BarCrawl

December 8, 2023

[1]:
```
%load_ext gams.magic
m = gams.exchange_container
```

[2]:
```
import pandas as pd
import random
import numpy as np
import math
```

1 Madison Holiday Bar Crawl Optimization

1.1 By: Elizabeth Travis and Katie Dums

1.2 Introduction

1.2.1 Problem Statement:

Our project, the Madison Holiday Bar Crawl, airms to determine the optimal route for a bar crawl that visits 10 bars that minimizes the total distance participants have to walk, while maximizing the amount of alcohol participants consume.Bar Crawls happen all over the world with all different themes and rules. Madison alone is host to many bar crawls that take advantage of the large student population and the close proximity of a large amount of bars. For example there was just a Holiday Bar Crawl in downtown Madison for seniors at University of Wisconsin Madison. Students paid $20 for a tshirt that got them into 9 different bars where they would receive special deals. Students could visit any number of these bars in any order. Inspired by this bar crawl, our project works to help organizers create a bar crawl.

1.2.2 Data:

This project created a model that picks a specified number of bars, in this model 10, from the set of 22 sites(bars): 20 bars, 1 starting location, and 1 ending location. To determine which bars should be chosen, parameters for the distance between each pair of sites(distance) and the average amount a participant's Blood Alcohol content(BAC)(bac_per_bar) will increase after a drink at each bar.

[3]:
```
barcrawl = pd.read_csv("BarCrawl.csv")
df = pd.DataFrame(barcrawl)
```

```
df = df.rename(columns = {"Unnamed: 0":'Bar'})
sites = list(df["Bar"])
df.index = sites
df = df.drop(columns = "Bar")
coords = ['Latitude', 'Longitude']
display(df)
```

```
                     Latitude  Longitude
Sconnie Bar          43.067752 -89.410211
Jordans Big ten      43.068028 -89.408223
Lucky's Brew Pub     43.067659 -89.408160
buckinghams          43.067863 -89.399369
Echo Tap             43.068286 -89.391403
Pauls Club           43.074901 -89.388327
Hawks Bar and Grill  43.074978 -89.392693
Genna's Lounge       43.072789 -89.384348
Tipsy Cow            43.074752 -89.381565
Kollege Klub         43.075689 -89.397134
Whiskey Jack's       43.074989 -89.394780
City Bar             43.075039 -89.396167
State Street Brats   43.074793 -89.395928
Nitty Gritty         43.071921 -89.395772
Mondays              43.074668 -89.394631
Vintage Spirits      43.073005 -89.395448
Red Rock Saloon      43.075172 -89.391667
Wandos               43.073447 -89.396027
Chasers 2.0          43.074092 -89.393214
Capitol Building     43.074905 -89.384221
Memorial Union       43.076256 -89.400009
The Double U         43.073396 -89.396780
```

This distance parameter was created using google maps latitude and longitude coordinates for each site and the haversine distance formula to create a table of distances between each site in the set of bar sites.

[4]:
```
def haversine_dist(a,b):
    R = 6371e3
    lat1 = df['Latitude'].loc[a]
    lon1 = df['Longitude'].loc[a]
    lat2 = df['Latitude'].loc[b]
    lon2 = df['Longitude'].loc[b]
# metres
    k = lat1 * math.pi/180
#  ,   in radians
    k2 = lat2 * math.pi/180
    d = (lat2-lat1) * math.pi/180
    d2 = (lon2-lon1) * math.pi/180
```

```
    a = math.sin(d/2) * math.sin(d/2) + math.cos(k) * math.cos(k2) * math.sin(d2/
 ↪2) * math.sin(d2/2)
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1-a))
# in metres
    d = R * c
#to feet
    d = d*3.28084
#to miles
    d = d*0.000189394
    return d
```

[5]:
```
distances = pd.DataFrame(index = sites, columns = sites)
for i in sites:
    for j in sites:
        distances[j].loc[i] = haversine_dist(i,j)
```

The bac_per_bar parameter was determined using a statistic found that says 1 drink will produce a BAC of between 0.02-0.04 for each participant.

[6]:
```
b_alcohol = np.random.uniform(.02,.04,len(sites))
b_alcohol[sites.index("Memorial Union")] = 0
b_alcohol[sites.index("Capitol Building")] = 0
```

[7]:
```
bars = m.addSet('bars',description='sites',records=sites)
distance = m.addParameter('distance',[bars,bars],records=gams.
 ↪from2dim(distances))
bac_per_bar = m.addParameter('bac_per_bar',[bars],records=b_alcohol)
```

Another thing included in the model is the max_bac scalar that indicates the max BAC a participant wants to reach at any point, or in other words the max amount of alcohol they want to consume. Lastly the model accessed a value for which a participants BAC level will decrease per mile they walk. This was found using the amount people's BAC level decrease per hour, .015,times the rate in miles per hour that an average person walks, 3 mi/hr.

[8]:
```
max_bac = m.addParameter('max_bac',records=0.08)
bac_decrease = m.addParameter('bac_decrease',records=((.015)/3))
%gams display bars, distance, bac_per_bar, max_bac, bac_decrease
%gams_lst -e
```

```
E x e c u t i o n


----     17 SET bars  sites

Sconnie Bar         ,    Jordans Big ten     ,    Lucky's Brew Pub    ,
buckinghams         ,    Echo Tap            ,    Pauls Club          ,    Hawks
Bar and Grill,    Genna's Lounge      ,    Tipsy Cow           ,    Kollege Klub
,    Whiskey Jack's      ,    City Bar            ,    State Street Brats ,
```

Nitty Gritty , Mondays , Vintage Spirits , Red Rock Saloon , Wandos , Chasers 2.0 , Capitol Building , Memorial Union , The Double U

---- 17 PARAMETER distance

	Sconnie B~	Jordans B~	Lucky's B~	buckingha~	Echo Tap	Pauls Club	Hawks Bar~	Genna's L~	Tipsy Cow	Kollege K~	Whiskey J~	City Bar	State Str~	Nitty Gri~	Mondays	Vintage S~	Red Rock ~	Wandos	Chasers 2~	Capitol B~	Memorial ~	The Doubl~
Sconnie Bar		0.102	0.104	0.547	0.950	1.210	1.015	1.351	1.525	0.858	0.926	0.869	0.870	0.784	0.920	0.829	1.067	0.817	0.963	1.402	0.781	0.782
Jordans Big ten	0.102		0.026	0.447	0.849	1.111	0.919	1.249	1.423	0.770	0.832	0.778	0.777	0.684	0.825	0.731	0.971	0.720	0.866	1.301	0.704	0.686
Lucky's Brew Pub	0.104	0.026		0.444	0.847	1.119	0.930	1.253	1.429	0.786	0.844	0.791	0.790	0.691	0.837	0.740	0.981	0.731	0.876	1.308	0.723	0.698
buckinghams	0.547	0.447	0.444		0.403	0.740	0.596	0.831	1.017	0.552	0.544	0.521	0.509	0.334	0.528	0.407	0.637	0.421	0.531	0.906	0.581	0.404
Echo Tap	0.950	0.849	0.847	0.403		0.483	0.467	0.473	0.668	0.588	0.494	0.525	0.504	0.334	0.470	0.385	0.476	0.426	0.411	0.584	0.701	0.445
Pauls Club	1.210	1.111	1.119	0.740	0.483		0.220	0.248	0.341	0.448	0.326	0.396	0.384	0.428	0.319	0.382	0.170	0.401	0.253	0.207	0.597	0.439
Hawks Bar and Grill	1.015	0.919	0.930	0.596	0.467	0.220		0.447	0.562	0.229	0.105	0.175	0.164	0.262	0.100	0.195	0.053	0.199	0.067	0.428	0.380	0.233
Genna's Lounge	1.351	1.249	1.253	0.831	0.473	0.248	0.447		0.195	0.676	0.548	0.616	0.601	0.580	0.535	0.560	0.404	0.591	0.456	0.146	0.826	0.629
Tipsy Cow	1.525	1.423	1.429	1.017	0.668	0.341	0.562	0.195		0.788	0.667	0.737	0.725	0.743	0.659	0.711	0.511	0.735	0.590	0.134	0.937	0.774

```
Kollege Klub              0.858     0.770     0.786     0.552     0.588
0.448     0.229     0.676     0.788               0.128     0.066
0.087     0.269     0.145     0.204     0.278     0.165     0.227
0.654     0.150     0.159
Whiskey Jack's            0.926     0.832     0.844     0.544     0.494
0.326     0.105     0.548     0.667     0.128               0.070
0.059     0.218     0.023     0.141     0.158     0.124     0.100
0.533     0.278     0.149
City Bar                  0.869     0.778     0.791     0.521     0.525
0.396     0.175     0.616     0.737     0.066     0.070
0.021     0.216     0.082     0.145     0.227     0.110     0.163
0.603     0.211     0.118
State Street Brats        0.870     0.777     0.790     0.509     0.504
0.384     0.164     0.601     0.725     0.087     0.059     0.021
0.199     0.066     0.126     0.217     0.093     0.145     0.591
0.229     0.106
Nitty Gritty              0.784     0.684     0.691     0.334     0.334
0.428     0.262     0.580     0.743     0.269     0.218     0.216
0.199               0.198     0.077     0.306     0.106     0.198
0.618     0.368     0.114
Mondays                   0.920     0.825     0.837     0.528     0.470
0.319     0.100     0.535     0.659     0.145     0.023     0.082
0.066     0.198               0.122     0.154     0.110     0.082
0.526     0.293     0.140
Vintage Spirits           0.829     0.731     0.740     0.407     0.385
0.382     0.195     0.560     0.711     0.204     0.141     0.145
0.126     0.077     0.122               0.243     0.042     0.135
0.582     0.322     0.072
Red Rock Saloon           1.067     0.971     0.981     0.637     0.476
0.170     0.053     0.404     0.511     0.278     0.158     0.227
0.217     0.306     0.154     0.243               0.250     0.108
0.376     0.428     0.286
Wandos                    0.817     0.720     0.731     0.421     0.426
0.401     0.199     0.591     0.735     0.165     0.124     0.110
0.093     0.106     0.110     0.042     0.250               0.149
0.604     0.279     0.038
Chasers 2.0               0.963     0.866     0.876     0.531     0.411
0.253     0.067     0.456     0.590     0.227     0.100     0.163
0.145     0.198     0.082     0.135     0.108     0.149
0.457     0.374     0.186
Capitol Building          1.402     1.301     1.308     0.906     0.584
0.207     0.428     0.146     0.134     0.654     0.533     0.603
0.591     0.618     0.526     0.582     0.376     0.604     0.457
0.802     0.642
Memorial Union            0.781     0.704     0.723     0.581     0.701
0.597     0.380     0.826     0.937     0.150     0.278     0.211
0.229     0.368     0.293     0.322     0.428     0.279     0.374
0.802               0.256
```

```
The Double U              0.782       0.686       0.698       0.404       0.445
0.439       0.233       0.629       0.774       0.159       0.149       0.118
0.106       0.114       0.140       0.072       0.286       0.038       0.186
0.642       0.256


----     17 PARAMETER bac_per_bar

Sconnie Bar         0.034,    Jordans Big ten     0.027,    Lucky's Brew Pub
0.032,    buckinghams         0.027,    Echo Tap            0.036,    Pauls Club
0.026,    Hawks Bar and Grill 0.026,    Genna's Lounge      0.032,    Tipsy Cow
0.035,    Kollege Klub        0.037,    Whiskey Jack's      0.021,    City Bar
0.035,    State Street Brats  0.027,    Nitty Gritty        0.033,    Mondays
0.025,    Vintage Spirits     0.036,    Red Rock Saloon     0.034,    Wandos
0.033,    Chasers 2.0         0.030,    The Double U        0.036


----     17 PARAMETER max_bac              =        0.080
           PARAMETER bac_decrease          =        0.005
```

1.3 Optimization Problem Approach

1.3.1 Model Type: Simple TSP Model

This model is based on a traveling salesman problem(TSP) with a tradeoff. It is more complicated than a regular TSP model because in a TSP problem you visit every location in the set. The project models a simple TSP in the first model 'visitallbars'. This model determines the shortest path to visit all the bars being looked at. ### Mathematical Model #### Variables x(i,j) = {0,1} 1 if the bar crawl will travel from bar i to bar j, 0 otherwise u(i) = the position that the bar is visited during the crawl Totaldistance = the overall distance traveled during the whole bar crawl #### Equations N=22 obj equation to minimize total distance traveled to reach all bars $ _{(i,j)} distance(i,j)*x(i,j)$ positive u(i) equation $u(i) 0 $ i=1…N total paths taken to reach alllocations $N-1 = \sum_{(i,j)} x(i,j)$ every bar should have atleast one path coming from it or going to it $ 1 = *{(i)} x(i,j)$ j..N $ 1 = {(j)} x(i,j)$ i..N the first site visited should be the memorial union $1 = u(i) i =$ "$MemorialUnion$" the last site where the bar crawl will end is the capitol building $N = u(i) i =$ "$MemorialUnion$" if you travel from bar i to bar j then the position of bar j should be 1 greater than bar i $N(1 - x(i,j)) \geq 1 + u(i) - u(j)$ i..N, j..N

[9]:
```
%%gams
alias (bars, i, j);

Positive variable u(i) "position in order of bar crawl for bar i";
Variable totalDistance "the total distance traveled by the group";
Binary variable x(i, j) "if they choose to travel from bar i to bar j";
```

```
Equation defobj_eqn;   defobj_eqn.. totalDistance =e= sum((i,j), distance(i,j)␣
 ↪* x(i,j));
Equation visit_all_eqn; visit_all_eqn.. card(i) - 1 =e= sum((i,j), x(i,j));
Equation assigned_from_eqn; assigned_from_eqn(j).. 1 =g= sum(i$(not␣
 ↪sameas(i,j)), x(i,j));
Equation assigned_to_eqn; assigned_to_eqn(i).. 1 =g= sum(j$(not sameas(i,j)),␣
 ↪x(i,j));
Equation source_eqn; source_eqn.. 1 =e= u("Memorial Union");
Equation sink_eqn; sink_eqn.. card(i) =e= u("Capitol Building");
Equation miller_tucker_zemlin_eqn; miller_tucker_zemlin_eqn(i,j).. card(i) * (1␣
 ↪- x(i,j)) =g= 1 + u(i) - u(j);

x.fx(i,i) = 0;

set tour(i,i);
option tour:0:0:1;

model visitallbars /defobj_eqn, visit_all_eqn, assigned_from_eqn,␣
 ↪assigned_to_eqn, source_eqn, sink_eqn, miller_tucker_zemlin_eqn/;
```

[10]:
```
%gams solve visitallbars using mip min totalDistance;
```

[10]:
```
  Solver Status        Model Status  Objective #equ #var Model Type Solver  \
0    Normal (1)  Optimal Global (1)     3.5264  532  507        MIP  CPLEX

  Solver Time
0       0.324
```

[11]:
```
%%gams
tour(i,j) = no;
tour(i,j)$(x.l(i,j) > 0.01) = yes;
```

1.3.2 Solution

The order the path for a bar crawl that visits all 20 bar options follows the order of the list below

[12]:
```
u = m['u'].records
u.rename(columns={'i':'bars'}, inplace=True)
tour = m['tour'].records
display(u.sort_values("level"))
```

	bars	level	marginal	lower	upper	scale
20	Memorial Union	1.0	0.0	0.0	inf	1.0
0	Sconnie Bar	2.0	0.0	0.0	inf	1.0
1	Jordans Big ten	3.0	0.0	0.0	inf	1.0
2	Lucky's Brew Pub	4.0	0.0	0.0	inf	1.0

```
3           buckinghams    5.0       0.0    0.0    inf    1.0
4              Echo Tap    6.0       0.0    0.0    inf    1.0
13         Nitty Gritty    7.0       0.0    0.0    inf    1.0
15      Vintage Spirits    8.0       0.0    0.0    inf    1.0
17                Wandos    9.0       0.0    0.0    inf    1.0
21         The Double U   10.0       0.0    0.0    inf    1.0
9          Kollege Klub   11.0       0.0    0.0    inf    1.0
11             City Bar   12.0       0.0    0.0    inf    1.0
12   State Street Brats   13.0       0.0    0.0    inf    1.0
10       Whiskey Jack's   14.0       0.0    0.0    inf    1.0
14               Mondays   15.0       0.0    0.0    inf    1.0
18          Chasers 2.0   16.0       0.0    0.0    inf    1.0
6   Hawks Bar and Grill   17.0       0.0    0.0    inf    1.0
16      Red Rock Saloon   18.0       0.0    0.0    inf    1.0
5             Pauls Club   19.0       0.0    0.0    inf    1.0
7        Genna's Lounge   20.0       0.0    0.0    inf    1.0
8             Tipsy Cow   21.0       0.0    0.0    inf    1.0
19     Capitol Building   22.0       0.0    0.0    inf    1.0
```

[13]:
```
objl = m['totalDistance'].toValue()
print('Total minimum distance to travel to all 20 bars will be ' + str(objl) + '␣
 ↪miles')
```

```
Total minimum distance to travel to all 20 bars will be 3.5264461761620414 miles
```

1.3.3 Visualization

Map of the path taken through downtown Madison, WI to visit each bar is below. The blue dots are the locations of the sites visited and the red line is the path the bar crawl will take between bars.

[14]:
```
from matplotlib import image
from matplotlib import pyplot as plt

def maps(data):
    new = pd.DataFrame(data)
    new = data[['bars','level']]
    new = new.loc[new['level'] > 0.0]
    new = new.set_index('bars')
    result = pd.concat([df, new], axis=1, join="inner")
    result = result.sort_values('level')
    newbox = (result['Longitude'].min(),result['Longitude'].
 ↪max(),result['Latitude'].min(),result['Latitude'].max())
    from matplotlib import image
    from matplotlib import pyplot as plt
    x_vals = []
    y_vals = []
```

```
    for i in range(len(result)):
        x_vals.append(result['Latitude'].iloc[i])
        y_vals.append(result['Longitude'].iloc[i])
    newbox = (-89.41021074, -89.38156536, 43.06775204, 43.0783)
    data = image.imread('map (6).png')
    plt.subplots(figsize = (10,10))
    # to draw first line from (100,400) to (500,100)
    # to draw second line from (150,100) to (450,400)
    plt.scatter(result['Longitude'],result['Latitude'], zorder=1, alpha= .7,␣
↪c='b', s=30)
    plt.plot(y_vals,x_vals, color="red", linewidth=2)
    # plt.axis('off')
    plt.imshow(data, zorder=0, extent = newbox, aspect= 'equal')
    plt.show()
```

[15]:
```
maps(u)
```

1.3.4 Model Type: complex TSP where not all sites are visited

In the second model 'barcrawl' the simple TSP model was changed to a more complex model that chooses a certain number of bars to visit(bars_visited) from the set so not all bars in the set need to be visited. This new model needed to consider that not all nodes would be visited so the order in which the bar is visited will be 0. The miller tucker equation to change the big M value to handle occasions where just because bar i doesn't flow directly to bar j doesn't mean they both don't have numerical positions in the path so big M needed to be bars_visited + 3. ### Mathematical Model #### Variables x(i,j) = {0,1} 1 if the bar crawl will travel from bar i to bar j, 0 otherwise u(i) = the position that the bar is visited during the crawl y(i) = {0,1} 1 if bar i is visited during the bar crawl. 0 otherwise Totaldistance = the overall distance traveled during the whole bar crawl #### Equations bars_visited = 10 N=22 obj equation to minimize total distance traveled to reach all bars $ _{(i,j)} distance(i,j)*x(i,j)$ positive u(i) equation $u(i) 0 $ i=1...N total paths taken to reach alllocations bars_visited +1 = $\sum_{(i,j)} x(i,j)$ if yi is equal to 0 x(i,j) needs to be 0

$y(i) \geq \sum_j x(i,j)$ j..N if yi is equal to 0 x(j,i) needs to be 0 $y(i) \geq \sum_i x(j,i)$ i..N the path should go from 1 bar to the capitol building $ 1 = *{(i)} x(i,"Capitol Building")$ the path should go from memorial union to 1 bar $ 1 = *{(j)} x("Memorial Union",j)$ the first site visited should be the memorial union $1 = u(i)i = "MemorialUnion"$ the last site where the bar crawl will end is the capitol building bars_visited $+2 = u(i)i = "CapitolBuilding"$ there should be 12 total sites visited bars_visited $ + 2 *{i} y(i)$ if you travel from bar i to bar j then the position of bar j should be 1 greater than bar i $N(1 - x(i,j)) \geq 1 + u(i) - u(j)$ i..N, j..N

[16]:
```
bars_visited = m.addParameter('bars_visited',records=10)
```

[17]:
```
%%gams
# GAMS Model
alias (bars, i, j);

Positive variable u2(i) "position in order of bar crawl for bar i";
Variable totalDistance2 "the total distance traveled by the group";
Binary variable x2(i, j) "if they choose to travel from bar i to bar j";
Binary variable y2(i) "if bar i is visited";

Equation defobj2_eqn "sums the total distance traveled between all bars visited␣
 ↪on crawl";   defobj2_eqn.. totalDistance2 =e= sum((i,j), distance(i,j) *␣
 ↪x2(i,j));
Equation visit_11paths_eqn "sums that the total paths taken between bars should␣
 ↪be 1 less than the total sites being visted"; visit_11paths_eqn..␣
 ↪bars_visited + 1 =e= sum((i,j), x2(i,j));
Equation yi_eqn "if yi is equal to 0 x(i,j) needs to be 0"; yi_eqn(i).. y2(i)␣
 ↪=g= sum(j, x2(i,j));
Equation second_yi_eqn "if yi is equal to 0 x(j,i) needs to be 0";␣
 ↪second_yi_eqn(i).. y2(i) =g= sum(j, x2(j,i));
Equation memorial_eqn; memorial_eqn.. 1 =e= sum(j, x2("Memorial Union", j));
Equation capitol_eqn; capitol_eqn.. 1 =e= sum(i, x2(i, "Capitol Building"));
Equation source2_eqn; source2_eqn.. 1 =e= u2("Memorial Union");
Equation sink2_eqn; sink2_eqn.. bars_visited + 2 =e= u2("Capitol Building");
Equation sum_position_eqn; sum_position_eqn.. sum(i, y2(i)) =l= bars_visited + 2;
Equation miller_tucker_zemlin2_eqn; miller_tucker_zemlin2_eqn(i,j)..␣
 ↪(bars_visited + 3)*(1 - x2(i,j)) =g= 1 + u2(i) - u2(j);

x2.fx(i,i) = 0;


set tour2(i,i);
option tour2:0:0:1;

model barcrawl /defobj2_eqn, visit_11paths_eqn, yi_eqn, second_yi_eqn,␣
 ↪memorial_eqn, capitol_eqn, source2_eqn, sink2_eqn, sum_position_eqn,␣
 ↪miller_tucker_zemlin2_eqn/;
```

[18]:
```
%gams u2.lo(i) = 0;
%gams u2.up(i) = bars_visited + 2;
```

[19]:
```
%gams solve barcrawl using mip min totalDistance2;
```

[19]:
```
  Solver Status        Model Status  Objective #equ #var Model Type Solver  \
0    Normal (1)  Optimal Global (1)     1.0503  535  529        MIP  CPLEX

  Solver Time
0       0.166
```

[20]:
```
%%gams
tour2(i,j) = no;
tour2(i,j)$(x2.l(i,j) > 0.01) = yes;
```

1.3.5 Solution

The order of the path this bar crawl will take to visit the 10 bars included is

[21]:
```
u2 = m['u2'].records
u2.rename(columns={'i':'bars'}, inplace=True)
display(u2.sort_values("level"))
```

	bars	level	marginal	lower	upper	scale
0	Sconnie Bar	0.0	-0.0	0.0	12.0	1.0
1	Jordans Big ten	0.0	-0.0	0.0	12.0	1.0
2	Lucky's Brew Pub	0.0	-0.0	0.0	12.0	1.0
3	buckinghams	0.0	-0.0	0.0	12.0	1.0
4	Echo Tap	0.0	-0.0	0.0	12.0	1.0
18	Chasers 2.0	0.0	-0.0	0.0	12.0	1.0
15	Vintage Spirits	0.0	-0.0	0.0	12.0	1.0
7	Genna's Lounge	0.0	-0.0	0.0	12.0	1.0
8	Tipsy Cow	0.0	-0.0	0.0	12.0	1.0
13	Nitty Gritty	0.0	-0.0	0.0	12.0	1.0
20	Memorial Union	1.0	0.0	0.0	12.0	1.0
9	Kollege Klub	2.0	0.0	0.0	12.0	1.0
11	City Bar	3.0	0.0	0.0	12.0	1.0
12	State Street Brats	4.0	0.0	0.0	12.0	1.0
21	The Double U	5.0	0.0	0.0	12.0	1.0
17	Wandos	6.0	0.0	0.0	12.0	1.0
14	Mondays	7.0	0.0	0.0	12.0	1.0
10	Whiskey Jack's	8.0	0.0	0.0	12.0	1.0
6	Hawks Bar and Grill	9.0	0.0	0.0	12.0	1.0
16	Red Rock Saloon	10.0	0.0	0.0	12.0	1.0
5	Pauls Club	11.0	0.0	0.0	12.0	1.0
19	Capitol Building	12.0	0.0	0.0	12.0	1.0

Bars with a 0 position in the path mean those bars weren't chosen to be visited at all. The minimum distance that must be walked to visit these 10 bars when starting at Memorial Union and ending at the Capitol Building is

[22]:
```
objl2 = m['totalDistance2'].toValue()
print('Total minimum distance to visit 10 of the 20 bars will be ' + str(objl2)␣
↪+ ' miles')
```

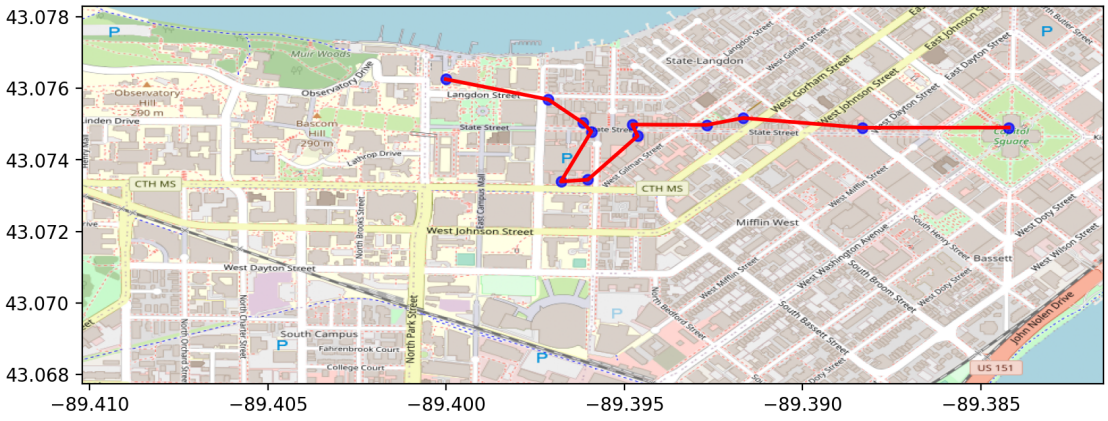
```
Total minimum distance to visit 10 of the 20 bars will be 1.050320033466992
miles
```

1.3.6 Visualization

Map of the path taken through downtown Madison, WI to visit each bar is below. The blue dots are the locations of the sites visited and the red line is the path the bar crawl will take between bars.

[23]:
```
new = u2[['bars','level']]
new = new.loc[new['level'] > 0.0]
new = new.set_index('bars')
# new
result = pd.concat([df, new], axis=1, join="inner")
result = result.sort_values('level')
```

[24]:
```
maps(u2)
```



1.3.7 Model Type: Complex TSP with tradeoff

For the last model we complicated the problem again by adding the tradeoff aspect to the problem. We did this by adding a constraint to maximize a participant's BAC level while keeping it below a desired threshold. When adding this constraint we also had to consider not only if a bar was

visited, but if it was visited before other bars. We also added the scalar(bac_decrease) of how much a bar crawl's participant's BAC level decreases per mile they walk during the bar crawl. To keep track of the bac level after visiting each bar we had to add another binary variable z(i,j) to say if bar i was visited before bar j. We needed this to sum up the distances and BAC levels from all the previous bars visited. We did this using another big M equation to set this binary variable to 0 if the position of bar j was less than or equal to bar i. ### Mathematical Model #### Variables x(i,j) = {0,1} 1 if the bar crawl will travel from bar i to bar j, 0 otherwise u(i) = the position that the bar is visited during the crawl y(i) = {0,1} 1 if bar i is visited during the bar crawl. 0 otherwise z(i,j) = {0,1} 1 if bar i is visited before bar j during the bar crawl, 0 otherwise Totaldistance = the overall distance traveled during the whole bar crawl #### Equations bars_visited = 10 N=22 obj equation to minimize total distance traveled to reach all bars $ _{(i,j)} distance(i,j)*x(i,j)$ positive u(i) equation $u(i) 0 $ i=1…N max_bac $\geq$ $(\sum_j(z(j,i)+x3(i,j)-1)*bacperbar(j)))-(\sum_j(z(j,i)+x3(i,j)-1)*bacdecrease*distance(i,j)))$ i..N total paths taken to reach alllocations bars_visited +1 = $\sum_{(i,j)} x(i,j)$ if yi is equal to 0 x(i,j) needs to be 0 $y(i) \geq \sum_j x(i,j)$ j..N if yi is equal to 0 x(j,i) needs to be 0 $y(i) \geq \sum_i x(j,i)$ i..N the path should go from 1 bar to the capitol building $ 1 = *{(i)} x(i,"Capitol Building")$ the path should go from memorial union to 1 bar $ 1 =* {(j)} x("Memorial Union",j)$ the first site visited should be the memorial union 1 = $u(i)i$ = "$MemorialUnion$" the last site where the bar crawl will end is the capitol building bars_visited +2 = $u(i)i$ = "$CapitolBuilding$" there should be 12 total sites visited bars_visited $ + 2 _{i} y(i)$ z(i,j) will be 0 if u(j) is less than or equal to u(i) $u3(j) \geq u3(i)+1-((barsvisited+3)*(1-z(i,j)))$ each position 1-12 should only be assigned to 1 bar $\sum_i u3(i) = 78$ z(i,j) will be 1 if u(j) is greater than u(i) $u3(j) \leq u3(i)+((barsvisited+3)*z(i,j))$ i..N, j..N if you travel from bar i to bar j then the position of bar j should be 1 greater than bar i $N(1-x(i,j)) \geq 1+u(i)-u(j)$ i..N, j..N

[25]:
```
%%gams
set baroptions(bars);
baroptions(bars) = yes;
baroptions("Memorial Union") = no;
baroptions("Capitol Building") = no;



Positive variable u3(i) "position in order of bar crawl for bar i";
Variable totalDistance3 "the total distance traveled by the group";
Binary variable x3(i, j) "if they choose to travel from bar i to bar j";
Binary variable y3(i) "if bar i is visited";
Binary variable z(i,j) "1 if bar i is visited before bar j,zero if at same␣
 ↪position because that means neither bar is part of the path"

Equation defobj3_eqn;   defobj3_eqn.. totalDistance3 =e= sum((i,j),␣
 ↪distance(i,j) * x3(i,j));
Equation max_bac_eqn; max_bac_eqn(baroptions(i)).. max_bac =g= sum(j,␣
 ↪(z(j,i)+x3(i,j) - 1)*bac_per_bar(j)) - sum(j, (z(j,i)+x3(i,j) -␣
 ↪1)*bac_decrease*distance(i,j));
Equation visit_11paths3_eqn; visit_11paths3_eqn.. bars_visited + 1 =e=␣
 ↪sum((i,j), x3(i,j));
```

```
Equation yi3_eqn "if yi is equal to 0 x(i,j) and x(j,i) need to be 0";␣
 ↪yi3_eqn(i).. y3(i) =g= sum(j, x3(i,j));
Equation second_yi3_eqn; second_yi3_eqn(i).. y3(i) =g= sum(j, x3(j,i));
Equation memorial3_eqn; memorial3_eqn.. 1 =e= sum(j, x3("Memorial Union", j));
Equation capitol3_eqn; capitol3_eqn.. 1 =e= sum(i, x3(i, "Capitol Building"));
Equation source3_eqn; source3_eqn.. 1 =e= u3("Memorial Union");
Equation sink3_eqn; sink3_eqn.. bars_visited + 2 =e= u3("Capitol Building");
Equation sum_position3_eqn; sum_position3_eqn.. sum(i, y3(i)) =l= 12;
Equation z_eqn; z_eqn(i,j).. u3(j) =g= u3(i) + 1 - ((bars_visited + 3)*(1 -␣
 ↪z(i,j)));
Equation z_balance; z_balance(i,j).. u3(j) =l= u3(i) + ((bars_visited +␣
 ↪3)*z(i,j));
Equation sum_u_eqn; sum_u_eqn.. sum(i, u3(i)) =e= 78;
Equation miller_tucker_zemlin3_eqn; miller_tucker_zemlin3_eqn(i,j)..␣
 ↪(bars_visited + 3)*(1 - x3(i,j)) =g= 1 + u3(i) - u3(j);

x3.fx(i,i) = 0;

set tour3(i,i);
option tour3:0:0:1;

model barcrawlBAClimit / defobj3_eqn, max_bac_eqn, visit_11paths3_eqn, yi3_eqn,␣
 ↪second_yi3_eqn, memorial3_eqn, capitol3_eqn, source3_eqn, sink3_eqn,␣
 ↪sum_position3_eqn, z_eqn, z_balance, sum_u_eqn, miller_tucker_zemlin3_eqn /;
u3.lo(i) = 0;
u3.up(i) = bars_visited + 2;
tour3(i,j) = no;
tour3(i,j)$(x3.l(i,j) > 0.01) = yes;
```

[26]:
```
%gams solve barcrawlBAClimit using mip min totalDistance3;
```

[26]:
```
  Solver Status        Model Status  Objective  #equ  #var Model Type Solver  \
0    Normal (1)  Optimal Global (1)     1.0503  1524  1013        MIP  CPLEX

  Solver Time
0       0.497
```

1.3.8 Solution

The order of the path this bar crawl will take to visit the 10 bars included that keeps their BAC level below the desired max is

[27]:
```
u3 = m['u3'].records
u3.rename(columns={'i':'bars'}, inplace=True)
display(u3.sort_values("level"))
```

	bars	level	marginal	lower	upper	scale
0	Sconnie Bar	0.0	-0.0	0.0	12.0	1.0
1	Jordans Big ten	0.0	-0.0	0.0	12.0	1.0
2	Lucky's Brew Pub	0.0	-0.0	0.0	12.0	1.0
3	buckinghams	0.0	-0.0	0.0	12.0	1.0
4	Echo Tap	0.0	-0.0	0.0	12.0	1.0
18	Chasers 2.0	0.0	-0.0	0.0	12.0	1.0
15	Vintage Spirits	0.0	-0.0	0.0	12.0	1.0
7	Genna's Lounge	0.0	-0.0	0.0	12.0	1.0
8	Tipsy Cow	0.0	-0.0	0.0	12.0	1.0
13	Nitty Gritty	0.0	-0.0	0.0	12.0	1.0
20	Memorial Union	1.0	0.0	0.0	12.0	1.0
9	Kollege Klub	2.0	0.0	0.0	12.0	1.0
11	City Bar	3.0	0.0	0.0	12.0	1.0
12	State Street Brats	4.0	0.0	0.0	12.0	1.0
21	The Double U	5.0	0.0	0.0	12.0	1.0
17	Wandos	6.0	0.0	0.0	12.0	1.0
14	Mondays	7.0	0.0	0.0	12.0	1.0
10	Whiskey Jack's	8.0	0.0	0.0	12.0	1.0
6	Hawks Bar and Grill	9.0	0.0	0.0	12.0	1.0
16	Red Rock Saloon	10.0	0.0	0.0	12.0	1.0
5	Pauls Club	11.0	0.0	0.0	12.0	1.0
19	Capitol Building	12.0	0.0	0.0	12.0	1.0

Bars with a 0 position in the path mean those bars weren't chosen to be visited at all. The minimum distance that must be walked to visit these 10 bars when starting at Memorial Union and ending at the Capitol Building while maintaing a BAC below the maximum level is

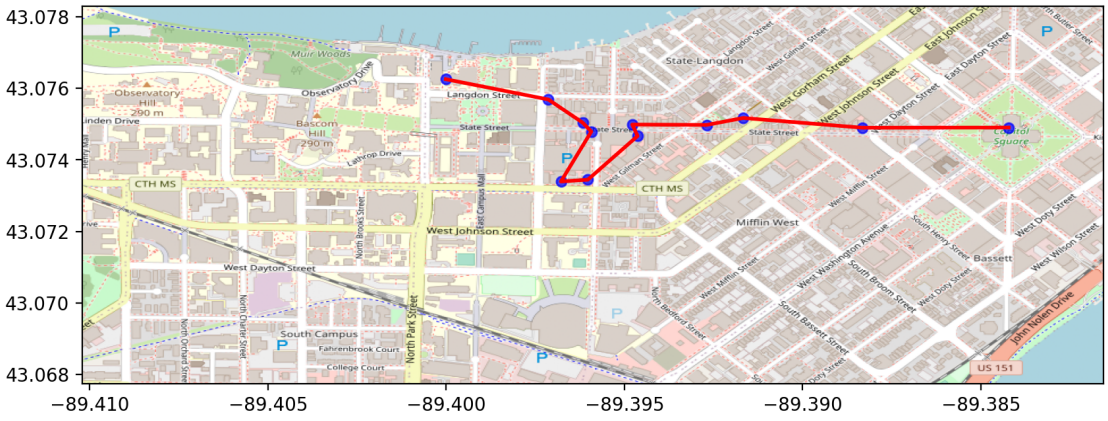
[28]:
```
obj13 = m['totalDistance3'].toValue()
print('Total minimum distance to visit 10 of the 20 bars will be ' + str(obj13)␣
↪+ ' miles')
```

```
Total minimum distance to visit 10 of the 20 bars will be 1.050320033466992
miles
```

1.3.9 Visualization

Map of the path taken through downtown Madison, WI to visit each bar is below. The blue dots are the locations of the sites visited and the red line is the path the bar crawl will take between bars.

[29]:
```
maps(u3)
```



To see how sensitive the model is to the max_bac level the model is also run with a lower max bac level of .15

[30]:
```
max_bac.setRecords(0.15)
%gams solve barcrawlBAClimit using mip min totalDistance3;
```

[30]:
```
   Solver Status        Model Status  Objective  #equ  #var Model Type Solver  \
0    Normal (1)  Optimal Global (1)     1.0503  1524  1013        MIP  CPLEX

   Solver Time
0        0.499
```

[31]:
```
u3 = m['u3'].records
u3.rename(columns={'i':'bars'}, inplace=True)
display(u3.sort_values("level"))
```

	bars	level	marginal	lower	upper	scale
0	Sconnie Bar	0.0	-0.0	0.0	12.0	1.0
1	Jordans Big ten	0.0	-0.0	0.0	12.0	1.0
2	Lucky's Brew Pub	0.0	-0.0	0.0	12.0	1.0
3	buckinghams	0.0	-0.0	0.0	12.0	1.0
4	Echo Tap	0.0	-0.0	0.0	12.0	1.0
18	Chasers 2.0	0.0	-0.0	0.0	12.0	1.0
15	Vintage Spirits	0.0	-0.0	0.0	12.0	1.0
7	Genna's Lounge	0.0	-0.0	0.0	12.0	1.0
8	Tipsy Cow	0.0	-0.0	0.0	12.0	1.0
13	Nitty Gritty	0.0	-0.0	0.0	12.0	1.0
20	Memorial Union	1.0	0.0	0.0	12.0	1.0
9	Kollege Klub	2.0	0.0	0.0	12.0	1.0
11	City Bar	3.0	0.0	0.0	12.0	1.0
12	State Street Brats	4.0	0.0	0.0	12.0	1.0
21	The Double U	5.0	0.0	0.0	12.0	1.0
17	Wandos	6.0	0.0	0.0	12.0	1.0

```
14              Mondays    7.0       0.0    0.0   12.0    1.0
10       Whiskey Jack's    8.0       0.0    0.0   12.0    1.0
6   Hawks Bar and Grill    9.0       0.0    0.0   12.0    1.0
16      Red Rock Saloon   10.0       0.0    0.0   12.0    1.0
5            Pauls Club   11.0       0.0    0.0   12.0    1.0
19     Capitol Building   12.0       0.0    0.0   12.0    1.0
```

As shown from the results above our model is not very sensitive to the max_bac level because the path stayed the same and the minimum distance traveled for the bar crawl stayed at 1.05 miles.

[32]:
```
max_bac.setRecords(0.02)
%gams solve barcrawlBAClimit using mip min totalDistance3;
```

[32]:
```
  Solver Status        Model Status  Objective  #equ  #var Model Type Solver  \
0    Normal (1)  Optimal Global (1)     1.0503  1524  1013        MIP  CPLEX

  Solver Time
0       0.519
```

1.4 Conclusion

In conclusion, with the choice between all 22 possible bars or a select 10, the maximum distance traveled was 3.526446 and 1.0503 miles, respectively. BAC and desired maximum BAC failed to have an impact on the distance traveled, whether this is because a mistake within the equations or the total amount walked during the bar crawl truly is enough to keep someone under the maximum BAC is unsure. We think it has to do with the walking amount cancelling out the alcohol consumed because we originally had the Lucky's bar located somewhere else in Wisconsin in our problem instead of the Lucky's in Madison and this caused a larger distance and when we lowered the bac and added it as a constraint both changes led to the distance and path changing where now they aren't affecting the model results when we have the correct Lucky's location included. Overall, the minimum distance to travel to 10 bars for a Madison bar crawl with a max BAC of .08 is 1.05 and visits the chosen bars in the following order.

[33]:
```
display(u3.sort_values("level"))
```

```
                 bars  level  marginal  lower  upper  scale
0          Sconnie Bar    0.0      -0.0    0.0   12.0    1.0
1       Jordans Big ten    0.0      -0.0    0.0   12.0    1.0
2      Lucky's Brew Pub    0.0      -0.0    0.0   12.0    1.0
3           buckinghams    0.0      -0.0    0.0   12.0    1.0
4              Echo Tap    0.0      -0.0    0.0   12.0    1.0
18         Chasers 2.0     0.0      -0.0    0.0   12.0    1.0
15      Vintage Spirits    0.0      -0.0    0.0   12.0    1.0
7        Genna's Lounge    0.0      -0.0    0.0   12.0    1.0
8            Tipsy Cow     0.0      -0.0    0.0   12.0    1.0
13         Nitty Gritty    0.0      -0.0    0.0   12.0    1.0
20       Memorial Union    1.0       0.0    0.0   12.0    1.0
9          Kollege Klub    2.0       0.0    0.0   12.0    1.0
```

11	City Bar	3.0	0.0	0.0	12.0	1.0
12	State Street Brats	4.0	0.0	0.0	12.0	1.0
21	The Double U	5.0	0.0	0.0	12.0	1.0
17	Wandos	6.0	0.0	0.0	12.0	1.0
14	Mondays	7.0	0.0	0.0	12.0	1.0
10	Whiskey Jack's	8.0	0.0	0.0	12.0	1.0
6	Hawks Bar and Grill	9.0	0.0	0.0	12.0	1.0
16	Red Rock Saloon	10.0	0.0	0.0	12.0	1.0
5	Pauls Club	11.0	0.0	0.0	12.0	1.0
19	Capitol Building	12.0	0.0	0.0	12.0	1.0

[34]:
```
%gams_cleanup --closedown
```

[ ]:

[ ]: