Single-Moment Analysis of Unreliable Trunk Networks Employing K-Shortest-Path Routing

Mark C. Sinclair[1]

1. Introduction

Traditionally, research into circuit-switched (C-S) networks has been divided into two areas, reliability and performance, each making little reference to the other. More recently, however, research has focused on unification of the two. One possible choice for a combined performance and reliability measure is the overall network grade-of-service (NGOS), i.e. the overall probability of blocking, averaged over all node pairs, taking into account both the offered traffic between node pairs and the link (trunk group) availabilities.

In a recent paper [1], two methods were given for the single-moment analysis of fixed-alternative-routing (FAR) circuit-switched communications networks with reliable nodes but unreliable links. The second of these, MKV_GOS, employs a Markov model for an unreliable link, in combination with earlier work [2-4], so as to greatly reduce execution time compared to existing algorithms.

In two subsequent papers [5,6], the MKV_GOS method was used as part of three heuristic routing table optimisation algorithms, which can be used to obtain routing tables such that the overall NGOS is minimised under different call control rules. These algorithms were then used to demonstrate the improvements in trunk network fault-tolerance (assessed in terms of the overall NGOS) that can be achieved both through routing table updating in the event of link failure [5], and through network augmentation i.e. the addition of extra links to a network, either in parallel with existing links, or between nodes that previously had no direct connection [6].

In the current paper, an extension to the MKV_GOS method, which was originally limited to OOC-SF (Originating Office Control with Spill-Forward) table-based routing [2], is described. This extension allows the analysis of networks employing two additional types of routing, K-shortest-disjoint-loopless-path and K-shortest-loopless-path [7]. The extended method is then demonstrated by comparing results for the two K-shortest-path routing types in terms of their overall NGOS for an example network. In addition, the execution time of the method for a series of networks of increasing size is presented.

2. Network Description

The following five elements are used to describe the model of an unreliable trunk network used in this paper:

1. Call Control Rule
2. Network Configuration Table
3. Link Capacity, Availability & Length Table
4. Busy-Hour Traffic Matrix
5. Loading

The Call Control Rule determines the type of routing used: whether K-shortest-disjoint-loopless-path or K-shortest-loopless-path, and the value of K.

The Network Configuration Table (e.g. Table 1) is an n x n array of link numbers, where n is the number of nodes in the network, except that the forward diagonal is empty. The (S,T) entry is either the link number of the link connecting node S to node T, or -1 if there is no direct link.

The Link Capacity, Availability & Length Table (e.g. Table 2) contains the capacities (i.e. number of circuits), availabilities and lengths (in arbitrary units) of the b links connecting the network.

The Busy-Hour Traffic Matrix (e.g. Table 3) is an n x n array, with an empty forward diagonal, of the busy-hour traffic in Erlangs between each node pair (S,T).

The Loading is used to obtain different traffic levels. Each entry in the Busy-Hour Traffic Matrix is multiplied by the loading, and thus, for example, with a loading of 1.1, the traffic is increased by 10% over the nominal level. Unless otherwise stated, the loading is always set to 1.

[1] British Telecom Lecturer in Communications Network Modelling , Department of Electronic Systems Engineering, University of Essex

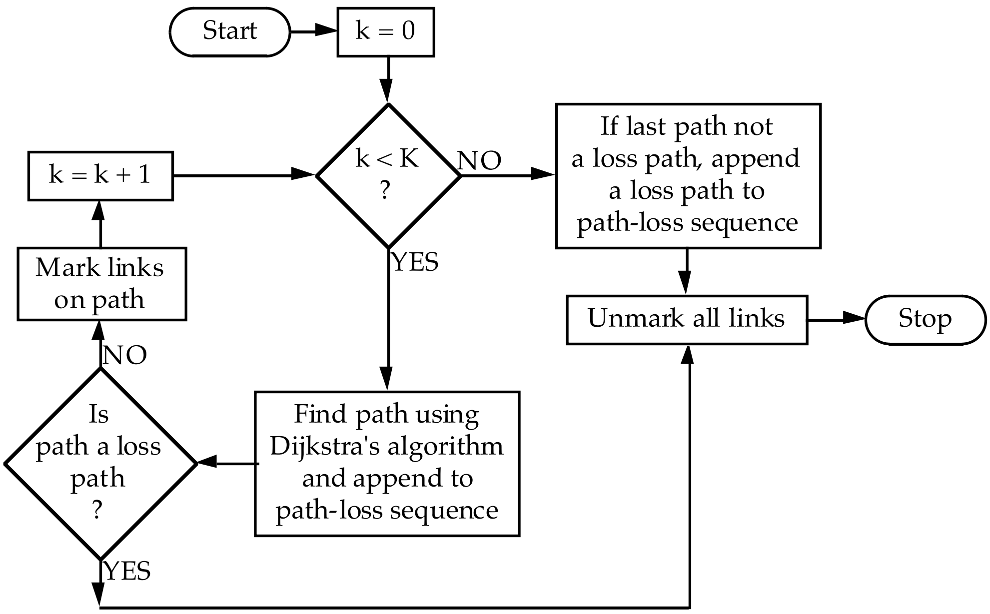
3. Shortest-Path Routing Algorithms

The shortest-path algorithm used as a subroutine by both the K-shortest-disjoint-loopless-path and K-shortest-loopless-path algorithms was a modification of Moore & Dijkstra's algorithm "Finding the shortest path from [a node] to the other [nodes] in a graph where the lengths are positive" [8]. The modification was simply to take the length of a link as infinite if it was found to be *marked* at the start of the algorithm. The use of link marking is described in the next two sections.

3.1. K-Shortest-Disjoint-Loopless-Path Algorithm

Flowchart 1 below shows the overall flow of control for the K-shortest-disjoint-loopless-path algorithm; its operation is straightforward given the above modification to Dijkstra's algorithm. By way of illustration, the path-loss sequence [1,2] produced with K=3 for node pair (3,1) of the Example Network (see §9 below) is {P1(3,1), P2(3,4,1), P3(3,2,1), L1(3,L)}, with the three path lengths being 12.0, 26.0 and 32.0, respectively. The first path, P1, is simply the direct link between nodes 3 and 1; with that link marked, the second path found is that via node 4; and with those two additional links marked, the third path is that via node 2. The algorithm concludes by appending the loss path L1 to the end of the sequence. By way of contrast, the path-loss sequence for node pair (3,4) is {P1(3,4), P2(3,1,4), L1(3,L)}, with path lengths 10.0 and 28.0, respectively. The algorithm has correctly failed to find a third disjoint path for this node pair, as with links 2, 3 and 6 marked, there are no paths between nodes 3 and 4.

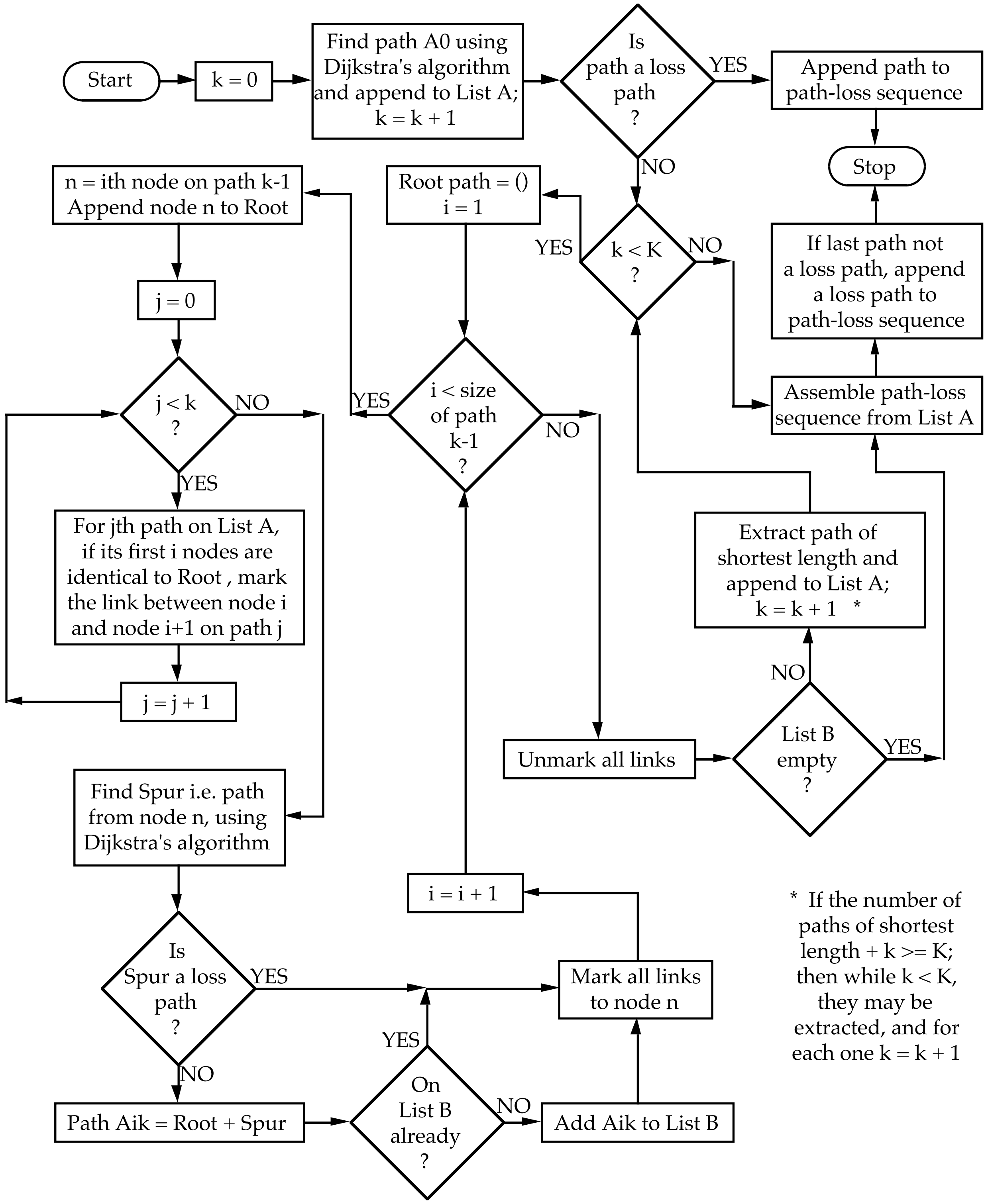
Flowchart 1 K-Shortest-Disjoint-Loopless-Path Algorithm



3.2. K-Shortest-Loopless-Path Algorithm

Flowchart 2 below shows the more complex overall flow of control of the K-shortest-loopless-path algorithm. This is essentially that of Yen [7] with three modifications (see [7] for a definition of the terms used). These were firstly, the use of link marking rather than setting and resetting the link lengths to infinity. Secondly, the realisation that the algorithm might not find a valid spur (i.e. Dijkstra's algorithm could return a loss path), especially with a large number of marked links, a possibility not considered in Yen's paper. Finally, that although the algorithm might find a valid path A_i^k from combining a root (R_i^k) and a spur (S_i^k), this might well be a duplicate of one already found and stored on List B, something Yen's paper also failed to consider. By way of illustration, the path-loss sequence produced with K=3 for node pair (3,1) is {P1(3,1), P2(3,4,1), P3(3,2,1), L1(3,L)}, i.e. the same as above. This is because these three paths, although disjoint, are also the three shortest paths (disjoint or not) for this node pair. For node pair (3,4), however, the path-loss sequence is {P1(3,4), P2(3,1,4), P3(3,2,1,4), L1(3,L)}, with path lengths 10.0, 28.0 and 48.0, respectively; the algorithm has found a third non-disjoint path.

Flowchart 2 K-Shortest-Loopless-Path Algorithm



4. Analysis of a Network Using a Markov Model for Unreliable Links

The *MKV_GOS* method for the single-moment overall NGOS analysis of OOC-SF FAR C-S communications networks with reliable nodes but unreliable links was described in [1]. It was developed using techniques given by Lin *et al.* [2] and Chan [3], in combination with a recursive algorithm for the generation of path-loss sequences from OOC-SF routing tables and a Markov model for unreliable links, both devised by this author. The method uses the iterative approach to network analysis first developed by Katz & Cooper [9], and it relies on the standard eight assumptions discussed by Lin *et al.*, as well as the assumption that all the nodes in the network are perfectly reliable. For details of the *MKV_GOS* method, and a comparison with a method based on earlier work, see [1].

The extension of *MKV_GOS* to incorporate K-shortest-disjoint-loopless-path and K-shortest-loopless-path routing was straightforward after the routing algorithms themselves had been implemented. Rather than carrying out the analysis based on path-loss sequences generated from the OOC-SF routing table, the K-shortest path algorithms were used to generate the path-loss sequences directly, and the remainder of the *MKV_GOS* method remains the same as in [1]. It should be noted that while Chan's Algorithm 2, for a "single-loss-route route sequence"[3], is applicable to K-shortest-path routing, Chan's more general Algorithm 1, for an "arbitrary route sequence", was used throughout the current version of *MKV_GOS* .

5. Results for the Example Network

The extended *MKV_GOS* method was used on the Example Network for a selection of different types of routing. Figure 1 below shows the overall NGOS at various link availabilities for shortest-path routing (i.e. K=1), K-shortest-loopless-disjoint routing (with K=2) and K-shortest-loopless-path routing (K=2). While a single figure is given on the graph for link availability, the actual link unavailabilities used were varied proportional to link length, with the figure given on the graph being that for a length of 10 units. The actual levels used for a nominal link availability of 0.98 are shown in Table 2 below. Clearly for this network, K-shortest-loopless-path routing (with K=2) results in the lowest overall NGOS at all levels of link availability examined.

Fig 1 NGOS for Example Network using Three Types of Routing

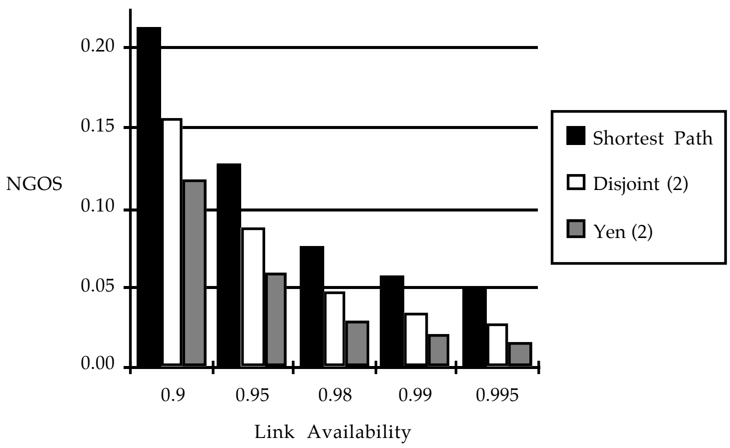
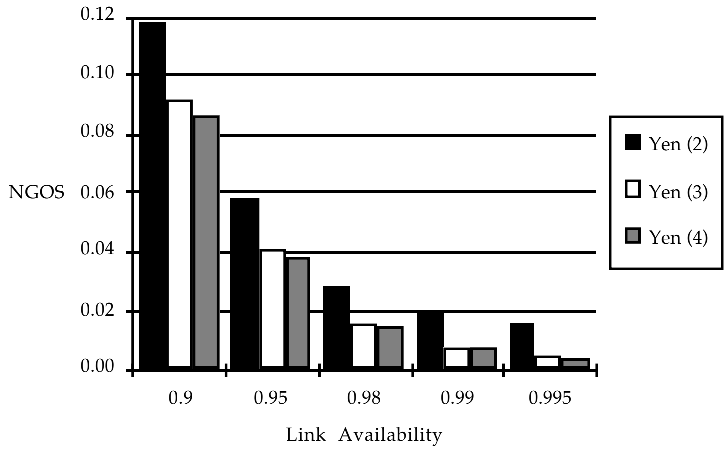


Figure 2 below shows the overall NGOS at various link availabilities for K-shortest-loopless-path routing with K = 2, 3 and 4. While there is some improvement between K=2 and 3, little is gained by moving up to K=4 for the Example Network.

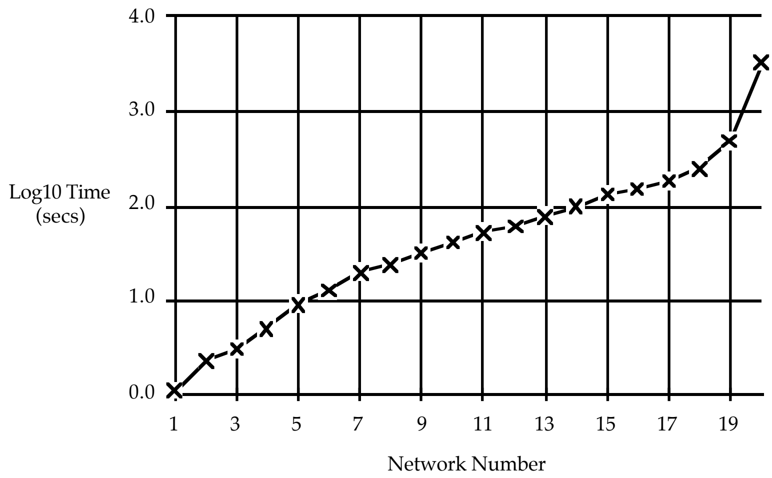
Fig 2 NGOS for Example Network using K-Shortest-Loopless-Path Routing



6. Execution Times

In order to establish the execution times of the extended *MKV_GOS* method for larger networks, a series of networks of increasing size were analysed. The code was written in C++ and executed on a SPARCstation ELC. As this exercise was also being used to build confidence in the implementation, the "Programming by Contract" assertions [10] were not compiled out, so the times obtained should be regarded as upper bounds. The graph below shows the mean of three runs (variance negligible), using K-shortest-loopless-path routing with K=3 and link availabilities of 0.98, on the series of networks described in [1]; for network numbers 1 to 20 these range in size from 4 to 42 nodes.

Figure 3 Log10 Execution Time for a Series of Networks



The sharp rise in execution times observed above 36 nodes was due to an increase in the number of iterations required to achieve convergence. The approach taken in the current version of *MKV_GOS* is simply to average the previous iteration's estimate for each link blocking probability with the current estimate, and then use that figure for the next iteration [1]. A more sophisticated approach may therefore prove necessary for large networks, although the convergence difficulties observed with the long ladder/chain networks used here may not be observed in more practical network topologies.

7. Conclusion

The extension of the *MKV_GOS* method to include both K-shortest-disjoint-loopless-path and K-shortest-loopless-path routing has been demonstrated, and in addition a number of useful omissions from Yen's original K-shortest-loopless-path algorithm have been presented.

8. References

1 Sinclair, M.C. **"Single-Moment Analysis of Unreliable Trunk Networks: Two Methods for the Network Grade-of-Service"** Proc 8th UK Teletraffic Symposium, Beeston, Apr 91, p8/1-8/11

2 Lin, P.M., Leon, B.J. & Stewart, C.R. **"Analysis of Circuit-Switched Networks Employing Originating-Office Control with Spill-Forward"** IEEE Trans Commun vCOM-26 n6 Jun 78 p754-765

3 Chan, W.S. **"Recursive Algorithms for Computing End-to-End Blocking in a Network with Arbitrary Routing Plan"** IEEE Trans Commun vCOM-28 n2 Feb 80 p153-164

4 Bonaventura, V., Cacopardi, S., Decina, M. & Roveri, A. **"Service Availability of Communication Networks"** Proc 1980 Nat Telecommun Conf, Houston, Dec 80, p15.2.1-15.2.6

5 Sinclair, M.C. **"Trunk Network Fault-Tolerance through Routing Table Updating in the Event of Trunk Failure"** Proc 9th UK Teletraffic Symposium, Univ. of Surrey, Guildford, Apr 92, p17/1-17/9

6 Sinclair, M.C. **"Trunk Network Fault-Tolerance through Network Augmentation"** Proc 4th Bangor Communications Symposium, University of Wales, Bangor, May 92, p252-256

7 Yen, J.Y. **"Finding the K Shortest Loopless Paths in a Network"** Manag. Sci. v17 n11 Jul 71 p712-716

8 Gondran, M. & Minoux, M. **"Graphs and Algorithms"** Wiley, 1984

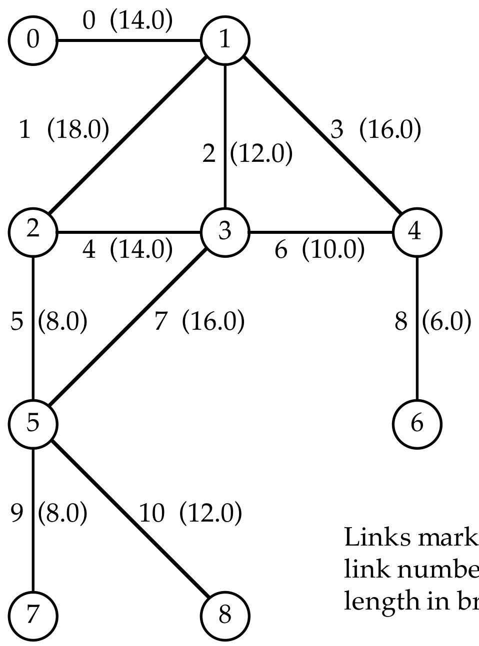
9 Katz, S. **"Statistical Performance Analysis of a Switched Communications Network"** Proc 5th Int Teletraffic Cong, New York, Jun 67, p566-575

10 Meyer, B. **"Object-Oriented Software Construction"** Prentice Hall, 1988

9. Appendix: The Example Network

Figure 4

A Circuit-Switched Communications Network



Links marked with link number and length in brackets

Table 1 Network Configuration Table

From \ To	0	1	2	3	4	5	6	7	8
0		0	-1	-1	-1	-1	-1	-1	-1
1	0		1	2	3	-1	-1	-1	-1
2	-1	1		4	-1	5	-1	-1	-1
3	-1	2	4		6	7	-1	-1	-1
4	-1	3	-1	6		-1	8	-1	-1
5	-1	-1	5	7	-1		-1	9	10
6	-1	-1	-1	-1	8	-1		-1	-1
7	-1	-1	-1	-1	-1	9	-1		-1
8	-1	-1	-1	-1	-1	10	-1	-1	

Table 2 Link Capacity, Availability & Length Table

Link Number	Capacity	Avail.	Length
0	30	0.972	14.0
1	30	0.964	18.0
2	30	0.976	12.0
3	30	0.968	16.0
4	30	0.972	14.0
5	30	0.984	8.0
6	30	0.98	10.0
7	30	0.968	16.0
8	30	0.988	6.0
9	30	0.984	8.0
10	30	0.976	12.0

Table 3 Busy-Hour Traffic Matrix

From \ To	0	1	2	3	4	5	6	7	8
0		0.94	0.94	1.88	0.94	0.94	0.47	0.47	0.47
1	0.94		1.88	3.76	1.88	1.88	0.94	0.94	0.94
2	0.94	1.88		3.76	1.88	1.88	0.94	0.94	0.94
3	1.88	3.76	3.76		3.76	3.76	1.88	1.88	1.88
4	0.94	1.88	1.88	3.76		1.88	0.94	0.94	0.94
5	0.94	1.88	1.88	3.76	1.88		0.94	0.94	0.94
6	0.47	0.94	0.94	1.88	0.94	0.94		0.47	0.47
7	0.47	0.94	0.94	1.88	0.94	0.94	0.47		0.47
8	0.47	0.94	0.94	1.88	0.94	0.94	0.47	0.47	