# Optimizing the PGA Tour

Matt Fisher

May 5, 2024

# 1 Intro

The Professional Golfers' Association (PGA) Tour is golf's main collection of professional golf tournaments. In 2024, they will hold 34 main tournaments (not including sub-tournaments) from January to September across the United States, Mexico, Canada, and Scotland. With this large number of tournaments, the question arises of how to most effectively schedule the tour. Most players who are on the tour play every one of the tournaments and will need to travel all across the world over 9 months. Along with this, there is a camera staff, caddies, commentators, and more who need to travel to every single tournament on the schedule. In order to both reduce travel time and mitigate the carbon footprint of the PGA Tour, we seek to minimize the total distance to travel and still hit every tournament of the tour.

# 2 Complications with scheduling

Most of the PGA Tour's current schedule is based on tradition. In the current schedule, they have a stint in Florida and Georgia in March and April because this would encourage journalists covering MLB Spring Training to stop by PGA events. Also, there is a stint of events in California in the winter months, this is because in the past, when country clubs in the northern states closed for the winter, people tended to also have memberships in California to play golf in the winter. Because of this and many other traditions, the PGA Tour has not changed substantially in the past years to accommodate for the growing number of events and increased travel.

Many of these traditions can be improved on to minimize travel distance, but some of them have become rules that are set in stone that we must work around when creating our model. One of the most important rules is that of the 4 major tournaments. These are the 4 most popular events, and they are always played on the same weekends every year. These events are The Masters in Augusta, Georgia which takes place on the weekend containing the second Sunday in April, The PGA Championship in Louisville, Kentucky on the weekend before Memorial Day in May, The U.S. Open in Pinehurst, North Carolina taking place on the weekend containing the third Sunday in June, and finally The Open Championship in Troon Scotland, which takes place on the weekend containing the third Friday in July. These 4 events can not be moved around the schedule due to the rules of the PGA. A few more events that must take place on certain weeks are The Players in Sawgrass, Florida which must be in March, as well as the FedEx Playoff tournaments. The FedEx events are the last 3 events of the tour, the 2024 events, in order, are the FedEx Championship in Memphis Tennessee, The BMW Championship in Castle Rock, Colorado, and the Tour Championship in Atlanta, Georgia.

Another complication that we will need to keep in mind is that of the weather. The schedule runs from January to September, meaning there will be events scheduled in the winter months. Given that golf can not be played when there is snow on the ground, we must work around this especially since the tour has events in Canada, Minnesota, and other places where snow can be a concern in the winter months.

# 3 Details of the original model used

As the PGA Tour includes 34 unique stops along with the various restrictions mentioned above, the model used to minimize travel time has various layers and complications, which we will describe here starting with the distance matrix. We needed to retrieve the distance between all 34 country clubs that will be played. In order to retrieve this data, we will utilize the Python package haversine. The haversine package will retrieve the straight line on a map distance between any 2 given points on Earth using latitude and longitude points. We found the latitude and longitude of all clubs on the Tour using Google Maps, then used Python to find the distance in miles between any two clubs.

The code for this, along with all AMPL code used, will be given in the Appendix at the end of this paper. Putting all of the distances into a distance matrix, we get the following:

```
param distance:
    SEN  SON  AEX  FIO  PBB  PHX  GEN  MEX  COG  API  PLY  VAL  HOU  VTO  MAS  RBC  ZNO  CJC  WFC  PGA  CSC  CAN  MEM  USO  TRV  RMC  JDC  GSO  TOP  TMP  WYN  FDX  BMW  TOR:=
SEN 0    81   2607 2543 2330 2858 2488 3302 4778 4670 4652 4601 3827 3651 4572 4658 4137 3732 4618 4334 3696 4580 4443 4693 4967 4428 4059 6671 6899 3921 4660 4118 3291 4440
SON 81   0    2666 2603 2379 2917 2545 3375 4842 4733 4713 4665 3891 3715 4630 4718 4200 3792 4674 4387 3757 4627 4494 4749 5015 4475 4108 6673 6899 3965 4715 4176 3343 4498
AEX 2607 2666 0    80   480  253  129  1122 2198 2081 2052 2017 1254 1087 1964 2052 1552 1129 2014 1745 1095 2055 1875 2090 2425 1890 1511 4840 5082 1459 2060 1512 748  1832
FIO 2543 2603 80   0    483  315  109  1117 2252 2137 2111 2072 1304 1135 2029 2115 1606 1189 2082 1817 1154 2132 1949 2158 2500 1967 1588 4918 5160 1538 2129 1578 826  1897
PBB 2330 2379 480  483  0    688  373  1599 2586 2459 2414 2403 1669 1515 2293 2393 1945 1504 2321 2018 1479 2251 2117 2393 2637 2097 1730 4775 5019 1597 2354 1839 971  2164
PHX 2858 2917 253  315  688  0    379  983  1946 1828 1799 1764 1004 840  1715 1801 1299 876  1767 1507 842  1837 1644 1844 2198 1670 1292 4702 4942 1272 1816 1264 558  1582
GEN 2488 2545 129  109  373  379  0    1227 2325 2207 2176 2144 1382 1216 2084 2174 1679 1253 2131 1856 1220 2152 1980 2206 2526 1989 1611 4885 5128 1541 2175 1630 839  1952
MEX 3302 3375 1122 1117 1599 983  1227 0    1643 1584 1625 1505 879  755  1677 1703 1136 1009 1787 1694 964  2134 1873 1859 2389 1977 1680 5223 5448 1827 1863 1369 1296 1569
COG 4778 4842 2198 2252 2586 1946 2325 1643 0    140  245  183  951  1127 474  369  647  1082 574  847  1109 1132 936  579  1108 1091 1167 3962 4152 1459 635  805  1668 538
API 4670 4733 2081 2137 2459 1828 2207 1584 140  0    120  79   845  1022 350  257  533  958  462  713  987  1023 812  480  1038 969  1029 3922 4118 1321 531  666  1534 401
PLY 4652 4713 2052 2111 2414 1799 2176 1625 245  120  0    166  840  1018 231  138  526  923  341  602  957  903  694  363  928  850  927  3823 4022 1217 411  592  1472 298
VAL 4601 4665 2017 2072 2403 1764 2144 1505 183  79   166     0 775  952  375  301  466  899  497  717  926  1055 831  526  1094 989  1019 3987 4185 1311 571  633  1487 400
HOU 3827 3891 1254 1304 1669 1004 1382 879  951  845  840  775  0    178  829  881  314  245  926  816  233  1261 997  1001 1511 1109 856  4356 4578 1071 998  492  857  708
VTO 3651 3715 1087 1135 1515 840  1216 755  1127 1022 1018 952  178  0    998  1055 492  260  1089 947  219  1382 1124 1165 1656 1223 931  4467 4693 1108 1158 625  770  874
MAS 4572 4630 1964 2029 2293 1715 2084 1677 474  350  231  375  829  998  0    118  540  848  130  380  889  680  463  187  760  620  716  3668 3874 1002 212  456  1331 132
RBC 4658 4718 2052 2115 2393 1801 2174 1703 369  257  138  301  881  1055 118  0    576  927  206  497  966  767  568  225  796  722  834  3696 3898 1120 273  554  1436 231
ZNO 4137 4200 1552 1606 1945 1299 1679 1136 647  533  526  466  314  492  540  576  0    445  652  636  466  1079 811  723  1270 950  800  4160 4374 1068 731  357  1062 436
CJC 3732 3792 1129 1189 1504 876  1253 1009 1082 958  923  899  245  260  848  927  445  0    917  722  48   1142 893  995  1440 979  672  4216 4443 852  977  418  629  717
WFC 4618 4674 2014 2082 2321 1767 2131 1787 574  462  341  497  926  1089 130  206  652  917  0    335  961  562  369  77   632  520  678  3539 3746 952  82   505  1352 219
PGA 4334 4387 1745 1817 2018 1507 1856 1694 847  713  602  717  816  947  380  497  636  722  335  0    770  448  181  393  718  314  343  3543 3763 623  344  325  1047 318
CSC 3696 3757 1095 1154 1479 842  1220 964  1109 987  957  926  233  219  889  966  466  48   961  770  0    1189 941  1038 1488 1026 717  4262 4489 889  1021 464  626  758
CAN 4580 4627 2055 2132 2251 1837 2152 2134 1132 1023 903  1055 1261 1382 680  767  1079 1142 562  448  1189 0    268  555  390  169  545  3096 3317 669  498  770  1314 695
MEM 4443 4494 1875 1949 2117 1644 1980 1873 936  812  694  831  997  1124 463  568  811  893  369  181  941  268  0    396  558  158  390  3363 3582 618  336  505  1153 447
USO 4693 4749 2090 2158 2393 1844 2206 1859 579  480  363  526  1001 1165 187  225  723  995  77   393  1038 555  396  0    578  537  733  3485 3690 998  61   582  1422 293
TRV 4967 5015 2425 2500 2637 2198 2526 2389 1108 1038 928  1094 1511 1656 760  796  1270 1440 632  718  1488 390  558  578  0    541  916  2907 3114 058  550  1031 1691 838
RMC 4428 4475 1890 1967 2097 1670 1989 1977 1091 969  850  989  1109 1223 620  722  950  979  520  314  1026 169  158  537  541  0    379  3248 3471 537  476  622  1152 603
JDC 4059 4108 1511 1588 1730 1292 1611 1680 1167 1029 927  1019 856  931  716  834  800  672  678  343  717  545  390  733  916  379  0    3551 3781 292  680  445  774  629
GSO 6671 6673 4840 4918 4775 4702 4885 5223 3962 3922 3823 3987 4356 4467 3668 3696 4160 4216 3539 3543 4262 3096 3363 3485 2907 3248 3551 0    244  3461 3457 3866 4149 3738
TOP 6899 6899 5082 5160 5019 4942 5128 5448 4152 4118 4022 4185 4578 4693 3874 3898 4374 4443 3746 3763 4489 3317 3582 3690 3114 3471 3781 244  0    3697 3664 4086 4388 3948
TMP 3921 3965 1459 1538 1597 1272 1541 1827 1459 1321 1217 1311 1071 1108 1002 1120 1068 852  952  623  889  669  618  998  1058 537  292  3461 3697 0    941  722  716  921
WYN 4660 4715 2060 2129 2354 1816 2175 1863 635  531  411  571  998  1158 212  273  731  977  82   344  1021 498  336  61   550  476  680  3457 3664 941  0    560  1383 295
FDX 4118 4176 1512 1578 1839 1264 1630 1369 805  666  592  633  492  625  456  554  357  418  505  325  464  770  505  582  1031 622  445  3866 4086 722  560  0    884  325
BMW 3291 3343  748 826  971  558  839  1296 1668 1534 1472 1487 857  770  1331 1436 1062 629  1352 1047 626  1314 1153 1422 1691 1152 774  4149 4388 716  1383 884  0    1205
TOR 4440 4498 1832 1897 2164 1582 1952 1569 538  401  298  400  708  874  132  231  436  717  219  318  758  695  447  293  838  603  629  3738 3948 921  295  325  1205 0;
```

Where all of the headings on the rows and columns are condensed names of the Tournament themselves. For example, The Masters is MAS, the Phoenix Open is PHX, and so on.

We will now discuss the model we produced in AMPL, a mathematical programming language, and the theory behind what we did. Starting with the variables we defined for this model, we have 3 variables defined as the following.

$$X_{i,j} = \begin{cases} 1, & \text{if tournament j follows tournament i} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$B_i = \begin{cases} 1, & \text{if location of tournament i has more than 15 inches of snow a year} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$T_i$ = number in the Tour cycle that tournament i is

Our objective function will be the following:

$$\sum_{i,j,i\neq j}^{n} D_{i,j} \cdot X_{i,j} \tag{3}$$

Where $D_{i,j}$ is the distance between tournament i and j, and n is the number of tournaments.

Now, we will go over the models' constraints. Something we need to be mindful of is that the tour makes $n-1$ trips between tournaments, where n is total number of tournaments (34 for the full tour). So we will implement this constraint into our model by doing the following.

$$\sum_{i,j,i\neq j}^{n} X_{i,j} = n - 1 \tag{4}$$

Along with this, we can observe that when we add up all the orders of tournaments in the tour (1 + 2 + ... + 34) we get n*(n+1)/2, this makes sure that our tournament order goes from 1 to 34 and does not take on any other values, this constraint can be added as:

$$\sum_{i}^{n} T_i = \frac{n*(n+1)}{2} \tag{5}$$

Next, we can observe 2 things. For every tournament attended, we will want at most one tournament to follow or precede that tournament. This constraint can prevent our model from assigning 2 or more tournaments arcing from one point, keeping a single straight line path between all 34 tournaments. We can do this by introducing the following:

$$\sum_{i}^{n} X_{i,j} \leq 1 \tag{6}$$

$$\sum_{j}^{n} X_{i,j} \leq 1 \tag{7}$$

These constraints guarantee that when summing over i, at most 1 tournament will precede any given tournament while summing over j guarantees at most 1 will follow.

Now that we have established that we will have one arc to and from every club, we will want the model to ensure that there are no cycles in the path. This will help confirm that every club is visited exactly once in our model, and we can implement it by doing the following.

$$T_i - T_j + (n+1) * X_{i,j} \leq n \ \ \forall \text{ i, j, i} \neq \text{j} \tag{8}$$

These constraints guarantee that there can be no cycles in this given path.

Another factor we will want to consider is the weather of the tournaments. With there being tournaments in the northern United States, Canada, and Scotland, where snow is a very large risk if played in the earlier months of the year, we need to implement a constraint to attempt to avoid tournaments being played in the snow. We can do this in the following way:

$$T_i \geq B_i * k \tag{9}$$

Where k is the 14th event in the Tour. This constraint ensures that any tournament that is at a location where the average snowfall is at or above 15 inches will be pushed back in the order to at least the 14th spot in the tour. This will push any of these tournaments back until at least mid-April, that way it is incredibly unlikely that snow will be a hindrance on the tour.

Finally, we need to carry out constraints so that all of the majors and FEDEX Playoff tournaments take place on their specific weeks. This is a fairly simple constraint that can be done in the following way:

$$T_{\text{Specific Tournament}} = i \tag{10}$$

Where i is the specific week that the tournament must fall on. With this constraint, we now have our finalized initial model.

# 4 Revisions of original model

Whwn testing the original model presented above, using n = 34, we used the CPLEX solver on a NEOS server. A problem arose in testing, as The NEOS server only allows for a given problem to use up to 3 GB of RAM to solve a problem, and our model used more. Because of this, we were given an error statement and were unable to get a result. We then had to rethink the model, in order to minimize the amount of memory used in the server so we can get our optimal result.

The first thing that came to mind was minimizing the amount of information that has to go through the optimization process, starting at the distance matrix, as that almost definitely is a factor in the high memory usage. We can start by attempting to disregard useless data. We can do this by getting rid of distance data that is over a certain threshold, say 900 miles. When minimizing the total distance traveled, it is very unlikely that there will be any travel paths over 900 miles unless there are specific circumstances. These exceptions being that we are traveling to or from one of the international tournaments in Scotland or Mexico, or if they are going to or from one of the 2 Hawaii tournaments. A way to add this constraint is by adding a new parameter to our model. This parameter will be the following.

$$B_{i,j} = \begin{cases} 1, & i \neq j \wedge (\text{i or j are one of the previously stated tournaments} \vee D_{i,j} \leq 900) \\ 0, & \text{otherwise} \end{cases}$$

With this new parameter, we will substitute $B_{i,j} = 1$ into all variables and constraints in which the condition $i \neq j$ was needed. We will substitute it into the objective function as well. This will lessen the amount of variables that the solver needs to intake, lowering the amount of RAM used.

Another adjustment made was removing a chunk of the tournaments. As we are using CPLEX, which uses the simplex algorithm, it can take very long, almost as long as brute force. The NEOS Server times out at 8 hours, which we have exceeded multiple times in testing. We removed a chunk of the beginning of the Tour, as this is mainly optimized for distance anyway. The first bunch of tournaments are in Hawaii, California, Arizona, and Mexico all places that are close together for the most part. We had to make some changes to the code itself, such as dropping the events from the distance matrix, adjusting our n, as well removing Hawaii and Mexico from the new parameter stated in the last paragraph. After testing various different models, we found that the most tournaments we can include is 24 (we will discuss this more in section 6), meaning we will drop 10.

The events dropped are

- The Sentry
- Sony Open
- The American Express
- Farmers Insurance Open
- AT&T Pebble Beach Pro-Am
- WM Pheonix Open
- The Genesis Invitational
- Mexico Open
- BMW Championship
- Tour Championship

Excluding the BMW Championship and the Tour Championship, these are all the first 8 tournaments. We chose to exclude these as the beginning of the tour mostly includes of tournaments that are fairly close together (It is mainly all on the West Coast), so optimizing the final portion will be more beneficial.

The last 2 events of the tour, BMW Championship and the Tour Championship, are excluded from the model, as we know these both follow the FedEx Championship, as the last 3 events are set in stone as previously mentioned. Now with this reduced mode, we can run it in the NEOS CPLEX server without error.

# 5 Results

When running our program using a CPLEX solver, we get the following order of tournaments, with our results on the left and original schedule on the right:

1. Zurich New Orleans Classic (New Orleans, LA) / Cognizant Classic (Palm Beach Gardens, FL)
2. Valspar Championship (Palm Harbor, FL) / Arnold Palmer Invitational (Orlando, FL)
3. **The Players Championship (Sawgrass, FL) / The Players Championship (Sawgrass, FL)**
4. Arnold Palmer Invitational (Orlando, FL) / Valspar Championship (Palm Harbor, FL)
5. Cognizant Classic (Palm Beach Gardens, FL) / Houston Open (Houston, TX)
6. RBC Heritage (Hilton Head, SC) / Valero Texas Open (San Antonio, TX)
7. **The Masters (Augusta, GA) / The Masters (Augusta, GA)**
8. Houston Open (Houston, TX) / RBC Heritage (Hilton Head, SC)
9. Valero Texas Open (San Antonio, TX) / Zurich New Orleans Classic (New Orleans, LA)
10. Charles Schwab Challenge (Fort Worth, TX) / The CJ Cup (McKinney, TX)
11. The CJ Cup (McKinney, TX) / Wells Fargo Championship (Charlotte, NC)
12. **PGA Championship (Louisville, KY) / PGA Championship (Louisville, KY)**
13. 3M Open (Blaine, MN) / Charles Schwab Challenge (Fort Worth, TX)
14. John Deere Classic (Silvis, IL) / RBC Canadian Open (Hamilton, Canada)
15. Wells Fargo Championship (Charlotte, NC) / The Memorial Tournament (Dublin, OH)
16. **US Open (Pinehurst, NC) / US Open (Pinehurst, NC)**
17. Wyndham Championship (Greensboro, NC) / Travelers Championship (Cromwell, CT)
18. The Memorial Tournament (Dublin, OH) / Rocket Mortgage Classic (Detroit, MI)
19. Rocket Mortgage Classic (Detroit, MI) / John Deere Classic (Silvis, IL)
20. RBC Canadian Open (Hamilton, Canada) / Genesis Scottish Open (North Berwick, Scotland)
21. **The Open (Troon, Scotland) / The Open (Troon, Scotland)**
22. Genesis Scottish Open (North Berwick, Scotland) / 3M Open (Blaine, MN)
23. Travelers Championship (Cromwell, CT) / Wyndham Championship (Greensboro, NC)
24. **FedEx Championship (Memphis, TN) / FedEx Championship (Memphis, TN)**
25. **BMW Championship (Castle Rock, CO) / BMW Championship (Castle Rock, CO)**
26. **Tour Championship (Atlanta, GA) / Tour Championship (Atlanta, GA)**

Ones in bold are major Tournaments that place in the schedule are set, and cannot change.

This is a fairly different result than what the PGA Tour currently uses. But even just by observation, we can see that distance is fairly minimized, with all of the events in Texas and Florida all bunched together, there are stints in the northern US, and so on. Something notable is that, much like the original schedule, places in the northern US are played later in the schedule, this most likely isn't due to our weather constraints. I likely believe that this is due to where the major tournaments are. The Players and The Masters are in Florida and Georgia respectively, and are both tournaments that are on a set week. In order to minimize travel distance, it is likely that the tournaments in the north, where snow is most prevalent, will not be played until later in the year.

# 6 Time Analysis

In testing, I wanted to see how many tournaments we could add to this model before the remote server could not take it anymore. So we started with the last 18 tournaments, that way we include all of the 4 major tournaments, as well as we include the last portion, as that is the section of the tour with the most distance traveled prior to optimization. The model with only 18 tournaments ran in less than one minute, so we slowly added in tournaments one at a time. This way when n = 19 it is the last 19 tournaments of the original PGA schedule, n = 20 is the last 20 tournaments of the original PGA schedule, and so on. We did this until it wouldn't allow us anymore and observed the run time, the following is the results of running this model with increasing n.

- n = 18: Less than a minute
- n = 19: 3 minutes
- n = 20: 8 minutes
- n = 21: 11 minutes
- n = 22: 17 minutes
- n = 23: 90 minutes
- n = 24: 6 hours 50 minutes
- n = 25: Exceeded allotted RAM

This does many things, not only did it help us settle on having a 24 Tournament model, but also illustrates the rate at which adding more tournaments increases the run time. Another observation can be made by looking at the number of simplex iterations and branch-and-bound nodes that are needed to be done to solve the problem. These are retrieved from the CPLEX output for the AMPL problems.

- n = 18: 61,096 simplex iterations, 8,879 branch-and-bound nodes
- n = 24: 1,023,371,273 simplex iterations, 75,182,228 branch-and-bound nodes

We can see just how fast this problem grows by increasing the n by only 6. When n = 18, it only does .00597% the amount of simplex iterations compared to n = 24. This also demonstrates why doing the full model of n = 34 would be impossible without heavily adjusting our model.

# 7 Future work

Many refinements can be added to this model in order to account for more scenarios that the Tour takes into consideration.

- Look into ways to condense variables so that we can run this code for the whole PGA Tour.
- The Super Bowl heavily affects all other TV ratings for the whole weekend that it airs. If we were able to access the TV ratings of the PGA Tour, which I have been unable to find, we could prioritize less popular tournaments to air the week of the Super Bowl. This way the PGA will minimize their revenue loss throughout the year.
- Investigate the possibility of switching haversine distance with flight/driving distance to more accurately measure the distance the players/crew will have to travel.

# 8 References

https://www.espn.com/golf/schedule
Used to find PGA Tour locations and events
https://www.pgatour.com/
Used to gather info on Tour traditions and customs
https://www.google.com/maps
Used for latitude and longitude of the event locations
https://s2.smu.edu/ olinick/cse3360/lectures/brute_force.htm
Used to gather info about CPLEX algorithms

# 9 Appendix

**Python Code Used For Distance Matrix**

```
!pip install haversine
from haversine import haversine, Unit
from haversine import haversine_vector, Unit
import numpy as np
SEN = (20.9946001248632, -156.653473270624541)
SON = (21.3069447444130, -157.858337667429032)
AEX = (33.6654015556922, -116.30798311112099)
FIO = (32.8419910471217, -117.27301882351397)
PBB = (38.1290731941269, -122.88434386277191)
PHX = (33.64070144886564, -111.90910886569488)
GEN = (34.04997103007352, -118.50219732895634)
MEX = (20.681043251427173, -105.28686333663033)
COG = (26.830633649876166, -80.13832785015691)
API = (28.458628889079844, -81.50963138724956)
PLY = (30.19889198964165, -81.3942001176103)
VAL = (28.111027597022275, -82.75374281522284)
HOU = (29.773931383507033, -95.42836188444284)
VTO = (29.666126567067618, -98.39446091355777)
MAS = (33.50238699892952, -82.02178538638701)
RBC = (32.13572691226164, -80.80936554405868)
ZNO = (29.90254329690623, -90.18820495776565)
CJC = (33.14092173707245, -96.71862829432696)
WFC = (35.11649093625573, -80.8414662538495)
PGA = (38.24273076940281, -85.47084527898363)
CSC = (32.716171589705326, -97.37133580781949)
CAN = (43.21957285400244, -79.97530827940336)
MEM = (40.1401518527829, -83.14097123665532)
USO = (35.19414802532097, -79.47333643635238)
TRV = (41.6320876375511, -72.63809930596864)
RMC = (42.42661261589339, -83.1248997311028)
JDC = (41.47744798225558, -90.39136269987951)
GSO = (57.83980492222196, -9.511702403890528)
TOP = (55.53282059279759, -4.648627367341988)
TMP = (45.17726081811669, -93.21209672316282)
WYN = (36.01449583864608, -79.88558216178946)
FDX = (35.05745887091684, -89.7781591538495)
BMW = (39.44078685686753, -104.89738163295364)
TOR = (33.74351942073119, -84.30191918638701)
place_vec = [SEN, SON, AEX, FIO, PBB, PHX, GEN, MEX, COG, API, PLY, VAL, HOU, VTO, MAS, RBC, ZNO, CJC, WFC, PGA, CSC, CAN, MEM, USO, TRV, RMC, JDC, GSO, TOP, TMP, WYN, FDX, BMW, TOR]
v = haversine_vector(place_vec, place_vec, Unit.MILES, comb=True)
v = np.around(v)
print((v))
```

**AMPL Code Used**

```
set Tournament;

set Weather;

param downfall{Weather, Tournament};

param distance{i in Tournament, j in Tournament: i!=j};


#if an arc is less then 900 miles dont include (unless one is in Scotland)
param allowed_travel{i in Tournament, j in Tournament}:=
 if (i!=j and (i == 'GSO' or i =='TOP'or j =='GSO' or j =='TOP'or  distance[i,j]<=900))
  then 1
  else 0;

###VARIABLE

var next_tournament{i in Tournament, j in Tournament: allowed_travel[i,j]=1} binary;

var included{Tournament} binary;

var Tournament_Order{Tournament} >= 0, <= 24;


subject to all_must_occur:     #Makes sure all Tournaments are included
sum {i in Tournament} included[i]  = 24;

# sum of all tournament orders should add up to n(n+1)/2
subject to sum_order:
sum {i in Tournament} Tournament_Order[i] = 300;

#Exactly n-1 arcs in total
subject to tournaments_following:
sum{i in Tournament, j in Tournament:
 allowed_travel[i,j]=1} next_tournament[i,j] = 23;

#at most one tournament follows
subject to at_most_one_tournament_follows {i in Tournament}:
sum {j in Tournament: allowed_travel[i,j]=1} next_tournament[i,j] <= 1;

#at most one tournament precedes
subject to at_most_one_tournament_precedes {j in Tournament}:
sum {i in Tournament: allowed_travel[i,j]=1} next_tournament[i,j] <= 1;

#Keep it a path, let no cycles form
subject to prevent_cycle {i in Tournament, j in Tournament:
 allowed_travel[i,j]=1}:  Tournament_Order[i] - Tournament_Order[j] +
 25*next_tournament[i,j] <= 24;


##if there is snow, play it after mid-April
subject to snowfall {i in Tournament}:
Tournament_Order[i] >= downfall["Snow", i]*7;
```

```
### If a Tournament is always played on a certain week, set that here.

subject to Players {i in Tournament}:
Tournament_Order ["PLY"] = 3;

subject to Masters {i in Tournament}:
Tournament_Order ["MAS"] = 7;

subject to PGA_Champ {i in Tournament}:
Tournament_Order ["PGA"] = 12;

subject to US_Open {i in Tournament}:
Tournament_Order ["USO"] = 16;

subject to The_Open {i in Tournament}:
Tournament_Order ["TOP"] = 21;

subject to FedEx {i in Tournament}:
Tournament_Order ["FDX"] = 24;

### OBJ Function
minimize total_distance: sum {i in Tournament, j in Tournament:
allowed_travel[i,j]=1}  distance[i,j]*next_tournament[i,j];
```

```
data;

set Tournament:= COG API PLY VAL HOU VTO MAS RBC ZNO CJC WFC PGA CSC CAN MEM USO TRV RMC JDC GSO TOP TMP WYN FDX;


set Weather:= Snow;

param distance:
    COG  API  PLY  VAL  HOU  VTO  MAS  RBC  ZNO  CJC  WFC  PGA  CSC  CAN  MEM  USO  TRV  RMC  JDC  GSO  TOP  TMP  WYN  FDX:=
COG 0    140  245  183  951  1127 474  369  647  1082 574  847  1109 1132 936  579  1108 1091 1167 3962 4152 1459 635  805
API 140  0    120  79   845  1022 350  257  533  958  462  713  987  1023 812  480  1038 969  1029 3922 4118 1321 531  666
PLY 245  120  0    166  840  1018 231  138  526  923  341  602  957  903  694  363  928  850  927  3823 4022 1217 411  592
VAL 183  79   166  0    775  952  375  301  466  899  497  717  926  1055 831  526  1094 989  1019 3987 4185 1311 571  633
HOU 951  845  840  775  0    178  829  881  314  245  926  816  233  1261 997  1001 1511 1109 856  4356 4578 1071 998  492
VTO 1127 1022 1018 952  178  0    998  1055 492  260  1089 947  219  1382 1124 1165 1656 1223 931  4467 4693 1108 1158 625
MAS 474  350  231  375  829  998  0    118  540  848  130  380  889  680  463  187  760  620  716  3668 3874 1002 212  456
RBC 369  257  138  301  881  1055 118  0    576  927  206  497  966  767  568  225  796  722  834  3696 3898 1120 273  554
ZNO 647  533  526  466  314  492  540  576  0    445  652  636  466  1079 811  723  1270 950  800  4160 4374 1068 731  357
CJC 1082 958  923  899  245  260  848  927  445  0    917  722  48   1142 893  995  1440 979  672  4216 4443 852  977  418
WFC 574  462  341  497  926  1089 130  206  652  917  0    335  961  562  369  77   632  520  678  3539 3746 952  82   505
PGA 847  713  602  717  816  947  380  497  636  722  335  0    770  448  181  393  718  314  343  3543 3763 623  344  325
CSC 1109 987  957  926  233  219  889  966  466  48   961  770  0    1189 941  1038 1488 1026 717  4262 4489 889  1021 464
CAN 1132 1023 903  1055 1261 1382 680  767  1079 1142 562  448  1189 0    268  555  390  169  545  3096 3317 669  498  770
MEM 936  812  694  831  997  1124 463  568  811  893  369  181  941  268  0    396  558  158  390  3363 3582 618  336  505
USO 579  480  363  526  1001 1165 187  225  723  995  77   393  1038 555  396  0    578  537  733  3485 3690 998  61   582
TRV 1108 1038 928  1094 1511 1656 760  796  1270 1440 632  718  1488 390  558  578  0    541  916  2907 3114 058  550  103
RMC 1091 969  850  989  1109 1223 620  722  950  979  520  314  1026 169  158  537  541  0    379  3248 3471 537  476  622
JDC 1167 1029 927  1019 856  931  716  834  800  672  678  343  717  545  390  733  916  379  0    3551 3781 292  680  445
GSO 3962 3922 3823 3987 4356 4467 3668 3696 4160 4216 3539 3543 4262 3096 3363 3485 2907 3248 3551 0    244  3461 3457 3866
TOP 4152 4118 4022 4185 4578 4693 3874 3898 4374 4443 3746 3763 4489 3317 3582 3690 3114 3471 3781 244  0    3697 3664 4086
TMP 1459 1321 1217 1311 1071 1108 1002 1120 1068 852  952  623  889  669  618  998  1058 537  292  3461 3697 0    941  722
WYN 635  531  411  571  998  1158 212  273  731  977  82   344  1021 498  336  61   550  476  680  3457 3664 941  0    560
FDX 805  666  592  633  492  625  456  554  357  418  505  325  464  770  505  582  1031 622  445  3866 4086 722  560  0;

param downfall:
        COG API PLY VAL HOU VTO MAS RBC ZNO CJC WFC PGA CSC CAN MEM USO TRV RMC JDC GSO TOP TMP WYN FDX:=
Snow    0   0   0   0   0   0   0   0   0   0   0   1   0   1   1   0   1   1   1   1   1   1   0   0;
```

AMPL Output:

```
Error at _cmdno 4 executing "solve" command
(file amplin, line 135, offset 6027):
error processing param distance:
        24 invalid subscripts discarded:
        distance['COG','COG']
        distance['API','API']
        distance['PL
Presolve eliminates 229 constraints and 71 variables.
Adjusted problem:
373 variables:
        355 binary variables
        18 linear variables
386 constraints, all linear; 1938 nonzeros
        2 equality constraints
        384 inequality constraints
1 linear objective; 355 nonzeros.

CPLEX 22.1.1.0: threads=4
CPLEX 22.1.1.0: optimal integer solution within mipgap or absmipgap; objective 12326
1023371273 MIP simplex iterations
75182228 branch-and-bound nodes
absmipgap = 1.23256, relmipgap = 9.99966e-05
Tournament_Order [*] :=
API  4   COG  5   GSO 22   MAS  7   PLY  3   TMP 13   USO 16   WFC 15
CAN 20   CSC 10   HOU  8   MEM 18   RBC  6   TOP 21   VAL  2   WYN 17
CJC 11   FDX 24   JDC 14   PGA 12   RMC 19   TRV 23   VTO  9   ZNO  1
;
```