the area itself. This means that, depending on the orientation and shape of the actual search area, the search planes may spend excessive amounts of time searching outside where the debris could be. Another weakness at the moment is that the model can only calculate current drift in the vicinity of where flight MH370 went down; we lack the data necessary to model ocean currents globally. Our model does not account for survivors floating on debris or in rafts moving the debris against the current and outside the search area as they head for land. Also, our model does not take into account the speed of the crashing plane when it loses power and begins to glide, so the actual search area might be larger than what we are currently estimating.

7.3 Future improvements

Due to the limited time available to us, there were some simplifications that we had to make to our model that we would like to expand upon, given more time. We would like to develop a way to incorporate local current data measured at the site of the last known point of contact and model the drift using that, rather than just a small section of data that we were able to find for flight MH370's area. We would also like to expand upon our "splash-down" shape by accounting for more possible types of loss-of-contact other than just loss of power. A third area we would like to improve in the model is to incorporate other search methods and continue comparing them until we have a definitive "best" method, rather than the best of three that we chose. The fourth way we would like to improve the model is to continue writing and integrating the program so that each day's search area is displayed with latitude and longitude coordinates at the corners for reference, and the search pattern the planes are to fly is traced onto the map as well for convenient visualization.

We would also like to add a feature that allows you to put more specific data into the model and thus narrow the search parameters. For example, future versions of our model may allow the user to provide the distance that search planes must fly to reach the search area as a parameter. It may also allow the user to provide the time since the crash occurred, eventually.

8 Conclusion

The model we have described in this paper is a robust and comprehensive model that can assist searchers throughout the search for a missing airplane. When the airplane goes down, our model can give a maximum bounding area of the plane's location, carefully limited by the airplane's turn radius. Every day, when given a search pattern and information about the search conditions, the model can predict the search area covered and report the probability that debris from the crash will be found. Each day, the model can precisely move the search area based on ocean current data, so searchers can know exactly where to focus their efforts, and once debris is found, the model can trace the debris back to the original location of the crash.

In addition, testing our model on different search methods revealed that there is no one perfect way to search for a missing plane. Depending on the weather, the area to search, and the number of planes available, different search algorithms may yield different results. But by testing these algorithms on a variety of scenarios, our model was able to generate recommended search methods for different search situations.

There is no one way to search for a missing airplane—there are simply too many unknowns, too many variables, too many things that can go wrong in different ways. But no matter the type of plane, the time it went down, or even the weather, we hope that our model could help searchers at every step of the way.

References

[1] Australian Transport Safety Bureau. (2014). *MH370 - Definition of underwater search areas*. Canberra, ACT 2601: Australian Transport Safety Bureau.

[2] National Search and Rescue Committee [NSARC]. (2000). *United States National Search and Rescue Supplement to the International Aeronautical and Maritime Search and Rescue Manual*. Washington, D.C.: U.S. Superintendent of Documents.

[3] Potomac Management Group, Inc. (2006). *A Simple Guide to Conducting Ground Search and Rescue Detection Experiments*. Washington, D.C.: United States Coast Guard Office of Search and Rescue.

[4] Steiner, Margaret F. *Ditching behavior of military airplanes as affected by ditching aids*. UNT Digital Library. http://digital.library.unt.edu/ark:/67531/metadc61115/. Accessed February 6, 2015.

[5] Sedgewick, R., & Wayne, K. (2011, August 6). *Convex hull*. Retrieved February 6, 2015, from http://algs4.cs.princeton.edu/99hull/

[6] Long, L., & Weiss, H. (2006). How Terminal Is Terminal Velocity? *The American Mathematical Monthly,* (113), 752-755.

[7] Carry-on Baggage. (n.d.). Retrieved February 7, 2015, from http://www.delta.com/content/www/en_US/traveling-with-us/baggage/during-your-trip/carry-on.html

[8] Aircraft Specifications. (n.d.). Retrieved February 7, 2015, from http://www.gateway-group.com/aircraft-specifications/

[9] P-3 Orion Specifications. (n.d.). Retrieved February 8, 2015, from http://www.lockheedmartin.com/us/products/p3/p-3-specifications.html

[10] 737 Airborne Early Warning and Control. (n.d.). Retrieved February 8, 2015, from http://www.boeing.com/assets/pdf/defense-space/ic/aewc/docs/737AEWC_overview.pdf

[11] Cessna 172. (n.d.). Retrieved February 6, 2015, from http://en.wikipedia.org/wiki/Cessna_172

[12] Lockheed Martin F-22 Raptor. (n.d.). Retrieved February 6, 2015, from http://en.wikipedia.org/wiki/Lockheed_Martin_F-22_Raptor

[13] Bombardier Challenger 300. (n.d.). Retrieved February 6, 2015, from http://en.wikipedia.org/wiki/Bombardier_Challenger_300

[14] The Boeing Next-Generation 737 family. (n.d.). Retrieved February 6, 2015, from http://www.boeing.com/boeing/commercial/737family/background.page

[15] Boeing 737. (n.d.). Retrieved February 6, 2015, from http://en.wikipedia.org/wiki/Boeing_737

[16] Boeing 737-300. (n.d.). Retrieved February 6, 2015, from http://www.airliners.net/aircraft-data/stats.main?id=92

[17] Lockheed C-5B Galaxy. (n.d.). Retrieved February 6, 2015, from http://en.wikipedia.org/wiki/Lockheed_C-5_Galaxy

[18] Airbus A330. (n.d.). Retrieved February 9, 2015, from http://en.wikipedia.org/wiki/Airbus_A33

[19] Forces of a Glider lab and lecture. (n.d.). Retrieved February 6, 2015, from http://www.t4t.org/wp-content/uploads/2013/08/FORCES-The_Glider.docx

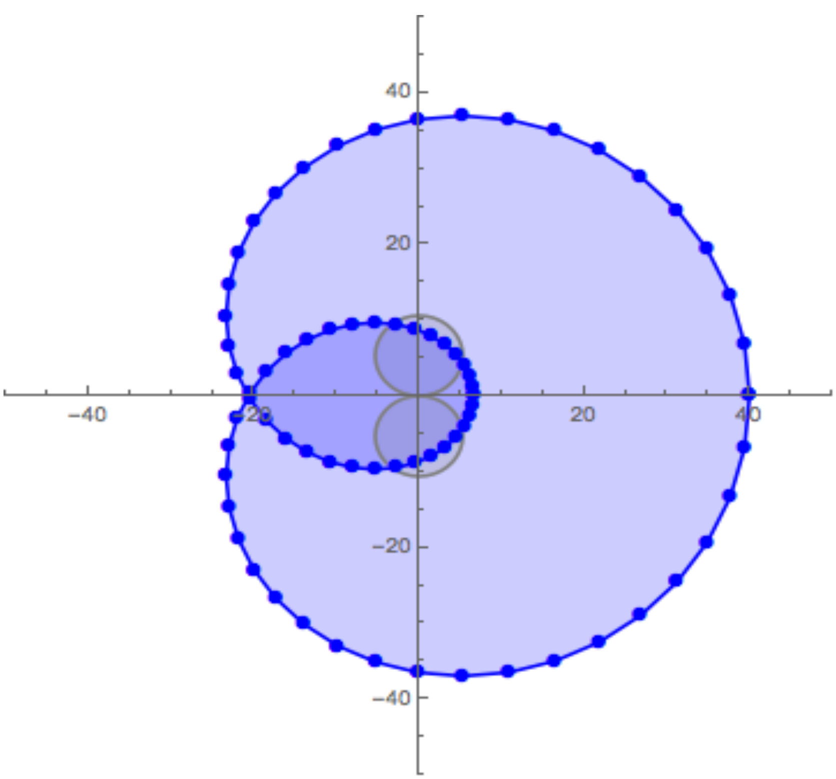
[20] OSCAR Project Office. "Data Download for Ocean Surface Currents." Retrieved February 5, 2015, from www.oscar.noaa.gov/datadisplay/oscar_datadownload.php.

[21] Hafner, Izidor. "South Indian Ocean Currents, Flight MH370." Demonstrations RSS. Wolfram, 21 Apr. 2014. Web. 05 Feb. 2015. ¡http://www.demonstrations.wolfram.com/SouthIndianOceanCurrentsFlightMH370/¿.

Addressing the Press Regarding the Plan for Future Searches

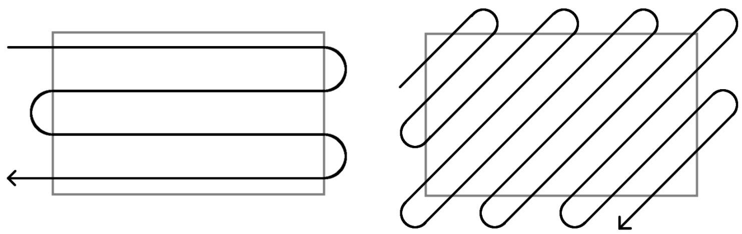
Our new plan for searching for planes that are lost over large bodies of water can be summarized by three stages: finding an approximate location for the site of where the plane went down, searching this location for evidence of the location of the plane, and analyzing the findings to obtain a better approximation for the location of the plane. This process is repeated until the plane is found. These stages will now be explained in further detail.

Upon receiving news that one of our planes has been lost over a large body of water, we will first gather information regarding the situation, such as the model of plane, the point of last contact, the time of last contact, the weather in the area, and the heading of the plane. We will then use this information to determine an area that the plane could have gone down in. This area is usually a circular shape.



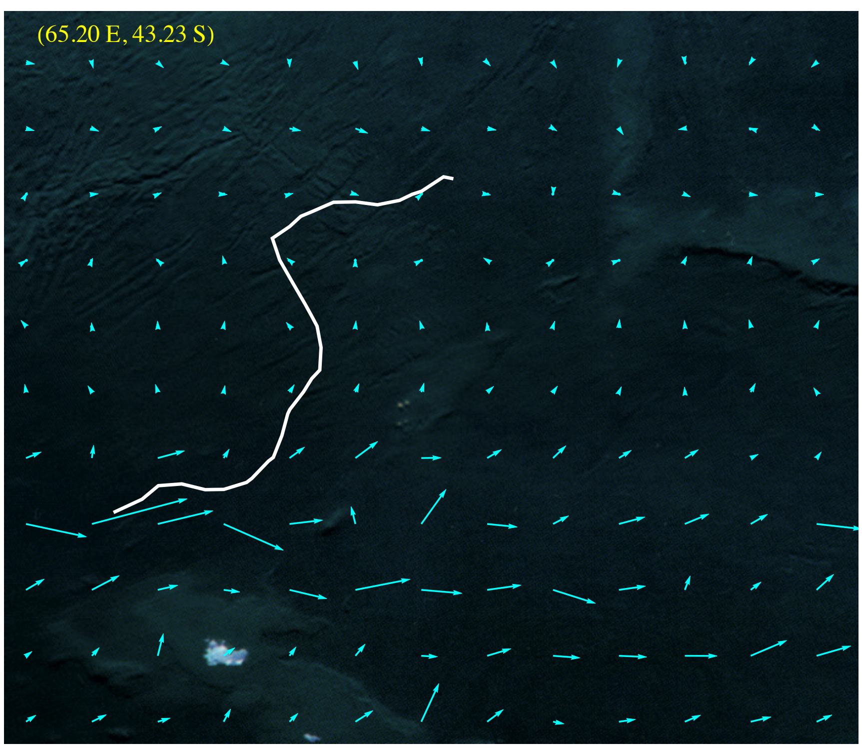
The full range of end locations that the plane could have crashed at

After determining the area that the plane could have gone down in, we will trace a rectangle around this area and search the rectangle with search planes. If few search planes are available, they will search in a straight-line pattern across the long side of the rectangle. If ten or more planes are available for the search, then the planes will fly diagonally across the rectangle to cover the area more efficiently. The planes will use visual techniques to spot evidence of the plane's location, such as debris.



The different search patterns

Every time debris is found, we will use data of ocean currents of that location to determine where that debris originated. This will give us a better approximation of the location of the plane, and we will start again with the first stage and repeat this process until the plane has been found.



A vector field showing how ocean currents move debris[1]