Optimization of Distribution Centers for Amazon Prime Air Service

Nolan Crook, Dustin James Harper, Matthew Wald

***Abstract*—This research develops a model for optimizing important parameters for the unmanned aerial vehicle (UAV) delivery service, Amazon Prime Air; specifically, the location of a distribution center in a smaller community, and number of UAVs needed at the distribution center. A simplified profit model was used to calculate the income and cost associated with the UAV distribution center. Variables for the model included distance to possible delivery points, real estate cost, and maintenance cost of UAVs. Inputs to the model were real estate cost, number of packages to deliver, profit of each package, average speed of UAVs, and initial purchase of UAVs. No-fly zones—controlled airspace—were incorporated in the distance calculation and no-build zones as location constraints. The city of Provo, UT was used to test the functionality of the model. The model is meant to be developed in more detail and eventually used as a tool that will help future planning of distribution centers in various cities.**

I. Introduction

UNMANNED aerial vehicles (UAVs) have been used by the military for several years, but the technology is now sufficiently affordable and accessible to allow businesses and consumers to use it in a wider variety of applications. One example is Amazon's Prime Air service, which is currently in development. This service will guarantee a package delivery time of just 30 minutes for select items; once it is fully implemented it will use UAVs as the delivery vehicle [1].

The first implementations of Amazon Prime Air will most likely start from existing fulfillment centers—the distribution warehouses where Amazon organizes and ships orders. As the service grows it will need to expand to smaller, more localized fulfillment centers in cities too far distant from existing centers. Before making this transition, Amazon must understand the business case associated with this change in infrastructure in order to implement the new technologies and systems in an optimal way.

Several other studies have developed tools to optimize the location of a distribution center [2], logistics inside a distribution center [3], and energy savings within a distribution center [4]. Similar to our research, these studies generally focus on development of an optimization model rather than finding a single optimal solution. Unlike the model developed in our research, all of these previous studies focus on traditional distribution warehouses with automobile delivery.

There have been previous studies about the economic viability of using UAVs for commercial purposes [5], but only for larger UAVs with different mission parameters. Our analysis presents a simplified business model that focuses on factors which are more significant to UAV package delivery. Unfortunately, specific information about these important factors is still largely unavailable due to FAA regulations, which restrict experimentation and testing. Several other variables and parameters associated with a UAV distribution center must also be approximated as outlined below. It is intended that this model will be modified to fit parameters defined by delivery information and future regulations once they become available.

II. Methodology

A. Development of Model

The simplified profit model used as the objective function is made of two main parts: costs and income. Since this is a maximization problem, the function must be negated, as shown in Equation 1. The model is a function of the longitudinal (East-West) location (x), latitudinal (North-South) location (y), and number of UAVs available at the distribution center (N).

$$Profit(x, y, N) = Cost - Income \tag{1}$$

We recognize that the truly optimal solution will include many more factors than just these design

variables. However, these three variables were chosen for their unique importance in a model of UAV distribution centers comparative to truck delivery systems. For example, the number of trucks available at a distribution center is less sensitive to the number and location of orders because a truck can load several packages to be delivered in one trip, while current UAV delivery prototypes are constrained to one package per trip. Similarly, location of the distribution center becomes more important for a UAV system with delivery vehicles traveling in a straight line to a target location, and therefore vary less in fuel consumption and travel time for each delivery.

1) Assumptions

Much of the data necessary to formulate an accurate model of future UAV delivery systems is still unavailable. The following is a list of assumptions made in this study:

- Each address has an equal probability of ordering a package to be delivered by a UAV.
- 526 packages were ordered and delivered in each 30-minute time interval. This estimation comes from information provided by the local US Postal Service and assuming peak orders [6].
- UAVs fly in a straight line (unless interfered by a no-fly zone) at top speed to each delivery location and return immediately along the same path.
- Each package delivered earns 30 dollars with no variation with value of goods delivered.

2) Distance Travelled

The distance travelled by the delivery UAVs indirectly corresponds to costs associated with power and maintenance. Together with the speed of the UAVs, distance also limits the number of packages that can be delivered in a 30-minute time frame. A list of over 15,000 potential delivery locations for the city of Provo, UT was gathered by using the Utah County land records website [7] and an online geocoding tool [8]. These addresses include both commercial and residential addresses. Each of these potential delivery locations are illustrated in Fig. 1.

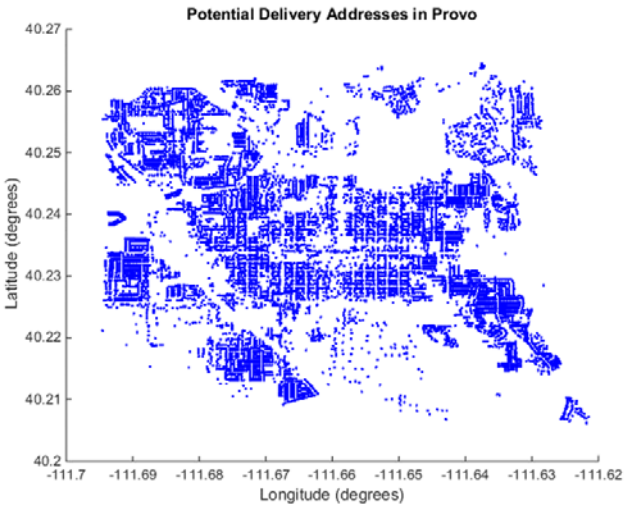


Fig. 1. Potential Delivery Addresses for Provo, UT. Each dot represents a registered address, including residential and commercial addresses. This data was obtained using the Utah County land records website [7] and an online batch geocoding tool [8].

A representative distance travelled for a 30-minute time period was calculated for each iteration of optimization by randomly selecting 526 addresses from the list of potential delivery addresses and summing the distances between each selected address location (x_t, y_t) and the delivery center location (x, y).

$$Distance(x,y) = \sum_{i=1}^{n} \sqrt{(x - x_t)^2 + (y - y_t)^2} \quad (2)$$

This process was used for each selected delivery point, except when the delivery path was interfered by a no-fly zone. When this occurred, the distance was calculated as the shortest path around the no-fly zone (see fig. 2). If a delivery is selected within the no-fly zone then the UAV would take a straight path, assuming that the delivery was ordered by those in authority of the no-fly zone. Source code for calculating the distance to each delivery point is available in Appendix II.

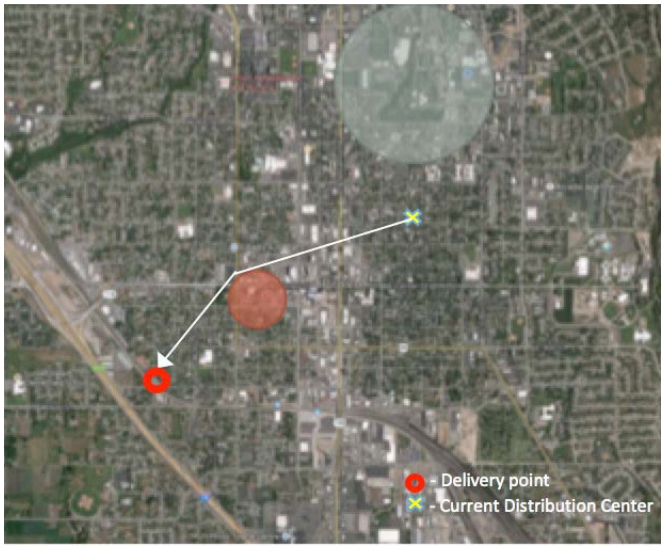


Fig. 2. Active "no-fly zone" example illustrating an exception to the normal distance calculation. Rather than calculating the direct distance between distribution center and target, the algorithm calculates an estimated shortest possible distance around restricted airspace to the target location.

B. Income

In this model, the income is based solely on the number of packages delivered; that is, each package delivered earns a set amount of money for Amazon. As stated above, our model uses an estimate based on package delivery information from the local post office for the number of packages delivered in 30 minutes (526). The amount of income for each package delivered was estimated from the average value of each Amazon order, with an added amount for the half-hour delivery.

Since the number of delivered packages can vary widely based on the time of day and day of the year, this model uses a value representative of a high demand for deliveries as a worst-case analysis. The Christmas season or the release of a popular video game are examples of when the demand for deliveries might be unusually high.

C. Cost

1) Real Estate Costs

To model the cost of land at a given location, we used actual real estate listings for the city of Provo, UT [9]. The mean listing price for each block was compiled into a 41-by-41 matrix. Fig.3 illustrates these mean listing prices. With data for each individual block, this portion of the cost function is a discrete function. Therefore, the objective function cannot be minimized using gradient-based methods when real estate is included.

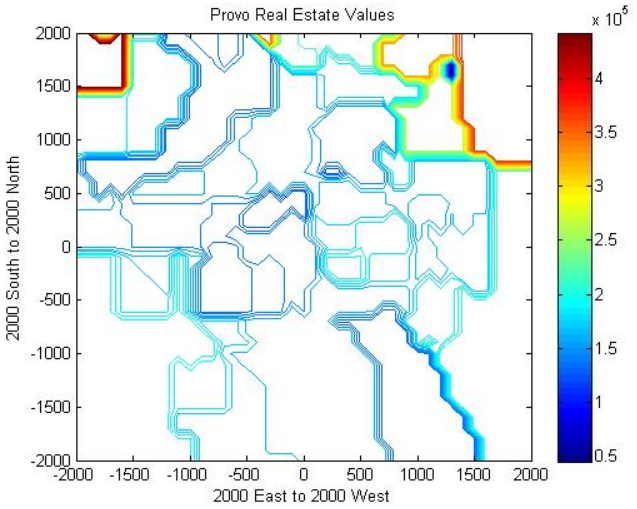


Fig. 3. Average Real Estate costs across the 40-by-40 block area of Provo, UT. The contours represent the mean listing price for each block as obtained from current real estate listings [9].

2) Initial UAV Purchase

Using information about existing commercially available UAVs, the initial purchase cost of a drone was estimated at $8000 [10]. Our preliminary model of this cost was calculated with a simple multiplication of cost times the number of drones, but this method presented several problems when optimizing for a 30-minute delivery time. Therefore, we scaled this parameter to make it a function of delivery time by amortizing the cost over its flight lifetime. According to preliminary research, the flight lifetime of a UAV is estimated at 1080 hours [11]. With these parameters we then estimated the purchase cost per hour of flight as shown in equation 3.

$$Drone\ Initial\ Cost\ = \frac{8000 * t_{delivery}}{1080} \tag{3}$$

where $t_{delivery}$ is defined as

$$t_{delivery} = \frac{Distance}{N * v_{ave}} \tag{4}$$

with v_{ave} representing the average velocity of a UAV, estimated at 35 miles per hour.

3) Penalty Costs

On each iteration, if the delivery time (see equation 4) did not fit within the 30-minute limit, a penalty was added as shown in equation 5.

$$Penalty = Penalty_{time} * Penalty_{cost} \tag{5}$$

Where $Penalty_{time}$ is defined as the absolute difference between the 30-minute limit and the actual delivery time in hours, with $Penalty_{cost}$ estimated at $100 per hour late.

4) Maintenance Costs

The cost associated with UAV maintenance is made of fixed and variable costs. A cost factor, C_{DT}, represents the maintenance and fuel cost for a UAV traveling a certain distance (dollars per mile), and is multiplied by the distance that the UAV is required to travel in a 30 minutes. The fixed cost, C_r, associated with routine maintenance for each drone and independent of distance travelled, is added to the variable cost.

$$Maintenance(x, y, N) = C_{DT} * \overline{DT}(x, y) + C_r * N \quad (6)$$

D. Constraints

A location constraint was used in order to exclude locations in which construction of a distribution center would not be permitted. These areas may be limited by zoning requirements or the unavailability of real estate. These "no-build zones" were modeled as inequality constraints as shown below:

$$D = \sqrt{(x_{nb} - x)^2 + (y_{nb} - y)^2} \quad (7)$$

$$c = r_{nb} - D \quad (8)$$

Where D is the Distance between a potential distribution center location (x, y) and the center of a no-build zone (x_{nb}, y_{nb}). The constraint value c is then calculated by subtracting the value D from the radius of the no-build zone r_{nb} (see equation 8). If c is positive, the potential distribution center location is located within the no-build zone, and that potential delivery location is not included. In the model of Provo, Brigham Young University campus was used as an example no-build zone (see fig.4).

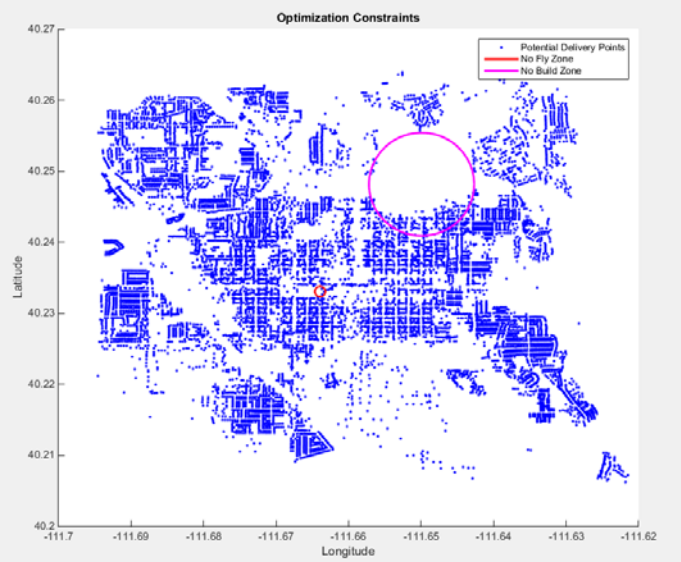


Fig. 4. Optimization Constraints. The larger ring covering the campus area of Brigham Young University signifies a direct constraint on distribution center location. i.e. an area where building is not permitted. The smaller ring is over the Provo Police Department, and signifies controlled airspace which would require a delivery UAV to fly around.

E. Optimization

The objective function and constraints, as explained above, were modeled using MATLAB. Source code for each function is available in Appendix II.

1) Algorithm

The model was optimized using the MATLAB genetic algorithm with a beginning population of 30. A gradient-free algorithm was necessary due to the discontinuous real-estate values of the objective function, and the population level was chosen as 10 units per design variable. The 30 initial population points were made up of randomly generated longitude and latitude points across the 40-by-40 block area of Provo and a range of 1 to 1000 delivery UAVs. The settings for mutation, stopping criteria, and other options were held at the MATLAB defaults.

We also attempted using the GODLIKE optimization algorithm, but found that it took much longer to solve and lacked much of the functionality of the MATLAB genetic algorithm.

III. Results

After 5 generations and 1680 objective function calls, the genetic algorithm located the optimum distribution center location at (-111.6506, 40.2328) with 28 operating UAVs (see fig.5). Appendix I shows a complete optimization history.

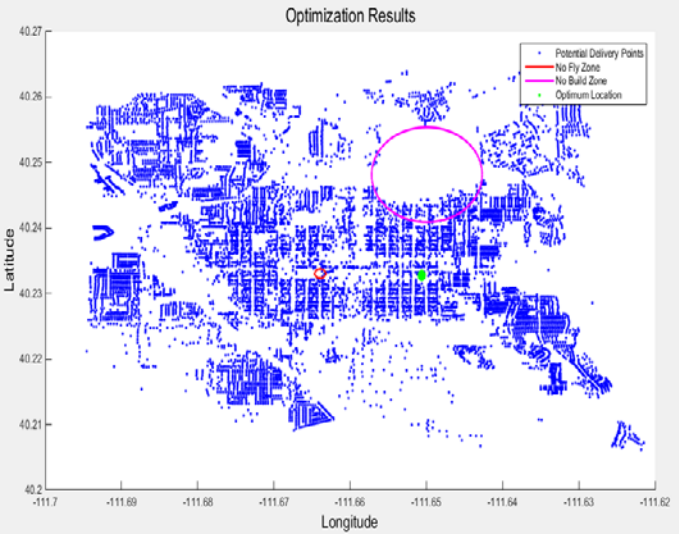


Fig. 5. Optimal Location Result. The optimal distribution center location is at (-111.6056, 40.2328), or around 500 E. Center Street, and is signified by a green dot. The optimal number of UAVs is 28. Note again that this implementation is an example of how a more detailed model could be used for optimization with more detailed parameters in a specific city.

As stated previously, these results are only useful as an approximation for the city of Provo. The important result is a useful model that can be used as a framework for future optimization studies.

IV. Conclusion

The model developed in this study was successful in optimizing the parameters of a fulfillment center for Amazon's future Prime Air service. The MATLAB genetic algorithm is considerably slower than gradient based methods, but was able to locate an optimum solution for the non-differentiable objective function. The optimizer located an optimum point of $(-111.6506, 40.2328)$, around 500 E. Center Street, with $N = 28$ operating UAVs while respecting all current location constraints. As expected with Gradient-free optimization methods, there is limited repeatability with each evaluation of the genetic algorithm, as each optimum point is slightly different.

The inaccuracies of this model come mostly from unavailable information. A true model of income and cost would be based on hundreds of variables, whereas this model only uses three. This model also focuses on the viability of operation within a 30-minute time frame, even though most costs occur over a much longer time. Future work would involve scaling or amortizing the real-estate cost to find the long-term profitability of the UAV delivery service and filling in details about other estimated parameters.

As more information about costs and regulations associated with UAV delivery systems become available, this study may be used as a framework to develop models that will find the optimal operation for a UAV package delivery system for specific cities.

References

[1] "Amazon Prime Air," Amazon, 2015. [Online]. Available: http://www.amazon.com/b?node=8037720011. [Accessed 20 March 2015].

[2] Q. Tang and F. Xie, "Distribution Center Optimization by Genitic Algorithm," in *5th International Conference on Natural Computation*, Tianjian, China, 2009.

[3] J. Yu, Y. Qin, L. Jia and Z. Xing, "Optimization of vehicles arrangement in distribution center," in *Proceedings of the 1st international conference on Logistics, Informatics and Service Science*, Beijing, 2011.

[4] J. Chen, W. Wu, Y. Guo and J. Huang, "The research of distribution center optimization addressing based on energy saving and emission reducting confine factors," in *2010 International Conference on Management and Service Science*, Wuhan, China, 2010.

[5] B. Litvinas and N. Dickerson, "Analyzing the Business Case and Economic Viability of Unmanned Aircraft within the Nation's Airspace," in *Aviation Technology, Integration and Operations Conference*, Los Angeles, 2013.

[6] Sharon, Interviewee, *USPS Deilvery rates in Provo.* [Interview]. 2 April 2015.

[7] "Real Property Address Search," Utah County, 7 April 2015. [Online]. Available: http://www.utahcountyonline.org/LandRecords/AddressSearchForm.asp. [Accessed 7 April 2015].

[8] D. B. Zwiefelhofer, "Batch Geocoding," 2015. [Online]. Available: http://www.findlatitudeandlongitude.com/batch-geocode/#.VSQRtvnF-WZ. [Accessed 7 April 2015].

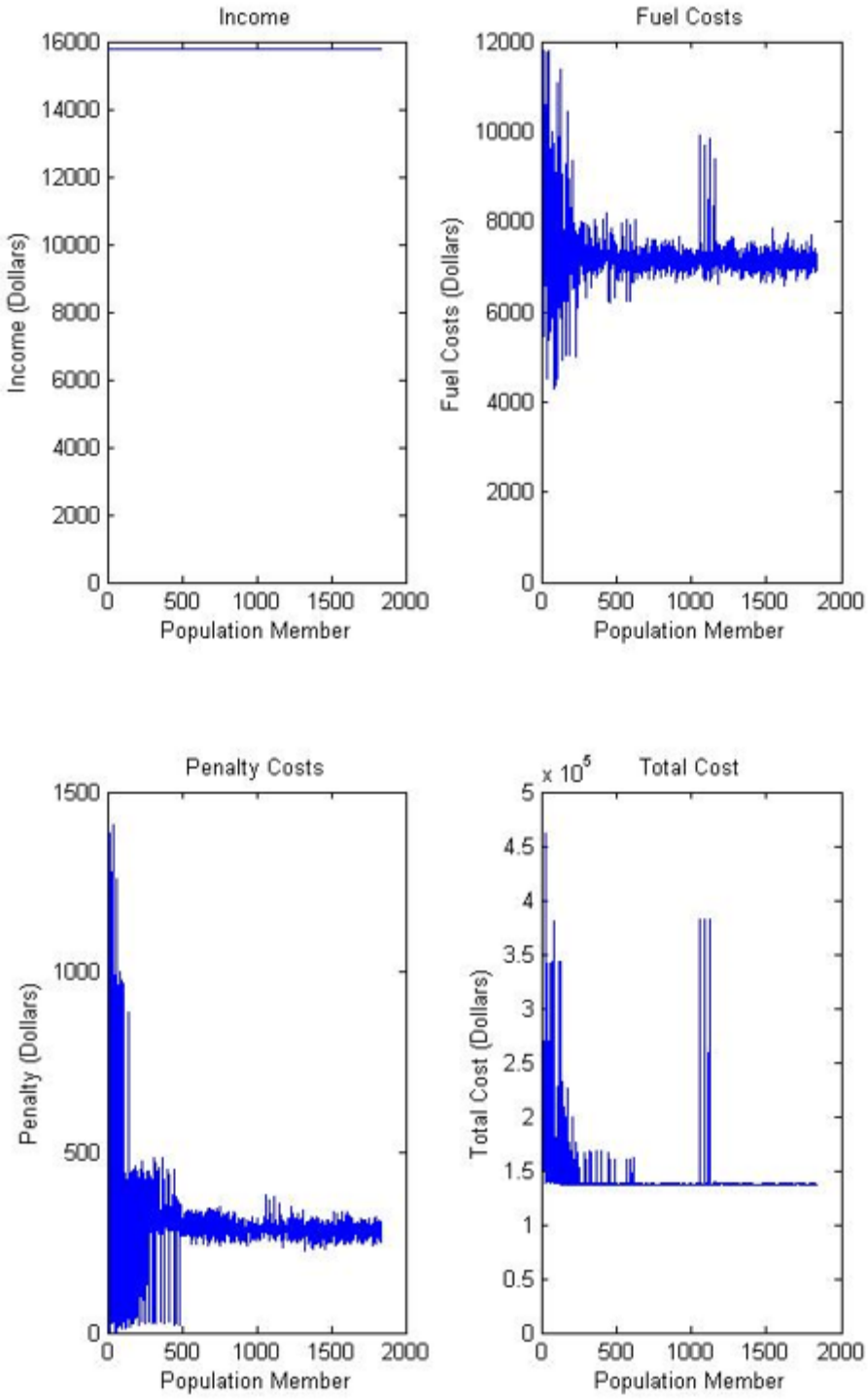
[9] "Provo Real Estate Overview," Trulia, [Online]. Available: http://www.trulia.com/real_estate/Provo-Utah/. [Accessed 15 March 2015].

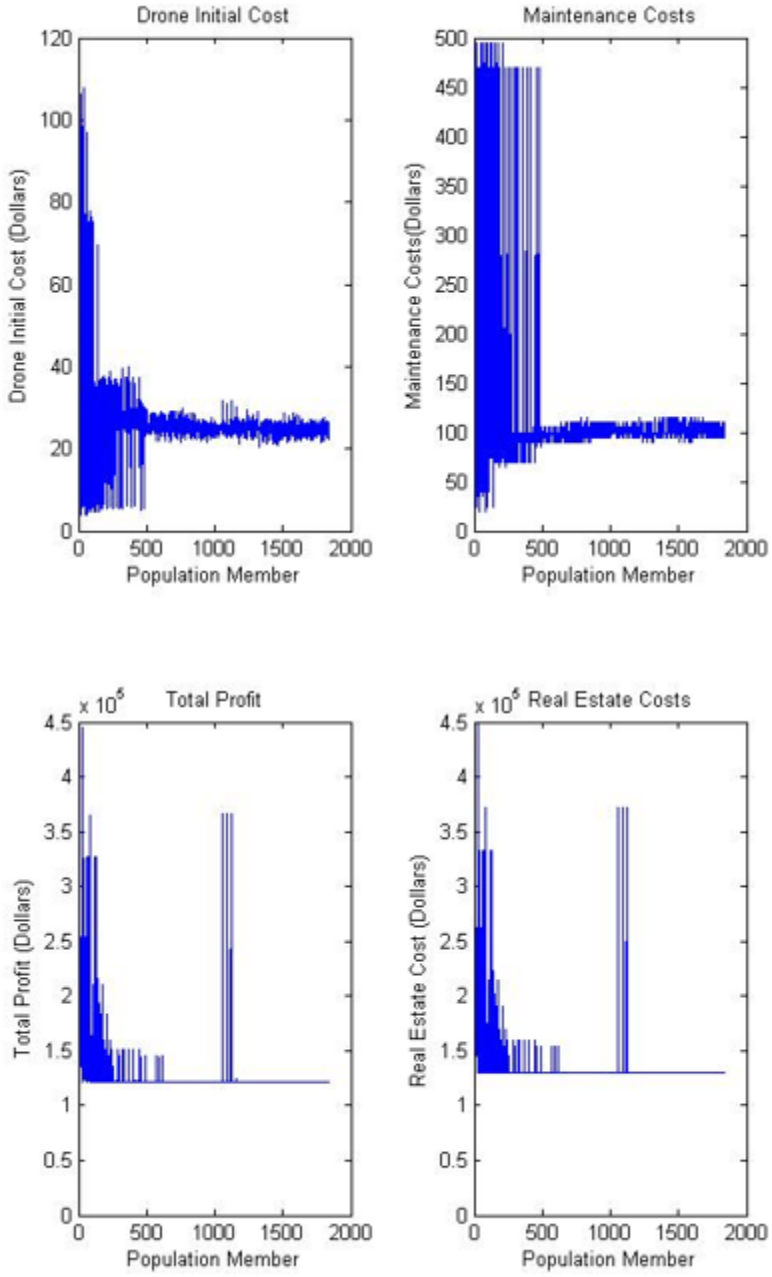
[10] "Draganflyer X6 UAV six rotor helicopter with 20.1 MP digital camera and Patented Folding Airframe," Draganfly Innovations Inc., 2015. [Online]. Available: http://www.rctoys.com/rc-products/DF-X6-PACKAGE.html?gclid=CjwKEAjw9bKpBRD-geiF8OHz4EcSJACO4O7TiYy33QTzHRP7YeTo6qlLhYO_xeVkG1vbpCRkQ7d7NRoCX2Hw_wcB. [Accessed 14 April 2015].

[11] Office of the Secretary of Defense, "Unmanned Aerial Vehicle Reliability Study," Office of the Secretary of Defense, Washington, 2003.

V. Appendix I

1) Optimization History





It is useful to observe from these figures that the penalty function is active even at the optimum point (approximately $300). Therefore, over the course of delivering 526 packages in 30 minutes, there is a total late delivery penalty of three hours.

VI. Appendix II

1) Matlab Code

```
--------------------------------Top Level Function-----------------
---------
function [xopt,fopt,exitflag,output] = main_new()
%xopt,fopt,exitflag,output
close all
clc
clear all
global counter_record
global counter_record2
persistent xrecord;
persistent counter;
persistent x0
counter_record = 0;
counter_record2 = 0;
persistent center2 ;      %BYU Campus
persistent radius2 ;
center2 = [-111.649933 40.248153];
radius2 = (1/2)/68.97;

nvars = 3;
xrecord=zeros(1,2);
counter = 0;
counter_record = 0;

%initalize starting point, upper bound, lower bound, no fly
zone location
%and radius

min_x=-111.6946;
min_y=40.2062;
max_x=-111.6215;
max_y=40.2644;
min_n=1;
max_n=100;
LB = [min_x;min_y;min_n]; %arbitary upper bound
UB = [max_x;max_y;max_n]; %arbitrary lower bound
InitPopRange = zeros(30,3);
for i = 1:30
   for j = 1:3

      if j==1
      a1 = min_x;
      b1 = max_x;
      r1 = (b1-a1).*rand(1) + a1;
      InitPopRange(i,j)=r1;
      end

      if j==2
      a2 = min_x;
      b2 = max_x;
      r2 = (b2-a2).*rand(1) + a2;
      InitPopRange(i,j)=r2;
      end

      if j==3

      r3 = randi(max_n);
      InitPopRange(i,j)=r3;
      end


   end
end

%----------------------Create Options for ga-------------------
options = gaoptimset(...
      'PopulationSize',30,...
      'Display','iter',...
      'InitialPopulation', InitPopRange,...
      'Generations',10,...
      'UseParallel','always');

[xopt,fopt,exitflag,output]=ga(@obj,nvars,[],[],[],[],LB,UB,
@con,options);
new_real_estate_plot(xopt,radius2,center2);

plot_record(2,0,0,0,0,0,0,0);

%Objective Function
   function val = obj(x)
      x(3)=ceil(x(3));
      val = Profit(x);
      if counter == 0
         xrecord = x;
         x0=x;
      else
         xrecord = [xrecord;x];
      end
      counter = counter +1;
   end

%Constraint function
   function [c,ceq] = con(x)

   distance=sqrt((center2(1)-x(1))^2 + (center2(2)-x(2))^2);
   c2 = radius2 - distance;
   c = [c2];
   ceq = [];
   end
end

%Profit function
function [ profit ] = Profit(vec)
%% Profit.m
% Objective function for optimizing parameters for a
drone-delivery
% distribution center.

%   profit = estimated profit over a year for the provided
parameters.
%   X = the longitudinal location in decimal form (i.e. -
111.263344)
%   Y = the latitudinal location in decimal form (i.e. -
46.258643)
%   N = the number of drones for optimal performance at
the given location

X=vec(1);
Y=vec(2);
```

```
N=vec(3);
%% User Defined Variables-----------------------------
D = 526; % Avg. number of packages delivered per week
2180
IncomePerDelivery = 30; %dollars
Weeks = 52; % Time for analysis (weeks)
ave_speed = 35; %average speed of a UAV in miles/hr
hours = .5; %working hours in a week - 8hrs * 5 days
penalty_hours_cost = 100; %dollars per hour penalty

%% Profit Functions

Distance = totalmean_with_noflyzone(X,Y,D);

Delivery_time = Distance/(N*ave_speed);
%Penalty function for taking too long to deliver

if ((abs(Delivery_time) > hours)||(abs(Delivery_time) <
hours))
   penalty_hours = abs(hours-Delivery_time);
end

Income = IncomePerDelivery*D;
RE_Cost = re_provo([X,Y]);
MDCost = Distance*3; % "Maintenence Distance": and
Battery cost for a travelled distance
DICost = (8000*Delivery_time)/(1080); % "Drone Initial":
cost of UAV purchase - amortized per hour of flight
FMCost = N*5; % "Fixed Maintenence": cost for
maintaining each drone each week regardless of flight
Penalty = penalty_hours*penalty_hours_cost;

Cost = MDCost+DICost+FMCost+Penalty+RE_Cost;
profit = Cost-Income;
plot_record(1,Income,MDCost,DICost,FMCost,Penalty,Co
st,profit,RE_Cost);
end

function [] =
plot_record(flag,Income,MDCost,DICost,FMCost,Penalty,
Cost,Profit,RE_Cost)
persistent Income_record
persistent MDCost_record
persistent DICost_record
persistent FMCost_record
persistent Penalty_record
persistent Cost_record
persistent RECost_record
persistent Profit_record
global counter_record
global counter_record2
if counter_record == 0

Income_record = 0;
MDCost_record = 0;
DICost_record = 0;
FMCost_record = 0;
Penalty_record = 0;
Cost_record = 0;
Profit_record = 0;
RECost_record = 0;
end

if flag == 1
Income_record = [Income_record;Income];
MDCost_record = [MDCost_record;MDCost];
DICost_record = [DICost_record;DICost];
FMCost_record = [FMCost_record;FMCost];
Penalty_record = [Penalty_record;Penalty];
Cost_record = [Cost_record; Cost];
RECost_record = [RECost_record; RE_Cost];
Profit_record = [Profit_record; Profit];

counter_record = counter_record+1;
counter_record2 = [counter_record2;counter_record];
end

if flag == 2
figure (3)
subplot (2,4,1)
plot(counter_record2,Income_record);
title('Income')
xlabel('Population Member');
ylabel('Income (Dollars)');
subplot (2,4,2)
plot(counter_record2,MDCost_record);
title('Fuel Costs')
xlabel('Population Member');
ylabel('Fuel Costs (Dollars)');
subplot (2,4,3)
plot(counter_record2,DICost_record);
title('Drone Initial Cost')
xlabel('Population Member');
ylabel('Drone Initial Cost (Dollars)');
subplot (2,4,4)
plot(counter_record2,FMCost_record);
title('Maintenance Costs')
xlabel('Population Member');
ylabel('Maintenance Costs(Dollars)');
subplot (2,4,5)
plot(counter_record2,Penalty_record);
title('Penalty Costs')
xlabel('Population Member');
ylabel('Penalty (Dollars)');
subplot (2,4,6)
plot(counter_record2,Cost_record);
title('Total Cost')
xlabel('Population Member');
ylabel('Total Cost (Dollars)');
subplot (2,4,7)
plot(counter_record2,Profit_record);
title('Total Profit')
xlabel('Population Member');
ylabel('Total Profit (Dollars)');
subplot (2,4,8)
plot(counter_record2,RECost_record);
title('Real Estate Costs')
xlabel('Population Member');
ylabel('Real Estate Cost (Dollars)');
counter_record = counter_record + 1;
end
end
```

```
----------Function for plotting output ----------

function [] = new_real_estate_plot(xopt,radius2,center2)

figure(2);

% X = -1000:50:1000;
% Y = -1000:50:1000;
% cost = zeros(length(X));
%
% for i = 1:length(X)
%    for j = 1:length(Y)
%        xvec = [X(j),Y(i)];
%        cost(i,j) = 1*totalmean(xvec)+1000*re_provo(xvec);
%    end
% end
% contour(X,Y,cost)
% hold on
% radius = 300; %value of the radius from the distribution
center to no fly zone
% radius2 = 100;
% nfz_center = [0,0]; %x and y location of no fly zone
% nfz2_center = [600,0];

data = dlmread('Provo_addresses_24March_lat_lon.txt','\t');
lon = data(:,1);
lat = data(:,2);
b = 100;

scatter(lat,lon,'b.');
hold on

center1 = [-111.663923 40.233052];      %Provo Police
radius1 = (1/20)/68.97;

ang=0:0.01:2*pi;
xp1=radius1*cos(ang);
yp1=radius1*sin(ang);
plot(center1(1)+xp1,center1(2)+yp1,'r','Linewidth',2);

xp2=radius2*cos(ang);
yp2=radius2*sin(ang);
plot(center2(1)+xp2,center2(2)+yp2,'m','Linewidth',2);
a = 1000;

%Plot ending point
scatter(xopt(1),xopt(2),a,'.','gr');

%scatter(x0(1),x0(2),a,'.','r');

title('Optimization Results');
xlabel('Longitude');
ylabel('Latitude');

legend('Potential Delivery Points','No Fly Zone','No Build
Zone','Optimum Location');
hold off;

----------Totalmean function----------
function [ mean ] = totalmean_with_noflyzone( x, y, n)

%Initialize variables
global ra;

center1 = [-111.663923 40.233052];      %Provo Police
radius1 = (1/20)/68.97;
center2 = [-111.649933 40.248153];      %BYU Campus
radius2 = (1/2)/68.97;
center3 = [-111.642122 40.25965];       %Provo Temple
radius3 = (1/4)/68.97;
C = [center1;center2;center3];
r = [radius1;radius2;radius3];

%read in addresses from text file and split into column
vectors
data = dlmread('Provo_addresses_24March_lat_lon.txt','\t');
targetsY = data(:,1);
targetsX = data(:,2);

%calculate random integer index value between 1 and
number of addresses
ra = randi([1 length(targetsX)],n,1);

%Shift the distrubution center to be the center of the grid
C(:,1) = C(:,1) - x;
C(:,2) = C(:,2) - y;
targetsX = targetsX - x;
targetsY = targetsY - y;
data(:,1) = data(:,1) - y;
data(:,2) = data(:,2) - x;

%Calculate the angle theta and distance to center of each
'no-fly-zone'
for i = 1:length(C)
    if C(i,1) > 0 && C(i,2) > 0                  %first quadrant
       theta(i) = atan(C(i,2)/targetsX(i));
   elseif C(i,1) < 0 && C(i,2) > 0               %2nd
quadrant
       hyp = sqrt(C(i,1)^2 + C(i,2)^2);
       theta(i) = acos(C(i,1)/hyp);
   elseif C(i,1) < 0 && C(i,2) < 0               %3rd quadrant
       theta(i) = pi + atan(C(i,2)/C(i,1));
   elseif C(i,1) == 0                            %check for dots
on y axis
       if C(i,2) > 0
          theta(i) = pi/2.0;
       else
          theta(i) = 3*pi/2.0;
       end
   elseif C(i,2) == 0                            %check for dots
on x axis (and 0,0)
       if C(i,1) > 0
          theta(i) = 0;
       else
          theta(i) = pi;
       end
   else
       hyp = sqrt(C(i,1)^2 + C(i,2)^2);
       theta(i) = 2*pi + asin(C(i,2)/hyp);
   end
```

```
   d(i) = sqrt(C(i,1)^2 + C(i,2)^2);
%Distance from distribution to center of no fly zone
   phi(i) = abs(atan(r(i)/d(i)));                    %Angle of
center to outside radius

end

%Calculate the angles for each point in the population
for o = 1:length(ra)
   data_ra(o,:) = data(ra(o),:);
   random = ra(o);
   targetsY_ra(o)=targetsY(random);
   targetsX_ra(o)=targetsX(random);
   if targetsX_ra(o) > 0 && targetsY_ra(o) > 0
%first quadrant
      angle(o) = atan(targetsY_ra(o)/targetsX_ra(o));
   elseif targetsX_ra(o) < 0 && targetsY_ra(o) > 0
%2nd quadrant
      hyp = sqrt(targetsX_ra(o)^2 + targetsY_ra(o)^2);
      angle(o) = acos(targetsX_ra(o)/hyp);
   elseif targetsX_ra(o) < 0 && targetsY_ra(o) < 0
%3rd quadrant
      angle(o) = pi + atan(targetsY_ra(o)/targetsX_ra(o));
   elseif targetsX_ra(o) == 0
%check for dots on y axis
      if targetsY_ra(o) > 0
         angle(o) = pi/2.0;
      else
         angle(o) = 3*pi/2.0;
      end
   elseif targetsY_ra(o) == 0                        %check
for dots on x axis (and 0,0)
      if targetsX_ra(o) > 0
         angle(o) = 0;
      else
         angle(o) = pi;
      end
   else
      hyp = sqrt(targetsX_ra(o)^2 + targetsY_ra(o)^2);
      angle(o) = 2*pi + asin(targetsY_ra(o)/hyp);
   end

end

%Calculate distance to each target from distribution point
around 'no-fly-zone'
distance = zeros(length(ra),1);
for k = 1:length(ra)
   for l = 1:length(theta)
      if norm(data_ra(k,:)) >= norm(C(l,:))
         if angle(k) > theta(l) - phi(l) && angle(k) < theta(l)
            rot = [cos(theta(l)) sin(theta(l));-sin(theta(l))
cos(theta(l))];
            t = rot*data_ra(k,:)';
            lilc = rot*C(l,:)' - [0;r(l)];
            distance1 = norm(lilc);
            distance2 = norm(t - lilc);
            distance(k) = distance1 + distance2;
         elseif angle(k) < theta(l) + phi(l) && angle(k) >
theta(l)
            rot = [cos(theta(l)) sin(theta(l));-sin(theta(l))
cos(theta(l))];
            t = rot*data_ra(k,:)';
            lilc = rot*C(l,:)' + [0;r(l)];
            distance1 = norm(lilc);
            distance2 = norm(t - lilc);
            distance(k) = distance1 + distance2;
         end
      end
   end
   if distance(k) == 0
      distance(k) = norm(data_ra(k,:));
   end
end
%Total distance in degrees
mean = sum(distance);
%Total distance in miles
mean = 2*mean*68.97;
end
```