Reliability Engineering and System Safety 150 (2016) 1–10

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

An improved algorithm for finding all minimal paths in a network

Guanghan Bai, Zhigang Tian*, Ming J. Zuo

Department of Mechanical Engineering, University of Alberta, Edmonton, Canada

ARTICLE INFO

Article history:
Received 20 September 2015
Received in revised form
5 January 2016
Accepted 8 January 2016
Available online 1 February 2016

Keywords:
Minimal path
Backtracking
Network reliability
Linked path structure
Recursive algorithm

ABSTRACT

Minimal paths (MPs) play an important role in network reliability evaluation. In this paper, we report an efficient recursive algorithm for finding all MPs in two-terminal networks, which consist of a source node and a sink node. A linked path structure indexed by nodes is introduced, which accepts both directed and undirected form of networks. The distance between each node and the sink node is defined, and a simple recursive algorithm is presented for labeling the distance for each node. Based on the distance between each node and the sink node, additional conditions for backtracking are incorporated to reduce the number of search branches. With the newly introduced linked node structure, the distances between each node and the sink node, and the additional backtracking conditions, an improved backtracking algorithm for searching for all MPs is developed. In addition, the proposed algorithm can be adapted to search for all minimal paths for each source–sink pair in networks consisting of multiple source nodes and/or multiple sink nodes. Through computational experiments, it is demonstrated that the proposed algorithm is more efficient than existing algorithms when the network size is not too small. The proposed algorithm becomes more advantageous as the size of the network grows.

1. Introduction

Minimal paths (MPs) play an important role in reliability evaluation for binary networks [14]. They can also be used in reliability evaluation for multistate networks [1,2]. Given a two-terminal network with one source node and one sink node, a path is a sequence of links or nodes that connect the source node to the sink node. A minimal path is such a path that removing any link or node will make it no longer a path. Given both the network and its components can only take two possible states, completely working or totally failed, the network is said to be operative if there is at least one operating path between the source and the sink [3]. Due to combinatorial explosion, finding all the MPs is an NP-hard problem. Over the past decades, many algorithms have been proposed to improve the efficiency for finding all the MPs. Chen and Lin [4] classified these algorithms into three types, including symbolic expression-based search algorithm [5,6], augmentation-based search algorithm [7,8], and directed search-based search algorithm [9–11].

The symbolic expression-based algorithm represents the paths as symbolic forms and generates MPs based on Boolean algebra and/or other algebraic operators. Rai and Aggarwal [5] proposed a symbolic expression-based method for finding all MPs using path polynomial and Boolean functions. Based on Universal Generating Function Method (UGFM), Yeh [6] defined a generalized composition operator to find all MPs. The augmentation-based search algorithm is a type of heuristic algorithm to search for MPs. It starts with only all the nodes of the original network, and add the link one at a time. Each time a link in included, new MPs are generated. All MPs can be found when all the links have been included. This heuristic algorithm was first proposed by Al-Ghanim [7]. Yeh [8] further improved Al-Ghanim's algorithm by eliminating the chance of generating duplicate MPs. The direct search-based algorithm implements the depth-first search (DFS) mechanism to find all the MPs, which is a simple and efficient search strategy in computer algorithm. It also uses simple data structures such as connection matrix, and linked list to represent the network. Most of the reported studies on finding MPs are within this category. Based on graph theory and dual principle, Shen [10] proposed an DFS algorithm to search all MPs using connection matrix. Kobayashi and Yamamoto [11] further improved Shen's algorithm by incorporating additional processes base on the level set of nodes. Colbourn [9] reported an DFS algorithm which has a time complexity of $O(m \cdot \pi)$, where n denotes the number of links, and π denotes the total number of MPs. Chen and Lin [4] proposed another DFS algorithm using backtracking, which is more efficient than Colbourn's algorithm. To the authors' knowledge, currently, Chen and Lin's algorithm is the best known DFS algorithm.

---

* Corresponding author. Tel.: +1 7804922822; fax: +1 7804922200.
E-mail address: ztian@ualberta.ca (Z. Tian).

http://dx.doi.org/10.1016/j.ress.2016.01.011
0951-8320/

Chen and Lin's algorithm [4] uses a linked path structure to represent a network in its directed from and conducts the search algorithm based on the concept of backtracking. It starts by picking one outgoing link from the source node and visiting the node pointed by the link. Then it picks one outgoing link of that visited node and visits the node pointed by that link, and so on. When the sink is reached, or a cycle is detected, or no links can be found to proceed, it will backtrack to the previous node and pick another outgoing link to restart the visit. Each time it reaches the sink, one MP is found. All MPs can be found when all the links have been visited. The pseudo-code of Chen and Lin's algorithm can be found in Section 3 of their paper [4]. We focus on “Algorithm 2” in [4], which searches for all MPs only.

However, the current version of Chen and Lin's algorithm [4] contains two limitations. The first limitation is the input data structure, which is shown in Step 1 of “Algorithm 2” in Ref. [4]. The input data structure is the linked path structure indexed by links, which only accepts directed networks. For undirected network, a transformation is required which adds one reverse directed link for each link that does not connect to the source node. Each time a MP is found, it needs to be transformed back to the undirected form, which is shown in Step 5 of “Algorithm 2” [4]. These transformations not only consume additional computational effort, but also create additional links for the network, resulting in more computational consumption. A MP can actually be represented either by a sequence of links or by a sequence of nodes. If we use a linked path structure indexed by nodes, it can accept directed network, undirected network, and mixed network. Thus, there is no need to create additional links and perform transformations, which can save a lot of computational effort. In addition, MP represented by a sequence of nodes contains both the nodes and the links (as pairs of ordered nodes), which allows both nodes and links to be failure prone in the network.

Another limitation of Chen and Lin's algorithm [4] is the lack of backtracking conditions. There are three conditions when a backtrack is triggered in the current version of Chen and Lin’s algorithm [4]: 1) a cycle is detected, as shown in Step 7 of “Algorithm 2” [4]; 2) the sink node is reached, i.e. an MP is found, as shown in Step 5 of “Algorithm 2” [4]; 3) No nodes can be found to proceed, as shown in Step 9 of “Algorithm 2” [4]. However, the three backtracking conditions did not take the structure of the network into full consideration. When all nodes that are adjacent to the sink have been visited already, it will keep visiting other nodes. It also keeps visiting other nodes when all nodes that are closer to the sink have been visited already. As pointed out by Kobayashi and Yamamoto [11], if all nodes with equal distance to the sink node have been searched under the current searching branch, searching nodes with larger distance is useless. Thus, under those search branches, no MPs can be found. As the network size becomes large, more and more search branches as such are in fact useless for the purpose of searching for MPs. Thus, there is a need to introduce additional backtracking conditions for Chen and Lin’s algorithm to eliminate these search branches.

With the two observations above, we believe that Chen and Lin’s algorithm [4] can be further improved. We first limit our discussions to two-terminal networks, and then extend to networks consisting of multiple source nodes and/or multiple sink nodes. The rest of the paper is organized as follows. The linked path structure indexed by nodes is introduced in Section 2. The definition of distance between each node and the sink node is given, together with a simple algorithm for labeling the distance of each node. Based on the definition of distance, a property is proposed for additional backtracking condition. Section 3 presents the improved algorithm to search for all MPs, together with an illustrative example. The space and time complexity of the proposed algorithm is also analyzed. Section 4 compares the efficiency of proposed algorithm with that of Chen and Lin’s algorithm [4] for two-terminal networks. An extension of the proposed algorithm to search for all the MPs in networks consisting of multiple source nodes and/or multiple sink nodes is discussed in Section 5. Section 6 concludes the study. This study is based on the work presented in the conference paper by Bai et al. [12].

1.1. Notations

$G=(V,E)$: the given network, where V is the set of all nodes, and E is the set of all links.
n: the number of nodes in the network.
m: the number of links in the network.
π: the number of MP in the network.
λ: the average number of links for each MP.
η: the average number of nodes for each MP.
s: the source node.
t: the sink node.
d: the distance between a pair of nodes.
$c/\overline{c}$: the number of cycles contained in the network/ the number of cycles visited by proposed algorithm.
$u/\overline{u}$: the current visiting node/ unpicked adjacent/outgoing node of the current node
$Ind/\overline{Ind}$: the set containing cardinality of each current node/ updated node.
P: the minimal path set.
$W=\{v_0,v_1,v_2,\cdots,v_n\}$: the linked path structure by links, where v_0 represents outgoing links from the source node, v_i, $i=1,2,\cdots n$ represents the outgoing links from the node pointed by link i.
$L=\{u_0,u_1,u_2,\cdots,u_n\}$: the linked path structure indexed by nodes, where u_0 represents adjacent nodes and outgoing nodes from the source node, u_i, $i=1,2,\cdots n$ represents the non-source adjacent nodes of node i if the link connected to node i is undirected, and the outgoing nodes of node i if the link connected to node i is directed.
$L_d=(d_0,d_1,d_2,\cdots,d_n)$: the distance between each node and the sink node, where d_0 is the distance between the sink node and the source node, and d_i is the distance between the sink node and node i
$Q=\left\{q_0,q_1,q_2,\cdots q_{d^{\max}}\right\}$: the set containing the numbers of nodes with the same distance, where q_i,$i=0,1,2,\cdots,d^{\max}$, is the number of nodes with distance $d=i$, and$d^{\max}=\max(L_d)$.
$S=\left\{s_0,s_1,s_2,\cdots,s_{d^{\max}}\right\}$: the distance checking list, where $s_i=0$, $i=0,1,2,\cdots,d^{\max}$, is the number of nodes with distance i that have been visited under the current search branch.

1.2. Assumptions

1) The network is coherent, in which the improvement of any component does not degrade the performance of the network [17].
2) The network contains no parallel links. For networks with parallel links, a reliability-preserving parallel reduction technique reported in [3] can be used to replace such parallel links by one single link.
3) There are no common-cause outages in the network [8].
4) The network satisfies the flow-conservation law [16].
5) There are no loops in the network [6].

2. Preliminaries

In this section, we focus on two-terminal networks. The linked path structure indexed by nodes is introduced. The definition of