Shanley Philip
CSCI 598D
Machine Learning

Discrete Event Simulation Practice

1. After modifying the san program to incorporate the following directed graph (Illustration 1) to determine the critical path. It can be seen by Illustration 3 that path 3 has the highest probability of being the critical path.

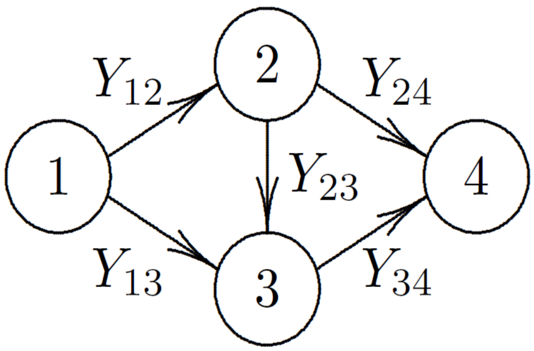

Illustration 1: Directed graph used for Practice 1

```
Path 1: -1--4-
Path 2: -2--5-
Path 3: -1--3--5-
```

Illustration 2: Possible paths calculated by the program

```
Critical path probabilities:
 -   1  -  -   2  -  -   3  -
 0.142000  0.142000  0.716000
```

Illustration 3: Probability that a given path is the critical path

2. Using the Monte Carlo method to train the function (Illustration 4) in an attempt to determine the ranges of the parameters *k* and *a* that minimizes the overshooting time greatly depends on the initial seed used.

$$y(t) = 1 - e^{-kt}cos(at)$$

Illustration 4: The response characteristic to low frequency input

It is not guaranteed that a longer execution time yields a better solution, only that a solution exists assuming that the function converges in the given time limit. Initially, the bounds of the the *k* and *a* parameters are [0,2000] and [-2000,2000] respectively. The following illustration (Illustration 5) shows the results after different seeds are used to initialize the program.

```
Enter seed (9 digits or less): 123456

Results:
After 123456 iterations...
k values        0       1550.07
a values        -921.841        1064.17
```

```
Enter seed (9 digits or less): 987654

Results:
After 987654 iterations...
k values        0       400.847
a values        -1      1118.94
```

```
Enter seed (9 digits or less): 58632

Results:
After 58632 iterations...
k values        0       635.853
a values        -1      1031.86
```

Illustration 5: The results after executing the simulation three times

3. The traveling salesperson problem involves a salesperson traveling to each city (excluding the salesperson's hometown) in a given area. The goal is to find the optimal path that minimizes travel time while maximizing the number of cities reached without visiting a previously visited city.

While the solution is easy to calculate when the number of cities is small, the solution becomes harder as the number of cities increases. For this exercise, it is assumed that each city has a path that can visit all other cities. The following illustrations show the optimal path that the salesperson should follow for the case where there are four cities to visit. Illustration 6a shows the path should the salesperson travel in a straight line and Illustration 6b shows the path assuming the salesperson can only travel rectilinearly.

```
Total number of paths: 6

Locations:
City #0 at (0.05, 0.13)
City #1 at (0.71, 0.30)
City #2 at (0.86, 0.14)
City #3 at (0.66, 0.60)

Path 0 : 0 3 1 2
Path 1 : 0 2 1 3
Path 2 : 0 3 2 -1
Path 3 : 0 1 -1 2
Path 4 : 0 2 3 1
Path 5 : 0 1 2 3

To travel 4 cities:
Time needed: 0.678
Optimal path:   0 1 2

Distance matrix
0.0000  0.6776  0.8056  0.7696
0.6776  0.0000  0.2254  0.3001
0.8056  0.2254  0.0000  0.5041
0.7696  0.3001  0.5041  0.0000
```

Illustration 6a: The optimal path that the saleperson should travel if the salesperson can travel in a straight line

```
Total number of paths: 6

Locations:
City #0 at (0.05, 0.13)
City #1 at (0.71, 0.30)
City #2 at (0.86, 0.14)
City #3 at (0.66, 0.60)

Path 0 : 0 2 1 3
Path 1 : 0 1 3 -1
Path 2 : 0 1 2 3
Path 3 : 0 3 1 2
Path 4 : 0 1 3 -1
Path 5 : 0 2 3 1

To travel 4 cities:
Time needed: 0.809
Optimal path:   0 3 1 2

Distance matrix
0.0000  0.8273  0.8093  1.0791
-0.8273 0.0000  -0.0180 0.2518
-0.8093 0.0180  0.0000  0.2697
-1.0791 -0.2518 -0.2697 0.0000
```

Illustration 6b: The optimal path that the saleperson should travel if the salesperson travels rectilinearly