

Network Probabilistic Connectivity: Exact Calculation with Use of Chains^{*}

Olga K. Rodionova¹, Alexey S. Rodionov¹, and Hyunseung Choo²

¹ Institute of Computational Mathematics and Mathematical Geophysics
Siberian Division of the Russian Academy of Science
Novosibirsk, RUSSIA +383-2-396211
`alrod@rav.sbcc.ru`

² School of Information and Communication Engineering
Sungkyunkwan University
440-746, Suwon, KOREA +82-31-290-7145
`choo@ece.skku.ac.kr`

Abstract. The algorithmic techniques which allow high efficiency in the exact calculation of reliability of an undirected graph with absolutely reliable nodes and unreliable edges are considered in this paper. The new variant of the branching algorithm that allow branching by chains is presented along with improvement of series-parallel reduction method that permits the reduction of a long chain by one step.

1 Introduction

The task of calculating or estimating the probability of whether the network is connected (often referred to as its *reliability*, is the subject of much research due to its significance in a lot of applications, communication networks included. The problem is known to be NP-hard irrelative of whether the unreliable edges or nodes or both are considered. Most explored is the case of absolutely reliable nodes and unreliable edges that corresponds to real networks in which the reliability of nodes is much higher than that of edges. The transport and radio networks are good examples. We show that, by taking into consideration some special features of real network structures and using modern high-speed computers, we can conduct the exact calculation of reliability for networks with dimension of a practical interest.

The well-known branching algorithm [1] uses branching on the alternative states of an arbitrary edge. Our first approach is to branch by the whole chain if it exists. Another well-known approach that uses series-parallel reduction owes to its spreading mostly to A.M. Shooman [2,3]. In the reduction of series this method uses consequent reduction of pairs of edges. We propose to reduce the entire chain at once thereby increasing in calculation speed.

^{*} This work was supported in parts by BK21, University ITRC and RFBR. Dr. H.Choo is the corresponding author.

The programming of the proposed algorithms is non-trivial. In this paper we are trying give a proper attention to this task. Special notice is given to the problem of computer storage economy.

2 Using Chains in the Calculation of Network Reliability

As the treating of dangling nodes, articulation nodes and bridges in the reliability calculation is well-known we consider the initial network structures that are free of them.

Our extended branching method (branching by chain) is based on the following theorem.

Theorem 1. *Let a graph G have a simple chain $Ch = e_1, e_2, \dots, e_k$ with edge reliabilities p_1, p_2, \dots, p_k , respectively, connecting nodes s and t . Then the reliability of G is equal to*

$$R(G) = \prod_{j=1}^k p_j \cdot R(G^*(Ch)) + \sum_{i=1}^k (1 - p_i) \prod_{j \neq i} p_j \cdot R(G \setminus Ch), \quad (1)$$

if e_{st} does not exist and

$$R(G) = \left[(p_1 + p_{st} - p_1 p_{st}) \prod_{j=2}^k p_j + p_{st} \sum_{i=2}^k (1 - p_i) \prod_{j \neq i} p_j \right] \times R(G^*(Ch)) + \quad (2)$$

$$\left[(1 - p_1)(1 - p_{st}) \prod_{j=2}^k p_j + (1 - p_{st}) \sum_{i=2}^k (1 - p_i) \prod_{j \neq i} p_j \right] \times R(G \setminus Ch \setminus e_{st}),$$

otherwise, where $G^*(Ch)$ is a graph obtained from G by contracting by a chain, $G \setminus Ch$ is a graph obtained from G by deletion of this chain with nodes (except for terminal ones), and p_{st} is the reliability of an edge directly connecting the terminal nodes of the chain.

A.M. Shoorman [2,3] has proposed substituting the parallel or subsequent pair of edges to one to speed up the reliability calculation. Thus the graph G is transformed to some graph G^* with lesser number of edge and, possibly, nodes. Reducing k parallel edges to one with reliability p is obvious and simple while the reducing of an consequent pair of edges leads to a graphs with a different reliability:

$$R(G) = r R(G^*), \quad (3)$$

$$p = \frac{p_1 p_2}{1 - (1 - p_1)(1 - p_2)} = \frac{p_1 p_2}{p_1 + p_2 - p_1 p_2}, \quad r = p_1 + p_2 - p_1 p_2.$$

Based on this result and the consequent reduction on pairs of edges for the chain with length $k > 2$ we derived the following

Theorem 2. *Let a graph $G(n, m)$ have a simple chain $Ch = e_1, e_2, \dots, e_k$ with edge reliabilities p_1, p_2, \dots, p_k , respectively, connecting nodes s and t . Then*

$$R(G(n, m)) = \prod_{i=1}^k p_i \left(\sum_{i=1}^k p_i^{-1} - k + 1 \right) R(G_2(n - k + 1, m - k + 1)), \quad (4)$$

where a graph $G_2(n - k + 1, m - k + 1)$ is derived from $G_1(n, m)$ by substituting the chain by a single edge with the probability of the edge existence

$$p = 1 / \left(\sum_{i=1}^k p_i^{-1} - k + 1 \right). \quad (5)$$

After substituting all chains by edges the reduced graph is calculated by the simple branching method. If during the process a new chain appears, then it is also substituted by an edge. Reducing all chains with consequent branching is faster than branching by chains as it leads to small-dimension graphs on earlier recursions.

3 Program Realization of the Algorithms and Case Studies

The problem of programming the proposed algorithms is not trivial by virtue of the high request to the memory, and of numerous recursions also. We discuss the following aspects in this section: (1) re-usage of memory in recursions; (2) finding chains for branching and reduction; (3) renumbering nodes; and (4) the final graphs that allow direct calculation.

The **re-usage of memory** is provided by considering the upper-left block of the same probability matrix on each recursion. To provide this we need **renumbering** of nodes: the chain should be contracted to a node with node number $n - k$ (dimension of the reduced graph), thus this number is assigned to one of its terminal nodes. The number $n - k + 1$ is assigned to the other one. Thus the numbers of nodes of the resolving chain (including terminal) should be $n - d, n - d + 1, \dots, n$ after renumbering, where d is the number of edges for the chain, and n is the number of nodes for the graph under reduction.

On execution of branching it is necessary to take into account all **possible variants of the resulting graphs**. While performing the classical branching method there are only 3 possible results: the derivation of a disconnected graph at deletion of an edge, a graph of small dimension simple for calculation at contracting and a graph that is connected but not possible for direct calculation yet, to which the operation of branching is applied again. At usage of the branching by chain or chain reduction it is necessary to take additional variants into account. They are: (1) the resulting graph is a cycle; (2) the resolving chain is a cycle; (3) the dangling node appears. We specially treat the case (4) "the resulting graph is disconnected". The last means that any edge in the deleted

chain is a bridge. Accordingly, by contracting we obtain a articulation point and the reliability of the graph is considered as the product of the reliabilities of two graphs G_1 and G_2 and probability of the existence of a resolving chain (or edge).

We conducted several experiments on the computer with the processor AMD Athlon 800MHz inside. We have made the comparisons among the algorithm with branching by chains (BC), basic branching algorithm (BB), branching algorithm with chain reduction (BR) and algorithm from [6] (RT).

In the example of the lattice (4×4) graph, that was used in [6], the number of basic recursion for RT is 2579141, time spent for calculation was about 47 seconds. Algorithm BC takes 0.17 seconds and only 407 recursions on this example. Note, that 200 chains were found during the calculation with average length 2.385. So on this example our algorithm is more than 200 times faster. The basic BB algorithm takes on this example 8.35 seconds, which is about 50 times slower than BC and takes 80619 recursions. However best results were shown by the BR algorithm which takes only 0.06 seconds on 93 recursions. When the dimension of a lattice was increased up to (5×5) the algorithm RT did not finished in 2 hours, and BB, BC and BR algorithms took 21 minutes, 15.05 and 2.47 seconds on 13817311, 51652 and 14581 recourses respectively.

Last we calculate the reliability of the graph with the structure of well-known ARPA network. This graph has 58 nodes and 71 edges. The algorithm BC takes approximately 20 minutes and BR – about one minute for calculation. With this the last algorithm takes only 31933 recursions.

Thus we can state that our modifications of branching method and method of parallel-subsequent reduction are faster than previous methods and allow the calculation of reliability of networks with tens of elements in reasonable time.

References

1. Moore, E.F., Shannon, C.E., "Reliable Circuits Using Less Reliable Relays," *J. Franklin Inst.*, 262, n. 4b, pp. 191-208, 1956.
2. Shooman, A.M., Kershenbaum, A., "Exact Graph-Reduction Algorithms for Network Reliability Analysis," *Proc. GLOBECOM' 91. Vol. 2*, pp. 1412-1420, 1991.
3. Shooman, A.M., "Algorithms for Network Reliability and Connection Availability Analysis," *Electro/95 Int. Professional Program Proc.*, pp. 309-333, 1995.
4. Rodionov, A.S., Rodionova, O.K., "On a Problem of Practical Usage of the Moore-Shennon Formula for Calculating the Reliability of Local Networks," *Proc. 2nd Int. Workshop INFORADIO-2000*, Omsk, pp. 67-69, 2000.
5. Rodionova, O.K., "Some Methods for Speed up the Calculation of Information Networks Reliability," *Proc. XXX International Conf. "IT in Science, Education, Telecommunications and Business," Ukraine, Gurzuf*, pp. 215-217, 2003.
6. Chen, Y., Li, J. Chen, J., "A new Algorithm for Network Probabilistic Connectivity," *Proc. MILCOM'99. IEEE, Vol. 2*, pp. 920-923, 1999.
7. Rodionova, O.K. "Application Package GRAPH-ES/3. Connectivity of the Multigraphs with Unreliable Edges (Atlas, procedures)," Preprint No. 356, Computing Center of the SB AS of the USSR, Novosibirsk, 1982. (in Russian)
8. T. Koide, S. Shinmori and H. Ishii, "Topological optimization with a network reliability constraint," *Discrete Appl. Math.*, vol. 115, Issues 1-3, pp. 135-149, November 2001.