shortPath nb

April 29, 2018

In [1]:
```
# Austin Griffith
# Python 3.6.5
# 4/25/2018

import pandas as pd
import numpy as np
from gurobipy import *
import matplotlib.pyplot as plt
import matplotlib.pylab as pylab
import networkx as nx
```

In [2]:
```
# set up plotting parameters
params = {'legend.fontsize': 20,
          'figure.figsize': (13,9),
         'axes.labelsize': 20,
         'axes.titlesize':20,
         'xtick.labelsize':15,
         'ytick.labelsize':15}
pylab.rcParams.update(params)
```

In [3]:
```
# graph all nodes and paths
def networkCompletePlot(solution,maxNode):
    G = nx.DiGraph()
    G.add_nodes_from(range(0,maxNode+1))
    for i,j in nodes:
        G.add_edge(i,j)

    # get solution nodes
    sp = [i for i,j in solution[1]]
    sp.append(end)

    colorNode = ['white' if not node in sp else 'red' for node in G.nodes()]
    title = 'Complete Network: Gamma = '+str(int(solution[0]))+', Opt Obj = '+str(round(
    nx.draw_networkx(G,node_color=colorNode,node_size=200)
    plt.axis('off')
    plt.title(title)
    plt.show()
```

```
# graph path, with costs on edges
def networkPathPlot(solution,maxNode,cost):
    # get solution nodes
    sp = [i for i,j in solution[1]]
    sp.append(end)

    # set up random position values
    a = np.arange(maxNode+1)
    b = np.arange(maxNode+1)
    np.random.shuffle(a)
    posArray = np.array([a,b]).transpose()

    positions = {}
    for p in range(0,len(sp)):
        L = posArray[p]
        positions[sp[p]] = (L[0],L[1])

    # set up network graph
    G = nx.DiGraph()
    G.add_nodes_from(sp)

    for i,j in tuplelist(solution[1]):
        G.add_edge(i,j)

    labels = {}
    for i in solution[1]:
        labels[i] = round(c[i],3)

    title = 'Optimal Path: Gamma = '+str(int(solution[0]))+', Opt Obj = '+str(round(solu
    nx.draw_networkx(G,positions,node_size=350)
    nx.draw_networkx_edge_labels(G,positions,edge_labels=labels)
    plt.axis('off')
    plt.title(title)
    plt.show()
```

In [4]:
```
# pull data
edges = pd.read_csv('edge_data.csv')
edges['i'] = np.int64(edges['i'])
edges['j'] = np.int64(edges['j'])

# create dictionaries of edge values
c = {}
d = {}
nodes = tuplelist()
for i in edges.index:
    c[edges['i'][i],edges['j'][i]] = edges['c(ij)'][i]
    d[edges['i'][i],edges['j'][i]] = edges['d(ij)'][i]
    nodes.append((edges['i'][i],edges['j'][i]))
```

```
maxNodes = max(edges['j'])
minNodes = min(edges['i'])
```

In [5]:
```
# choose start and end nodes
start = 0
end = 49

# allowed edge congestions
gend = 4
gammas = np.linspace(0,gend,gend+1)
print('Allowed Congestions:')
print(gammas)
```

```
Allowed Congestions:
[ 0.  1.  2.  3.  4.]
```

In [6]:
```
# initialize model
model = Model('Shortest_Path')

# set up x binary variables, set to each location/movement
xVars = model.addVars(nodes, vtype=GRB.BINARY, name='move')
y0 = model.addVar(vtype=GRB.CONTINUOUS, name='y0')
zVars = model.addVars(nodes, lb=0.0, vtype=GRB.CONTINUOUS, name='cong')
model.update()
```

In [7]:
```
# constrain all entrance and exit nodes
enterStart = []
leaveStart = []
enterEnd = []
leaveEnd = []
for n in nodes:
    # for start nodes
    if n[0] == start:
        leaveStart.append(xVars[n])
    elif n[1] == start:
        enterStart.append(xVars[n])
    # for end nodes
    if n[0] == end:
        leaveEnd.append(xVars[n])
    elif n[1] == end:
        enterEnd.append(xVars[n])

model.addConstr(quicksum(leaveStart) == 1)
model.addConstr(quicksum(enterStart) == 0)
model.addConstr(quicksum(leaveEnd) == 0)
model.addConstr(quicksum(enterEnd) == 1)
model.update()
```

In [8]:
```
# gather all paths
paths = []
for i in range(minNodes+1,maxNodes):
    pathFrom = []
    pathTo = []
    for n in nodes:
        if n[0] == i:
            pathFrom.append(xVars[n])
        elif n[1] == i:
            pathTo.append(xVars[n])
    paths.append([pathFrom,pathTo])
model.update()

for p in paths:
    model.addConstr(quicksum(p[0]) - quicksum(p[1]) == 0.0)
model.update()

print('Example of Path Constraint for a Given Node:')
print(quicksum(p[0]) - quicksum(p[1]))
```

```
Example of Path Constraint for a Given Node:
<gurobi.LinExpr: move[48,0] + move[48,1] + move[48,2] + move[48,3] + move[48,4] + move[48,5] + m
```

In [9]:
```
# objective function
costObj = []
for n in nodes:
    costObj.append(xVars[n]*c[n])
    model.addConstr(zVars[n] >= xVars[n]*d[n] - y0)
model.update()

print('Example of Congestion Constraint:')
print(zVars[n],' >= ',xVars[n]*d[n] - y0)
```

```
Example of Congestion Constraint:
<gurobi.Var cong[49,48]>  >=  <gurobi.LinExpr: 3.9051301780000003 move[49,48] + -1.0 y0>
```

In [10]:
```
# iterate optimization through various gammas (congestions)
output = []
for g in gammas:
    # optimize
    objective = quicksum(costObj) + g*y0 + quicksum(zVars)
    model.setObjective(objective, GRB.MINIMIZE)

    model.optimize()

    # order the printout of optimal edges
    moves = []
```

```python
            for m in xVars:
                if xVars[m].x != 0:
                    moves.append(m)
            order = [moves[0]]
            for i in range(len(moves)):
                for m in moves:
                    if order[i][1] == m[0]:
                        order.append(m)
            output.append([g,order,model.objVal])
```

```
Optimize a model with 2261 rows, 4419 columns and 11045 nonzeros
Variable types: 2210 continuous, 2209 integer (2209 binary)
Coefficient statistics:
  Matrix range     [8e-05, 5e+00]
  Objective range  [1e-05, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 0.2538690
Presolve removed 2211 rows and 3869 columns
Presolve time: 0.01s
Presolved: 50 rows, 550 columns, 1100 nonzeros
Variable types: 0 continuous, 550 integer (550 binary)

Root relaxation: objective 1.331944e-01, 22 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

*    0     0               0       0.1331944    0.13319  0.00%     -    0s

Explored 0 nodes (22 simplex iterations) in 0.03 seconds
Thread count was 8 (of 8 available processors)

Solution count 2: 0.133194 0.253869

Optimal solution found (tolerance 1.00e-04)
Best objective 1.331943590000e-01, best bound 1.331943590000e-01, gap 0.0000%
Optimize a model with 2261 rows, 4419 columns and 11045 nonzeros
Variable types: 2210 continuous, 2209 integer (2209 binary)
Coefficient statistics:
  Matrix range     [8e-05, 5e+00]
  Objective range  [1e-05, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]

Loaded MIP start with objective 4.13319

Presolve removed 365 rows and 1409 columns
```

```
Presolve time: 0.07s
Presolved: 1896 rows, 3010 columns, 17106 nonzeros
Variable types: 1164 continuous, 1846 integer (1846 binary)

Root relaxation: objective 7.420269e-01, 63 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.74203    0   34    4.13319    0.74203  82.0%     -    0s
H    0     0                       1.9536830    0.74203  62.0%     -    0s
H    0     0                       1.9184585    0.74203  61.3%     -    0s
H    0     0                       1.9105642    0.74203  61.2%     -    0s
H    0     0                       1.6807535    0.75255  55.2%     -    0s
     0     0    0.81623    0   16    1.68075    0.81623  51.4%     -    0s
     0     0    0.94160    0   27    1.68075    0.94160  44.0%     -    0s
H    0     0                       1.3586585    0.94160  30.7%     -    0s
     0     0    1.01801    0   18    1.35866    1.01801  25.1%     -    0s
     0     0    1.10836    0   20    1.35866    1.10836  18.4%     -    0s
H    0     0                       1.3528823    1.11922  17.3%     -    0s
     0     0    1.17071    0   19    1.35288    1.17071  13.5%     -    0s
     0     0    1.18298    0   13    1.35288    1.18298  12.6%     -    0s
H    0     0                       1.2992082    1.18298  8.95%     -    0s
     0     0    1.28002    0   13    1.29921    1.28002  1.48%     -    0s
     0     0    1.28091    0    3    1.29921    1.28091  1.41%     -    0s
     0     0     cutoff    0         1.29921    1.29921  0.00%     -    0s

Cutting planes:
  Gomory: 2
  MIR: 2

Explored 1 nodes (270 simplex iterations) in 0.28 seconds
Thread count was 8 (of 8 available processors)

Solution count 8: 1.29921 1.35288 1.35866 ... 4.13319

Optimal solution found (tolerance 1.00e-04)
Best objective 1.299208152000e+00, best bound 1.299208152000e+00, gap 0.0000%
Optimize a model with 2261 rows, 4419 columns and 11045 nonzeros
Variable types: 2210 continuous, 2209 integer (2209 binary)
Coefficient statistics:
  Matrix range     [8e-05, 5e+00]
  Objective range  [1e-05, 2e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]

Loaded MIP start with objective 1.91931
```

```
Presolve removed 91 rows and 178 columns
Presolve time: 0.01s
Presolved: 2170 rows, 4241 columns, 10600 nonzeros
Variable types: 2121 continuous, 2120 integer (2120 binary)

Root relaxation: objective 7.612361e-01, 77 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.76124    0   40    1.91931    0.76124  60.3%     -    0s
H    0     0                       1.8339424    0.76124  58.5%     -    0s
     0     0    1.00051    0   27    1.83394    1.00051  45.4%     -    0s
     0     0    1.01980    0   25    1.83394    1.01980  44.4%     -    0s
     0     0    1.13039    0   26    1.83394    1.13039  38.4%     -    0s
     0     0    1.13039    0   26    1.83394    1.13039  38.4%     -    0s
H    0     0                       1.8229833    1.13385  37.8%     -    0s
     0     0    1.14204    0   23    1.82298    1.14204  37.4%     -    0s
     0     0    1.21474    0   22    1.82298    1.21474  33.4%     -    0s
     0     0    1.23511    0   26    1.82298    1.23511  32.2%     -    0s
     0     0    1.25863    0   26    1.82298    1.25863  31.0%     -    0s
     0     0    1.34503    0   25    1.82298    1.34503  26.2%     -    0s
     0     0    1.34503    0   33    1.82298    1.34503  26.2%     -    0s
     0     0    1.34503    0   24    1.82298    1.34503  26.2%     -    0s
     0     0    1.34503    0   25    1.82298    1.34503  26.2%     -    0s
H    0     0                       1.6457191    1.34503  18.3%     -    0s
     0     0    1.34503    0   20    1.64572    1.34503  18.3%     -    0s
     0     0    1.34503    0   20    1.64572    1.34503  18.3%     -    0s
     0     0    1.34503    0   18    1.64572    1.34503  18.3%     -    0s
     0     0    1.34503    0   23    1.64572    1.34503  18.3%     -    0s
     0     0    1.35059    0   20    1.64572    1.35059  17.9%     -    0s
     0     0    1.59872    0   17    1.64572    1.59872  2.86%     -    0s
     0     0    1.59872    0   25    1.64572    1.59872  2.86%     -    0s
     0     0    1.59872    0   22    1.64572    1.59872  2.86%     -    0s
     0     0    1.59872    0   25    1.64572    1.59872  2.86%     -    0s
     0     0    1.59872    0    7    1.64572    1.59872  2.86%     -    0s
     0     0    1.62779    0    9    1.64572    1.62779  1.09%     -    0s
     0     0     cutoff    0         1.64572    1.64572  0.00%     -    0s

Cutting planes:
  Gomory: 3
  Clique: 1
  MIR: 1

Explored 1 nodes (513 simplex iterations) in 0.28 seconds
Thread count was 8 (of 8 available processors)

Solution count 4: 1.64572 1.82298 1.83394 1.91931
```

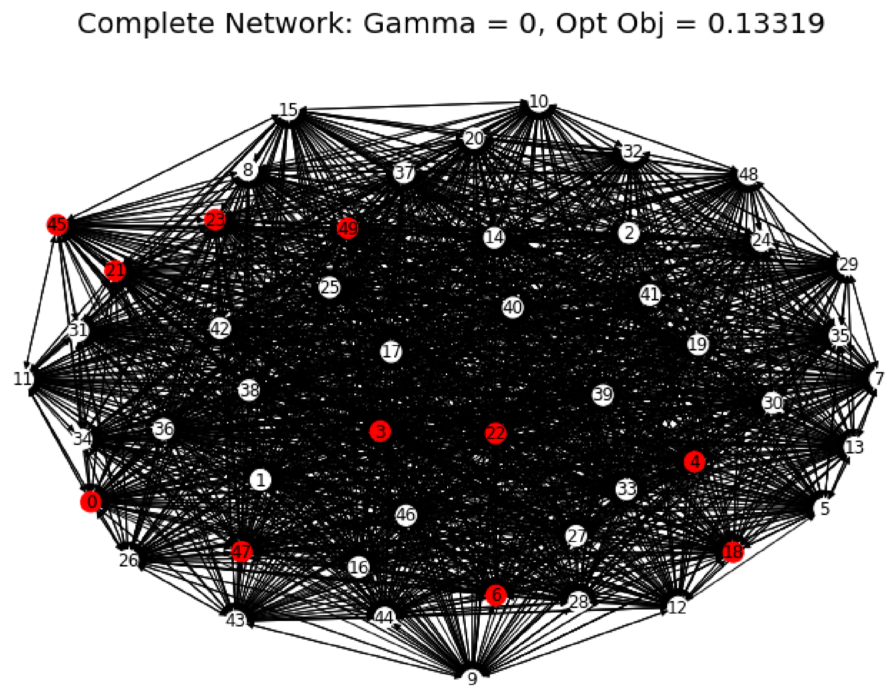
```
Optimal solution found (tolerance 1.00e-04)
Best objective 1.645719055000e+00, best bound 1.645719055000e+00, gap 0.0000%
Optimize a model with 2261 rows, 4419 columns and 11045 nonzeros
Variable types: 2210 continuous, 2209 integer (2209 binary)
Coefficient statistics:
  Matrix range     [8e-05, 5e+00]
  Objective range  [1e-05, 3e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]

Loaded MIP start with objective 1.99223

Presolve removed 91 rows and 178 columns
Presolve time: 0.01s
Presolved: 2170 rows, 4241 columns, 10600 nonzeros
Variable types: 2121 continuous, 2120 integer (2120 binary)

Root relaxation: objective 8.701595e-01, 88 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.87016    0   57    1.99223    0.87016  56.3%     -    0s
H    0     0                       1.8339424    0.87016  52.6%     -    0s
     0     0    1.22721    0   28    1.83394    1.22721  33.1%     -    0s
     0     0    1.22721    0   28    1.83394    1.22721  33.1%     -    0s
     0     0    1.36495    0   36    1.83394    1.36495  25.6%     -    0s
H    0     0                       1.7690089    1.36495  22.8%     -    0s
     0     0    1.43516    0   27    1.76901    1.43516  18.9%     -    0s
     0     0    1.54196    0   29    1.76901    1.54196  12.8%     -    0s
     0     0    1.54196    0   16    1.76901    1.54196  12.8%     -    0s
     0     0    1.65454    0   21    1.76901    1.65454  6.47%     -    0s
     0     0    1.66588    0   15    1.76901    1.66588  5.83%     -    0s
     0     0    1.75053    0    6    1.76901    1.75053  1.04%     -    0s
     0     0     cutoff    0         1.76901    1.76901  0.00%     -    0s

Explored 1 nodes (373 simplex iterations) in 0.15 seconds
Thread count was 8 (of 8 available processors)

Solution count 3: 1.76901 1.83394 1.99223

Optimal solution found (tolerance 1.00e-04)
Best objective 1.769008875000e+00, best bound 1.769008875000e+00, gap 0.0000%
Optimize a model with 2261 rows, 4419 columns and 11045 nonzeros
Variable types: 2210 continuous, 2209 integer (2209 binary)
Coefficient statistics:
  Matrix range     [8e-05, 5e+00]
```

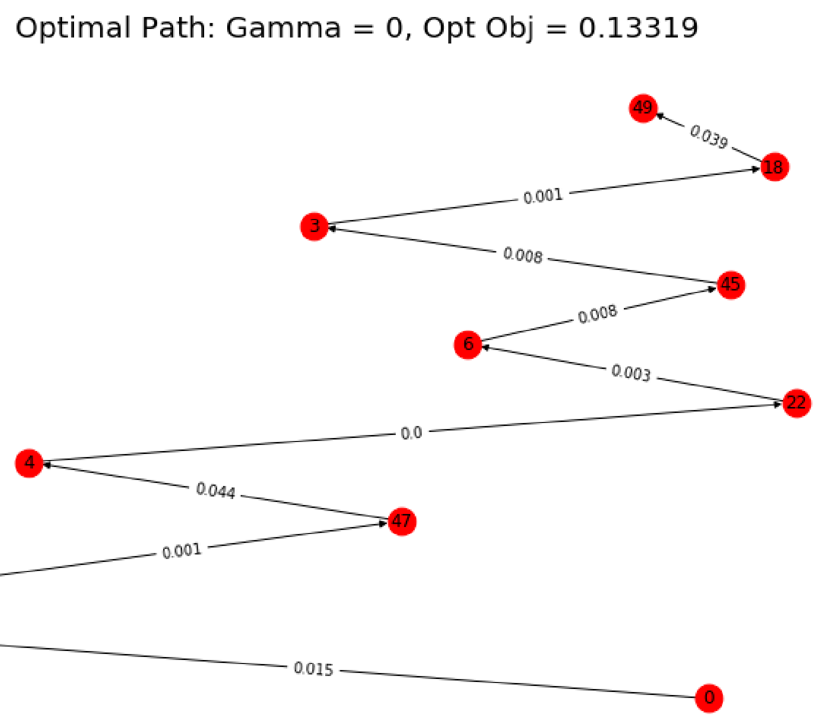
```
  Objective range  [1e-05, 4e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]

Loaded MIP start with objective 1.8923

Presolve removed 91 rows and 178 columns
Presolve time: 0.01s
Presolved: 2170 rows, 4241 columns, 10600 nonzeros
Variable types: 2121 continuous, 2120 integer (2120 binary)

Root relaxation: objective 9.929632e-01, 67 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.99296    0   50    1.89230    0.99296  47.5%     -    0s
H    0     0                       1.8339424    0.99296  45.9%     -    0s
     0     0    1.41127    0   38    1.83394    1.41127  23.0%     -    0s
     0     0    1.41127    0   31    1.83394    1.41127  23.0%     -    0s
     0     0    1.49119    0   31    1.83394    1.49119  18.7%     -    0s
     0     0    1.49119    0   29    1.83394    1.49119  18.7%     -    0s
     0     0    1.64322    0   27    1.83394    1.64322  10.4%     -    0s
     0     0    1.64322    0   16    1.83394    1.64322  10.4%     -    0s
     0     0    1.72029    0   21    1.83394    1.72029  6.20%     -    0s
     0     0    1.72029    0   11    1.83394    1.72029  6.20%     -    0s
     0     0    1.80893    0    3    1.83394    1.80893  1.36%     -    0s
     0     0    1.80893    0    1    1.83394    1.80893  1.36%     -    0s
     0     0     cutoff    0         1.83394    1.83394  0.00%     -    0s

Cutting planes:
  Gomory: 1
  MIR: 1

Explored 1 nodes (433 simplex iterations) in 0.15 seconds
Thread count was 8 (of 8 available processors)

Solution count 2: 1.83394 1.8923

Optimal solution found (tolerance 1.00e-04)
Best objective 1.833942446000e+00, best bound 1.833942446000e+00, gap 0.0000%
```

In [11]:
```python
# print optimal values and paths, plot network
for o in output:
    print('\nFor Gamma: '+str(o[0]))
    print('Path:')
    print(o[1])
```

```
    print('Cost of Movement (Objective):')
    print(o[2])
    networkCompletePlot(o,maxNodes)
    networkPathPlot(o,maxNodes,c)
```

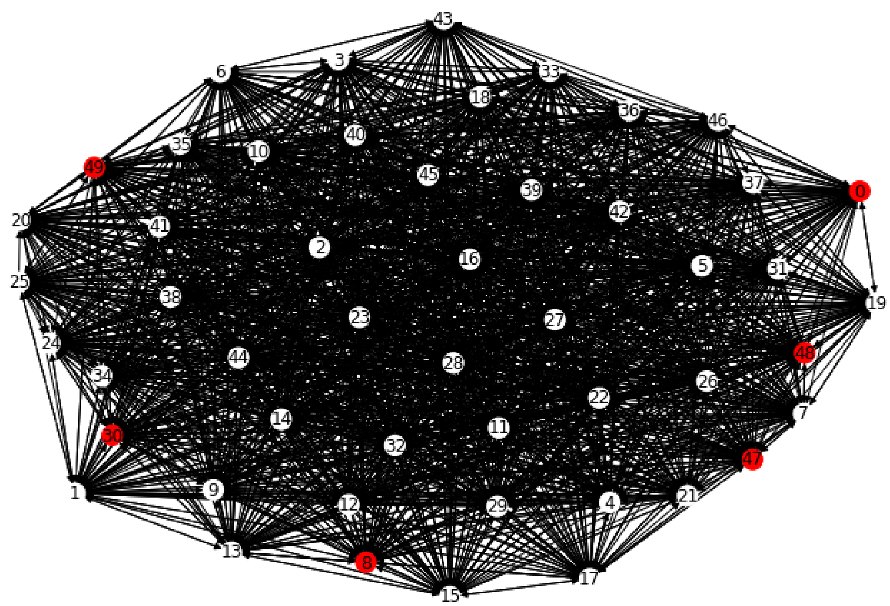
```
For Gamma: 0.0
Path:
[(0, 23), (23, 21), (21, 47), (47, 4), (4, 22), (22, 6), (6, 45), (45, 3), (3, 18), (18, 49)]
Cost of Movement (Objective):
0.133194359
```



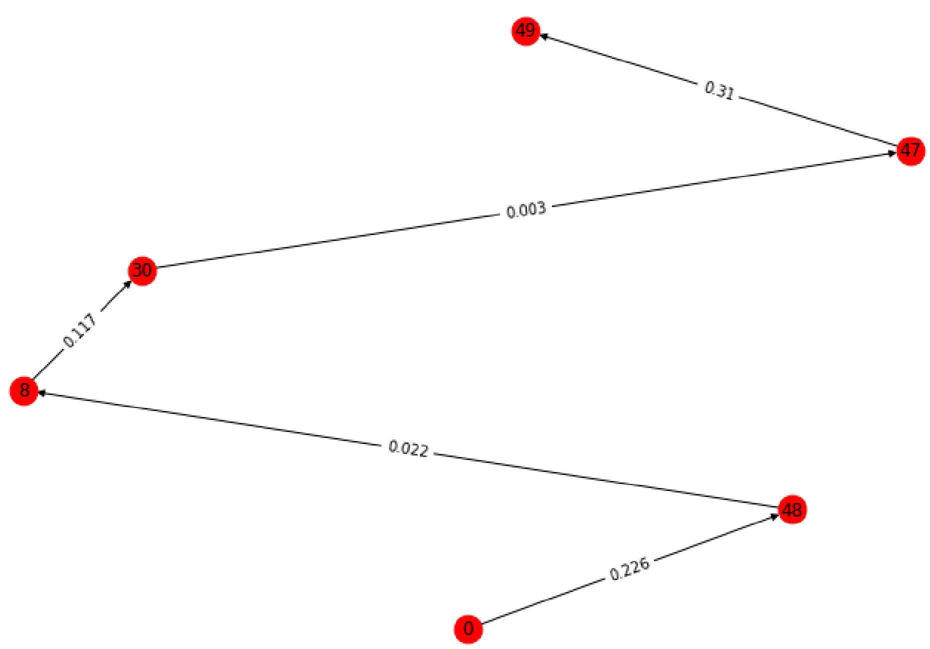



```
For Gamma: 1.0
Path:
[(0, 48), (48, 8), (8, 30), (30, 47), (47, 49)]
Cost of Movement (Objective):
1.299208152
```

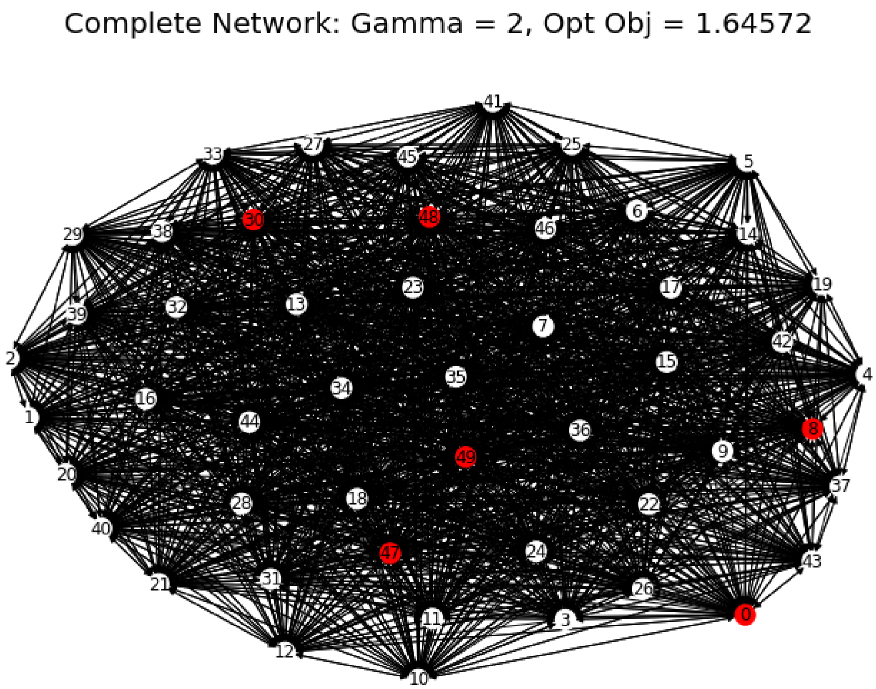
Complete Network: Gamma = 1, Opt Obj = 1.29921

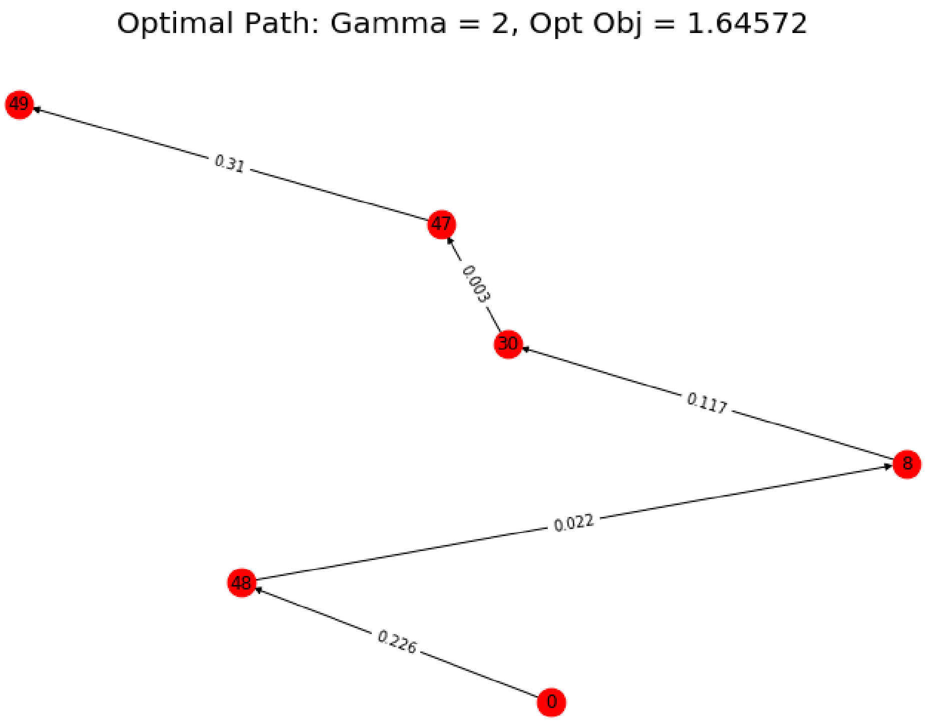


Optimal Path: Gamma = 1, Opt Obj = 1.29921



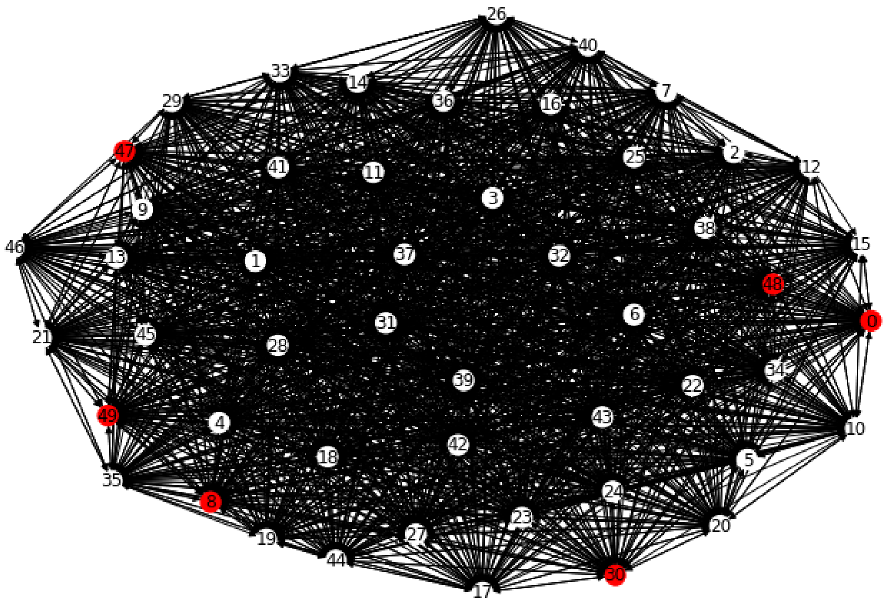
```
For Gamma: 2.0
Path:
[(0, 48), (48, 8), (8, 30), (30, 47), (47, 49)]
Cost of Movement (Objective):
1.6457190550000003
```



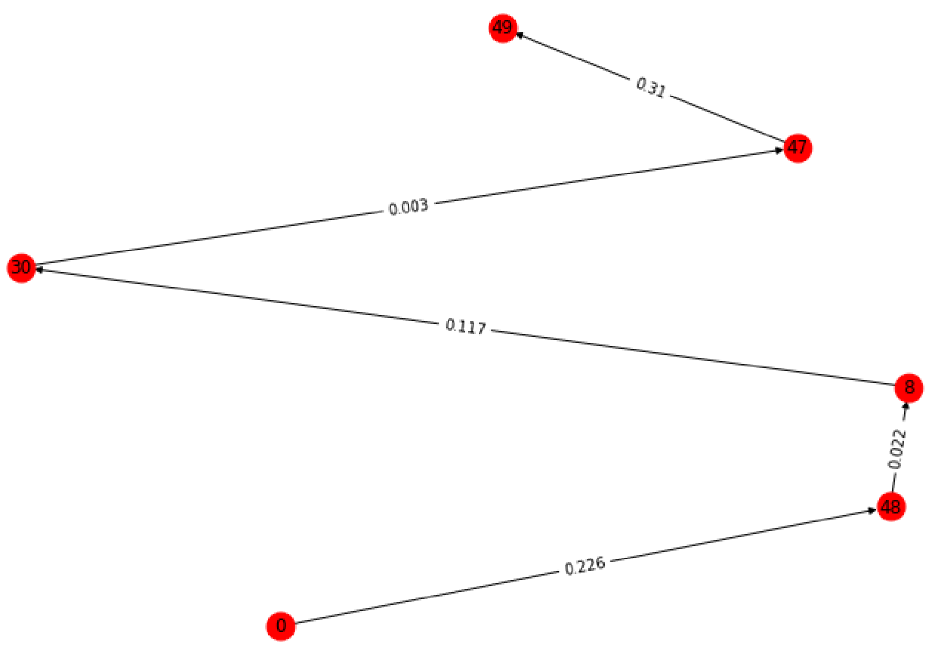



```
For Gamma: 3.0
Path:
[(0, 48), (48, 8), (8, 30), (30, 47), (47, 49)]
Cost of Movement (Objective):
1.7690088749999995
```

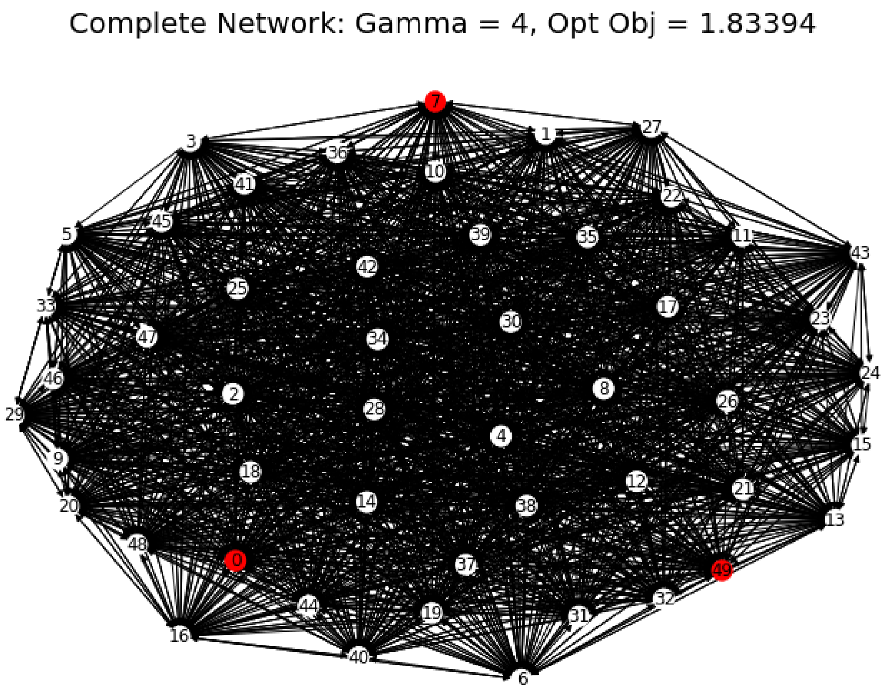
Complete Network: Gamma = 3, Opt Obj = 1.76901



Optimal Path: Gamma = 3, Opt Obj = 1.76901



```
For Gamma: 4.0
Path:
[(0, 7), (7, 49)]
Cost of Movement (Objective):
1.833942446
```



Optimal Path: Gamma = 4, Opt Obj = 1.83394