For office use only	Team Control Number	For office use only
T1 _____	US-7211	F1 _____
T2 _____		F2 _____
T3 _____		F3 _____
T4 _____		F4 _____

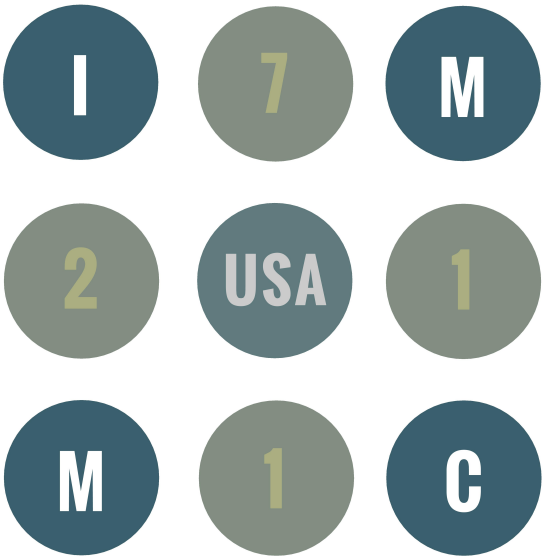
2017

USA Regional Round
The International Mathematical Modeling Challenge (IM^2C)
Summary Sheet
(Attach a copy of this page to the front of your solution paper.)

USA 7211: Say "No" to Jet Lag

In the modern world, technology creates global connections, enabling companies to have employees in all the corners of the world. However, at times physical meeting are necessary for optimal productivity. As this requires employees to congregate in one location, employees must travel vast distances, often crossing timezones. Arising problems are abnormal expenses and jet lag, which affects employees' productivity and compromises the value of these meetings. Our team attempts to solve this problem, using techniques from modeling airplane tickets prices based on distance and time of year to modeling individual neurons in employee brains. Many aspects of this problem were taken into consideration.

Our model selected Irkutsk, Russia for test scenario 1 and Astana, Kazakhstan for scenario 2.


I 7 M
2 USA 1
M 1 C


72
11
US
IMMC

Contents

1 Assumptions

Assumption 1: Participants will not be treated for jet lag

Several treatments for jet lag, such as individual schedule management, hypnotic agents, and chemicals exist (Choy & Salbu, 2011). However, we disregard these treatments. *Justification*: Regardless of treatment, we must minimize the effects of jet lag.

Assumption 2: The Earth is perfectly spherical

Justification: We assume that irregularities in the earths structure will not change our results. The largest irregularities on the earths surface are less than 0.1% of the earths radius.

Assumption 3: Jet lag is due to geographical change

Primary cause of jet lag is the change in geographical location rather than political time zone.
Justification: The oscillators in our brain which determine circadian rhythm are affected by geographical and environmental factors, (e.g. hours of sunlight).

Assumption 4: Mean flight cost is calculated by averaging available annual flight data

Justificiation: Although we do not have all ticket data, we have a rather large dataset, and averaging the flight data that we have accumulate for one year should approximate the true average of all ticket costs.

Assumption 5: Flights are always available between any two airports

Justification: As test inputs do not give year of meeting, finding actual flight data is impossible, and for ease of computation, we assume that flights are always available.

2 Model

Given an input of the time of year and locations of participants, we used Google Maps API to convert city names to coordinates and created a grid of (lattitude, longitude) non-nautical coordinates using extrema points. Each point in the grid was evaluated and given a fitness rating for each participant. This rating was calculated by weighing jetlag, distance, and cost fitness. All the point's fitness values were averaged to find its overall fitness value. The point with the highest overall fitness was chosen for the meeting location.

2.1 Jetlag

The suprachiasmatic nucleus (SCN) is a region of the mammalian hypothalamus that contains around 10,000 neurons. The firing of vasoactive intestinal polypeptide in the neurons of SCN indicates rhythm of the "body clock". Thus, it is responsible for circadian rhythms, which cyclically control sleep, physical activity, alertness, hormone levels, body temperature, immune function, and digestive activity. These factors contribute to productivity and mood, and therefore, it is essential for the neurons of SCN to be in synchrony with environmental conditions of the time zone. Because circadian rhythm is cyclic, we can model neurons in the SCN with individual oscillators.

A reputable dynamical system for modeling multiple oscillators is the Kuramoto Model, given by

$$\dot{\theta}_i = \omega_i + \frac{K}{N}\sum_{j=1}^{N}(\sin\theta_j - \theta_i) + \zeta_i$$

where each θ_i is an oscillator, and ω_i is the natural frequency of oscillator θ_i. Average ω_i is $\frac{2\pi}{24}$ and each ω_j is generated using a random Cauchy distribution centered on $\omega = \frac{2\pi}{24}$. Although each oscillator has its own natural frequency, over time, all oscillators tend to approach the same frequency, $\frac{K}{N}\sum_{j=1}^{N}(\sin(\theta_j - \theta_i))$. The constant K is known as coupling strength. As K increases, the oscillators more readily tend towards a uniform frequency. At the initialization of the system, note that $\theta_j = \theta_i$, and $\sin(\theta_j - \theta_i) = 0$. The coupling strength is quite weak at initialization, but it grows stronger as t increases. At initialization, ζ_i and ω_i have the largest influence on the frequency of the system. ζ_i is simply a noise function. In this model, the noise function accounted for the change in daylight due to geographic change.

$$\zeta_i = F\sin(\theta_i - \sigma t - \Delta(p))$$

where $\Delta(p)$ is the radial longitudinal difference in position after travel and $\frac{2\pi}{\sigma}$ was the daylight period in the new location. If $\Delta(p)$ is a negative value, we see that ζ_1 has

a more significant effect in phase offset. This subtlety results in significant differences between eastward and westward jet lag. We can use

$$R = \frac{1}{N}\sum_{j=1}^{N}(e^{\theta_j - \sigma t - \Delta(p)})$$

to model the average value of all θ_i. Childs and Strogatz showed that $\frac{\mathrm{d}R}{\mathrm{d}t}$ is

$$\dot{R} = \frac{(KR+F) - R^2(\overline{KR+F})}{2} - (\Delta + i\Omega)R$$

Where F, Δ, Ω are constants—due to the research of Lu et al—we selected as

$$\delta = 20$$

$$\Omega = 1.4\delta$$

$$K = 4.5\delta$$

$$f = 3.5\delta$$

for a normal human. The stability point of $\dot{R}$ is R_{sp}. At R_{sp}, the coupling factor has "cancelled out" the noise function, and the participant feels no jet lag. This is the point that $\dot{R}$ and θ_i tend towards. Since R_{sp} is the point of no jet lag, we quantify the amount of jetlag at point t as $|R - R_{sp}|$ So, at initialization, the starting points of θ_i are:

$$R_{sp} \cdot \text{rotation}$$

In this case, the rotation aroound the origin corresponds to the radian measure of longitudinal $\Delta(p)$ that the participant travels. This model's algorithm can be found in subsection 4.1.

2.2 Spherical Distance

We know

$$\vec{v_1} \cdot \vec{v_2} = \|\vec{v_1}\|\|\vec{v_2}\|\cos(\theta)$$

where θ is the angle in between the two vectors, also the distance between two points on a unit sphere. We use this to find spherical distance between points. Given longitude and lattitude, we convert coordinates to unit vectors using the formulas:

$$x = \cos(\text{lat}) \cdot \cos(\text{long})$$

$$y = \cos(\text{lat}) \cdot \cos(\text{long})$$

$$z = \sin(\text{lat})$$

We then get:

$$\vec{v_1} \cdot \vec{v_2} = 1 \cdot 1 \cdot \cos(\theta)$$

$$\arccos(\vec{v_1} \cdot \vec{v_2}) = \arccos(\cos(\theta))$$

$$\theta = \arccos(\vec{v_1} \cdot \vec{v_2})$$

We then multiply this by the Earth's circumference to find distance on Earth.

2.3 Cost

We also consider cost. To determine cost function, we analyze flight data from Los Angeles LAX to Tokyo NRT using SkyScanner's API. Data include date of departure and cost. We used this data to determine a function of cost in terms of departure date. Regression calculation procedure is described in 3.3.1-3.3.3.

As average air fare depends on season, we consider cost based on time of year. We analyzed international flight data from LAX to Tokyo NRT, which include time of departure and date. We used this information to regress a polynomial $f(t)$, where t is departure date. Mean cost was interpreted as

$$m(t) = \frac{0}{364} \int_1^{365} f(t)\mathrm{d}t$$

.

To estimate mean cost based on distance, we use 1,780,832 price points from Rome2Rio.com. Airfare cost follows a linear graph with the equation

$$C(x) = 50 + (x \cdot 0.11)$$

where C(x) is the cost in USD of flying x miles. Then,

$$C(d) = 50 + (d \cdot 0.177)$$

where C(d) is the cost in USD of flying d kilometers. We multiply our $f(t)$ by $c(d)$ to get a cost function based on flight distance and departure date, which we define as $C_m(d, t)$.

2.3.1 Multi-variable linear regression and Polynomial regression

The general hypothesis for a linear model of n variables is

$$h_\theta(x_0, x_1, x_2, \ldots, x_n) = \theta_0 \cdot x_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \ldots \theta_n \cdot x_n$$

We write all θ_i in a matrix:

$$\theta = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 & \ldots & \theta_n \end{bmatrix} \in \mathbb{R}$$

and all x_i in a matrix:

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

We write

$$h(x_0, x_1, x_2, \ldots, x_n) = \theta \cdot x = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 & \ldots & \theta_n \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= \theta_0 \cdot x_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \cdots + \theta_n \cdot x_n$$

for i input values we model input/output as the following:

m	x_0	x_1	x_2	x_3	$\ldots$	x_n	y
1	$x_0^{(1)}$	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$\ldots$	$x_n^{(1)}$	$y^{(1)}$
2	$x_0^{(2)}$	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	$\ldots$	$x_n^{(2)}$	$y^{(2)}$
3	$x_0^{(3)}$	$x_1^{(3)}$	$x_2^{(3)}$	$x_3^{(3)}$	$\ldots$	$x_n^{(3)}$	$y^{(3)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
m	$x_0^{(m)}$	$x_1^{(m)}$	$x_2^{(m)}$	$x_3^{(m)}$	$\ldots$	$x_n^{(m)}$	$y^{(m)}$

Here, $x_m^{(0)}$ are the inputs of $\theta_m(x)$. For instance,

$$x^{(2)} = \begin{bmatrix} x_0^{(2)} \\ x_1^{(2)} \\ x_2^{(2)} \\ x_3^{(2)} \\ \vdots \\ x_n^{(2)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

We use average mean square difference for cost function of regression, which is given by

$$j(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)})^2)$$

2.3.2 Gradient descent

We optimize the formula given above using gradient descent. The general algorithm for gradient descent is

$$\theta_i \coloneqq \theta_i - \alpha \frac{\partial}{\partial \theta_i}(j(\theta))$$

where $j(\theta)$ is a cost function and α is the learning rate. For multiple variables, algorithm is simultaneously evaluated for all θ_i. For example, a ten variable equation would be evaluated using:

$$\theta_0 \coloneqq \theta_0 - \alpha \frac{\partial}{\partial \theta_0}(j(\theta))$$

$$\theta_1 \coloneqq \theta_1 - \alpha \frac{\partial}{\partial \theta_1}(j(\theta))$$

$$\theta_2 \coloneqq \theta_2 - \alpha \frac{\partial}{\partial \theta_2}(j(\theta))$$

$$\vdots$$

$$\theta_{10} \coloneqq \theta_{10} - \alpha \frac{\partial}{\partial \theta_{10}}(j(\theta))$$

When we substitute in $j(\theta)$, we get

$$\theta_i \coloneqq \theta_i - \alpha \frac{\partial}{\partial \theta_i}(\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})))$$

which simplifies to

$$\theta_j \coloneqq \theta_j - \alpha \frac{1}{m}(h_\theta^{(i)} - y^{(i)}) \cdot x_j^{(i)}$$

2.3.3 Gradient descent in polynomial regression

We use the same input/output and algorithm in multi-variable linear regression, but we use 1 input and create more variables by increasing its power. The input would be as follows

m	x^0	x^1	x^2	x^3	...	x^n	y
1	$(x^{(1)})^0$	$x^{(1)}$	$(x^{(1)})^2$	$(x^{(1)})^3$	...	$(x^{(1)})^n$	$y^{(1)}$
2	$(x^{(2)})^0$	$x^{(2)}$	$(x^{(2)})^2$	$(x^{(2)})^3$	...	$(x^{(2)})^n$	$y^{(2)}$
3	$(x^{(3)})^0$	$x^{(3)}$	$(x^{(3)})^2$	$(x^{(3)})^3$	...	$(x^{(3)})^n$	$y^{(3)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
m	$(x^{(m)})^0$	$x^{(m)}$	$(x^{(m)})^2$	$(x^{(m)})^3$	...	$(x^{(m)})^n$	$y^{(m)}$

The algorithm for this model is described in Section 4.2.

3 Model Implementation

3.1 Kuramoto Model

We now find algorithm to implement model in Section 3.1. To find stability point of R, we solve zeroes of system $\dot{R}$, given by equation

$$\dot{R} = \frac{(KR + F) - R^2(\overline{KR + F})}{2} - (\Delta + i\Omega)R$$

Since starting values of θ_i, which are $\theta_i(0)$ and are centered about $R_{sp} \cdot \text{rotation} = R_{sp} \cdot e^{\Delta p}$, and $\frac{\mathrm{d}\theta_i}{\mathrm{d}t}$ are calculatable, we use Euler's method to find $\theta_i(0 + \mathrm{d}t)$. The algorithm is as follows:

Algorithm 1: Using Euler's method to find relative jetlag

input : R_{sp}, $\mathrm{d}t$, $\Delta(p)$
set $t = 0$;
initialization;
calculate *initial values of all* θ_i;
calculate *initial values of all* ω_i;
while $|R - R_{sp}|$ *is decreasing* **do**
  for *each* θ_i **do**
    calculate $\frac{\mathrm{d}\theta_i}{\mathrm{d}t}$;
  end
  set $\theta_i = \theta_i + \frac{\mathrm{d}\theta_i}{\mathrm{d}t} \cdot \mathrm{d}t$;
  set $t = t + \mathrm{d}t$;
  calculate *new* R;
  calculate $|R - R_{sp}|$;
end
return $t \cdot \mathrm{d}t$

3.2 Polynomial regression

As mentioned above, the cost function for multi-variable linear regression is

$$j(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)})^2$$

and its derivative is

$$\frac{\partial}{\partial \theta_j} j(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h(\theta^{(i)}) - y^{(i)}) x_j^{(i)}$$

Its algorithm is as follows:

Algorithm 2: Gradient descent to optimize n-th degree polynomial regression and return polynomial

input: x, y, α, r = minimum R^2
generate $n+1$ *random coefficients* θ_i *in* L_0;
while $R^2 < r$ **do**
  generate *empty list* L_1 *of coefficients*;
  for *each coefficient* θ **do**
    $\Delta = 0$;
    for *each input* x **do**
      calculate *predicted output* $f(x)$;
      m =**calculate** *difference between* $f(x)$ *and* y;
      $\Delta = \Delta + m$;
    end
    New Coefficient $= \theta_i - \alpha \frac{1}{m} \Delta$;
    store *New Coefficient in* L_1;
  end
  $L_0 = L_1$;
end
return L_0

Note that the coefficients are not updated immediately after calculation, thus referred to as simultaneous gradient descent.

3.3 Location selection

Given an input of annual flight data and locations of participants, we select the optimal meeting point using this weighting—$20\% \cdot (\text{cost fitness}) + 20\% \cdot (\text{distance fitness}) + 60\% \cdot (\text{jetlag fitness})$. The algorithm is as follows:

Algorithm 3: Selecting optimal meeting point

input: (Flight Dates, Flight Costs), locations, dt, K, F, Δ, Ω
$P(x)$ = **Algorithm 2** *(Flight Dates, Flight Costs)*;
for *lattitude* **range** ($\lfloor$**min** *(lattitudes)*$\rfloor$, $\lfloor$**max** *(lattitudes)*$\rfloor$, step $= 5$) **do**
  for *longitude* **range** ($\lfloor$**min** *(longitudes)*$\rfloor$, $\lfloor$**max** *(longitudes)*$\rfloor$, step $= 5$) **do**
    check *Is (lattitude, longitude) water?*;
    Fitness $= 0$;
    for *location* **do**
      Fitness + $(0.6 \cdot$ **Algorithm 1** $+ 0.2 \cdot \frac{\text{distance}}{2\pi} + 0.2 \cdot P(d,t))$
    end
    Fitness $= \frac{\text{fitness}}{\text{Count}(locations)}$;
    rank *(latitude, longitude)*
  end
end
return *Top ten ranked (latitude, longitude)*

4 Model Results

Given a coordinate input of the locations of participants and the time of year, our model optimizes the location for a meeting and predicts the total ticket costs. Since the exact coordinate is not necessarily a city, the nearest city with an airport city was selected as the best location.

The top coordinates for each scenario are shown, and if a nearby airport exists, it is shown as well.

Scenario 1
52.1863724, 93.1053239 - Kyzyl Airport, Kyzyl, Russia
52.1863724, 88.1053239
52.1863724, 98.1053239
47.1863724, 88.1053239 - Altay Airport, Altay, Xinjiang, China
52.1863724, 103.105324 - International Airport Irkutsk, Irkutsk, Russia
47.1863724, 93.1053239
52.1863724, 83.1053239
47.1863724, 83.1053239 - Tacheng Airport, Tacheng Prefecture, China
52.1863724, 108.105324 - Baikal International Airport, Ulan-Ude, Russia

Scenario 2
52.1863724, 73.9411199 - Astana International Airport, Astana, Kazakhstan
52.1863724, 68.9411199 - Kokshetau Airport, Kokshetau, Kazakhstan
52.1863724, 78.9411199
52.1863724, 63.9411199 - Kostanay Airport, Kostanay, Kazakhstan
47.1863724, 73.9411199 - Balkhash Airport, Balkhash, Kazakhstan
47.1863724, 78.9411199 -
47.1863724, 68.9411199 - Zhezhazgan Airport, Zhezqazgan, Kazakhstan

Typically, the predicted latitudinal coordinate $\approx$ average latitudinal location of the participants. However, the optimal longitudinal location tended towards the western side so that there would be more western travel than eastern travel. In our model, this is due to the dependence of the severity of jet lag begin on direction. Similarly to a study conducted by Lu et al, our model showed that eastward travel caused more loss of productivity from jet lag than westward travel. We conducted a paired t-test to compare the severity of jet lag from eastward travel and westward travel. The severity of jet lag from eastward jet lag was significantly greater than the severity of westward jet lag ($p = 0.002$, df $= 58$, $t = 2.00$). Locations that required less eastward travel were more optimal. Below is a map of our results and the input for both scenarios.

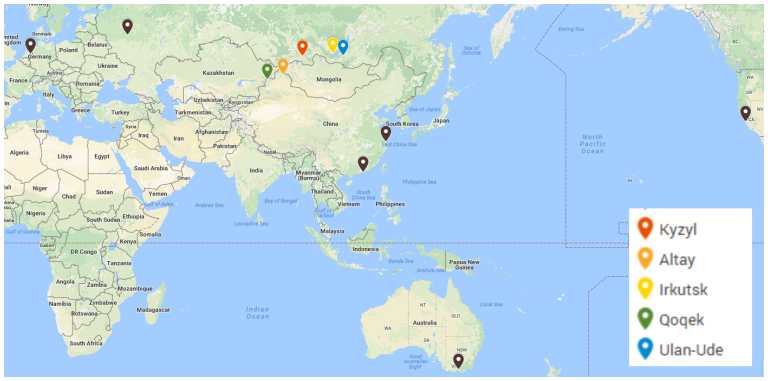


Figure 1: Above are the model's results for scenario one. We chose the optimal locations by considering the resulting amount of jetlag, distance, and expenses. The most optimal locations, in decreasing order, are Kyzyl, Altay, Irkutsk, Tacheng, Baikal. However, our team recommends we use Irkutsk, due to its relatively high availability of infrastructure.

Figure 2: Above are the model's results for scenario two. We chose the optimal locations by considering the resulting amount of jetlag, distance, and expenses. The most optimal locations, in decreasing order, are Astana, Kokshetau, Kostanay, Balkash, Zhezhazgan. Our team recommends we use Astana, due to its relatively high fitness rating.

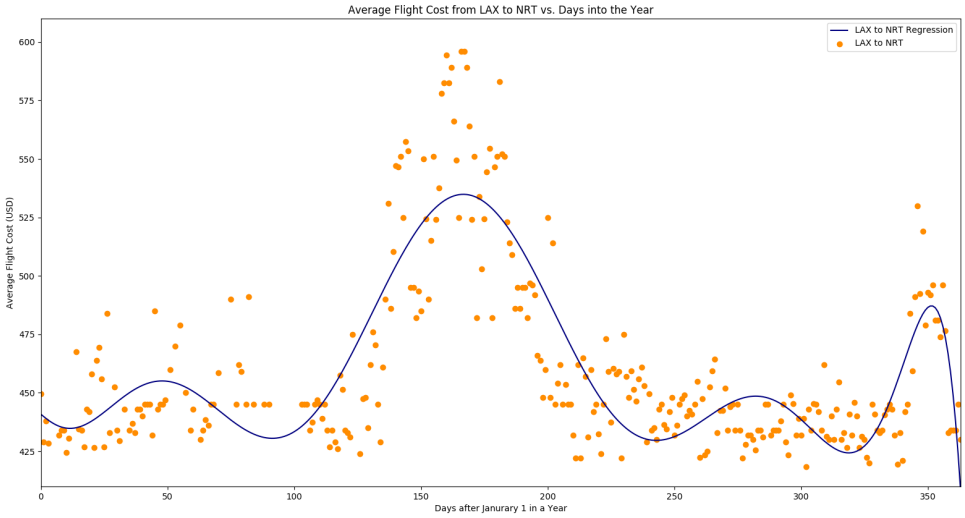


Figure 3: Graphed above are the flight cost data and the function our gradient descent polynomial regressor outputted. Note the slight underfit. Our team intentionally prevented the algorithm from fully convergence due to an extreme variance in ticket pricing.

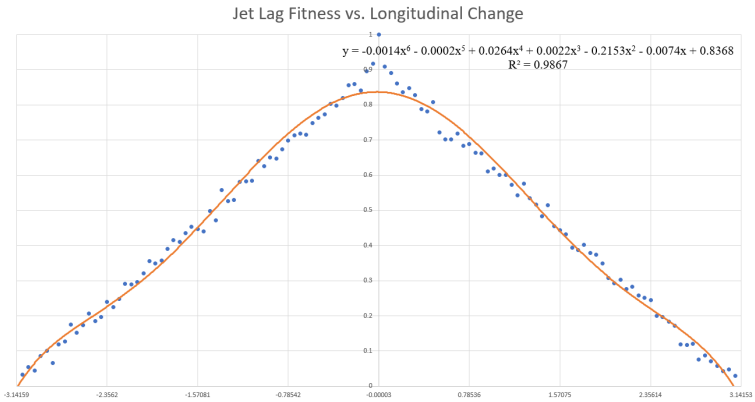


Figure 4: Above are the fitness values for certain $\Delta(P)$ that a participant travels. Note that this function is not even and that the results are not strictly decreasing on $[0, \pi]$ and not strictly increasing on $[-\pi, 0]$. In these ranges, a slight change in angle does not gaurantee a predictable change in fitness. As in real life, random chance affected amount of jet lag experienced. Unforutnately, initial conditions such as mood, nutrition, and mental stability, which may vary from person to person, cannot be modeled, and the best way to account for this is to introduce variability in the results. This algorithm is further explained in subsections 3.1 and 4.1

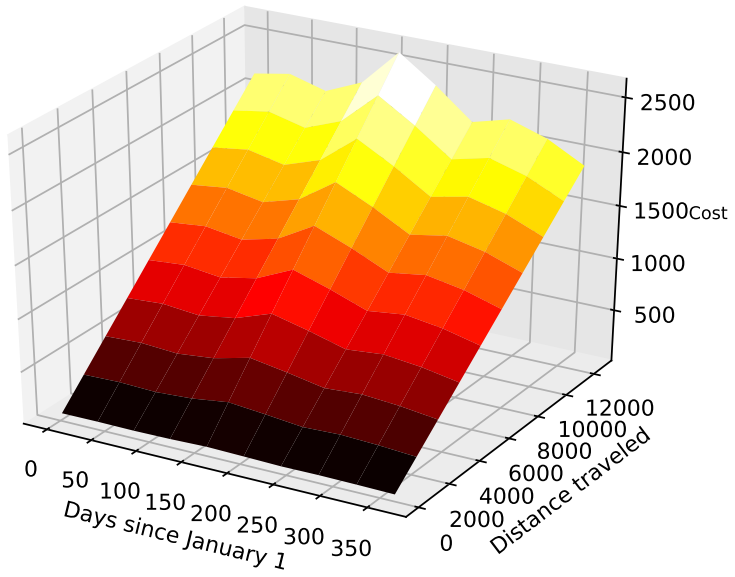


Figure 5: Above is a surface plot of the Number of Days since January 1 and Distance in Miles traveled vs. Ticket Cost

IMMC
US
7211

5 Model Discussion

As mentioned prior, our calculated results appear to be rather close to the "average" of all the locations. The model seems to have worked apparently well.

5.1 Kuramoto

Similarly to Lu et al's noise function, we ensured that eastward travel resulted in greater time needed for recovery than westward travel. Upon traveling west n time zones, one would stay up n hours longer than usual, but traveling east would require sleeping n hours earlier. Logically speaking, staying up later is easier than sleeping earlier, and as mentioned before, we assume that participants do not take treatment for jet lag, including sleeping pills. The Kuramoto model was able to account for a large number of factors, such as the change in daylight, geographic location, and variance of circadian rhythms in different persons.

Kuramoto model weaknesses

If the values of the natural frequencies of neurons in the SCN, ω_i, are not randomly generated, $\frac{\mathrm{d}\theta_i}{\mathrm{d}t}$ and $\theta_i(0)$ will be identical for all i, and our model will not work. Unfortunately, random chance is a significant factor in determining the final outcome, leading to inconsistent results. Additionally, the Kuramoto model does not account for travel time, but only recovery time. Thus, this had to be accounted for with another algorithm.

5.2 Gradient Descent for multi-dimensional regression

Although our team could have used the polynomial regressor in Microsoft Excel, we chose to create our own regression algorithm. Our algorithm enabled us to precisely control the fitting of the algorithm. In the case of flight cost data, which highly varies due to an extremity of variables in calculating ticket prices, a slight underfit is required. Thus, we terminated the gradient descent before its maximum convergence, leaving us with a polynomial regression with $R^2 = 0.68$. Another benefit of gradient descent of Excel polynomial regression is we can regress an unlimited number of degrees, while Excel allows for only up to 6.

Gradient Descent weaknesses

The largest issue with using gradient descent for regression rather than least square polynomials (as done by Microsoft Excel) is the agonizing amount of computation time our algorithm required, and this is time not readily available in real-world scenarios.

Another issue is gradient descent's tendency to converge to local minima. Since the algorithm is

$$\theta_i \coloneqq \theta_i - \alpha \frac{\partial}{\partial \theta_i}(j(\theta))$$

It will stop updating when $\frac{\partial}{\partial \theta_i}(j(\theta)) = 0$, or at a local minima of the cost function. Convergence does not necessarily guarantee that our calculated regression is near the most optimal regression found at the global minima.

5.3 Grid brute force algorithm

Using the algorithms described and an input of cities, we created a grid of (latitude, longitude) coordinates which were spaced 5°apart. From these coordinates, we then removed all the coordinates in the ocean by posting get requests to Google Maps API for images of maps and analyzing the pixels for blue color. This algorithm was incredibly efficient.

Grid brute force algorithm weaknesses

The grid algorithm calculated fitness for points spaced in increments 5°. Unfortunately, this on average corresponds to 345 miles, which is an large interval. Coordinates with more optimal fitness values may exist, but our algorithm cannot find them.

Another weakness is its inability to detect islands. Coordinates nearby islands but in water will be excluded, as the majority of nearby pixels in a map would be blue. Unfortunately, this removes major islands such as parts of Japan, the Phillipines, and Hawaii from consideration.

Although these compromises may result in a loss of optimization, our team decided that they were worth the benefit of computational ease. This model may be used to serve impatient customers, and since the Google Maps API only provides a limited number of queries to a user per day, requests and computations must be minimized.

5.4 Conclusions and comments on the overall model

Unfortunately, our model could not account for variables such as weather, availability of flights, and the accessibility of infrastructure.

Nevertheless, the model was able to make a seemingly accurate prediction. The individual parts of the model accounted for variables the other parts could not cover. For instance, the distance algorithm accounts for the fact that the Kuramoto model assumes that travel time does not affect productivity. Our model, rather than account for only the recovery time for jet lag, also included ticket price and exhaustion due to long plane rides. Thus, it seeked to optimize the productivity of the participants, achieving a more advantageous objective.

References

[1] "Airline Fare Analysis: Comparing Cost per Mile." *Rome2rio*. N.p., 02 Jan. 2013. Web. 15 Apr. 2017.

[2] Childs, Lauren and Strogatz. "Stability Diagram For The Forced Kuramoto Model". *Chaos*. AIP Publishing, 2015. Web. 16 Apr. 2017.

[3] Choy, Mary and Salbu. "Jet Lag: Current and Potential Therapies." *Pharmacy and Therapeutics*. MediMedia USA, Inc., Apr. 2011. Web. 15 Apr. 2017.

[4] Lu, Zhixin, et al. "Resynchronization of circadian oscillators and the east-west asymmetry of jet-lag." *Chaos*. AIP Publishing, 2016. Web. 15 Apr. 2017.

[5] Ng, Andrew . "Multivariate Linear Regression." *Machine Learning*. Stanford University, Web. 15 Apr. 2017.

[6] "Spherical Distance." *Wolfram MathWorld*. Wolfram Inc., Web. 15 Apr. 2017.

6 Appendix

The code for all algorithms is as follows: