

USING MINIMAL PATH ALGORITHM FOR MEASUREMENT OF NETWORK RELIABILITY

Kunjali Bhardwaj, Dr. Neerja Jindal

Abstract: The mathematical theory of reliability has grown out of the demand of modern technology and particularly out of the experience with complex military equipment and systems. The early Advance Research Project Agency (ARPA) network, the forerunner of today's Internet and World Wide Web, motivated this. Although, the problem of computing the probability that distinguished nodes in a network would remain connected is an NP-hard problem, the area has continued to catch the attention of researchers all over the globe to explore and develop efficient algorithms. The main objective is to enhance the ability of such complex network systems. For this research, reliability is an important consideration. Ideally, we design the network models and algorithms and take as input and produce as output of network design model. Minimal Paths and Minimal Cuts are the most popular connectivity methods and many researchers address many network design problems. This work proposes an algorithm to generate all the minimal paths of the general flow network, which is a novel approach and it is based on the principle of backtracking; we can further evaluate the network reliability using any existing SDP based approach.

Index term : Principal of Backtracking, SDP base approach, flow network, Distributed communication network, multi source and multi sink communication network, path set and cut set based algorithm, Graphs connectivity and network.

◆

1. Introduction

When systems grow in size then complexity increases and they become more prone to failures so it is necessary to improve the performance of the system through reliability analysis. There are many applications to determine the reliability of the system and whose components have more prone to failures such as power system, telecommunication system, computer communication systems, mechanical systems and large software structure. These systems are in the form of network in which there are several interconnections. In computer communication or a telecommunication, there are number of vertices and edges in which vertices represent the locations of transmitters or receivers and edges represent the communication links between them. The network is totally dependent on the vertices and edges, if any vertices or edges fail or work, then the network itself considered to be in working state or in the failed state.

Now-a-days, communication between the specified vertices of the different networks is an important issue. Mostly networks are designed in such a way that there are several minimal paths between the two vertices of the network that's why network should be highly reliable. A network is designed in such a way that if there are two vertices in the network then there should be few disjoint paths set between the two vertices. A major problem is to determine the reliability of the network. Several researchers provide different definitions of the system reliability. In Network reliability, problem is to search for all the Minimal Paths and Minimal Cuts in the network. There are different types of algorithms for searching the Minimal Paths in the network such as direct search based, augmentation based

and symbolic expression based. In symbolic expression based algorithm, symbolic terms and operators are defined and produce MP using algebraic manipulations. This type of approach mainly causes inefficient search for both computation time and storage.

There are three types of reliability measures such as 2-terminal reliability, k-terminal reliability and g-terminal reliability. 2-terminal reliability defines the probability that two specified pair of nodes can communicate with each other. G-terminal reliability means that every node in the network can communicate with each other and k-terminal reliability means a specified set of nodes can communicate with each other in the network.

Mostly methods are very simple but they become impractical when the number of spanning trees increase and their exponential growth also starts increase in the network size. Recently, some researchers, like Kansal & Devi proposed an algorithm which is based on the minimal cut-sets and generates many cut-sets so it does not provide the correct results. The 2-terminal is an important reliability measure and those methodologies is to deal with the 2-terminal have factoring theorem, sum of disjoint products, state enumeration, Boolean algebra and transformations. Besides, there are two categories for computing terminal reliability are path-sets and cut-sets based algorithms and they are more superior than path-sets based algorithm with respect to the computation time.

In this thesis, we are studying about the different areas of reliability such as cloud computing, Grid computing, multicast communication, spread sensor networks, parallel computing and under water communication. Today's

single source and terminal is not only bound in the communication network so it is more important to increase the reliability of the complex network, we can explore in the multi-source and multi-terminal network. The ultimate objective of this research is to enumerate all the minimal paths and cycles for multi sources in a general flow network using the principle of Backtracking.

2. Objective

The objective of the ongoing research is to find the method by which an important performance parameter i.e. reliability can be find out by efficiently and by doing less computation. Thus it will save both time and energy.

Based on these observations the following major objectives are set:

1. Study the various communication general networks.
2. Study the various approaches used for enumerating the all minimal path sets for general networks.
3. Study the various communication network reliability evaluation methods.
4. To propose a modified algorithm to enumerate multi-terminal path sets for evaluating reliability of multi-source and multi sink communication networks.

3. Reliability Evolution

When design the multi component system then many problems occurs of using available sources so as to maximize the consumption of resources for multi component systems achieving specific reliability goals. There are several methods to improve the reliability of the system are:

- (1) To reduce the system complexity
- (2) Component's reliability allocation
- (3) Redundancy alone or combined with reliability allocation.

4. Proposed Algorithm

The highlights of the method presented here can be summarized as:

1. It can find minimal paths for a specified pair of nodes.
2. Path sets are obtained in increasing order.
3. If Terminal Numbering Convention is used, where the numbering begins at source node and terminates at

terminal node and it is desired to enumerate all the path sets in terms of branch/edge numbers

Step 1: Mark the arcs in network from 1 to n , source nodes being S_1, S_2, \dots, S_n and destination nodes being $-1, -2, -3, \dots, -n$.

Step 2: Create a linked data structure $L = \{(Arcs\ of\ source\ node\ 1), (Arcs\ of\ source\ node\ 2), \dots, (Arcs\ of\ Source\ node\ N)\}, \{(Outgoing\ arcs\ from\ arc\ 1), (Outgoing\ arcs\ from\ arc2), \dots, (Outgoing\ arcs\ from\ arcN)\}$ of the network with the first "n" node of the link representing the outgoing arcs from each source node S_1 to S_n . The remaining nodes form a link with every arc being linked to its source arc.

Step 3: In order to define the minimal paths in the network, traverse the linked data structure from the source arc ($S_1 \dots S_n$) and follow until the destination is reached.

Step 4: Create a working set (R) for pointing to the current arcs being considered in order to find if the arc is one amongst the minimal path.

Step 4a: Point to the first arc in the working set (R_i).

Step 4b: Check if the arc being pointed to (p) is the destination arc. If yes, print the minimal path, pop elements from stack (S) which has been added during the current traversal, delete the current set of arcs from queue (W) and backtrack.

Step 4c: If p is not the destination arc, traverse through the working set until the node pointed by p is reached.

Step 4d: The node being pointed is the current set of arcs ($V_1 \dots V_n$) being considered for minimal paths.

Step 5: Find if the current set of arcs (V_1 to V_n) has already been visited.

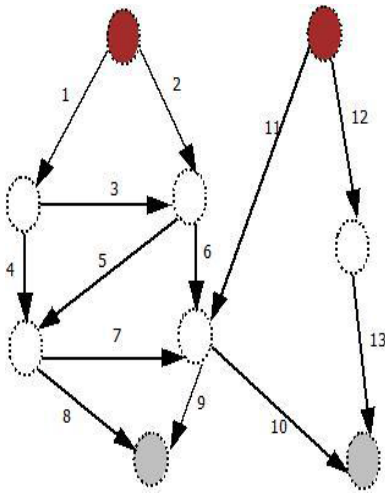
Step 5a: Check if (V_1 to V_n) is already present in the queue (W) which contains the visited set of arcs.

Step 5b: If yes, Cycle is detected. Print the cycle, add the set of arcs (V_1 to V_n) to the cycle queue (C), delete the current set of arcs from queue (W) and backtrack. If not, move to step 6.

Step 6: Push the current arc being considered (p) to Stack (S) and add the current set of arcs being considered (V_1, \dots, V_n) to the queue (W) and loop to step 4.

Step 7: Loop through 4 to 6 for rest of the arcs in working set (R_i).

Example



Step 1: Consider a network with two source (*S1 and S2*), two destination (*-1 and -2*) nodes and 13 arcs. Mark the arcs from 1 to 13.

Step 2: The linked data structure for the example network is created as follows:

$\{(1,2), (11,12), (3,4), (5,6), (7,8), (9,10), (-1, -2), (9,10), (13), \}$

Here, the first two nodes represent the outgoing arcs from source nodes while the rest of the nodes link each source arc with its outgoing arc.

Step 3: Select the first source node (*S1*) to find the minimal path to reach the destination.

Step 4: Working set is the set of arcs being considered for the first set of source nodes. In this example, the working set for the first set of source nodes is

$R = \{(1,2), (3,4), (5,6), (7,8), (9,10), (-1), (-1), (-2), (13), \}$

Point *p* to the first arc ($p=1$). Since *p* doesn't point to the destination node, traverse through *R* to the node pointed by *p*. $v = \{3, 4\}$.

Step 5: Since $v = \{3, 4\}$ has not been visited and $w = \text{NULL}$, move to step 6.

Step 6: Add 1 to *S* and *v* to *W*. Currently, $S = \{1\}$ and $W = \{(3, 4)\}$.

Step 4: Since all arcs are not yet visited in the current working set, point *p* to the arc in current working set (3, 4). So, $p=3$. Since *p* doesn't point to the destination node, traverse through *R* to the node pointed by *p*. $v = \{5, 6\}$.

Step 5: Since $v = \{5, 6\}$ has not been visited and $w = \{(3, 4)\}$, move to step 6.

Step 6: Add 3 to *S* and *v* to *W*. Currently, $S = \{3, 1\}$ and $W = \{(5, 6), (3, 4)\}$.

Step 4: Since all arcs are not yet visited in the current working set, point *p* to the arc in current working set (5, 6). So, $p=5$. Since *p* doesn't point to the destination node, traverse through *R* to the node pointed by *p*. $v = \{7, 8\}$.

Step 5: Since $v = \{7, 8\}$ has not been visited and $w = \{(3, 4), (5, 6)\}$, move to step 6.

Step 6: Add 5 to *S* and *v* to *W*. Currently, $S = \{5, 3, 1\}$ and $W = \{(3, 4), (5, 6), (7, 8)\}$.

Step 4: Since all arcs are not yet visited in the current working set, point *p* to the arc in current working set (7, 8). So, $p=7$. Since *p* doesn't point to the destination node, traverse through *R* to the node pointed by *p*. $v = \{9, 10\}$.

Step 5: Since $v = \{9, 10\}$ has not been visited and $w = \{(3, 4), (5, 6), (7, 8)\}$, move to step 6.

Step 6: Add 7 to *S* and *v* to *W*. Currently, $S = \{7, 5, 3, 1\}$ and $W = \{(3, 4), (5, 6), (7, 8), (9, 10)\}$.

Step 4: Since all arcs are not yet visited in the current working set, point *p* to the arc in current working set (9, 10). So, $p=9$. Since *p* doesn't point to the destination node, traverse through *R* to the node pointed by *p*. $v = \{-1\}$.

Step 5: Since $v = \{-1\}$ has not been visited and $w = \{(3, 4), (5, 6), (7, 8), (9, 10)\}$, move to step 6.

Step 6: Add 9 to *S* and *v* to *W*. Currently, $S = \{9, 7, 5, 3, 1\}$ and $W = \{(3, 4), (5, 6), (7, 8), (9, 10), (-1)\}$.

Step 4: Since all arcs are not yet visited in the current working set, point *p* to the arc in current working set (-1). So, $p=-1$. Since *p* has now reached the destination, pop elements from stack and delete the nodes from *W*. So, Minimal Path is {1, 3, 5, 7, 9}. Backtrack to the node (3,4) to consider the other outgoing arcs from the arc 3.

Loop through for the rest of arcs to get the minimal path for first source (*S1*) to be:

$\{(1,3,5,7,9), (1,3,5,7,10), (1,3,5,8), (1,3,6,9), (1,3,6,10), (1,4,7,9), (1,4,7,10), (1,4,8), (2,5,7,9), (2,5,7,10), (2,5,8), (2,6,9), (2,6,10), (11,9), (11,10), (12,13)\}$.

Step 3 and 4: Once the minimal paths for first source is found out, move towards the second source and create the working set.

$R = \{(11,12), (3,4), (5,6), (7,8), (9,10), (-1), (-2), (13), \}$

Follow the steps from 4 to 7 for this working set to find the minimal paths for source (*S2*) to be:

$\{(9, 11), (10, 11), (12, 13)\}$

4. Result

We design the network models and algorithms and take as input and produce as output of network design model. We have found several minimal paths from proposed algorithm and implemented C language. This is short and simple method to determine minimal path for make the system highly reliable and no need to determine cost matrix and probability of components to make the system highly reliable. Minimal Paths and Minimal Cuts are the most popular connectivity methods and many researchers address many network design problems. This work proposes an algorithm to generate all the minimal paths of the general flow network, which is a new approach and it is based on the principle of backtracking further we can evaluate the network reliability using any existing SDP based approach. Here, we used term general means that through this approach we can find the minimal paths with multi sources and multi sinks in the network.

5. Future Scope & conclusion

This paper concerns with the development of efficient algorithm for enumerating multi source and multi terminal path sets and reliability evaluation of complex networks and implementation of these algorithms in a computer program. Its has been solving examples of large networks of various sizes and complexity. The algorithm can be implemented using the simplest data-structure-arrays, in any programming language. This work proposes an algorithm to generate all the minimal paths of the general flow network, which is a new approach and it is based on the principle of backtracking further we can evaluate the network reliability.

References

- [1]. R. C. Chen and C. M. Lin, "Adaptive end-to-end delay equalizations for TCP virtual path transmissions in Internet environments", *International Journal of Innovative Computing, Information and Control*, vol.6, no.3 (A), pp.1069-1078, 2010.
- [2]. J. F. Lin, "Performance analysis and discussion on a heuristic approach for Scheduling multiprocessor tasks in a grid computing environment", *International Journal of Innovative Computing, Information and Control*, vol.6, no.12, pp.5451- 5462, 2010.

- [3]. J. F. Lin, "Scheduling parallel tasks with intra-communication overhead in a grid Computing Environment", *International Journal of Innovative Computing, Information and Control*, vol.7, no.2 pp.881-896, 2011.
- [4]. C. C. Jane, J. S. Lin and J. Yuan, "Reliability evaluation of a limited-own network in terms of minimal cutsets", *IEEE Transactions on Reliability*, vol.42, no.3, pp.354-361, 1993.
- [5]. W. C. Yeh, "Multistate network reliability evaluation under the maintenance cost constraint", *International Journal of Production Economics*, vol.88, no.1, pp.73-83, 2004.
- [6]. L. D. Bodin, B. L. Golden, A. A. Assad and M. O. Ball, "Routing and scheduling of vehicles and crews" The state of the art, *Computers and Operations Research*, vol.10, no.2, pp.63-212, 1983.
- [7]. B. L. Golden and T. L. Magnanti, "Deterministic network optimization: A bibliography, *Networks*", vol.7, no.2, pp.149-183, 1977.
- [8]. Y. K. Lin, "Reliability evaluation for an information network with node failure under cost Constraint", *IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans*, vol.37, no.2, pp.180-188, 2007.