Minimizing Travel Cost Project

Annie Gilliam, Trinity Walsh, Lexi Barry

Introduction:

The Minimizing Travel Costs Project presents the issue of determining where in the continental U.S. where the ClawedAI Corporation should locate its office in order to minimize travel costs between Middlebury, VT, Dallas, TX, Tampa, FL, and Detroit, MI. Five technicians from this corporation fly together to Middlebury 3 times per month, to Dallas two times per month, to Tampa 4 times per month and to Detroit once per month. The group travels on a corporate jet, the SyberJet SJ30i, which costs $6.74 per mile to operate. First, we modelled the U.S. as a flat 2-dimensional plane, using the latitude and longitude coordinates of each location to build an (x,y) coordinate system by dividing the coordinates by their respective miles/degree values to obtain mileage coordinates.

Overview/Known Information:

- Staff: 5
- Travel (Midd:3, Dallas:2, Tampa: 4, Detroit: 1)
- Land in small or large airports
- Office, travel, office (within a day)
- Cost: $6.74 / mile on Syber Jet
- 70 miles / degree of latitude and 50 miles / degree of longitude
- Plane flies in straight lines between paths and location
- Home office can be anywhere
- What locations will minimize monthly total cost? What town is near the ideal location
- Hessian Matrix: second order partial derivative to find curvature

City Coordinates: We first gathered these city coordinates through a google search.

City	Latitude	Longitude	Monthly Trips
Midd	44 N	73.2 W	3
Dallas	32.8 N	96.8 W	2
Tampa	27.9 N	82.5 W	4
Detroit	42.3 N	83 W	1

Then, we determined the milage coordinates by multiplying the latitude and longitudes for each city by their respective miles/degree values.

X = 50 (longitude) north is positive while south is negative , y = 70 (latitude) east is positive and west is negative

City	x	y
Midd	-3660	3080
Dallas	-4840	2296
Tampa	-4125	1953
Detroit	-4150	2961

We then made these milage coordinates correspond to x and y variables for each city. Temporary x and y variables are initialized to be used in future equations. We also created variables for the frequency of trips for each city, as given.

```
%temporary variables
syms x y

%Middlebury
xm = -3660;
ym= 3080;


%number of trips
mid = 3;
dal = 2;
tamp = 4;
detr = 1;

%coordinates
%Dallas
xd = -4840;
yd=2296;

%Tampa
xt = -4125;
yt = 1960;

%Detroit
xdt = -4150;
ydt = 2961;
```

City Distances:

Distance for each travel trip: $d(x, y) = \sqrt{(x - x_1)^2 + (y - y_1)^2}$ This equation is just the distance formula which will have the shortest "birds-eye" travel between two locations. We will use this equation to create distance equations between each city and some arbitray (unknown) office, where the x_1 and y_1 values will be the city coordinates and the x and y variables will be the arbitrary office coordinates.

```
%cost of going to Midd
M = sqrt((x-xm)^2 + (y-ym)^2);

%cost of going to Dallas
Da = sqrt((x-xd)^2 + (y-yd)^2);

%cost of going to Tampa
T = sqrt((x-xt)^2 + (y-yt)^2);

%cost of going to Detroit
Dt = sqrt((x-xdt)^2 + (y-ydt)^2);
```

Total monthly travel Distance:

Then to calculate the total monthly travel distance, we needed to incorporate the number of monthly trips for travel to/from each city with our sum of the distances. Note that because each given trip frequency must be there and return from the office, we multiplied the whole function by 2. This gives the equation $D(x, y) = 2(\sum (\text{trip frequency})(\text{distance}))$

```
%total monthly flight distance eqn
%multiply by 2 bc each trip is there and back
D = 2*(mid*M + dal*Da + tamp*T + detr*Dt);
```

Cost Function:

Now to calculate the cost to operate this monthly travel distance, we need to just consider the given cost of $6.74/mile. Since the total distance equation, D, is in miles, we just multiply the function by the cost/mile to get the total monthly operating cost, which we denote C.

```
%monthly operating cost
C = 6.74*D;
```

Minimize Cost Function:

Now we want to minimize the cost function to find the cheapest location for the office between all of the cities. Therefore, we want to determine where the total monthly operating cost stops increasing or decreasing. In single variable calculus, this would be the First Derivative Test. However, since we are using both x and y coordinates in this function, we need to take the multivariable approach and take the partial derivatives of the cost function.

```
%now want to minimize cost function
%use gradient
Cx = diff(C,x);
Cy = diff(C,y);
```

By setting the gradient of C equal to a solution of $\langle 0,0 \rangle$, we can evaluate the critical points (maximum and minimum values). Matlab functions can help us recieve the solution efficiently.

```
%find critical point
%use vpasolve to numerically calculate solution
soln = vpasolve([Cx == 0, Cy == 0], [x, y]);
```

Solutions + Coordinate Conversions:

The critical value solutions for the x and y coordinates are expressed as decimal valeus by taking the doubles so that they can be interpreted and analyzed as coordinates.

```
%note minimum coordinates of solution as a double to be precise
xMin = double(soln.x)
```

```
xMin = -4.1540e+03
```

```
yMin = double(soln.y)
```

```
yMin = 2.1794e+03
```

Because these coordinates are still in miles, we need to convert back to longitude and latitude coordinates by dividing each by the given miles/degrees longitude and latitude conversions.

```
%get latitude/longitude from these minimum coordinates
% ex)long will be the number of degrees of long (xMin) / miles per degree
long = xMin/50
```

```
long = -83.0796
```

```
lat = yMin/70
```

```
lat = 31.1338
```

Longitude: 83.0796N

Latitude: 31.1338W

Now with these longitude and latitude values, we googled the corresponding location. The address is *Theo Lane, Lanier County, GA, United States of America*, which is within the city **Lakeland, GA**. Logically, without even considering specific costs or coordinates, this location makes sense from a map view as it is between all of the cities.

Now to find the actual minimum cost of the funciton, we can substitute the minimum x and y coordinates (the mile values, not the latitude/longitude coordinates) into the cost function.

```
%use this to calculate minimum monthly cost as a double
%plug the min values into the cost eqn's x and y values w/ subs function
minCost = double(subs(C, [x,y],[xMin, yMin]))
```

```
minCost = 8.2768e+04
```

Minimum Cost: $82768.00

Verifying Minimum with Hessian Matrix and Determinant for "Second Derivative Test" for multivariable calculus:

Now, to actually verify that these critical points are minimum values, and not maximums or saddle points, we must do the Second Derivative Test, which tells us how quickly the slopes change in different directions. In multivariable calculus, this test is done with a Hessian Matrix, which takes second partial derivatives of the cost function gradient. For this application, the Hessian Matrix will describe the local curvature of the cost function in every direction around the point, helping us idenitfy if it is a minimum.

```
%verify that this is a minimum w/ Hessian matrix
Cxx = diff(Cx,x);
Cyy = diff(Cy,y);
%by Clairaut's Thm, Cyx=Cxy, so just calculate one
Cxy = diff(Cx,y);
%now construct Hessian
H = [Cxx, Cxy; Cxy, Cyy];
```

We evaluate the Hessian Matrix at the critical point (local minimum coordinate values we solved for) to evaluate the curvature at that exact location.

```
%now evaluate Hessian at the critical point values
hMin = double(subs(H,[x,y],[xMin,yMin]))
```

```
hMin = 2x2
    0.2881    0.0214
    0.0214    0.0509
```

To determine if the values are a local minimum, we need to take the determinant of the Hessian Matrix AND evaluate the second partial of the cost function, C_{xx} in relation to 0.

```
%calc determinant
clear det
detH = det(hMin)
```

```
detH = 0.0142
```

```
%check if detH > 0 (will return 1 if true, 0 if false)
detH > 0
```

```
ans = logical
   1
```

```
%check second derivative test
CxxMin = double(subs(Cxx, [x,y], [xMin,yMin]))
```

```
CxxMin = 0.2881
```

```
%check if CxxMin > 0 (will return 1 if true, 0 if false)
CxxMin > 0
```

```
ans = logical
   1
```

Both the detH > 0 and $C_{xx} > 0$, telling us that the critical point is a local minimum!! Therefore, we correctly minimized the cost and this location should be used for the office.

Finally, we need to confirm that the city locations themselves wouldn't be the ideal office location, so calculated the monthly operating cost with the coordinates for each location.

```
%can confirm critical point as local minimum if Hessian determinant is
%positive and Cxx >0 @ critical point
midCost = double(subs(C,[x,y],[xm,ym]))
```

```
midCost = 1.1038e+05
```

```
dalCost = double(subs(C,[x,y],[xd,yd]))
```

```
dalCost = 1.1281e+05
```

```
tampCost = double(subs(C,[x,y],[xt,yt]))
```

```
tampCost = 8.3838e+04
```

```
detrCost = double(subs(C,[x,y],[xdt,ydt]))
```

```
detrCost = 1.0022e+05
```

These costs are all greater than the cost for the office in Lakeland, GA, confirming that it is the location we should use!

Conclusion:

Therefore, by minimizing the total monthly travel costs equation, we found the mileage coordinates for the location of the home base office. We then converted these mileage coordinates into latitude and longitude coordinates to determine the location of Lakeland, GA as the home office location that minimizes travel costs on the SyberJet SJ30 for the

ClawedAI Corporation. The closest airport in Lakeland, GA is Valdosta Regional Airport. It is about 20 miles out and will be the most cost effective for our company.