

# An improved algorithm for the quickest path reliability problem

Majid Forghani-elahabad<sup>1</sup>

CMCC, Universidade Federal do ABC, Santo André, SP

**Abstract.** The quickest path reliability problem (QPRP) emerged for extending the quickest path problem to the system reliability. This problem aims to assess the reliability of a multi-state network (MSN) for transmitting a given  $d$  units of a commodity from a source node to a destination node through one single path in the network within  $T$  units of time. Most of the proposed algorithms in the literature for this problem need all of the network's minimal paths as an input. Here, we propose a fast algorithm based on the node-child matrix of the network to address the QPRP that does not need any minimal paths in advance. We discuss the correctness of the algorithm and provide two benchmark examples to illustrate the algorithm and explain its superiority compared to some existing algorithms in the literature.

**Keywords.** Quickest path reliability problem, Minimal paths, Algorithms.

## 1 Introduction

The shortest path problem (SPP) is an old and very attractive problem in the operations research area [1]. Several variants of this problem, including the quickest path problem (QPP), have been studied [1, 9]. A multi-state network (MSN) is a network whose arcs (and possibly nodes) may have several states [2, 8, 12, 14, 19], and accordingly, the arcs' capacities are stochastic. In a dynamic network, the arcs have another attribute called *lead time* which refers to the needed time for transmitting one unit of a commodity from the source node to the destination node [11]. In such a case, for a deterministic network whose arcs' capacities are fixed, the SPP turns to the QPP, whose aim is to determine a path through which  $d$  units of commodities can be transmitted from a source node to a sink node within the minimum time [1, 5, 9, 11, 15–17].

In an MSN, the arcs' capacities vary randomly; therefore, the required time to transmit  $d$  units of a commodity through a path varies. In such a case, the QPP was extended to the network reliability leading to the quickest path reliability problem (QPRP) [13]. This problem assesses the probability of sending  $d$  units of the commodity through one path from the source node to the sink node within  $T$  units of time, that is  $R_{d,T}$  [18]. Many studies have been made on this problem so far [4, 5, 7, 9–11, 13, 18]. However, in the proposed algorithms, it is usually required to have all the minimal paths (MP) of the network in advance. Here, we first provide some preliminaries in Section 2. Then, in Section 3, we propose a simple improved algorithm by employing the node-child matrix structure of an MSN, which prevents the need for any minimal paths in advance. We employ a benchmark example to illustrate the algorithm and show its efficiency compared to the existing algorithms in the literature. Moreover, we show the algorithm to be correct.

---

<sup>1</sup>m.forghani@ufabc.edu.br

## 2 Preliminaries

Let  $G(N, A, M, L)$  be a multi-state network (MSN) with  $N = \{1, 2, \dots, n\}$  as the set of nodes (so,  $n$  is the number of nodes),  $A = \{a_1, a_2, \dots, a_m\}$  as the set of arcs (so,  $m$  is the number of nodes),  $M = (M_1, M_2, \dots, M_m)$  as the maximum capacity vector in which the integer-valued number  $M_i$  denotes the maximum capacity of arc  $a_i$ , for  $i = 1, 2, \dots, m$ ,  $L = (l_1, l_2, \dots, l_m)$  as the lead time vector in which integer-valued number  $l_i$  denotes the lead time of arc  $a_i$ , for  $i = 1, 2, \dots, m$ . Nodes 1 and  $n$  are the source and destination nodes, respectively. Let also  $X = (x_1, \dots, x_m)$  be the current system state vector in which  $x_i$  shows the current capacity of arc  $a_i$ , taking values from  $\{0, 1, \dots, M_i\}$ , for  $i = 1, 2, \dots, m$ .

The number of incoming arcs to a node is called its *in-degree* and the number of outgoing arcs from a node its *out-degree*. Although the considered MSNs here are undirected, we note that the in-degree of node 1 and the out-degree of node  $n$  are zero. Let  $I_i$  and  $O_i$  be respectively *in-degree* and *out-degree* of node  $a_i$ , for  $i = 1, 2, \dots, m$ . Therefore, we have  $I_1 = O_n = 0$  and  $I_i = O_i$ , for  $i = 2, 3, \dots, n - 1$ , for any MSN in this work. A path is a set of adjacent arcs through which the commodity can be sent from node 1 to node  $n$ . A minimal path is a path with no proper subsets being path anymore. Let  $P_1, \dots, P_h$  be all the minimal paths in the network. The capacity of a minimal path is the minimum capacity of its links and the lead time of a minimal path is the sum of the lead times of its links. Let  $CP_j(X)$  be the capacity of  $P_j$  under system state vector of  $X$  and  $LP_j$  be the lead of  $P_j$ , for  $j = 1, \dots, h$ . Hence, assuming that  $P_j = \{a_{j_1}, a_{j_2}, \dots, a_{j_{n_j}}\}$  is a minimal path and  $X = (x_1, x_2, \dots, x_m)$  is the current system state vector, we have [11]

$$CP_j(X) = \min\{x_{j_1}, x_{j_2}, \dots, x_{j_{n_j}}\}, \quad LP_j = \sum_{r=1}^{n_j} l_{j_r}. \quad (1)$$

The transmission time to transmit  $d$  units of data on arc  $a_i$  with capacity of  $x_i$ , lead time of  $l_i$  is equal to

$$t = l_i + \lceil \frac{d}{x_i} \rceil, \quad (2)$$

where  $\lceil \eta \rceil$  is the smallest integer number not less than  $\eta$ . Similarly, the required time to transmit  $d$  units of the commodity (data or flow) from the source to the destination through  $P_j = \{a_{j_1}, a_{j_2}, \dots, a_{j_{n_j}}\}$  under  $X$  is equal to [11]

$$\tau(P_j, d, X) = LP_j + \lceil \frac{d}{CP_j(X)} \rceil. \quad (3)$$

As it is assumed that the commodity can be transmitted through only one path, the transmission time to send  $d$  units of the commodity through the network from node 1 to node  $n$  under  $X$  is equal to

$$Z(d, X) = \min_{j=1:h} \tau(P_j, d, X). \quad (4)$$

Assuming that  $\Psi_d = \{X \leq M \mid Z(d, X) \leq T\}$  and  $\Psi_{d,\min} = \{X^1, X^2, \dots, X^\sigma\}$  the set of all the minimal vectors in  $\Psi_d$ , and letting  $A_r = \{X \mid X \geq X^r\}$ , for  $r = 1, 2, \dots, \sigma$ , we have [5]

$$R_{d,T} = \Pr \cup_{r=1}^{\sigma} A_r = \sum_{r=1}^{\sigma} \Pr(A_r) - \sum_{l=2}^{\sigma} \sum_{r=1}^{l-1} \Pr(A_r \cap A_l) + \dots + (-1)^{\sigma+1} \Pr(\cap_{r=1}^{\sigma} A_r), \quad (5)$$

where  $\Pr(A_r) = \sum_{X \in A_r} \Pr(X)$  and  $\Pr(X) = \prod_{i=1}^m \Pr(x_i)$ . As a result, with the set of  $\Psi_{d,\min}$  at hand, the rest of the reliability assessment of the network is just calculating a union probability given in Equation (5). Hence, the focus of our proposed algorithm is to determine the vectors of  $\Psi_{d,\min}$ , which are called *d-QRPs*. In the next section, we propose an efficient algorithm to find all the *d-QRPs* in an MSN.

### 3 The proposed algorithm

One can use the Equations (1) and (3) to check the conditions  $LP_j < T$  and  $\tau(P_j, d, M) \leq T$ , respectively, for a given minimal path, say  $P_j$ . If it satisfies the conditions, then one can determine the minimal vector,  $X^j = (x_1, x_2, \dots, x_m)$ , corresponding to  $P_j$  through the following system [5]

$$\text{For } i = 1, 2, \dots, m, x_i = \begin{cases} \lceil \frac{d}{T-LP_j} \rceil & \text{if } a_i \in P_j, \\ 0 & \text{if } a_i \notin P_j. \end{cases} \quad (6)$$

Without loss of generality, let  $P_1, P_2, \dots, P_\sigma$  be all the minimal paths which satisfy the conditions. One notes that  $\sigma \leq h$ . This way, one can see that  $\Psi_{d,\min} = \{X^1, \dots, X^\sigma\}$ , where  $X^j$  is the minimal vector corresponding to  $P_j$ , for  $j = 1, \dots, \sigma$  [11, 17]. Therefore, one approach is first to find all the network's minimal paths, then check everyone for the conditions mentioned above, and finally calculate the corresponding vectors through Equation (6). However, the weakness of this approach is that it needs all the minimal paths in advance. Here, by using the node-child matrix of an MSN [3], we propose an efficient algorithm that does not need any minimal paths in advance. The node-child matrix of an MSN is an  $n \times q$  matrix, where  $q = \max\{O_i \mid i = 1, 2, \dots, n - 1\}$ . Each row in the matrix corresponds to a node in the network and shows its children. For instance, the node-child matrix associated with the given network in Figure 1 is the following.

$$\mathbf{B} = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

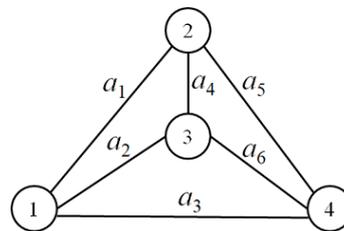


Figure 1: A benchmark network example taken from [6].

We note that the *out-degree* of the different nodes are not necessarily equal, and hence we put 0 in the node-child matrix when the *out-degree* of the corresponding node is less than  $q$ . For instance, in Figure 1, we have  $q = O_1 = 3$ , however, as  $O_2 = O_3 = 2$ , we put a 0 in the last column of the corresponding rows to the nodes 2 and 3 in the network. Moreover, as always  $O_n = 0$ , the last row in the node-child matrix is always a zero row (refer to [3] for more details on the concept of the node-child matrix). Having this matrix at hand, one can use a backtracking procedure to find all the minimal paths [3]. We use this procedure to find all the  $d$ -QRPs by adding one more condition to the procedure for checking the lead time of the under-construction minimal path. Whenever its lead time equals  $T$ , we stop the construction and go back to construct the following minimal path.

According to Equation (1), the capacity of a minimal path decreases during its construction. By adding a new arc to a path, its capacity may stay unchanged or be decreased. At the same time, by adding a new arc to a path, its lead time increases. As a result, according to Equation (3), the transmission time increases during the construction of a minimal path and may pass the threshold  $T$ . Therefore, the algorithm checks for the lead time after adding each arc to the

under-construction path. Accordingly, the proposed algorithm, Algorithm 1 below, is stated. In this algorithm,  $L(s, t)$  denotes the lead time of the arc from node  $s$  to node  $t$ .

**Algorithm 1**

**Input:** A multi-state network  $G(N, A, M, L)$  with time limit of  $T$  and demand value of  $d$ .

**Output:** All the  $d$ -QRPs

**Step 0.** Let  $\Psi = \{\}$ ,  $i = 1, s = 1, k = 1, f(r) = 1$  for  $r = 1, \dots, n, lt = 0, kap = \infty, Rp = 0, Q = (1, 0, \dots, 0)$  and  $K = (0, 0, \dots, 0)$ , where  $Q$  and  $K$  are  $n$ -tuple vectors.

**Step 1.** Determine the node-child matrix  $B$ .

**Step 2.** Let  $t = B(s, f(s))$ .

**Step 3.** If  $t$  is equal to one of the components in  $Q$ , then it is repeated, let  $Rp = 1$ .

**Step 4.** If  $t = 0$ , then

**Step 4.1** If  $s = 1$ , then stop.  $\Psi$  is the set of all the  $d$ -QRPs.

**Step 4.2** If  $s = n$ , then  $Q$  is a desired minimal path. Determine its corresponding system state vector,  $X^k$ , as follows:

$$x_{i,j} = \begin{cases} \alpha, & \text{if } (i, j) \in Q \\ 0. & \text{otherwise.} \end{cases} \tag{7}$$

Add  $X^k$  into  $\Psi$ . Now, if  $i = 2$ , then stop.  $\Psi$  is the set of all the solutions. Otherwise, let  $k = k + 1, f(Q(i - 1)) = 1, lt = lt - L(Q(i - 1), Q(i)) - L(Q(i - 2), Q(i - 1)), Q(i) = 0, Q(i - 1) = 0, K(i - 1) = 0, K(i - 2) = 0, s = Q(i - 2)$ , and  $i = i - 2$ . Now, if  $i = 1$ , let  $kap = \infty$ , else  $kap = \min\{K(j) \mid j = 1, \dots, i - 1\}$ . Go to Step 2.

**Step 4.3.** If  $s \neq 1, n$ , then let  $f(s) = 1, lt = lt - L(Q(i - 1), Q(i)), Q(i) = 0, K(i - 1) = 0, s = Q(i - 1)$ , and  $i = i - 1$ . Now, if  $i = 1$ , let  $kap = \infty$ , else  $kap = \min\{K(j) \mid j = 1, \dots, i - 1\}$ . Go to Step 2.

**Step 5.** If  $Rp = 1$ , then let  $f(s) = f(s) + 1, Rp = 0$ , and go to Step 2.

**Step 6.** If  $lt + L(s, t) < T$ , then let  $\alpha = \lceil \frac{d}{T - lt - L(s, t)} \rceil$ . If  $lt \geq T - L(s, t)$  or  $\alpha > \min\{kap, M(s, t)\}$ , then let  $f(s) = f(s) + 1$ , else let  $lt = lt + L(s, t), kap = \min\{kap, M(s, t)\}, K(i) = M(s, t), f(s) = f(s) + 1, i = i + 1, Q(i) = t$ , and  $s = t$ . Go to Step 2.

Algorithm 1 is a developed algorithm based on the proposed algorithm in [3]. The algorithm checks the required conditions during the construction of each minimal path to ensure that only the valid minimal paths are constructed. Besides, it checks that the  $d$ -QRP, say  $X^j$ , corresponding to the constructed minimal path, satisfies the condition  $X^j \leq M$ . Therefore, Algorithm 1 calculates all the  $d$ -QRPs correctly, and the following result is at hand.

**Theorem 3.1.** Algorithm 1 calculates all the  $d$ -QRPs for a given MSN  $G(N, A, M, L)$  with a time limit of  $T$  and demand value of  $d$ .

**3.1 An illustrative example**

Figure 1 depicts a network with  $N = \{1, 2, 3, 4\}$  and  $A = \{a_1, a_2, a_3, a_4\}$ . Let  $M = (5, 4, 6, 4, 3, 6)$  and  $L = (4, 4, 1, 4, 3, 1)$  respectively be the maximum capacity and lead time vectors. Use Algorithm 1 to calculate all the  $d$ -QRPs for this network assuming  $T = 7$  and  $d = 4$ .

**Solution:**

Step 0. Let  $\Psi = \{\}$ ,  $i = 1, s = 1, k = 1, f(r) = 1$  for  $r = 1, 2, 3, 4, lt = 0, kap = \infty, Rp = 0, Q = (1, 0, 0, 0)$  and  $K = (0, 0, 0, 0)$ .

Step 1.

$$\mathbf{B} = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Step 2. Let  $t = B(s, f(s)) = B(1, 1) = 2$ .

Step 3.  $t = 2$  is not equal to any component of  $Q$ .

Step 4.  $t = 2 \neq 0$ .

Step 5.  $Rp = 0 \neq 1$ .

Step 6. Since  $lt + L(1, 2) = 0 + 4 = 4 < 7$ , let  $\alpha = \lceil \frac{4}{7-0-4} \rceil = 2$ . Since  $lt = 0 < 3 = T - L(1, 2)$  and  $\alpha = 2 < 5 = \min\{kap, M(1, 2)\}$ , let  $lt = lt + L(1, 2) = 0 + 4 = 4$ ,  $kap = \min\{kap, M(1, 2)\} = 5$ ,  $K(1) = M(1, 2) = 5$ ,  $f(1) = f(1) + 1 = 2$ ,  $i = i + 1 = 2$ ,  $Q(2) = 2$ , and  $s = 2$ . The transfer is made to Step 2.

Step 2. Let  $t = B(2, f(2)) = B(2, 1) = 3$ .

Step 3.  $t = 3$  is not equal to any component of  $Q$ .

Step 4.  $t = 3 \neq 0$ .

Step 5.  $Rp = 0 \neq 1$ .

Step 6.  $lt + L(2, 3) = 4 + 4 = 8 \not< 7$ . Since  $lt = 4 \geq 3 = T - L(2, 3)$ , let  $f(2) = f(2) + 1 = 2$ . The transfer is made to Step 2.

⋮

The final solutions are  $\{(0, 2, 0, 0, 0, 2), (0, 0, 1, 0, 0, 0)\}$ .

There are only five minimal paths in this network example. It is not very time-consuming to determine all the minimal paths, check each for the required conditions, and calculate the corresponding  $d$ -QRPs to the valid minimal paths. However, we note that the number of minimal paths in large-sized networks increases exponentially with the network's size. Accordingly, it is not so efficient to determine all the minimal paths while a few of them are valid. For instance, there are 1274 minimal paths for the given network in Figure 2 while with  $M = (14, 18, 19, 16, 44, 26, 32, 40, 20, 49, 49, 30, 10, 45, 26, 35, 22, 48, 26, 33, 20, 19, 11, 33, 43, 41, 48, 40, 37, 40, 30, 29, 36, 43, 15, 10, 25, 27, 42, 45)$ ,  $L = (9, 5, 8, 8, 5, 5, 5, 6, 7, 7, 5, 10, 7, 7, 10, 10, 8, 10, 8, 10, 8, 6, 9, 10, 9, 10, 6, 6, 9, 5, 9, 9, 9, 9, 10, 6, 9, 7, 9, 9)$ ,  $T = 64$ , and  $d = 11$ , there are only nine  $d$ -QRPs for this network. Hence, Algorithm 1 is superior to the existing algorithms in the literature that need all the minimal paths in advance.

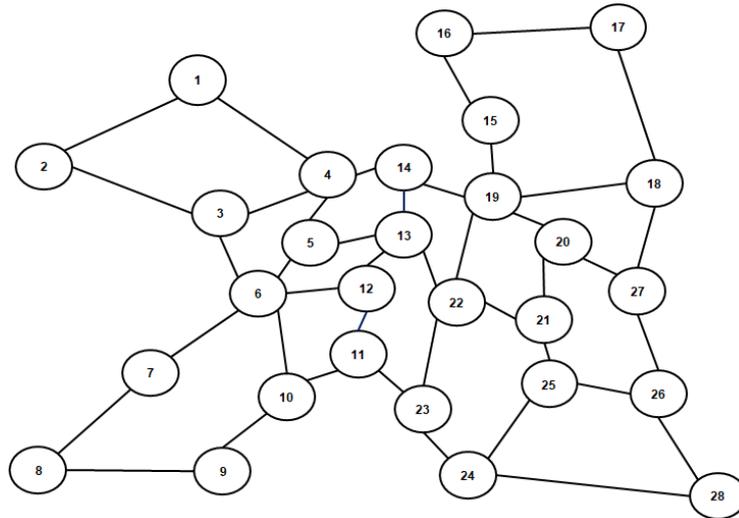


Figure 2: Pan European benchmark example.

## 4 Conclusions

The quickest path reliability problem is an exciting problem that extends the quickest path problem to the system reliability. The problem aims to evaluate the reliability of a multi-state network for transmitting  $d$  units of a commodity from a source node to a destination node within  $T$  units of time. Several proposed algorithms in the literature on this problem contain three main stages; (1) determine all the minimal paths for the network, (2) check every minimal path for the required conditions to find the valid ones, and (3) calculate the corresponding system state vectors to the valid minimal paths. However, determining all the minimal paths in the network is NP-hard by itself. Here, we proposed an approach that uses the structure of the node-child matrix to solve the problem without the need for any minimal path in advance. We demonstrated the correctness of the algorithm and illustrated it through a benchmark example. Moreover, we provided a large-sized benchmark to show the superiority of our proposed algorithm.

## Acknowledgement

The author thanks CNPq (grant 306940/2020-5) and the Federal University of ABC for supporting this work.

## References

- [1] Yen L Chen and Yeo H Chin. “The quickest path problem”. In: **Computers & Operations Research** 17.2 (1990), pp. 153–161.
- [2] H Salehi Fathabadi and M Forghani-elahabadi. “A note on “A simple approach to search for all d-MCs of a limited-flow network””. In: **Reliability Engineering & System Safety** 94.11 (2009), pp. 1878–1880.
- [3] H Salehi Fathabadi et al. “Determining all minimal paths of a network”. In: **Australian Journal of Basic and Applied Sciences** 3.4 (2009), pp. 3771–3777.

- [4] M Forghani-elahabad and N Mahdavi-Amiri. “A simple algorithm to find all minimal path vectors to demand level  $d$  in a stochastic-flow network”. In: **5-th Iranian Conference on Applied Mathematics, September 2–4, 2013 Bu-Ali Sina University**. 2013.
- [5] Majid Forghani-Elahabad. “3 The Disjoint Minimal Paths Reliability Problem”. In: **Operations Research**. CRC Press, 2022, pp. 35–66.
- [6] Majid Forghani-elahabad and Nelson Kagan. “Assessing reliability of multistate flow networks under cost constraint in terms of minimal cuts”. In: **International Journal of Reliability, Quality and Safety Engineering** 26.05 (2019), p. 1950025.
- [7] Majid Forghani-elahabad and Nezam Mahdavi-Amiri. “A New Algorithm for Generating All Minimal Vectors for the  $q$  SMPs Reliability Problem With Time and Budget Constraints”. In: **IEEE Transactions on Reliability** 65.2 (2015), pp. 828–842.
- [8] Majid Forghani-elahabad and Nezam Mahdavi-Amiri. “A new efficient approach to search for all multi-state minimal cuts”. In: **IEEE Transactions on Reliability** 63.1 (2014), pp. 154–166.
- [9] Majid Forghani-elahabad and Nezam Mahdavi-Amiri. “An efficient algorithm for the multi-state two separate minimal paths reliability problem with budget constraint”. In: **Reliability Engineering & System Safety** 142 (2015), pp. 472–481.
- [10] Majid Forghani-Elahabad, Nezam Mahdavi-Amiri, and Nelson Kagan. “On multi-state two separate minimal paths reliability problem with time and budget constraints”. In: **International Journal of Operational Research** 37.4 (2020), pp. 479–490.
- [11] Majid Forghani-elahabad and Wei-Chang Yeh. “An improved algorithm for reliability evaluation of flow networks”. In: **Reliability Engineering & System Safety** (2022), p. 108371.
- [12] Zhifeng Hao et al. “General multi-state rework network and reliability algorithm”. In: **Reliability Engineering & System Safety** 203 (2020), p. 107048.
- [13] Yi-Kuei Lin. “Extend the quickest path problem to the system reliability evaluation for a stochastic-flow network”. In: **Computers & Operations Research** 30.4 (2003), pp. 567–575.
- [14] Seyed Mehdi Mansourzadeh et al. “A Comparative Study of Different Approaches for Finding the Upper Boundary Points in Stochastic-Flow Networks”. In: **International Journal of Enterprise Information Systems (IJEIS)** 10.3 (2014), pp. 13–23.
- [15] Yi-Feng Niu, Can He, and De-Qiang Fu. “Reliability assessment of a multi-state distribution network under cost and spoilage considerations”. In: **Annals of Operations Research** (2021), pp. 1–20.
- [16] Yi-Feng Niu et al. “Finding all multi-state minimal paths of a multi-state flow network via feasible circulations”. In: **Reliability Engineering & System Safety** 204 (2020), p. 107188.
- [17] Wei-Chang Yeh. “A fast algorithm for quickest path reliability evaluations in multi-state flow networks”. In: **IEEE Transactions on Reliability** 64.4 (2015), pp. 1175–1184.
- [18] Wei-Chang Yeh, Wei-Wen Chang, and Chuan-Wei Chiu. “A simple method for the multi-state quickest path flow network reliability problem”. In: **2009 8th International Conference on Reliability, Maintainability and Safety**. IEEE. 2009, pp. 108–110.
- [19] Wei-Chang Yeh et al. “Novel binary-addition tree algorithm for reliability evaluation of acyclic multistate information networks”. In: **Reliability Engineering & System Safety** 210 (2021), p. 107427.