ALGORITHMIC ASPECTS

OF TELECOMMUNICATION

NETWORKS

PROJECT – 1

“AN APPLICATION TO NETWORK DESIGN”

**Submitted By –**

Navaneeth Venugopala Rao

2021223814

[nbv140130@utdallas.edu](mailto:nbv140130@utdallas.edu)

1. Abstract

**CONTENTS**

2. Flowchart

3. Graphical Observations

4. Outputs of the Algorithm

5. Network Topology Generated

6. Dijkstra’s Algorithm

7. Fast Solution Method

8. Source Code

9. Modules of Source Code

10. References

**ABSTRACT**

 Objective: Implement basic network design module that generates a network topology with capacities assigned to links, according to the model, An Application to Network Design, using shortest path based fast solution. The implementation need to take the input of number of nodes and traffic demand values (bij) between pair of nodes and unit cost values of potential links (aij).

Input:

• Number of nodes (36)

• Traffic demand values (bij)

• Unit cost values of potential links (aij) which are computed according to the problem statement

Output:

• Run through k = 3 to 15, the implementation generates a network topology according to the model cited in “An Application to Network Design”

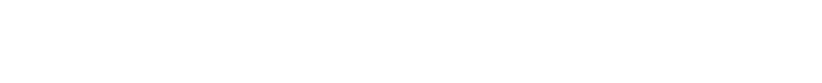
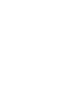
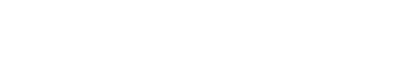
 Makes use of the Djikstra’s algorithm that computes shortest

paths between each pair of nodes for each value of k.

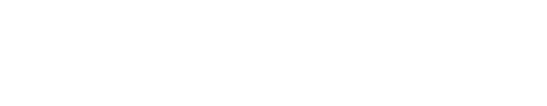
 For each value of k, MaxCost ie. sum total cost of all edges that form shortest paths in the topology is computed along with the density.

 Total cost and Density are analyzed and plotted for varying values of k

**FLOWCHA T**



**R**



START

Take input as randomly generated values of demand for each pair of

nodes b[i][j]

Repeat for all values of k

RepeaAt for all pairs of nodes

Compute shortest path between pair of nodes by

using Djikstra’s algorithm



Compute minimum cost between pair of nodes

C A D E

B

B

C B A D E



B

Increment total cost Ma ost

C

xC

Compute Ma ost

xC

Compute Density

Output k, MaxCost, De sity

n

STOP



**GRAPHICAL OBSERVATIONS**

 Maximum Cost(MaxCost) versus Maximum Outgoing

Degree :

8000

MaxCost(Zopt)

7000

6000

5000

4000

3000

2000

1000

0

3 4 5 6 7 8 9 10 11 12 13 14 15 16

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

K

 **Explanation** : As the maximum outgoing degree for each node is k increases, the optimum cost of the network attributed by all the links forming the shortest paths(MaxCost) should decrease ( as exhibited by the graph) because with increasing k, more routes are discovered to other nodes and hence more shorter paths are found out. Hence their sum MaxCost should obviously decrease.

 Density versus Maximum Outgoing Degree(k) :

0.5

Density

0.45

0.4

0.35

0.3

0.25

0.2

0.15

0.1

0.05

0

3 4 5 6 7 8 9 10 11 12 13 14 15 16

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

K

 **Explanation**: As the maximum outgoing degree for each node (k) increases, the density of the network i.e. the number of links forming the shortest paths to the total number of links possible should also increase ( as exhibited by the graph) because with increasing k, more routes are discovered to other nodes , thus cluttering the network more. Hence the density should obviously increase.

**OUTPUTS OF THE ALGORITHM**

0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0

0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0

0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1

0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1

0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0

0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0

0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0

0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1

0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Count is 106 NODE =36

k = 3

MaxCost(Zopt) = 4478

Density = 0.08412699

0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0

0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0

0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 1 0

0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0

0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0

0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1

0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0

0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1

0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 1 0 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1

0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0

0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 1

0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0

0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0

0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0

0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0

0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0

0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0

0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0

0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1

0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0

0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0

0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1

0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0

Count is 225 NODE =36

k = 4

MaxCost(Zopt) = 5278

Density = 0.17857143

0 0 0 0 1 0 1 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0

0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 1

0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0

0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0

0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 1 0 0 0

0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0

0 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0

0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0

0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 1 1 0 0 0 0 0 0

0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0

0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1

0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1

0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1

0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0

0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0

0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 1 1 1 1 0

0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0

0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 1

0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1

0 1 1 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0

0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 1 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1

0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1

0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0

0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0

0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 1 0 0 1

0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1

0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 1 0 1 0 1 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1

0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0

Count is 290 NODE =36

k = 5

MaxCost(Zopt) = 4654

Density = 0.23015873

0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 1 0

0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1

0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0

0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0

0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 1 0 1 1 0 1

0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0

0 1 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0

0 1 0 0 0 1 0 0 1 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0

0 0 1 1 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1 0

0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 1 1 0 1 0 0 0 0

0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0

0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 0 0 0

0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 1

0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 1

0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 1 1 1 0 0 0 0 0 1 0

0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 0 0 0 0 0 1 1 0 0

0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0

0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 1 0 1

0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0

0 0 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0

0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1

0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0 1 1 0 1

0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1

0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0

0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0

0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 1 1

0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 1 0 0

0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1

0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 0 1 0 0 0

0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1

0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0

0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0

0 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 1 0 0 0 0

Count is 352 NODE =36

k = 6

MaxCost(Zopt) = 4170

Density = 0.2793651

0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 1 1 0

0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0

0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0

0 0 1 0 0 0 0 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0

0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 1 1 1 0 1

0 0 1 0 0 0 0 1 1 1 0 0 1 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0

0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 1 0 1 0 0 0

0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0

0 1 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 1

0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 1 1 0 0 0 0

0 0 0 0 1 0 1 0 0 0 0 1 1 0 0 1 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0

0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0

0 1 0 0 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 1 0 0

0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1

0 1 0 1 0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 0 0 0 0 0

0 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0

0 0 0 0 1 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0 1 1 1 1 0 0

0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 1 0 1 0 0 1 0 1 0 1 0 0 1 0 0 0

0 1 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 1

0 1 0 0 1 0 1 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0

0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 1 0 0 0 1 0

0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0

0 0 0 1 1 1 0 1 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0

0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 1 1 0 0 0 0 1 0 0 1 0 1

0 0 0 0 0 0 0 1 1 0 1 0 1 0 1 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 1

0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 1 0 0 0

0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 1 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0

0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 1 0 0 0

0 0 1 0 1 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1

0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1

0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1

0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 1 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0

0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 1 0 0 1 0 0 1 0 0 0 1 0

0 0 0 1 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0

0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 1 1 0 0 0 0 1 0 1 0 0 1 0 0

0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0

Count is 396 NODE =36

k = 7

MaxCost(Zopt) = 3897

Density = 0.31428573

0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 1 0 1 1 0 0 1 1 0 1 1 0

0 0 0 0 1 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0

0 0 0 0 1 1 1 0 0 0 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 0 0

0 0 0 0 0 0 1 1 1 0 1 1 1 0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0

0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 1 1 1 0 1 1

0 0 1 0 0 0 1 0 1 1 0 0 1 0 0 1 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 1 1 0 1 0

0 1 1 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0

0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 0

0 1 0 0 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 0 0 1

0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 0 1 1 1 1 0 0 1 0

0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 1 1 0 0

0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 0 1 0 0 0 1 0 0 0 1 1 1 0 0 0

0 1 0 0 0 0 0 1 1 0 1 0 0 0 1 0 0 1 1 1 1 0 0 1 1 0 0 1 0 0 1 0 0 1 0 1

0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 1 0 1 0 0 1 1 0 0 0

0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0 0 0 1

0 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1

0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1

0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 1 1 1 1 1 0 1 0 1 0 0 0 0 0 1 0 0

0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0

0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 1 0 0 0

0 1 0 0 0 1 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0

0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 0 1 0 0

0 0 1 1 0 1 1 1 0 0 0 1 0 0 0 1 0 1 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0

0 1 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 1 1 1 0 0 1 1 0 0 1 0 0

0 1 0 0 0 1 0 0 1 0 1 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 1 0 1 0 1 0 0 0 1 0

0 1 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1 1 1 0 0 1 1 0 0 0

0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1

0 0 1 1 1 0 1 1 0 1 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0

0 0 1 0 1 0 0 1 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0

0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 1 0 1

0 0 1 1 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 1 1 1 0 0 1 0 0 0 0 0 1 0 0 1 0

0 0 0 0 0 0 0 1 1 0 1 1 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0

0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 0 1 1 1 0 1 0 0 0 1 0

0 1 1 0 1 0 0 1 0 1 1 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0

0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 1 1 1 0 1 0 0 0 0 1 1 0 0 0 1 0 0

0 1 0 0 0 1 1 1 1 0 0 0 0 0 1 0 1 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0

Count is 451 NODE =36

k = 8

MaxCost(Zopt) = 3685

Density = 0.3579365

0 0 0 1 0 1 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 1 0 0 0 1 0

0 0 0 1 1 1 0 1 1 0 0 0 0 1 1 0 1 1 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0

0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 0 1 0 0 1 0 1 0 1 1 1 0

0 0 1 0 0 0 1 0 1 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 1 1 1 1 1

0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 1

0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 1 0

0 1 1 1 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 1 0 1 0

0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 0

0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 0 1 1 0 1 1 0 0 0 0

0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 1 0 1 1 0 0 0 1 1 1

0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 1 1 0 0 1 0 1 1 0 1

0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 1 0 1 0

0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 1 0 0 0 1 1 1 0 1 0 0 1 1 0 0 1 1

0 1 0 0 1 1 0 1 0 0 1 0 0 0 1 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 1 0

0 1 0 0 0 0 1 0 1 1 0 0 1 1 0 1 1 0 1 0 1 0 0 1 0 1 0 0 0 0 1 1 0 0 0 1

0 0 1 1 0 0 1 0 0 0 1 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 1 0 1

0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0 1 1 0 0 0 1 0 0 0 1 1 0 0 1 1 0 1 1

0 0 0 1 0 0 0 0 0 1 1 0 1 1 1 1 0 0 0 1 1 1 0 1 0 0 1 0 0 0 0 1 1 1 0 0

0 1 0 0 0 1 0 1 0 1 1 1 0 0 1 1 0 0 0 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0

0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0 1 0 1 0 1 0 0 1 0 0 0 0

0 1 0 0 0 1 1 0 0 1 1 1 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 0

0 1 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 0 1 1 0

0 1 1 0 1 0 1 0 0 0 0 0 1 0 1 1 1 1 0 1 0 1 0 0 0 1 0 1 0 0 1 0 0 1 0 0

0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 1 1 1

0 1 0 0 0 1 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 1 1 1 1 1 0 1 0 1 0

0 0 1 1 0 1 1 1 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 1 1 0 0 1 1 0 0 1 0 0 0 1

0 0 0 1 1 1 0 1 0 1 1 0 0 1 0 0 0 0 0 1 1 0 1 0 1 1 0 1 0 1 0 0 0 0 1 1

0 1 0 1 1 0 1 1 0 1 1 0 0 0 1 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1

0 0 0 1 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 1 1 1 1 0 0 1 0 0 1 0 0

0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 1 0 1 0 0 1 0 0 1 0 0 0 1 0 1 0

0 0 0 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 0 1 0 1 0 1 0 1 0 0 1 0 0 0 0 1 1 0

0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 1 1 1 1 1 1 0 0 0 1 0

0 1 1 0 1 1 0 1 0 0 1 1 0 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0

0 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 0 1 0 0 1 0 0 0 1 0 1

0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 1 1 0 0 0 0 1 0 1 0 0

Count is 498 NODE =36

k = 9

MaxCost(Zopt) = 3531

Density = 0.3952381

0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 1 0 1 1 1 0 1 1 0 0 1 0

0 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 1 1

0 0 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 1 1 0 1 0 0 1 0 1 1 1 1 1 0

0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 1 1 1 0 0 0 0 0 1 0 1 1 1 0 1

0 0 1 0 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1

0 1 1 0 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 0 1 0 0 1 0 0 1 1 1 0 1 1 0 1 0

0 0 1 1 0 1 0 0 1 0 0 1 1 0 1 0 1 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 0

0 1 1 0 0 1 1 0 1 0 1 0 0 0 0 1 0 1 0 1 1 1 0 0 0 0 1 1 0 0 0 1 0 1 0 0

0 0 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 0 0

0 1 0 1 0 0 1 1 0 0 0 1 0 0 0 1 0 1 1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 1 0 1

0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 1 1 1 0 1 0 1 0 1 0 1 1 0 1 1 1

0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1 0 0 1 1 1 1 0 0 1 0

0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0

0 1 1 1 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 1

0 1 0 0 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1 1 1 1 1 1 0 0 0 1 1 1 0 0 1 0 1 0

0 1 0 1 0 1 1 0 0 1 1 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1

0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 1 0 0 1 1 1 0 1 1 0 1 1 0

0 1 0 0 1 0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 0 0 0

0 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1

0 0 0 0 1 1 0 1 0 0 1 0 1 1 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 0

0 1 0 1 0 1 1 0 0 1 0 0 1 1 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 0 0 0 0 1

0 1 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0

0 1 0 0 1 0 0 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 0

0 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0

0 0 0 0 1 0 1 0 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 0 0 1 1 0 1 0 0 1 0 0 1

0 0 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 1 1 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0

0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 0 1 0 1

0 1 0 1 1 0 0 1 0 1 1 1 0 0 0 1 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 1 1 0 0 1

0 1 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 1 1 0 0 0 0 1 0 1 0 1 1 0 0 0 1 1

0 0 0 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 1

0 1 0 0 1 0 1 0 0 0 1 1 0 1 0 0 0 1 1 1 0 0 1 0 1 0 0 0 1 0 0 0 1 0 1 1

0 0 0 1 0 0 1 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0

0 0 0 0 0 0 0 1 0 1 0 1 1 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 1 1 0 0 1 1 0

0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0 0 1 0 0 0 0 0

0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0 1

0 0 0 0 1 0 1 0 1 1 0 0 0 1 0 1 0 1 0 1 0 0 0 1 1 0 1 1 0 0 0 0 0 1 0 0

Count is 519 NODE =36

k = 10

MaxCost(Zopt) = 3359

Density = 0.41190475

0 0 1 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 1 0 0 1 1 1 1 0 1

0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 0 1 0 0 1 1 1 1 0 0 0 0 1 0 1 1

0 0 0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 1 1 0 0 1 1 0 1 1 1 0 1 1 1 1 0 1 0

0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 1 1 0 0 1 0 1 0 0 0 0 0 1

0 0 1 0 0 0 1 1 1 1 0 1 1 0 0 1 0 0 0 1 1 1 0 0 0 0 1 0 0 1 1 0 1 0 1 1

0 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 0 1 0 0 0 1 0 0 1 1 0 0 0 0 0 1 1 0

0 1 1 1 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0

0 0 1 0 0 1 1 0 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 1 0 0 0 0 1 1 1 0 1

0 0 0 1 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 0 0 1

0 1 0 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0

0 1 1 0 1 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 1 1 1 1 1 0

0 0 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 0 0

0 0 0 1 0 1 1 1 0 1 0 0 0 1 0 1 0 1 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0 1 1

0 0 1 1 1 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 1 1 0 1 0 0

0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 1 1 1 0 0 0 1 1 0 1 0 1 0 1 1 0 1 0 0 1 0

0 1 0 1 0 1 1 1 0 1 0 0 0 0 0 0 1 0 1 0 1 1 1 1 0 0 1 0 0 1 0 0 0 0 0 1

0 1 0 0 1 0 1 1 1 1 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1 0 1 1 0 0

0 1 0 0 1 1 1 1 1 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 0 0 1 0 0 1

0 0 0 1 0 0 0 1 1 0 0 1 1 1 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0 1 0 0 0 1 1

0 1 0 0 1 1 0 1 0 1 1 0 0 1 0 1 0 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 1 0

0 0 0 0 0 1 0 1 0 1 0 1 1 1 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 0 0 0 1

0 1 1 1 0 0 0 0 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 0 0

0 0 1 0 0 0 0 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1

0 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0

0 0 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 1 0 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0

0 1 1 0 1 0 1 0 1 1 0 1 0 0 1 0 0 1 1 1 0 1 1 1 1 0 0 1 0 0 0 1 0 0 0 1

0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 1 0 1 1 1 1 0 1 0 0 0 0 1 1 0 1 0 1 1 0

0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 0 1 1 1 0 1 0 0 1 1 0 0 0

0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 0 0 0

0 0 0 1 0 1 1 0 0 0 1 0 0 1 1 1 0 0 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0

0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 1 0 0 0 0 1 1 0 1 0

0 0 0 1 1 0 1 1 1 0 1 0 1 0 1 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 1

0 1 0 0 0 1 0 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 0 0

0 0 0 0 0 1 0 0 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 1 0 1 1 0 1 1 0 0 0 1

0 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 0 1 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0

0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 1 0 0

Count is 551 NODE =36

k = 11

MaxCost(Zopt) = 3260

Density = 0.43730158

0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0 1 1 1 1 1

0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 1 0 1 0 1 0 1 0 1 0

0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 0 1 1 0 1 0 1 1 0 1 1

0 0 0 0 1 1 0 0 1 1 1 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 1 0 0 1 1 1 1 0 1

0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 0 0 1 1 1 1 0 0 1 1

0 0 1 0 0 0 1 1 1 0 0 0 1 1 0 0 1 1 1 1 0 1 0 1 0 0 1 0 1 0 0 0 1 1 0 1

0 1 1 1 1 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 1 1 0 0

0 0 1 0 0 0 1 0 1 1 0 1 1 0 0 1 0 1 1 1 1 1 1 0 1 0 0 0 0 0 1 0 0 1 0 1

0 0 0 0 0 1 0 1 0 0 0 1 0 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0

0 0 0 1 1 0 1 0 0 0 1 0 1 0 1 1 1 0 0 0 1 0 1 0 0 1 1 0 0 1 1 0 1 1 0 1

0 1 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 0 1 0 0 0

0 0 1 0 0 1 1 1 1 1 1 0 0 0 0 1 1 0 1 1 1 1 1 0 1 1 0 1 1 0 0 0 1 0 0 0

0 1 1 1 0 0 0 1 1 1 1 0 0 1 0 1 0 1 0 1 1 1 1 1 0 1 0 1 0 0 1 0 0 0 1 0

0 0 1 0 1 1 0 0 1 1 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 1 0 1 0 1 1 0

0 0 0 0 1 0 1 0 1 0 0 1 0 1 0 1 1 1 1 0 1 0 1 1 1 0 1 0 1 0 1 1 0 1 1 0

0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 0 0 1 0

0 0 1 0 1 1 1 0 1 1 1 1 0 0 1 1 0 1 0 1 1 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0

0 0 0 0 1 1 1 0 1 0 1 1 0 0 1 1 0 0 0 1 0 1 1 1 0 1 0 0 1 1 0 0 0 0 1 1

0 0 0 1 0 1 0 1 1 0 0 1 1 1 1 0 1 0 0 1 0 0 1 1 1 1 1 1 1 0 0 0 0 1 0 0

0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0 0 1 1 0

0 0 1 0 1 0 1 0 1 1 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 0 0

0 1 1 1 0 0 1 0 1 0 0 1 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1

0 0 1 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 1

0 0 1 0 0 0 0 1 1 1 1 0 0 0 1 0 1 1 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 1 1 0

0 0 1 0 1 0 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 0 0 1 1 1 0 1 1

0 1 0 0 1 1 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 1 0 1 1 0 0 1 0 0 1 1 1 1 0 1

0 0 0 1 1 0 0 1 0 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0

0 0 1 0 0 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 0 0 0 0 1 1 0

0 0 0 0 1 1 0 0 0 1 0 1 0 1 0 0 1 1 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 1 0 1

0 0 0 1 0 1 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 0 1 0 0 0 1 1 0 0 1 1 0 0 1

0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 0 1 0 0

0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0

0 0 0 0 0 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 1 0 1 0 1

0 0 1 1 0 0 1 1 0 0 0 1 0 0 1 1 0 1 1 1 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0

0 0 1 1 0 0 1 0 1 0 1 1 0 1 1 1 0 1 0 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0

0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0

Count is 580 NODE =36

k = 12

MaxCost(Zopt) = 3187

Density = 0.46031746

0 0 0 1 0 1 1 1 1 0 0 1 1 0 0 1 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1

0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 1 0 0 1 1 0 0 1 1 1 0 1 0

0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 1 0 1 0 1 0 0 1 1 1 0 0 0

0 1 0 0 1 1 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 0 1

0 0 0 0 0 0 0 1 0 1 1 1 0 1 1 0 1 0 0 0 0 0 1 0 1 1 1 0 1 1 0 1 0 0 0 1

0 0 0 1 0 0 0 1 1 0 0 1 1 0 0 0 1 0 1 1 0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1

0 1 0 0 1 0 0 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0 1 0 1 0 1 0 1

0 0 1 0 1 1 1 0 1 0 0 1 1 1 0 1 0 1 1 1 0 1 1 0 0 0 0 1 1 0 1 0 1 1 0 1

0 0 0 0 0 1 1 1 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0 0 1

0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 1 0 1 0 0 1 0 1 1 0 0

0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 1 1 1 1 0 0 1 1 1 1 0 0 0 1 0 0 0 0 0 1

0 0 1 0 0 1 1 1 1 1 1 0 0 0 1 0 1 1 0 1 1 1 1 0 1 0 0 1 1 1 0 0 1 0 1 0

0 1 1 0 0 0 0 0 1 1 1 1 0 0 0 1 1 0 1 0 1 0 0 0 0 1 1 1 0 0 1 1 1 0 0 1

0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 0 0 0 1 1 1 0 1 0 1 1 0 0 0 1 1 1 1 0 1 0

0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 1 1 0 1 1 0

0 0 1 1 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 1 1 1 1 0 0 1 0

0 1 0 0 1 1 1 1 0 1 0 0 0 0 1 1 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1 1 0 0

0 1 1 0 0 1 1 0 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 0 1 0 1 1 1 1 0 0

0 1 0 1 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 1 0 1

0 0 1 1 1 0 0 0 0 0 0 1 1 0 1 1 1 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0

0 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0

0 0 0 1 0 0 1 0 1 0 1 1 0 1 0 1 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 1 1 1

0 1 0 0 0 1 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 1 0 1

0 1 0 1 1 0 0 1 0 1 0 0 0 0 1 1 0 0 0 1 1 1 0 0 1 0 1 0 0 0 1 1 0 0 0 1

0 0 1 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 1 0 0 1 0 1 1 1 0 0 0 1 0 0 0 0

0 1 0 0 1 1 0 1 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 0

0 0 0 1 1 0 0 1 0 0 1 0 1 0 1 0 0 0 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 0 0 1

0 0 1 0 0 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 0 0 0 0 1 1 0

0 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 1 0 1 0 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0

0 1 0 1 0 0 0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0

0 1 1 0 0 1 0 1 0 0 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 1 0

0 1 0 1 0 0 1 1 1 0 1 0 0 0 1 0 1 0 0 0 1 0 1 1 0 0 0 1 0 0 1 0 1 1 0 0

0 0 0 0 0 1 1 0 1 0 0 0 1 1 1 0 1 1 0 1 0 1 1 1 0 0 1 1 1 0 1 0 0 1 0 0

0 1 0 1 0 0 1 1 1 1 0 0 1 1 0 0 0 1 1 0 0 0 1 0 0 0 1 1 1 0 1 0 0 0 0 0

0 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 1 0 0 0

0 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 0 0

Count is 580 NODE =36

k = 13

MaxCost(Zopt) = 3130

Density = 0.46031746

0 1 0 1 1 0 0 1 0 1 0 0 1 1 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 1 0 1 1 0 1 0

0 0 1 0 1 0 1 1 1 1 0 0 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1 1 1 0 1 0 1 0 0 0

0 1 0 1 0 0 1 1 0 1 0 0 1 1 1 0 0 1 0 1 1 0 1 1 0 0 1 0 1 0 1 1 0 1 0 1

0 0 0 0 1 1 1 0 1 1 1 1 0 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0 0 1 0 0 0 1 0 0

0 0 0 1 0 1 1 1 0 0 1 0 1 1 1 1 0 1 0 1 1 0 0 0 1 0 1 0 1 1 0 1 1 0 0 0

0 0 0 1 1 0 0 1 0 0 1 0 1 1 1 1 1 1 1 1 0 0 0 0 0 1 0 1 1 0 1 0 0 1 1 1

0 1 1 1 0 0 0 1 0 0 1 1 1 0 1 1 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 0 1 0 1

0 0 1 0 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 0 1 1 1 1 1 1 0 0 0 0 0 0

0 1 0 0 1 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 1 0 1 1

0 1 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 0 1 1 0 1 1 1

0 1 1 0 0 1 1 1 0 0 0 0 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0

0 1 1 1 0 1 0 0 0 1 1 0 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0

0 1 0 0 0 0 1 0 1 1 0 1 0 1 0 1 1 0 0 1 0 1 1 1 0 0 0 0 1 0 1 1 1 0 0 1

0 1 1 0 0 1 0 0 0 0 1 1 1 0 1 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 0 1 0 1 1 1

0 0 1 1 1 1 1 0 0 0 0 0 1 0 0 0 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1 1 0 0 1 0

0 0 1 1 0 0 1 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 0 1 0 1 1 1 1

0 1 1 0 0 0 1 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 1 0 0 1 0 0 0 0 0 1 0 0 0 0

0 0 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0

0 0 0 1 1 1 0 0 1 1 1 0 1 1 1 1 1 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0 1 0

0 1 1 1 1 0 1 0 1 1 0 1 0 0 1 1 1 1 1 0 1 0 1 0 1 1 0 1 1 0 0 0 0 0 1 0

0 0 1 1 0 1 1 0 1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0

0 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0

0 0 1 1 0 0 1 0 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 1 1

0 0 0 1 0 1 0 0 0 1 1 1 1 1 0 0 1 1 0 0 0 1 0 0 1 0 1 0 1 1 1 1 0 0 0 0

0 0 1 0 0 0 0 1 0 1 1 1 1 1 1 0 0 1 0 0 0 1 0 1 0 1 1 1 0 1 1 0 1 1 0 1

0 0 0 0 1 1 1 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 0 0

0 1 0 1 1 0 1 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 1 1 0 0 1 1 0 1 0 0 1 1

0 0 1 0 1 0 0 1 0 0 0 1 1 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 0 1 0 0 0 0 1 0

0 0 0 0 1 0 0 1 0 1 1 1 1 1 0 1 1 0 0 1 0 1 0 0 0 1 1 0 0 0 1 0 0 0 1 1

0 0 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 0 0 0 1 0 0 1

0 1 1 0 1 1 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 1 0 1 0 0

0 0 1 0 0 0 1 0 1 1 1 1 1 1 1 0 0 1 0 1 1 0 0 0 0 1 1 1 1 1 0 0 1 0 0 0

0 1 0 0 1 1 1 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 0 1 0 0 1 0 0 1 1 0 0 1 0 1

0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1

0 0 1 1 1 0 1 0 0 0 0 1 0 0 1 0 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 0 1 0 0

0 0 1 1 0 1 1 0 0 0 1 0 0 0 0 1 0 1 1 0 1 0 0 1 0 0 0 1 0 0 1 1 0 0 1 0

Count is 605 NODE =36

k = 14

MaxCost(Zopt) = 3003

Density = 0.48015872

0 1 0 0 1 1 1 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0

0 0 1 0 1 1 0 1 1 1 1 0 1 0 1 0 1 1 1 1 1 0 0 0 1 0 1 1 1 0 1 1 1 1 1 0

0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 0 0

0 0 1 0 0 0 1 0 1 0 0 1 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1 0 1 0 1 0 0 1 0 1

0 1 0 1 0 1 1 1 1 0 0 0 1 1 0 1 0 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 1

0 1 1 0 1 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0 1 1 0 1 0 1 0 1 1 0 1 1 1 1 1 1

0 1 1 1 0 1 0 1 0 0 1 0 0 1 0 1 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 1 0 0 0 1

0 1 0 0 1 0 1 0 1 1 0 1 0 0 1 1 0 1 1 1 1 0 1 0 1 0 1 0 0 1 0 0 1 0 1 1

0 1 0 1 1 1 1 1 0 1 0 1 1 1 0 0 0 0 0 1 1 1 0 1 1 1 0 1 1 1 0 0 1 0 1 0

0 0 1 0 0 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 1 1 0 1 1 1 0 1 1 0 1 0 1 1

0 0 1 0 0 1 0 1 1 1 0 0 1 0 0 0 1 0 1 1 1 1 0 1 0 1 1 0 0 0 0 1 1 0 0 0

0 1 0 1 0 1 1 0 1 1 1 0 0 1 1 1 1 0 0 1 0 0 0 1 1 1 0 1 0 0 0 0 0 1 1 1

0 0 1 1 1 0 1 1 1 1 0 1 0 1 0 1 1 1 0 0 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0

0 0 0 0 0 1 1 0 1 0 1 0 1 0 0 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 1

0 1 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 0 0 0 1 0

0 0 0 1 0 0 0 1 0 1 1 1 1 1 1 0 1 0 0 0 0 1 1 1 1 0 1 1 1 0 0 0 0 1 0 1

0 0 0 1 1 0 1 0 1 1 1 0 0 1 0 1 0 0 0 1 1 0 0 1 1 1 0 1 1 0 1 1 0 1 0 0

0 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 1 0 1 1 0 1 1 1 1 1 0 1 1 0 0 0 0 1 1 0 1 1 1 0 0 0 0 1 0 1 0 0 0 0

0 1 0 1 1 1 1 0 1 1 0 0 1 0 0 0 1 1 1 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 1

0 0 0 1 0 1 1 0 1 0 1 1 0 1 1 1 1 1 0 1 0 1 0 0 1 0 0 0 0 1 0 1 0 1 1 1

0 0 0 1 1 0 0 0 0 1 0 1 1 1 0 0 0 0 1 0 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0

0 0 0 1 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 1 1 0 0 0 1 1 1 0 1 0 0 1 1 0 0 1

0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 1 1 1 1 1 0 0 1 1

0 0 0 1 0 0 0 1 1 0 0 1 1 1 1 1 0 0 0 0 1 0 0 1 0 1 0 1 1 1 1 0 1 1 1 0

0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 0 1 0 0 1 1 0

0 0 1 0 1 0 1 0 1 1 0 0 0 1 1 1 1 0 1 1 1 0 0 1 0 1 0 0 1 1 0 0 1 1 1 0

0 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 1 1

0 0 1 0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 0 0 0 1 0 0 0 1 1 0 0 1 1 0 0 1 0 1

0 0 0 0 1 1 0 1 1 1 0 0 1 0 0 1 0 0 0 0 1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 1

0 0 1 0 1 0 1 0 1 1 0 0 1 0 1 0 1 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 0 1 0 0

0 0 1 0 0 1 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 0 0 1 0 0

0 1 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 1 1 1 1 1 0 0 0 1 0 0 0 1 0 0 0 1 1 1

0 1 0 1 0 1 0 1 0 0 1 1 1 0 1 1 1 1 0 1 0 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0

0 0 1 1 1 1 0 0 1 0 1 1 0 1 1 1 0 1 1 1 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0

0 1 1 0 0 0 1 1 1 0 0 1 0 1 1 1 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0 0

Count is 643 NODE =36

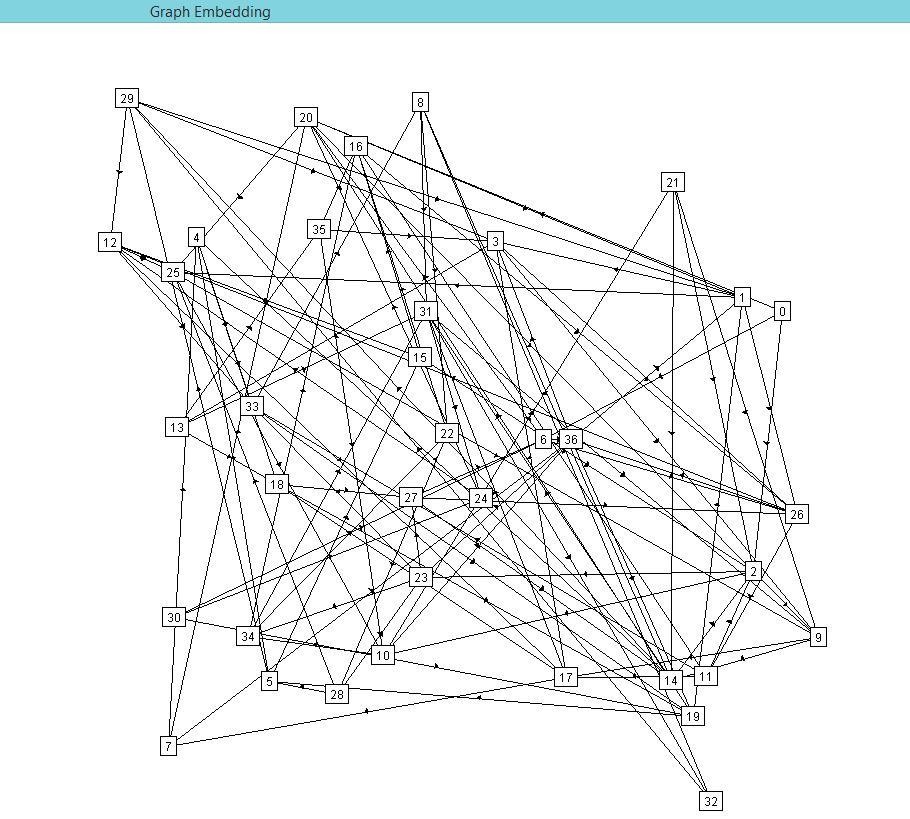
k = 15

MaxCost(Zopt) = 2989

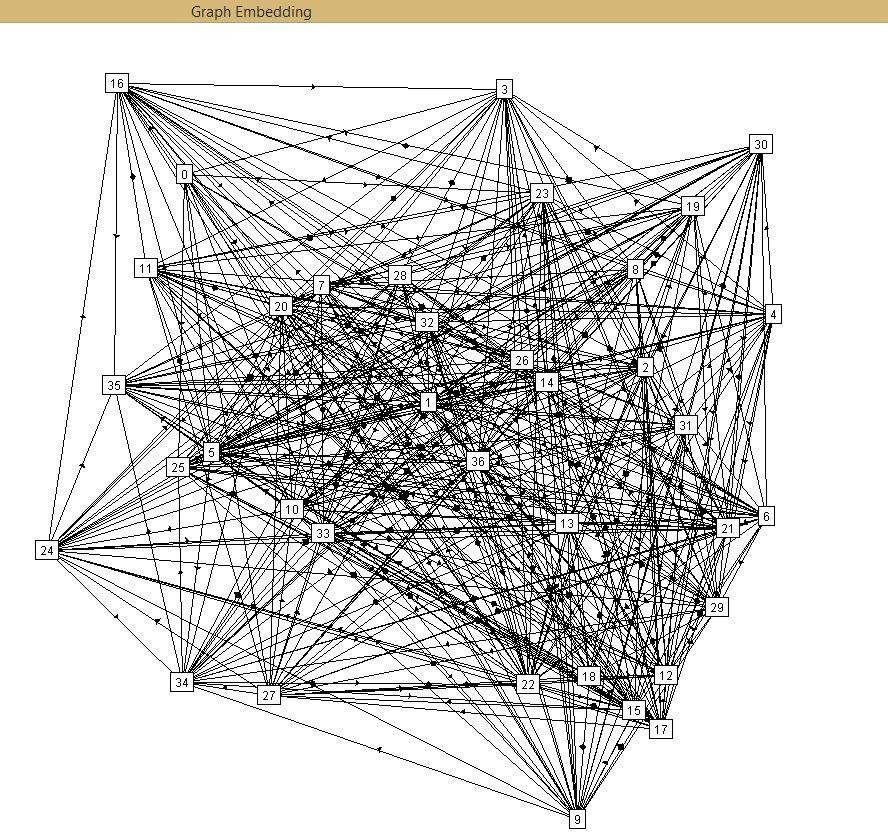
Density = 0.51031744

**Network Topology Generated**

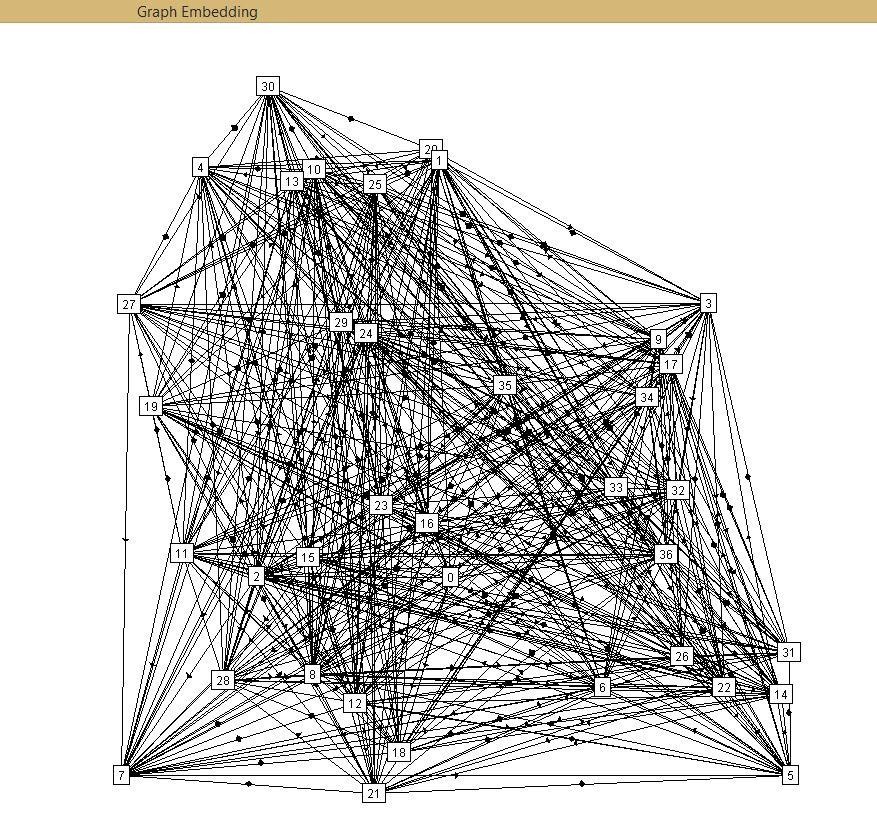
**K = 3**



**K = 10**



**K = 15**



**Dijkstra’s algorithm**

**Dijkstra's algorithm** is a [graph search algorithm t](http://en.wikipedia.org/wiki/Graph_search_algorithm)hat solves the single-source shortest paths for a graph with nonnegative path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms.

I have made use of Dijkstra’s algorithm to compute the shortest paths between each pair of nodes under consideration.

**Pseudo Code:**

• Initialize the cost of each node to ∞

• Initialize the cost of the source to 0

• While there are unknown nodes left in the graph

Select the unknown node b with the lowest cost

Mark b as known

For each node a adjacent to b

 a’s cost = min(a’s old cost, b’s cost + cost of (b, a))

**Implementation:**

We use the shortest path Dijkstra’s algorithm to construct the network topology. Below are the steps,

• User inputs the number of nodes (36) and value of K

• All cost and traffic data structures are initialized

• An array of size k is taken to store k random values of j.

• Set aij as, if, aij = 299 if j! = element in k array

• Set aij as, if, aij = 1 if j == element in k array

• set bij to a random variable in [0, 1, 2, 3]

• Run Dijkstra’s algorithm for each node in the available nodes as source node

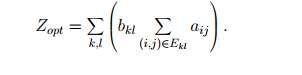
• The shortest path tree is generated by taking the minimum value of the Dijkstra’s algorithm.

**FAST SOLUTION METHOD**

 Find a minimum cost path between each pair k, l of nodes, with edge weights aij: This can be done by any standard shortest-path algorithm. Here I have employed Dijkstra’s algorithm.

 Set the capacity of link (I, j) to the sum of those bkl values for which (i,j) is on the min cost path found for (k,l):

 The optimum cost can be expressed explicitly. Let Ekl be the set of edges that are on the min cost k to l path. Then, according to the above, the optimal cost is:



**SOURCE CODE**

**NetworkGenerator:**

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Random;

**public** **class** NetworkGenerator {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

**int** NODE=36; //number of nodes given

**int** b[][]=**new** **int**[NODE][NODE]; //traffic demand between pairs of nodes

**int** a[][]=**new** **int**[NODE][NODE]; //unit cost of traffic between pairs of nodes

//possible values to choose from for demand

**int** GivenArray[]={0,1,2,3}; //array to store possible values for bij

Random generator = **new** Random(); //creating object of Random library class

**try**

{

**for**(**int** i=0;i<NODE;i++)

{

**for**(**int** j=0;j<NODE;j++)

{

b[i][j] = generator.nextInt(GivenArray.length);

//generating a random value from GivenArray

}

}

}

**catch**(ArrayIndexOutOfBoundsException e)

{

System.***out***.println("Array out of bounds");

}//iterating through each and every value of k - 3,4,5, ..... ,15

**for**(**int** k=3; k<16; k++)

{

**int** MaxCost=0;

**float** density;

**int** count=0;

//Creating an object of ShortestPath class

DjikstraShortestPath sp=**new** DjikstraShortestPath();

//iterating through every value from 0 to n-1

**for**(**int** i=0;i<NODE;i++)

{

//creating an array to store 0 to n-1

**int** indices[]=**new** **int**[NODE];

**for**(**int** l=0;l<NODE;l++)

{

indices[l]=l;

}

//creating an array list called randomList

List<Integer> randomList = **new** ArrayList<Integer>();

//creating an object of RandomGeneration

RandomGenerator r=**new** RandomGenerator();

//retrieve k random indices from RandomGeneration class

randomList=r.createRandomIndices(indices,k,i,NODE);

//System.out.println("Back to Main");

//iterating through 0 to NODE-1

**for**(**int** j=0;j<NODE;j++)

{

// System.out.println("In printing method\n");

//setting diagonal elements to zero

**if**(i==j)

{

a[i][j]=0;

}

//set cost to 1 for indices that are picked in randomlist

**if**(randomList.contains(j)&& i!=j)

{

a[i][j]=1;

}

**else**

//setting a[i][j] to 299 if j is not contained in randomlist

{

**if**(i!=j){

a[i][j]=299;

}

}

//System.out.print(a[i][j]);

//System.out.print("\t\t\t");

// System.out.println("Done printing");

//Invoking findpath method for every pair i,j

sp.findpath(i,j,a);

**int** hopCount=sp.getNumberOfHops();

**int** path[]=sp.getPath();

//retrieve distance to get from i to j

**int** distance=sp.getDistance();

//computing demand times unit cost for each link

MaxCost+=b[i][j]\*distance;

}

}

//counter variable to store number of edges making up shortest path

count=0;

//retrieving path matrix

**int** c[][]=sp.getCount();

//incrementing that link in matrix if it exists

**for**(**int** x=0;x<c.length;x++)

{

**for**(**int** y=0;y<c.length;y++)

{

**if**(c[x][y]>0)

count++;

}

}

**for** (**int** i = 0; i < c.length; i++)

{

System.***out***.print(" ");

**for** (**int** j = 0; j < c.length; j++)

{

**if**(c[i][j]!=0)

{

System.***out***.print("1");

**if**(j!=c.length)

System.***out***.print(" ");

}

**else**

{

System.***out***.print("0");

**if**(j!=c.length)

System.***out***.print(" ");

}

}

System.***out***.println(" ");

**if**(i!=c.length)

System.***out***.print(" ");

}

System.***out***.println("Count is "+ count+" NODE ="+NODE);

density=(**float**)(count)/(NODE\*(NODE-1));

System.***out***.println("k = "+k+"\n"+"MaxCost(Zopt) = "+MaxCost+"\n"+"Density = "+density);

System.***out***.println("\n");

}

}

}

DjikstraShortestPath:

**public** **class** DjikstraShortestPath {

**final** **int** Temp=0;

**final** **int** PERM= 1;

**final** **int** NODE= 36;

**final** **int** Value=299;

**int** NumberOfHops;

**int** path[];

**int** distance;

**int** Count[][]= **new** **int**[36][36];

**public** **int**[][] getCount() {

**return** Count;

}

**public** **void** setC(**int**[][] NewCount) {

**this**.Count = NewCount;

}

**public** **int** getDistance() {

**return** distance;

}

**public** **void** setDistance(**int** distance) {

**this**.distance = distance;

}

**public** **int** getNumberOfHops() {

**return** NumberOfHops;

}

**public** **void** setNumberOfHops(**int** numberOfHops) {

**this**.NumberOfHops = numberOfHops;

}

**public** **int**[] getPath() {

**return** path;

}

**public** **void** setPath(**int**[] path) {

**this**.path = path;

}

**public** DjikstraShortestPath()

{

**for**(**int** x=0; x<Count.length; x++)

{

**for**(**int** y=0; y<Count.length; y++)

{Count[x][y]=0;

}

}

}

**void** findpath(**int** s,**int** d,**int** a[][])

{

**int** i,min,count=0;

**int** current,newdist,u,v,n=NODE;

**int** setdist=0;

**int** path[]=**new** **int**[n];

NodeGenerator[] node=**new** NodeGenerator[n];

/\* Make all nodes temporary \*/

**for**(i=0; i<n; i++)

{

node[i] = **new** NodeGenerator();

node[i].setPredecessor(-1);

node[i].setDistance(Value);

node[i].setStatus(Temp);

}

/\*Source node should be permanent\*/

node[s].setPredecessor(-1);

node[s].setDistance(0);

node[s].setStatus(PERM);

/\*Starting from source node until destination is found\*/

current=s;

**while**(current!=d)

{

**for**(i=0; i<n; i++)

{

/\*Checks for adjacent temporary nodes \*/

**if** ( a[current][i] > 0 && node[i].getStatus() == Temp)

{

newdist=node[current].getDistance() + a[current][i];

/\*Checks for Relabeling\*/

**if**( newdist < node[i].getDistance() )

{

node[i].setPredecessor(current);

node[i].setDistance(newdist);

}

}

}/\*End of for\*/

/\*Search for temporary node with minimum distance make it current node\*/

min=Value; current=-1;

**for**(i=1; i<n; i++)

{

**if**(node[i].getStatus() == Temp &&

node[i].getDistance() < min)

{

min = node[i].getDistance();

current=i;

}

}/\*End of for\*/

**if**(current==-1) /\*If Source or Sink node is isolated\*/

**return** ;

node[current].setStatus(PERM);

}/\*End of while\*/

/\* Getting full path in array from destination to source \*/

**while**( current!=-1 )

{

// count++;

path[count++]=current;

current=node[current].getPredecessor();

}

/\*Getting distance from source to destination\*/

**for**( i=count-1; i>0; i--)

{

u=path[i];

v=path[i-1];

setdist= setdist + a[u][v];

Count[u][v]=Count[u][v]+1;

}

**this**.setNumberOfHops(count);

**this**.setPath(path);

**this**.setDistance(setdist);

**this**.setC(Count);

}/\*End of findpath()\*/

}

RandomGenerator:

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.List;

**public** **class** RandomGenerator {

//

**public** List<Integer> createRandomIndices(**int**[] indices,**int** k,**int** i,**int** n)

{

//creating a list to store 0 to n-1

List<Integer> list = **new** ArrayList<Integer>();

Integer arr[] = **new** Integer[36];

//creating list to store random k values

List<Integer> randomList = **new** ArrayList<Integer>();

**for**(**int** j=0;j<n;j++)

{

// System.out.println("In for loop");

arr[j] = **new** Integer(indices[j]);

list.add(arr[j]);

}

//System.out.println("Done adding to list\n");

//removing the possibility of selecting j=i

list.remove(i);

//shuffling the list to facilitate randomization

Collections.*shuffle*(list);

**for**(**int** j=0; j<k; j++)

{

{

//Retrieving the first k elements one by one

**int** z= (**int**) list.get(j);

//Adding retrieved element to randomList

randomList.add(z);

}

}

//return randomlist to main method

**return** randomList;

}

}

Node Generator **:**

**public class** NodeGenerator {

//integer to identify predecessor node

**int** Pred;

**int** dist; /\*minimum distance of node from source\*/

//stores status - permanent or temporary

**int** status;

**public** NodeGenerator() {

**super**();

}

**public int** getDistance() {

**return** dist;

}

**public void** setDistance(**int** dist) {

**this**.dist = dist;

}

**public int** getPredecessor() {

**return** Pred;

}

**public void** setPredecessor(**int** pred) {

**this**.Pred = pred;

}

**public int** getStatus() {

**return** status;

}

**public void** setStatus(**int** status) {

**this**.status = status;

}

}

Note : Total cost decreases with increase in k and total density increases with increase in k.

**Explanation of Modules**

 NetworkGenerator: Contains the main method, which calls the RandomGenerator and DjikstraShortestPath computation functions.

 RandomGenerator: Contains the createRandomIndices method which generates random k values from the i values passed without repetition.

 DjikstraShortestPath: Contains the method findPath which computes the shortest path between each pair of nodes using Djikstra’s algorithm. It Returns the shortest path and number of hops between the nodes.

 NodeGenerator: Has member variables- predecessor, status and distance. Also contains getters and setters for the same for usage within the DjikstraShortestPath class. These three properties apply to each node of the topology.

**REFERENCES**

 Dijkstra’s algorithm was taken off the shelf source available on algolist.com. It was a JAVA language implementation.

 The description of Dijkstra’s algorithm was taken as a basis

from Wikipedia.org and then translated into my own words.

 The network topology is generated from the website [http://www.cs.rpi.edu/research/groups/pb/graphdraw/headpa](http://www.cs.rpi.edu/research/groups/pb/graphdraw/headpage.html)

[ge.html.](http://www.cs.rpi.edu/research/groups/pb/graphdraw/headpage.html)

 The formulae for computation of the Zopt(MaxCost) and Density values was borrowed from material posted by Dr Andras Farago in his lecture notes.