An Algorithm for Calculating Multi-State Network Reliability with Arbitrary Capacities of the Links

Marija Mihova[1], Natasha Maksimova[2], Zaneta Popeska[1]
[1]*Faculty of natural sciences and mathematics, Skopje, Macedonia*
[2]*Faculty of Informatics, Stip, Macedonia*

***Abstract**: The problem that we regard in this paper is known as the multi-state two-terminal reliability computation, and we consider how the concept of the minimal path and cut vectors is used in modeling the reliability of these types of systems. The main focus is to develop an algorithm for obtaining minimal path vectors for multi-state two-terminal network. The proposed algorithm differs from the other known algorithms for this problem, because it does not request any restrictions for the values of the capacities of the links. Some examples to illustrate the algorithm are included.*

***Keywords:** Reliability, multi-state systems, network reliability, minimal path vectors.*

1. INTRODUCTION

Two-terminal network reliability (2TR) for binary network elements has been studied in various ways. For the binary case, it assumes a network and its components, be in either in a working or in a failed state. Nevertheless, for some systems the binary approach has proven to be insufficient in obtaining reliability models that resemble the true reliability behaviour of the system. Such systems are telecommunication systems, transportation systems, water distribution, gas and oil production and hydropower generation systems [1]. In those systems elements may operate in any of several intermediate states, and better results may be obtained using a multi-state reliability approach.

The algorithm proposed in [1] takes into consideration the multi-state nature of the network and its components. The authors developed a multi-state approach for exact computation of multi-state two-terminal reliability at demeaned level d ($M2TR_d$). $M2TR_d$ is defined as the probability that a demand of d units can be supplied from source to sink nodes through multi-state arcs. The methodology relies on a network reduction technique to obtain all possible multi-state minimal paths. It has been applied so far only to relatively small networks. In [2] is proposed other methodology that effectively solves the problem of generating multi-state minimal cut vector (MMCV) for the multi-state two-terminal network. In [3] is developed another enumerative methodology that provides the multi-state version of minimal path sets. However, weakly homogeneous components are required. That is, components can have different numbers of states, but the first j states must all be the same. Thus, for systems where components are heterogeneous the methodology may not be able to provide the exact reliability.

The main focus in this paper is to develop an algorithm for obtaining minimal path vectors for multi-state two-terminal network. The proposed algorithm differs from the other known algorithms for this problem, because it does not require any restrictions for the values of the capacities of the links.

2. BASIC DEFINITIONS

In this section we will introduce the problem of reliability of multi state networks and we will give some basic definitions. Most of the definitions which are connected with multi-state network are given in [1], [2] and [3].

Let G = (N, A) represent a stochastic capacitated network with known demand d from a specified source node s to a specified sink node t. N represents the set of nodes and A = $\{a_i \mid 1 \leq i \leq n\}$ represents the set of arcs. The set of available capacities of the arc a_i is denoted by S_i, where $S_{ij} \in \mathbf{R}$ $S_{i1}=0$ (no capacity), $S_{il}=M_i$ (full capacity) and $S_{ij}<S_{ik}$ for $j<k$. The set S_i is known as **set space** of the arc a_i. The set space of the whole network is note by S. Let x_i be the state of the arc a_i, then the vector $\vec{x} = (x_1, x_2, \ldots, x_{|A|})$ denotes the state of all the arcs of the network and it is called **state vector**. Let E be the set of all state vectors, i.e. E= $S_1 \times S_2 \times \ldots \times S_n$. The function φ: E→ S where $\varphi(\vec{x})$ is available capacity from source to sink under system state vector $\vec{x}$ is called **multi-state structure function**.

Multi-state two terminal reliability of level *d* ($M2TR_d$) is the probability that a flow equal to or greater to d can be successfully delivered form source node to sink node.

$$M2TR_d = P(\varphi(\vec{x}) \geq d)$$

A vector $\vec{y}$ is said to be less than $\vec{x}$, $\vec{y} < \vec{x}$**,** (or dominated by $\vec{x}$**)** if $\forall i,\ y_i \leq x_i$ and for some k, $y_k < x_k$.

A vector $\vec{x} \in$E is said to be a **minimal path vector to level *d*** if $\varphi(\vec{x}) \geq d$ and for every other $\vec{y} < \vec{x}$**,** $\varphi(\vec{y}) < d$.

A vector $\vec{x} \in$E is said to be a **minimal cut vector to level *d*** if $\varphi(\vec{x}) < d$ and for every other $\vec{y} > \vec{x}$, $\varphi(\vec{y}) \geq d$.

3. ALGORITHM

The proposed algorithm requires that binary minimal path sets for the system with the same structure be known a priori. Since the binary system can be regarded as a multi-state system with two possible states (0 and 1), minimal paths can have vector representations. We assume that these vectors are known. For example consider the network given by Figure 1. The minimal path sets for binary system are $\{a_1, a_2\}$, $\{a_4, a_5\}$, $\{a_1, a_3, a_5\}$ and $\{a_4, a_3, a_2\}$. If we consider the system in the multi-state context, the minimal path vectors are: (1,1,0,0,0), (0,0,0,1,1), (1,0,1,0,1) and (0,1,1,1,0), and these vectors are also minimal path vectors for level 1.

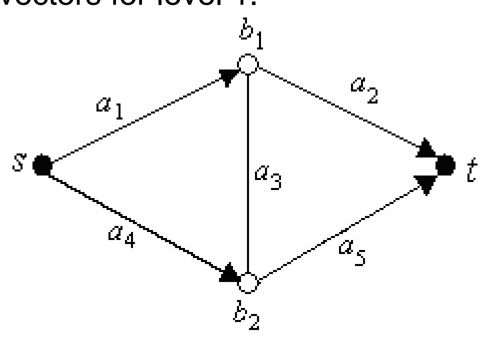

Figure 1

If a link is bidirectional we will choose one of the directions to be positive i.e. the component is equal to 1 for that direction and -1 for the other. In that way, from each minimal path we obtain a new vector, minimal direction path vector (MDP). In the Figure 1, a_3 is a bidirectional link, so the direction form b_1 to b_2 can be chosen as a positive direction and the opposite direction will be negative direction. Now, the minimal direction path vectors are: (1,1,0,0,0), (0,0,0,1,1), (1,0,1,0,1) and (0,1,-1,1,0).

Next we define some specific vectors and set that are used in the algorithm.

Definition 1: Let $\vec{x}$ be a minimal path vector of level d, then **MDP of level *d*** MPD_d is obtained from $\vec{x}$ by getting $-x_i$ always when the i-th link is bidirectional and it is used in negative direction (transport through this link is at the negative direction).

The difference between the MP_d and MPP_d is in that the elements from MP_d are also elements in E and for the elements in MPP_d it is not required to be in E. Note that if $\vec{x} \in E$ then $\vec{x}$ is a minimal path vector to level d.

Definition 2: A vector $\vec{x}$ is said to be a **minimal performance path vector to level *d*** MPP_d, if $\varphi(\vec{x}) = d$ and for every other $\vec{y} < \vec{x}$, $\varphi(\vec{x}) < d$.

Note that for each $\vec{x} \in MP_d$ there is a vector $\vec{y} \in MPP_d$ such that $\vec{y} < \vec{x}$. Moreover, $\vec{x}$ is the smallest vector in E such that $\vec{y} < \vec{x}$.

Let C' be a set of binary minimal direction path vectors and C is a set of binary minimal path vectors.

Definition3: Let S_i be the state vector of the arc a_i and $S_i' = \{S_{i_2}, \ldots, S_{i_{|S_i|}}\}$. For each $\vec{x} \in \mathbf{R^n}$, we define set $B_{\vec{x}}$ as:

$B_{\vec{0}} = S_i'$,

Let $\vec{x} \in \mathbf{R^n}$ with $B_{\vec{x}} = \{B_{\vec{x}}^1, \ldots, B_{\vec{x}}^n\}$ and $\vec{y} \in C$ such that $B_{\vec{x}}^i \neq \emptyset$ for all i for which $y_i = 1$. Then for $d \leq \min_{i, y_i = 1} B_{\vec{x}}^i(l)$ ($B_{\vec{x}}^i(l)$ is the l-th element in $B_{\vec{x}}^i$).

$B_{\vec{x}+d\cdot\vec{y}} = \{B_{\vec{x}+d\cdot\vec{y}}^1, \ldots, B_{\vec{x}+d\cdot\vec{y}}^n\}$, where

$$B_{\vec{x}+d\cdot\vec{y}}^i = \begin{cases} B_{\vec{x}}^i, & y_i = 0 \\ \{B_{\vec{x}}^i(1) - d, \ldots, B_{\vec{x}}^1(|B_{\vec{x}}^1|) - d\} & y_i = 1, d > B_{\vec{x}}^i(1) \\ \{B_{\vec{x}}^i(2) - d, \ldots, B_{\vec{x}}^1(|B_{\vec{x}}^1|) - d\} & y_i = 1, d = B_{\vec{x}}^i(1) \end{cases}$$

Steps of the algorithm:

Step 1: Find a set C',

Step 2: $C' = \{\vec{y}^1, \vec{y}^2, \ldots \vec{y}^L\}$, we defined function $f : C' \to \{1, 2, \ldots L\}$ with $f(\vec{y}^i) = i$

Step 3: $\forall \vec{y}^i \in C'$, we formed set $K_i = \{f(\vec{z}) \mid \vec{z} \in C', \vec{y}_j^i \cdot \vec{z}_j \geq 0, j = 1, \ldots n\}$

Step 4: Find a set $C = \{\vec{x}^1, \vec{x}^2, \ldots \vec{x}^L\}, \vec{x}^i = (|y_1^i|, \ldots, |y_n^i|)$

Step 5: Set $V = \{(\vec{0}, 0)\}, V' = \emptyset, H_{\vec{0}} = \{1, 2, \ldots L\}$ and find a set $B_{\vec{0}}$.

Step 6: $\forall \vec{v}$ such that $\{\vec{v}, d\} \in \mathrm{V}$ and $\forall i \in \mathrm{H}_{\vec{v}}$ we find $c = \{j \mid \vec{x}^i \in \mathrm{C}, x_j^i \neq 0\}$ and $m = \min\{\mathrm{B}_{\vec{v}}^j(1) \mid \mathrm{B}_{\vec{v}}^j(1) > 0 \text{ and } j \in \mathrm{C}\}$

The element $(\vec{v} + m\vec{x}^i, m + d)$ is put into V'.

The set $\mathrm{B}_{\vec{v}+m\cdot\vec{x}^i}$ is found using Definition 3.

The set $\mathrm{H}_{\vec{v}+m\cdot\vec{x}^i} = (\mathrm{H}_{\vec{v}} \cap \mathrm{K}_i) / \{r \mid \text{for } \mathrm{B}_{\vec{v}+m\cdot\vec{x}^i}^r = \emptyset\}$

Step 7: Delete all duplicate elements from V'.

Step 8: $d' = \min\{d \mid \{\vec{v}, d\} \in \mathrm{V}'\}$

Step 9: $\mathrm{MPP}_{d'} = \{\vec{v} \mid \{\vec{v}, d\} \in \mathrm{V}'\}$

Step 10: Using the set $\mathrm{MPP}_{d'}$ construct the set $\mathrm{MP}_{d'}$ on the following way: for each element $\vec{v} \in \mathrm{MPP}_{d'}$ found the smallest elements $\vec{y} \in \mathrm{E}$ which is larger then $\vec{v}$.

Step 11: Delete all elements $\vec{x} \in \mathrm{MP}_{d'}$, for which exists $\vec{y} \in \mathrm{MP}_{d'}$, $\vec{x} \geq \vec{y}$

Step 12: $\mathrm{V} \leftarrow \{\{\vec{v}, d\} \in \mathrm{V}' \mid d = d'\}$, and appropriate sets B and H are assigned accordingly.

Step 13: $\mathrm{V}' \leftarrow \mathrm{V}' \setminus \mathrm{V}$

Step 14: Repeat steps 6, 7, 8, 9, 10, 11, 12 and 13 until $\mathrm{V}' \neq \emptyset$.

Example: Let us have the network in Figure 1, and suppose that S_1={0,3,6}, S_2={0,3,6}, S_3={0,2}, S_4={0,4,8} and b_5={0,4,8}. The probabilities of the components are p_1=(0.1,0.1,0.8), p_2=(0.2,0.2,0.6), p_3=(0.2,0.8), p_4=(0.1,0.2,0.7) and p_5=(0.1,0.2,0.7). The set of binary minimal direction path vectors is C'={(1,1,0,0,0), (0,0,0,1,1), (1,0,1,0,1), (0,1,-1,1,0)}. The ordering of the vectors and the set of vectors that may be added to that one are given in the table Tab.1 :

Tab. 1: Binary vectors

C	(1,1,0,0,0)	(0,0,0,1,1)	(1,0,1,0,1)	(0,1,1,1,0)
F	1	2	3	4
K_i	{1,2,3,4}	{1,2,3,4}	{1,2,3}	{1,2,4}

We will give the first few steps of the algorithm. The initial values for the sets V, $\mathrm{B}_{\vec{v}}$ and $\mathrm{H}_{\vec{v}}$ are given in Tab. 2.

Tab. 2: Initial values

V		B_v	H_v
V	d	1 2 3 4 5	
(0,0,0,0,0)	0	3 3 2 4 4 6 6 8 8	{1,2,3,4}

The first 3 steps of the algorithm are present in the tables Tab. 3, Tab. 4 and Tab. 5. The first and the third columns give the vectors that are used in the procedure for finding minimal path vectors for next levels; while the second one gives the flow that can be delivered throw the system, when it is in the state **v**. m is the maximal flow that can be delivered using the *i*-th binary vector and system with set spaces $\mathrm{B}_{\vec{v}}$. The obtained vector is **x**, while m+d is the the flow that can be delivered throw the system, when it is in the state **x**.

Tab. 3

V						B_{v+mx}	H_{v+mx}	
v	d	$i \in H_v$	m	**x**	m+d	1 2 3 4 5		
(0,0,0,0,0)	0	1	3	(3,3,0,0,0)	3	3 3 2 4 4 8 8	{1,2,3,4}	V'
		2	4	(0,0,0,4,4)	4	3 3 2 4 4 6 6	{1,2,3,4}	V'
		3	2	(2,0,2,0,2)	2	1 3 0 4 2 4 6 8 6	{1,2}	V
		4	2	(0,2,2,2,0)	2	3 1 0 2 4 6 4 6 8	{1,2}	V
MP_2={(3,3,0,0,0), (0,0,0,4,4), (2,0,2,0,2), (0,2,2,2,0)}								
$M2TR_2$=0,96552								

Tab. 4.

V						B_{v+mx}	H_{v+mx}	
V	d	$i \in H_v$	m	**x**	m+d	1 2 3 4 5		
				(3,3,0,0,0)	3	3 3 2 4 4 8 8	{1,2,3,4}	V
				(0,0,0,4,4)	**4**	**3 3 2 4 4 6 6**	{1,2,3,4}	V'
(2,0,2,0,2)	2	1	1	(3,1,2,0,2)	3	3 2 0 4 2 5 8 6	{1,2}	V
		2	2	**(2,0,2,2,4)**	**4**	**1 3 0 2 4 4 6 6**	**{1,2}**	V'
(0,2,2,2,0)	2	1	1	(1,3,2,2,0)	3	2 3 0 2 4 5 6 8	{1,2}	V
		2	2	**(0,2,2,4,2)**	**4**	**3 1 0 4 2 6 4 6**	**{1,2}**	V'
MP_3={(3,3,0,0,0), (0,0,0,4,4)}								
$M2TR_3$=0,9468								

Tab. 5

V						B_{v+mx}	H_{v+mx}	
V	d	$i \in H_v$	m	**x**	m+d	1 2 3 4 5		
				(0,0,0,4,4)	4	3 3 2 4 4 6 6	{1,2,3,4}	V
				(2,0,2,4,4)	4	1 3 0 2 4 4 6 6	{1,2}	V
				(0,2,2,4,2)	4	1 3 0 2 4 4 6 6	{1,2}	V
(3,3,0,0,0)	3	1	3	**(6,6,0,0,0)**	**6**	**0 0 2 4 4 8 8**	**{2}**	V'
		2	4	**(3,3,0,4,4)**	**7**	**3 3 2 4 4**	**{1,2,3,4}**	V'
		3	2	**(5,3,2,0,2)**	**5**	**1 3 0 4 2 8 6**	**{1,2}**	V'
		4	2	**3,5,2,2,0)**	**5**	**3 1 0 2 4 6 8**	**{1,2}**	V'
(3,1,2,0,2)	3	2	2	**(3,1,2,2,4)**	**5**	**3 2 0 2 4 5 6**	**{1,2}**	V'
(1,3,2,2,0)	3	2	2	**1,3,2,4,2)**	**5**	**2 3 0 4 2 5 6**	**{1,2}**	V'
MP_4={(0,0,0,4,4),(6,6,0,0,0),(6,3,2,0,4),(3,6,2,4,0)}								
$M2TR_4$=0,91704								

The minimal path vectors for rest of the levels are given in Tab. 6.

Tab. 6: Minimal path vectors

D	Minimal Path Sets Level *d*	$M2TR_d$	D	Minimal Path Sets Level *d*	$M2TR_d$
	a_1 a_2 a_3 a_4 a_5			a_1 a_2 a_3 a_4 a_5	
5	6 6 0 0 0 6 3 2 0 4 3 6 2 4 0 3 3 0 4 4 3 0 2 4 8 0 3 2 8 4 0 0 0 8 8	0,85656	9	6 6 0 4 4 3 3 0 8 8 6 3 2 4 8 3 6 2 8 4	0,53104
6	6 6 0 0 0 3 3 0 4 4 3 0 2 4 8 3 0 2 8 4 0 0 0 8 8	0,84072	10	6 6 0 4 4 3 3 0 8 8	0,50640
7	3 3 0 4 4 0 0 0 8 8	0,72040	11	3 3 0 8 8	0,35280
8	0 0 0 8 8 6 6 0 4 4 6 3 2 4 8 3 6 2 8 4	0,66824	12	6 6 0 8 8	0,23520
			14	6 6 0 8 8	0,23520

4. CONCLUTION

This paper presents a new algorithm for calculating a minimal path vectors for multi-state two-terminal network. Its main advantage consists of the possibility of setting real numbers as values for the capacities of links. The main distinction between this algorithm and the other known algorithms is that it does not require any restriction for the values of the capacities of the links.

5. REFERENCES

[1] Ramirez-Marquez, J.E. and Coit, D. (2003), "Alternative Approach for Analyzing Multistate Network Reliability", *IERC Conference Proceedings* 2003

[2] Ramirez-Marquez, J.E., Coit, D. and Tortorella, M., "Multi-state Two-terminal Reliability: A Generalized Cut-Set Approach", Rutgers University IE Working Paper

[3] Lin, Y. (2001). "A Simple Algorithm for Reliability Evaluation of a Stochastic-Flow Nework with Node Failure." Computers and Operations Research, Vol 28, pp. 1277-1285

[4] Mihova,M., Synagina,N., "An algorithm for calculating multi-state network reliability using minimal path vectors", The 6th international conference for Informatics and Information Technology (CIIT 2008)