shortPath nb updating values

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_values.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 = int(maxNodes)

        # 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[88,57] + move[88,55] + move[88,64] + move[88,30] + move[88,49] + move[88,6
```

```
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[90,42]>  >=  <gurobi.LinExpr: 3.88801471279467 move[90,42] + -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 = []
            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 6061 rows, 11939 columns and 29729 nonzeros
Variable types: 5970 continuous, 5969 integer (5969 binary)
Coefficient statistics:
  Matrix range     [9e-05, 5e+00]
  Objective range  [2e-04, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 0.9533653
Presolve removed 5971 rows and 6294 columns
Presolve time: 0.03s
Presolved: 90 rows, 5645 columns, 11180 nonzeros
Variable types: 0 continuous, 5645 integer (5645 binary)

Root relaxation: objective 9.273644e-02, 24 iterations, 0.00 seconds

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

*    0     0               0       0.0927364    0.09274  0.00%     -    0s

Explored 0 nodes (24 simplex iterations) in 0.06 seconds
Thread count was 8 (of 8 available processors)

Solution count 2: 0.0927364 0.953365

Optimal solution found (tolerance 1.00e-04)
Best objective 9.273644178749e-02, best bound 9.273644178749e-02, gap 0.0000%
Optimize a model with 6061 rows, 11939 columns and 29729 nonzeros
Variable types: 5970 continuous, 5969 integer (5969 binary)
Coefficient statistics:
  Matrix range     [9e-05, 5e+00]
  Objective range  [2e-04, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
```

```
Loaded MIP start with objective 4.09274

Presolve removed 199 rows and 1126 columns
Presolve time: 0.18s
Presolved: 5862 rows, 10813 columns, 56035 nonzeros
Variable types: 5137 continuous, 5676 integer (5676 binary)

Root relaxation: objective 7.614694e-01, 108 iterations, 0.02 seconds

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

     0     0    0.76147    0   48    4.09274    0.76147  81.4%     -    0s
H    0     0                        2.4704548    0.76147  69.2%     -    0s
H    0     0                        2.3521988    0.76147  67.6%     -    0s
H    0     0                        1.7104797    0.76147  55.5%     -    0s
H    0     0                        1.5548262    0.76157  51.0%     -    0s
     0     0    0.81954    0   28    1.55483    0.81954  47.3%     -    0s
     0     0    0.93375    0   25    1.55483    0.93375  39.9%     -    0s
H    0     0                        1.5058401    0.93375  38.0%     -    0s
     0     0    0.95309    0   22    1.50584    0.95309  36.7%     -    0s
H    0     0                        1.4460993    0.95309  34.1%     -    0s
     0     0    0.95709    0   27    1.44610    0.95709  33.8%     -    0s
H    0     0                        1.2989292    0.95709  26.3%     -    0s
H    0     0                        1.2815288    0.95709  25.3%     -    0s
     0     0    1.02650    0   29    1.28153    1.02650  19.9%     -    0s
     0     0    1.02650    0   24    1.28153    1.02650  19.9%     -    0s
     0     0    1.05961    0   25    1.28153    1.05961  17.3%     -    0s
     0     0    1.09182    0   25    1.28153    1.09182  14.8%     -    0s
     0     0    1.23169    0   24    1.28153    1.23169  3.89%     -    0s
     0     0    1.23169    0   18    1.28153    1.23169  3.89%     -    0s
     0     0    1.23169    0   29    1.28153    1.23169  3.89%     -    0s
     0     0    1.23169    0   14    1.28153    1.23169  3.89%     -    0s
     0     0    1.24060    0   16    1.28153    1.24060  3.19%     -    0s
     0     0    1.27684    0    6    1.28153    1.27684  0.37%     -    0s
     0     0     cutoff    0         1.28153    1.28153  0.00%     -    0s

Cutting planes:
  Gomory: 1
  MIR: 2

Explored 1 nodes (427 simplex iterations) in 0.81 seconds
Thread count was 8 (of 8 available processors)

Solution count 9: 1.28153 1.29893 1.4461 ... 4.09274

Optimal solution found (tolerance 1.00e-04)
Best objective 1.281528811296e+00, best bound 1.281528811296e+00, gap 0.0000%
```

```
Optimize a model with 6061 rows, 11939 columns and 29729 nonzeros
Variable types: 5970 continuous, 5969 integer (5969 binary)
Coefficient statistics:
  Matrix range     [9e-05, 5e+00]
  Objective range  [2e-04, 2e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]

Loaded MIP start with objective 1.64177

Presolve removed 68 rows and 132 columns
Presolve time: 0.03s
Presolved: 5993 rows, 11807 columns, 29400 nonzeros
Variable types: 5904 continuous, 5903 integer (5903 binary)

Root relaxation: objective 8.678581e-01, 105 iterations, 0.00 seconds

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

     0     0    0.86786    0   56    1.64177    0.86786  47.1%     -    0s
     0     0    0.98548    0   45    1.64177    0.98548  40.0%     -    0s
     0     0    0.98548    0   39    1.64177    0.98548  40.0%     -    0s
H    0     0                        1.5697475    0.98548  37.2%     -    0s
     0     0    1.02430    0   34    1.56975    1.02430  34.7%     -    0s
     0     0    1.02430    0   31    1.56975    1.02430  34.7%     -    0s
     0     0    1.05457    0   37    1.56975    1.05457  32.8%     -    0s
     0     0    1.08872    0   32    1.56975    1.08872  30.6%     -    0s
     0     0    1.11275    0   33    1.56975    1.11275  29.1%     -    0s
     0     0    1.21582    0   29    1.56975    1.21582  22.5%     -    0s
     0     0    1.29376    0   21    1.56975    1.29376  17.6%     -    0s
     0     0    1.33737    0   14    1.56975    1.33737  14.8%     -    0s
     0     0    1.33928    0   12    1.56975    1.33928  14.7%     -    0s
     0     0    1.38501    0    4    1.56975    1.38501  11.8%     -    0s
     0     0    1.51625    0    7    1.56975    1.51625  3.41%     -    0s
     0     0     cutoff    0          1.56975    1.56975  0.00%     -    0s

Cutting planes:
  Gomory: 1
  MIR: 1

Explored 1 nodes (521 simplex iterations) in 0.41 seconds
Thread count was 8 (of 8 available processors)

Solution count 2: 1.56975 1.64177

Optimal solution found (tolerance 1.00e-04)
Best objective 1.569747518826e+00, best bound 1.569747518826e+00, gap 0.0000%
```

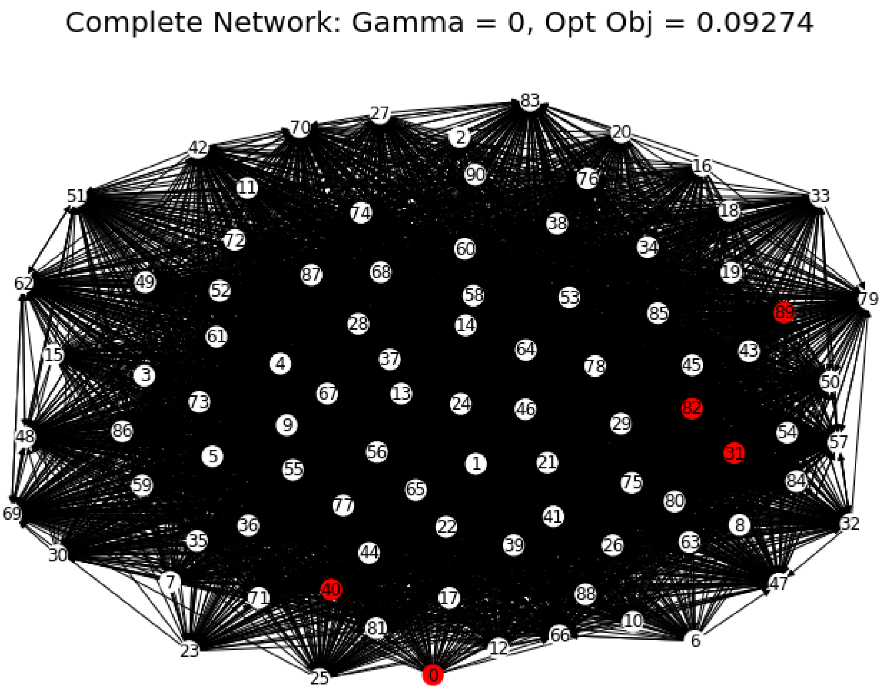
```
Optimize a model with 6061 rows, 11939 columns and 29729 nonzeros
Variable types: 5970 continuous, 5969 integer (5969 binary)
Coefficient statistics:
  Matrix range     [9e-05, 5e+00]
  Objective range  [2e-04, 3e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]

Loaded MIP start with objective 1.84057

Presolve removed 68 rows and 132 columns
Presolve time: 0.03s
Presolved: 5993 rows, 11807 columns, 29400 nonzeros
Variable types: 5904 continuous, 5903 integer (5903 binary)

Root relaxation: objective 9.211145e-01, 133 iterations, 0.01 seconds

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

     0     0    0.92111    0   69    1.84057    0.92111  50.0%     -    0s
     0     0    1.15557    0   34    1.84057    1.15557  37.2%     -    0s
     0     0    1.15557    0   47    1.84057    1.15557  37.2%     -    0s
     0     0    1.24425    0   28    1.84057    1.24425  32.4%     -    0s
     0     0    1.27932    0   28    1.84057    1.27932  30.5%     -    0s
     0     0    1.30883    0   26    1.84057    1.30883  28.9%     -    0s
     0     0    1.30883    0   41    1.84057    1.30883  28.9%     -    0s
     0     0    1.30883    0   23    1.84057    1.30883  28.9%     -    0s
     0     0    1.32785    0   35    1.84057    1.32785  27.9%     -    0s
H    0     0                          1.8005042    1.32785  26.3%     -    0s
     0     0    1.35446    0   28    1.80050    1.35446  24.8%     -    0s
     0     0    1.35446    0   35    1.80050    1.35446  24.8%     -    0s
     0     0    1.35446    0   29    1.80050    1.35446  24.8%     -    0s
     0     0    1.35446    0   28    1.80050    1.35446  24.8%     -    0s
     0     0    1.36510    0   27    1.80050    1.36510  24.2%     -    0s
     0     0    1.36524    0   25    1.80050    1.36524  24.2%     -    0s
     0     0    1.50659    0   25    1.80050    1.50659  16.3%     -    0s
     0     0    1.72042    0   25    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   28    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   28    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   24    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   20    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   19    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   18    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   16    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   15    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   16    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   16    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   16    1.80050    1.72042  4.45%     -    0s
     0     0    1.72042    0   14    1.80050    1.72042  4.45%     -    0s
     0     0    1.72883    0    8    1.80050    1.72883  3.98%     -    0s
     0     0    1.73773    0    7    1.80050    1.73773  3.49%     -    0s
     0     0    1.76343    0    1    1.80050    1.76343  2.06%     -    0s

Cutting planes:
  MIR: 1

Explored 1 nodes (851 simplex iterations) in 0.49 seconds
Thread count was 8 (of 8 available processors)

Solution count 2: 1.8005 1.84057

Optimal solution found (tolerance 1.00e-04)
Best objective 1.800504231753e+00, best bound 1.800504231753e+00, gap 0.0000%
Optimize a model with 6061 rows, 11939 columns and 29729 nonzeros
Variable types: 5970 continuous, 5969 integer (5969 binary)
Coefficient statistics:
  Matrix range     [9e-05, 5e+00]
  Objective range  [2e-04, 4e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]

Loaded MIP start with objective 2.03126

Presolve removed 68 rows and 132 columns
Presolve time: 0.03s
Presolved: 5993 rows, 11807 columns, 29400 nonzeros
Variable types: 5904 continuous, 5903 integer (5903 binary)

Root relaxation: objective 9.600741e-01, 131 iterations, 0.00 seconds

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

     0     0    0.96007    0   70    2.03126    0.96007  52.7%     -    0s
H    0     0                       2.0194844    1.01923  49.5%     -    0s
     0     0    1.29705    0   42    2.01948    1.29705  35.8%     -    0s
     0     0    1.29705    0   54    2.01948    1.29705  35.8%     -    0s
     0     0    1.36126    0   42    2.01948    1.36126  32.6%     -    0s
     0     0    1.39308    0   38    2.01948    1.39308  31.0%     -    0s
     0     0    1.44811    0   27    2.01948    1.44811  28.3%     -    0s
     0     0    1.44811    0   46    2.01948    1.44811  28.3%     -    0s
     0     0    1.44811    0   36    2.01948    1.44811  28.3%     -    0s
     0     0    1.44811    0   27    2.01948    1.44811  28.3%     -    0s
     0     0    1.44811    0   24    2.01948    1.44811  28.3%     -    0s
     0     0    1.46207    0   27    2.01948    1.46207  27.6%     -    0s
```

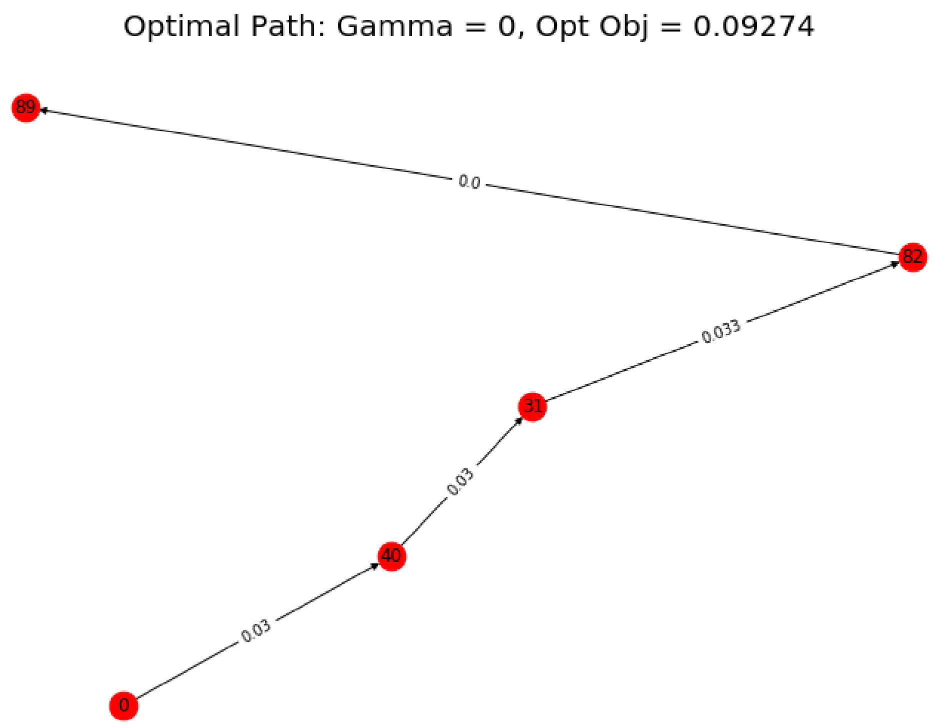
```
H    0     0                    1.9370508    1.46207  24.5%     -    0s
     0     0    1.47896    0   24    1.93705    1.47896  23.6%     -    0s
     0     0    1.49347    0   22    1.93705    1.49347  22.9%     -    0s
     0     0    1.50208    0   26    1.93705    1.50208  22.5%     -    0s
     0     0    1.50208    0   41    1.93705    1.50208  22.5%     -    0s
     0     0    1.50208    0   28    1.93705    1.50208  22.5%     -    0s
     0     0    1.50512    0   27    1.93705    1.50512  22.3%     -    0s
     0     0    1.52360    0   26    1.93705    1.52360  21.3%     -    0s
     0     0    1.53626    0   26    1.93705    1.53626  20.7%     -    0s
H    0     0                    1.9330727    1.53626  20.5%     -    0s
     0     0    1.92971    0   26    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   30    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   27    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   24    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   27    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   28    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   25    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   21    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   18    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   26    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   21    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   21    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   23    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   20    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   20    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   19    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   16    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   13    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   14    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   13    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0   14    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0    8    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0    4    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0    4    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0    4    1.93307    1.92971  0.17%     -    0s
     0     0    1.92971    0    4    1.93307    1.92971  0.17%     -    0s
     0     0 infeasible    0         1.93307    1.93307  0.00%     -    0s

Cutting planes:
  Gomory: 3
  MIR: 1

Explored 1 nodes (1223 simplex iterations) in 0.55 seconds
Thread count was 8 (of 8 available processors)

Solution count 4: 1.93307 1.93705 2.01948 2.03126

Optimal solution found (tolerance 1.00e-04)
```

```
Best objective 1.933072729796e+00, best bound 1.933072729796e+00, gap 0.0000%
```

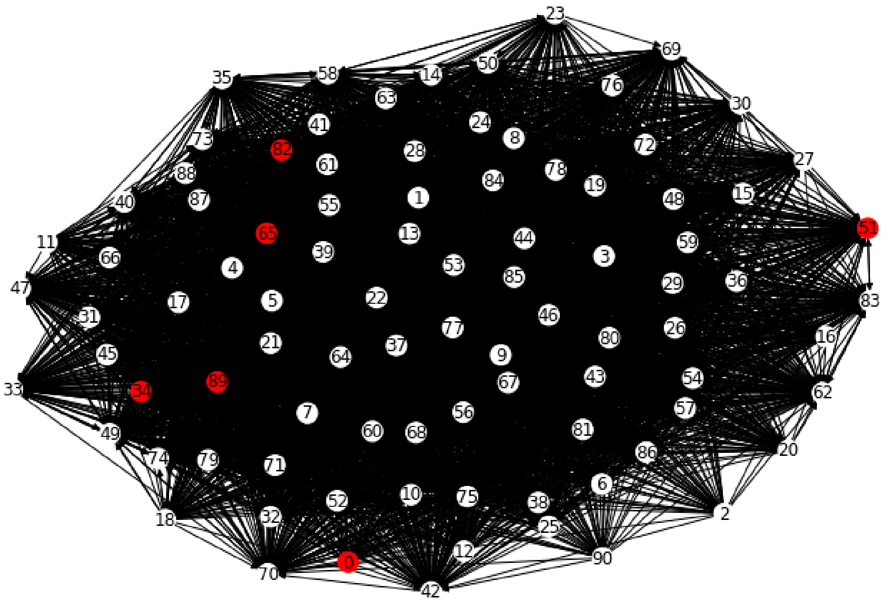
```
In [11]: # print optimal values and paths, plot network
         for o in output:
             print('\nFor Gamma: '+str(o[0]))
             print('Path:')
             print(o[1])
             print('Cost of Movement (Objective):')
             print(o[2])
             networkCompletePlot(o,maxNodes)
             networkPathPlot(o,maxNodes,c)
```

```
For Gamma: 0.0
Path:
[(0, 40), (40, 31), (31, 82), (82, 89)]
Cost of Movement (Objective):
0.09273644417874903
```

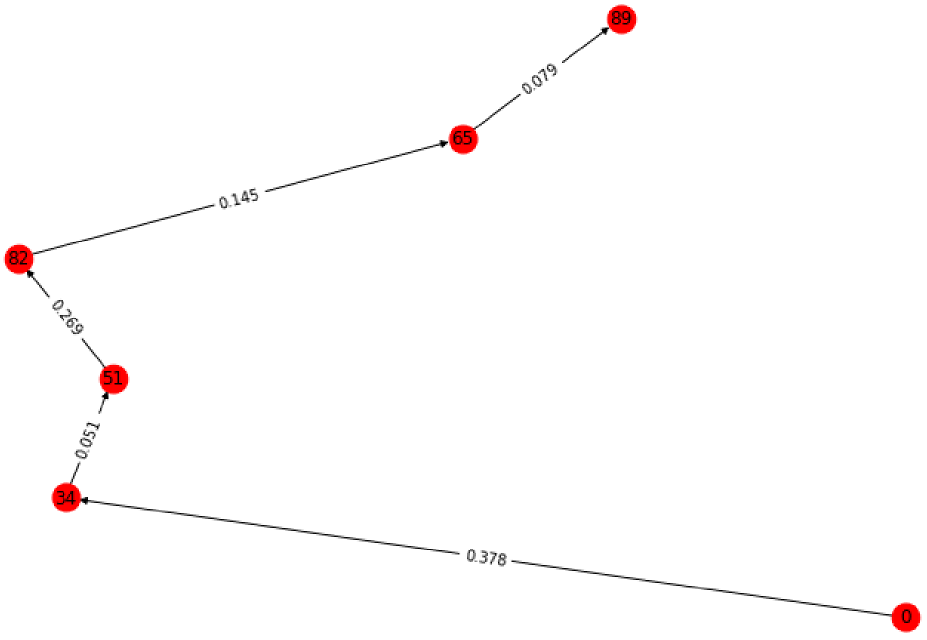



Optimal Path: Gamma = 0, Opt Obj = 0.09274

```
For Gamma: 1.0
Path:
[(0, 34), (34, 51), (51, 82), (82, 65), (65, 89)]
Cost of Movement (Objective):
1.2815288112962624
```



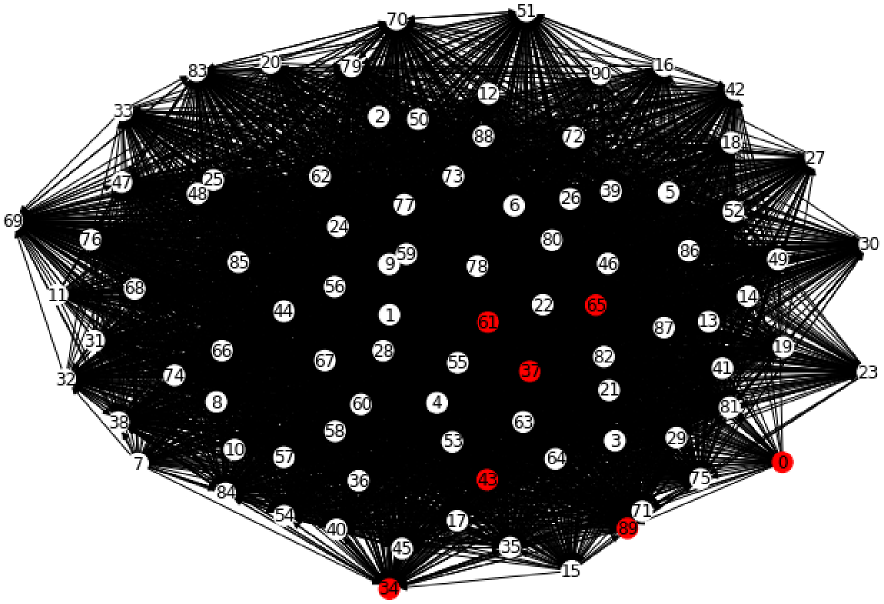
Complete Network: Gamma = 1, Opt Obj = 1.28153

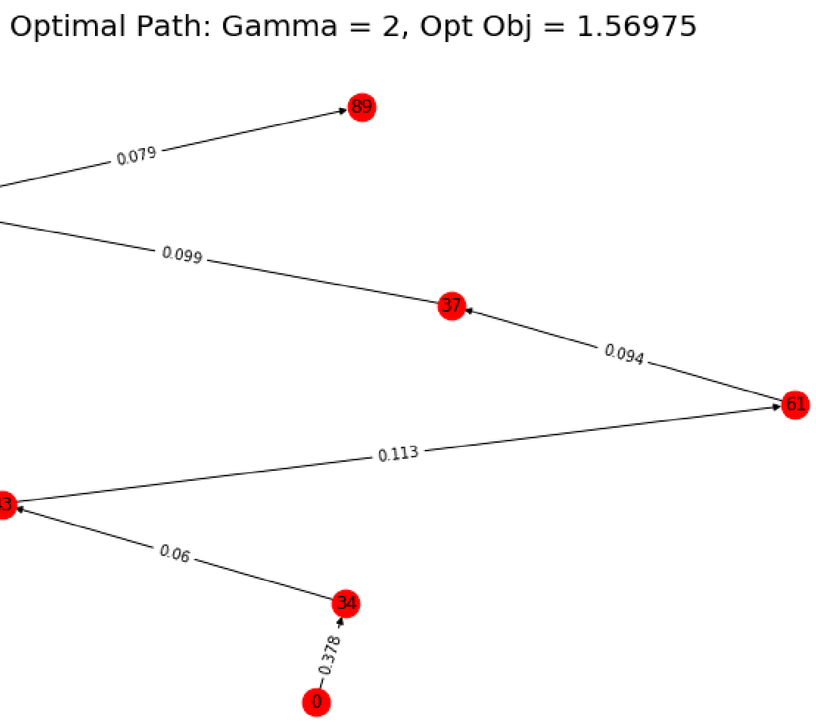


Optimal Path: Gamma = 1, Opt Obj = 1.28153

```
For Gamma: 2.0
Path:
[(0, 34), (34, 43), (43, 61), (61, 37), (37, 65), (65, 89)]
Cost of Movement (Objective):
1.5697475188262433
```

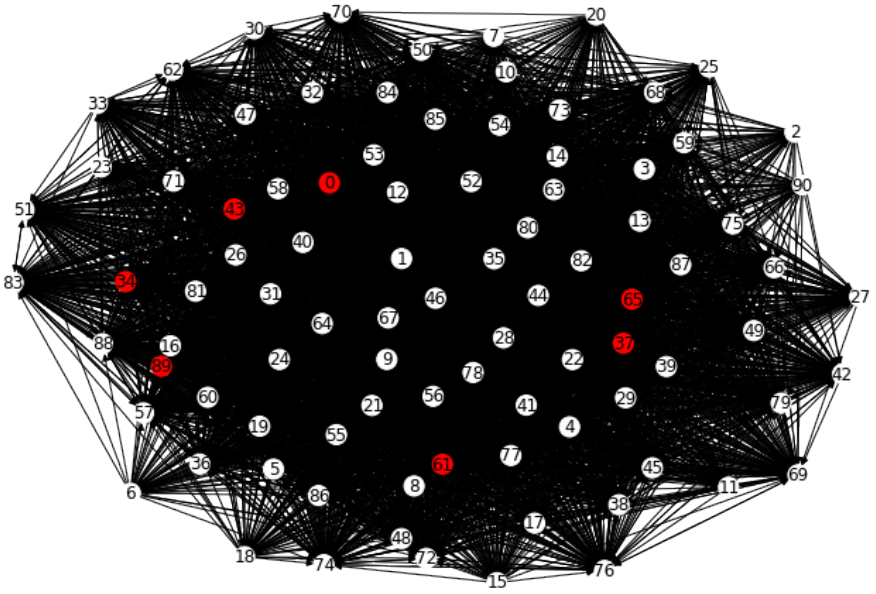
Complete Network: Gamma = 2, Opt Obj = 1.56975



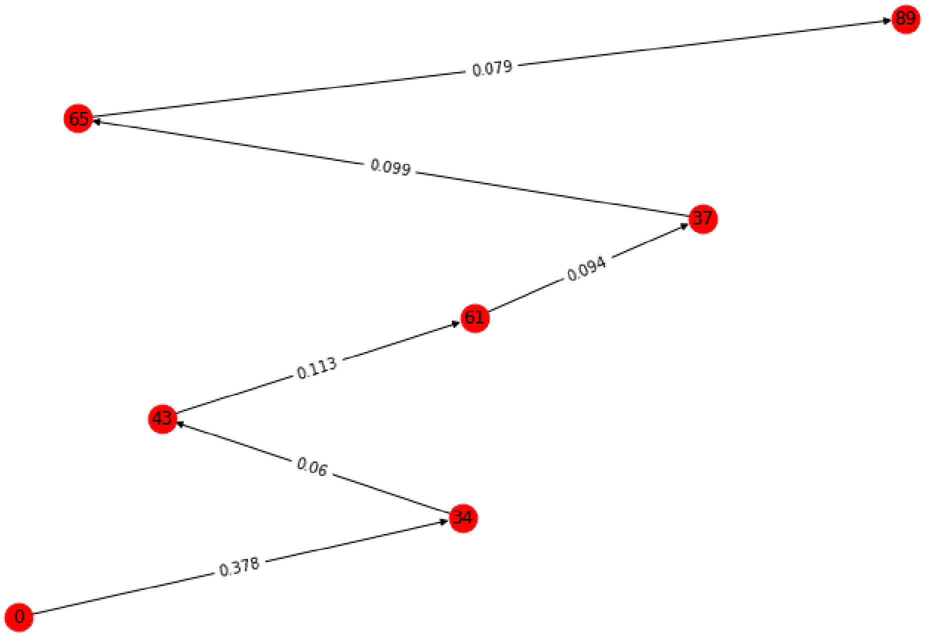


```
For Gamma: 3.0
Path:
[(0, 34), (34, 43), (43, 61), (61, 37), (37, 65), (65, 89)]
Cost of Movement (Objective):
1.8005042317534627
```

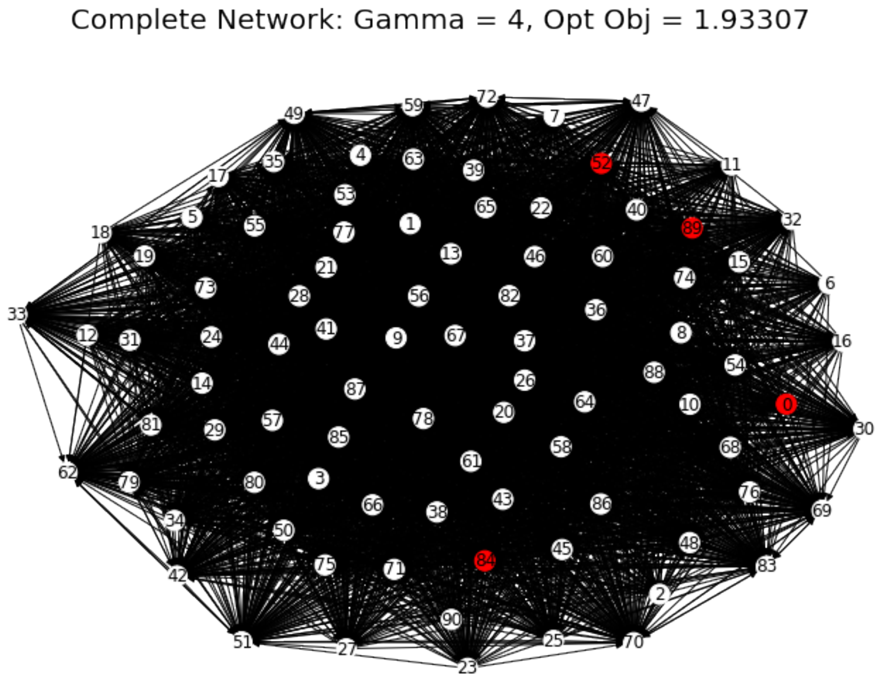
Complete Network: Gamma = 3, Opt Obj = 1.8005



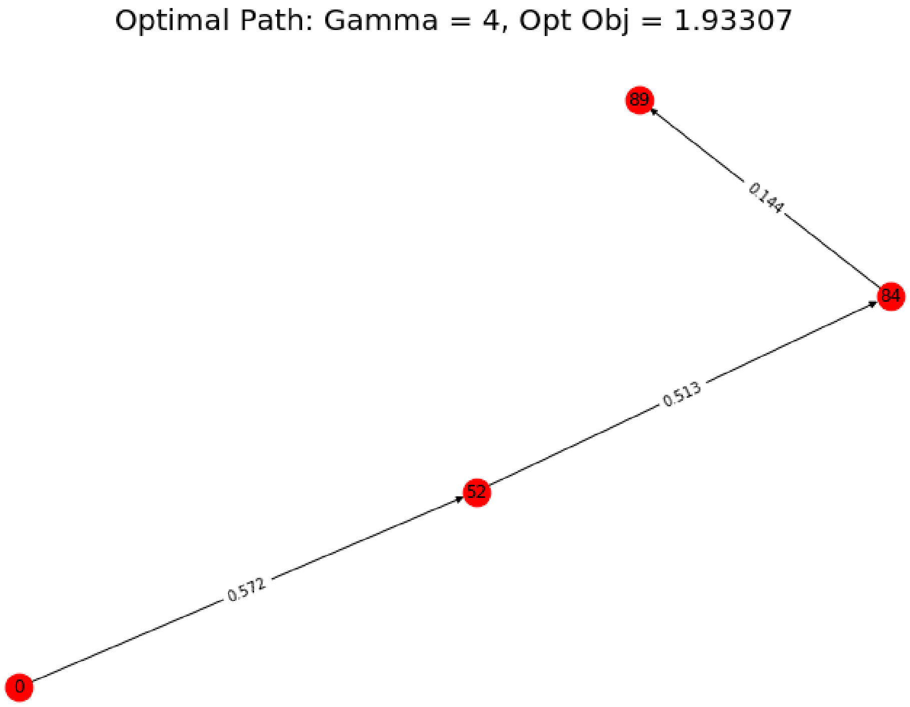
Optimal Path: Gamma = 3, Opt Obj = 1.8005



```
For Gamma: 4.0
Path:
[(0, 52), (52, 84), (84, 89)]
Cost of Movement (Objective):
1.9330727297958064
```



Complete Network: Gamma = 4, Opt Obj = 1.93307


Optimal Path: Gamma = 4, Opt Obj = 1.93307
89
0.144
84
0.513
52
0.572
0