AN ALGORITHM FOR CALCULATING MULTI-STATE NETWORK RELIABILITY USING MINIMAL PATH VECTORS

Marija Mihova
Faculty of Natural Sciences and Mathematics
"Ss. Cyril and Methodius" University
Skopje, Republic of Macedonia

Nina Synagina
Institute of Parallel Processing
BAN
Sofia, Bulgaria

ABSTRACT

We describe methodologies for the reliability analysis of multi-state systems. Especially, we are concentrating on the concept of a minimal path and a cut vectors. The problem of interest is known as the multi-state two-terminal reliability computation, and we regard how the concept of the minimal path and cut vectors is used for modelling the reliability of these types of systems. The main focus of this presentation is to develop an algorithm for obtaining minimal path vectors for multi-state two-terminal network with integer capacity of any link. The proposed algorithm used minimal path vectors for a binary system with the same structure as the regarded one. It is based on addition of vectors, so it is simply for understanding and fast. Examples are included to illustrate the algorithm.

Key words: Reliability, multi-state systems, network reliability, minimal path vectors

I. INTRODUCTION

Traditionally, network reliability has been analysed from a binary perspective. That assumes that the system and its components can be in either two states: completely functioning or failed. However, binary state theory does not fully describe some systems as telecommunication systems, transportation systems, water distribution, and gas and oil production [1]. Usually, the elements of these systems may operate in any of several intermediate states, so for analyzing its reliability is used multi-state reliability theory. These kinds of systems are regarded in several papers, especially from Ramirez-Marquez, J.E. and Coit, D [2, 3, 4, 5].

Here we will regard one method for obtaining a minimal path sets for any level for two terminal multi state systems M2TR. This method assumes that the minimal path set for binary system with the same structure is known. Other techniques for improving reliability are also proposed in [2,3,5]. The algorithm in [2,3] deal with minimal cut sets.

II. PROBLEM DESCRIPTION

In this section we will introduce the problem of reliability of multi state networks and we will give some basic definitions. More of the definitions connected with the basic theory of multi-state reliability systems are given in [1, 6]. The others, which are connected with multi-state network, are given in [2].

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.

A **multi-state link** is defined as an arc of a network having a set of states, S={0,…,100%}. State 0 corresponds to the case where no flow can be delivered through that link. State 100% represents the link can supply its full capacity, intermediate states represent degradation expressed as a reduction of flow capacity. The vector that reflects the state of a component is called a component **state space set**. For example, let a network arc may functioning with 100%, 50% and 0%. Then, the vector (0, 0.5, 1) is the component state space set of that arc. The states of the link can be enumerated such that the new states and the new state vector represent the capacity of the link. This vector is called **capacity state set** and it is obtained as the product of full capacity of the component and its states. Assume that the arc described previously can deliver an amount of flow equal to 6 under perfect conditions. Then its capacity vector is equal to $S_i=(6\cdot 0,\ 6\cdot 0.5,\ 6\cdot 1)=(0, 3, 6)$. For entire system we define **system capacity state set,** S, as the set of all available capacities from source to sink. Sometimes, for some simplification, the states of the links can be enumerated in different ways. For example, we can suppose that the perfect functioning corresponds to level 2, 50% corresponds to level 1 and total failure to level 0 and we will obtain the vector (0, 1, 2).

A vector **X** that describes the state of all the system's components is called a **state vector**. The set of all state vectors is denote by E, $E=S_1\times S_2\times\ldots\times S_{n|}$. The function φ: $E\to S$ maps the state vector into a system state. This function is known as **structure function**.

In a binary case, the reliability of the system is defined as the probability that the system works. Appropriately, for multi-state systems the reliability of level d, R_d is defined as the probability that the system works with level greater or equal to d. In fact,

$$R_d=P(\varphi(x)\geq d) \qquad (1)$$

For a network with multi-state links, **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.

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

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

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

III. RELIABILITY COMPUTATION

Based on these vectors $M2TR_d$ can be computed through the inclusion/exclusion formula [3]. For binary reliability this formula is defined as:

$$R = P\left(\bigcup_{h=1}^{t} P_h\right) = \sum_{h=1}^{t} P(P_h) - \sum_{h<k}^{t} P(P_h \cap P_k) + \ldots (-1)^t P(P_1 \cap \ldots \cap P_t) \quad (2)$$

where t is the number of minimal path sets and P_h = minimal path set h.

This formula has to be extended to account for the new vector structure of the minimal sets. For the multi-state case $M2TR_j$ can be obtained with the following modification of the inclusion/exclusion formula:

$$\text{M2TR}_j = \sum_{h=1}^{T} P(\mathbf{x} \geq \mathbf{y}_h) - \sum_{h<k}^{T} P(\mathbf{x} \geq \mathbf{y}_h \wedge \mathbf{x} \geq \mathbf{y}_k) + \ldots (-1)^t P(\mathbf{x} \geq \mathbf{y}_1 \wedge \ldots \wedge \mathbf{x} \geq \mathbf{y}_T) \quad (3)$$

where T is the number of MPV_j and $\mathbf{y}_h \in \text{MPV}_j$. Using following notation

$\max(\mathbf{z}_1,\ldots,\mathbf{z}_s)=(\max(z_1^{(1)}, \ldots, z_s^{(1)}), \ldots, \max(z_1^{(l)}, \ldots, z_s^{(l)}),$ (4)

where $z_u^{(v)}$ is the v-th coordinate of $\mathbf{z}_u$.

The equation (3) can be write as:

$$\text{M2TR}_j = \sum_{h=1}^{T} P(\mathbf{x} \geq \mathbf{y}_h) - \sum_{h<k}^{T} P(\mathbf{x} \geq \max(\mathbf{y}_h, \mathbf{y}_k)) + \ldots (-1)^t P(\mathbf{x} \geq \max(\mathbf{y}_1, \ldots \mathbf{y}_T)) \quad (5)$$

The reliability of a system that follows the two-terminal rationale can be obtained based on the transformed inclusion/exclusion formula. We use the formula (5) for calculation of the reliability of level j.

IV. AN ALGORITHM FOR NETWORKS WITH COMPONENT CAPACITY STATE SET $\{0,1,2,\ldots M_I\}$

In this chapter we will consider systems with components with the added constraint that the capacity of arc i is an integer-valued variable taking values $\{0, 1, 2, \ldots, Mi\}$. In this case, the capacity of the entire network is an integer value from the set $\{0, 1, 2, \ldots, M\}$. Set $\boldsymbol{M}=(M_1, M_2,\ldots,M_n)$ be the vector of maximal states of the system.

The initial step of the proposed algorithm requires that binary minimal cut 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), of the components, minimal paths can be represented by the vectors. We assume that these vectors are known. Moreover, these vectors are minimal path vectors for level 1 in the multi-state network. 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 regard 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 this vectors are also minimal path vectors for level 1.

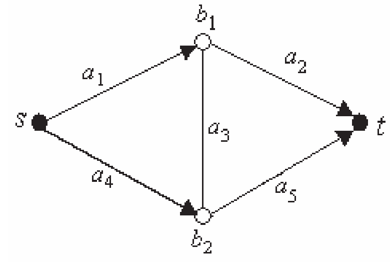

Figure 1: An example.

If a link is bidirectional we will chose one of the directions to be positive and we can take 1 for that direction and -1 for the other. In that way, from the each minimal path we get a new vector, minimal direction path vector (MDP). In the pervious example, the bidirectional link is a_3, and in this case we can choose the direction from b_1 to b_2 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). Note that each bidirectional link is used only in one way. In fact, the flow trough one bidirectional link goes from the node in which capacity is greater then capacity of the other links to the other node. In opposite, some elements will pass from one to another node and back.

Definition 1 Let **x** be a minimal path vector of level j, then **MDP of level *j*** MPD_j is obtained from x by getting $-x_i$ always when the i-th link is bidirectional and it is used in negative direction (transport true this link is in the negative direction)

We will illustrate the basic idea of the algorithm on the network given with Figure 1. An individual unit flows from the source to the sink using one of the paths from Figure 2. Suppose that one unit flows using path p1 Then the minimal required performance of the network are (1,1,0,0,0), and this is the minimal path vector for level 1, Figure 2.

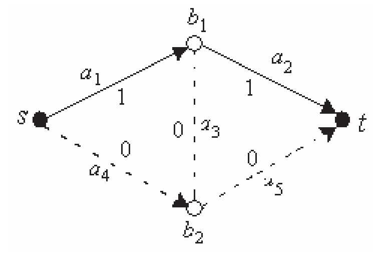

Figure 2

The second unit also flows using some of the paths p_1, p_2, p_3, and p_4. Suppose it is the path p_2. Now the minimal required performance of the network that are used from this two units are (2,1,1,0,1), Figure 4. The vector (2,1,1,0,1) is a minimal path vector for level 2. Continuing on these way k-times, we will obtain minimal path vector of level k.

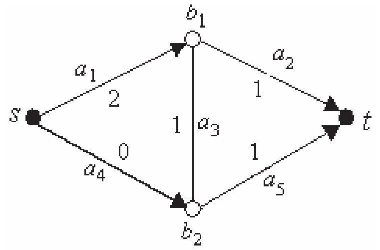

Figure 3

Proposition 1 Let **x** be a MDP of level j and **y** be a MDP of level k. If the vector **z**=**x**+**y**<**M**, then, it is a MDP of level $j+k$.

Using Proposition 1 we construct the following algorithm for obtaining all minimal path vectors for each level of the system:

*Step*1: Obtaining all minimal path vectors for level 1
Step 2: Obtaining all MDP of level 1, MDP_1
Step 3: Construct the set $MDP'_{j+1}=\{x+y|\ x\in MDP_j$ and $y\in MDP_1\}$
Step 4: Finding Abs(**x**)=($|x_1|$, $|x_2|$, … , $|x_n|$).
Step 5: The set MDP_j is obtain from the set MDP'_j by deleting duplicate elements and elimination all elements **x** such that Abs(**x**) is not smaller or equal to ***M***.
Step 6: The set MP_j is obtained from the set {Abs(**x**)| **x**∈MDP_j} by elimination the elements that appear more then once.
Step 7: Repeat steps 3, 4, 5 and 6 for all $j\in\{2,\ldots,M\}$.
Step 8: Calculating reliability using including-excluding formula
Step 9: When the minimal path sets are obtained, the reliability is calculated using including-excluding formula.

Example 1 Regard the network given by Figure 1 such that b_1={0,1,2}, b_2={0,1}, b_3={0,1}, b_4={0,1,2} and b_5={0,1,2,3}. The vector ***M***=(2,1,1,2,3). The probabilities of the components are p_1=(0.1,0.1,0.8), p_2=(0.1,0.9), p_3=(0.2,0.8), p_4=(0.1,0.1,0.8) and p_5=(0.1,0.05,0.05,0.8). The procedure will be explained stepwise:
MP_1={(1,1,0,0,0), (0,0,0,1,1), (1,0,1,0,1), (0,1,1,1,0)}
MDP_1={(1,1,0,0,0), (0,0,0,1,1), (1,0,1,0,1), (0,1,-1,1,0)}
For the reliability function we obtain: $M2TR_1$=0.97686.
j=2:

(1,1,0,0,0) + (1,1,0,0,0) = **(2,2,0,0,0)**
(1,1,0,0,0) + (0,0,0,1,1) = (1,1,0,1,1)
(1,1,0,0,0) + (1,0,1,0,1) = (2,1,1,0,1)
(1,1,0,0,0) + (0,1,-1,1,0) = **(1,2,-1,1,1)**
(0,0,0,1,1) + (0,0,0,1,1) = (0,0,0,2,2)
(0,0,0,1,1) + (1,0,1,0,1) = (1,0,1,1,2)
(0,0,0,1,1) + (0,1,-1,1,0) = (0,1,-1,2,1)
(1,0,1,0,1) + (1,0,1,0,1) = **(2,0,2,0,2)**
(1,0,1,0,1) + (0,1,-1,1,0) = (1,1,0,1,1) duplic
(0,1,-1,1,0) + (0,1,-1,1,0) = **(0,2,-2,2,0)**

When elements **x** such that Abs(**x**) is not smaller or equal to ***M*** (the bold elements) and elements that appear more then once are eliminate, we obtain MDP_2={(1,1,0,1,1), (2,1,1,0,1), (0,0,0,2,2), (1,0,1,1,2), (0,1,-1,2,1)} and MP_2={(1,1,0,1,1), (2,1,1,0,1), (0,0,0,2,2), (1,0,1,1,2), (0,1,-1,2,1)}. The reliability of level 2 is $M2TR_1$=0.84614.
j=3:

(1,1,0,1,1) + (1,1,0,0,0) = **(2,2,0,1,1)**
(1,1,0,1,1) + (0,0,0,1,1) = (1,1,0,2,2)
(1,1,0,1,1) + (1,0,1,0,1) = (2,1,1,1,2)
(1,1,0,1,1) + (0,1,-1,1,0) = **(1,2,-1,2,1)**
(2,1,1,0,1) + (1,1,0,0,0) = **(3,2,1,0,1)**
(2,1,1,0,1) + (0,0,0,1,1) = (2,1,1,1,2) duplic.
(2,1,1,0,1) + (1,0,1,0,1) = **(3,1,2,0,2)**
(2,1,1,0,1) + (0,1,-1,1,0) = **(2,2,0,1,1)**
(0,0,0,2,2) + (1,1,0,0,0) = (1,1,0,2,2) duplic.
(0,0,0,2,2) + (0,0,0,1,1) = **(0,0,0,3,3)**
(0,0,0,2,2) + (1,0,1,0,1) = (1,0,1,2,3)
(2,1,1,0,1) + (0,1,-1,1,0) = **(2,2,0,1,1)**
(1,0,1,1,2) + (1,1,0,0,0) = (2,1,1,1,2) duplic.
(1,0,1,1,2) + (0,0,0,1,1) = (1,0,1,2,3) duplic.
(1,0,1,1,2) + (1,0,1,0,1) = **(2,0,2,1,3)**
(1,0,1,1,2) + (0,1,-1,1,0) = (1,1,0,2,2) duplic.
(0,1,-1,2,1) + (1,1,0,0,0) = **(1,2,-1,2,1)**
(0,1,-1,2,1) + (0,0,0,1,1) = **(0,1,-1,3,2)**
(0,1,-1,2,1) + (1,0,1,0,1) = (1,1,0,2,2) duplic.
(0,1,-1,2,1) + (0,1,-1,1,0) = **(0,2,-2,3,1)**

We have MDP_3={(1,1,0,2,2), (2,1,1,1,2), (1,0,1,2,3)} and MP_3={(1,1,0,2,2), (2,1,1,1,2), (1,0,1,2,3)}. The reliability of level 3 is $M2TR_1$=0.64584.
j=4:

(1,1,0,2,2) + (1,1,0,0,0) = **(2,2,0,2,2)**
(1,1,0,2,2) + (0,0,0,1,1) = **(1,1,0,3,3)**
(1,1,0,2,2) + (1,0,1,0,1) = (2,1,1,2,3)
(1,1,0,2,2) + (0,1,-1,1,0) = **(1,2,-1,3,2)**
(2,1,1,1,2) + (1,1,0,0,0) = **(3,2,1,1,2)**
(2,1,1,1,2) + (0,0,0,1,1) = (2,1,1,2,3) duplic.
(2,1,1,1,2) + (1,0,1,0,1) = **(3,1,2,1,2)**
(2,1,1,1,2) + (0,1,-1,1,0) = **(2,2,0,2,2)**
(1,0,1,2,3) + (1,1,0,0,0) = (2,1,1,2,3) duplic.
(1,0,1,2,3) + (0,0,0,1,1) = **(1,0,1,3,4)**
(1,0,1,2,3) + (1,0,1,0,1) = **(2,0,2,2,4)**.
(1,0,1,2,3) + (0,1,-1,1,0) = **(1,1,0,3,3)**

When elements x such that Abs(x) is not smaller or equal to M (the bold elements) and elements that appear more then once are eliminate, we obtain MDP4={(2,1,1,2,3)} and MP4={(2,1,1,2,3)}. The reliability of level 4 is M2TR1=0.3686.

Note that we obtain a lot of duplicate elements. Also a lot of obtained elements are bigger then the maximal vector **M**. So, we want to find a way to reduce unnecessary elements. For that reason we analyze when such a vectors are obtained.
1. Duplicate vector is obtained whenever vectors with opposite sign are added in a same coordinate. That property follows from Remark 1. So we can test the signs of the vectors and add only vectors with the same signs. But, if there is a little number of bidirectional links, it is not necessary. In fact, that test can take more time then test of replication.
2. Duplicate vectors are also obtained when we add the same set of vectors from MDP_1 in different order. For example, let see why the vector (2,1,1,1,2) from Example 1 is obtain 3

times as the minimal path vector for level 3. The first time it is obtained as

(2,1,1,1,2) = ((1,1,0,0,0)+(0,0,0,1,1))+(1,0,1,0,1),

the second time as

(2,1,1,1,2) = ((1,1,0,0,0)+(1,0,1,0,1)) +(0,0,0,1,1),

and the third time as

(2,1,1,1,2)= ((0,0,0,1,1)+(1,0,1,0,1)) +(1,1,0,0,0).

In fact all three times we add vectors (0,0,0,1,1), (1,0,1,0,1) and (1,1,0,0,0), but in different order. So, we can make the algorithm such that each set of vectors is taking only once.

In construction of the algorithm we will use the following proposition:

Proposition 2: If **x** is a minimal direction path vector for level j, then there are vectors $\mathbf{x}_k \in MDP_1$, $k=1,\ldots,j$, such that $x = \sum_{k=1}^{j} x_k$. Moreover, each vector $\mathbf{y} \in MDP_1$ is not appearing more then $\min_{y_i=1} M_i$.

For each vector in $\mathbf{y} \in MDP_1$ we take an indicator, $c_y = \min_{|y_i|=1} M_i$. Then in order to obtain all minimal path vectors to level j we construct all j-sets of elements from MDP_1 such that each vector $\mathbf{y} \in MDP_1$ appears at last c_y times.

Example 1 (continue) Lets find the indicators for elements in MDP_1={(1,1,0,0,0), (0,0,0,1,1), (1,0,1,0,1), (0,1,-1,1,0)}. We had ***M***=(2,1,1,2,3), so $c_{(1,1,0,0,0)}=1$, $c_{(0,0,0,1,1)}=2$, $c_{(1,0,1,0,1)}=1$ and $c_{(0,1,-1,1,0)}=1$. Now, for j=2 we have

(1,1,0,0,0) + (0,0,0,1,1) = (1,1,0,1,1)
(1,1,0,0,0) + (1,0,1,0,1) = (2,1,1,0,1)
(1,1,0,0,0) + (0,1,-1,1,0) = **(1,2,-1,1,1)**
(0,0,0,1,1) + (0,0,0,1,1) = (0,0,0,2,2)
(0,0,0,1,1) + (1,0,1,0,1) = (1,0,1,1,2)
(0,0,0,1,1) + (0,1,-1,1,0) = (0,1,-1,2,1)
(1,0,1,0,1) + (0,1,-1,1,0) = (1,1,0,1,1) duplic.

In fact we obtain one duplicate element, and only one bigger then ***M***. With the previous procedure we had 3 elements bigger them ***M***.

For level j=3 we have:

(1,1,0,0,0) + (0,0,0,1,1) + (0,0,0,1,1) = (1,1,0,2,2)
(1,1,0,0,0) + (0,0,0,1,1) + (1,0,1,0,1) = (2,1,1,1,2)
(1,1,0,0,0) + (0,0,0,1,1) + (0,1,-1,1,0) = **(1,2,-1,2,1)**
(1,1,0,0,0) + (1,0,1,0,1) + (0,1,-1,1,0) = **(2,2,0,1,1)**
(0,0,0,1,1) + (0,0,0,1,1) + (1,0,1,0,1) = (1,0,1,2,3)
(0,0,0,1,1) + (0,0,0,1,1) + (0,1,-1,1,0) = **(0,1,-1,3,2)**
(0,0,0,1,1) + (1,0,1,0,1) + (0,1,-1,1,0) = (1,1,0,2,2) duplic.

Now, instead 20 vectors we obtain only 7, one duplicate and 3 bigger then ***M***.

For j=4:

(1,1,0,0,0) + (0,0,0,1,1) + (0,0,0,1,1) + (1,0,1,0,1)=(2,1,1,2,3)
(1,1,0,0,0) + (0,0,0,1,1) + (1,0,1,0,1) + (0,1,-1,1,0)=**(2,2,0,2,2)**
(1,1,0,0,0) + (0,0,0,1,1) + (1,0,1,0,1) + (0,1,-1,1,0)=**(2,2,0,2,2)**
(0,0,0,1,1) + (0,0,0,1,1) + (1,0,1,0,1) + (0,1,-1,1,0)=**(1,1,0,3,3)**

In fact, now we obtain only 4 elements, and before we had 12.

It can be concluded than by using the indicator function we obtain a smaller number of vectors. Also, this algorithm not requires minimal path sets for smaller levels to be known, so we can use it in the situation when we want to obtain the reliability of one particular level. In fact this algorithm directly determines the minimal path set for given level k.

V. Complexness of the Algorithm

In this section we will calculate the complexity of the algorithm that directly determines the minimal path set for given level j, in respect to number of binary minimal path sets (i.e. number of minimal path vectors for level 1).

Let m be the number of minimal path vectors for level 1. Then, the following theorem is true:

Lemma 1 Let we have a two-terminal network with m minimal path vectors for level 1. Then, the number of sets that contain j minimal path vectors for level 1 is equal to

$$\binom{m+j-1}{j} \tag{3}$$

Proof. The proof follows directly from fact that there is a $\binom{m+j-1}{j}$ ways to put j elements in m classes.

To obtain all minimal path vectors of level j, we need to calculate all sums $x = \sum_{k=1}^{j} x_k$ such that $\mathbf{x}_k \in MDP_1$. Because there is $\binom{m+j-1}{j}$ number of such sums, and $o\left(\binom{m+j-1}{j}\right) = o(m^j)$, we have following theorem:

Theorem 1 Let we have a two-terminal network with m minimal path vectors for level 1. Then, the complexness of the algorithm for directly determination of the minimal path set for level j is smaller then $o(m^j)$.

VI. An Algorithm for Networks that Has Arc Capacities with Integer Flows

The proposed algorithm can be modified for systems in which the capacity of arc i is an integer-value x_i, $0 \le x_i \le M_i$. The vector of maximal states of the system is ***M***= $(M_1, M_2,\ldots,M_n)$.

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

The difference between the MP_j and MPP_j is in that the elements from MP_j are also elements in E and from the

elements in MPP_j it is not required to be in E. Note that if $\mathbf{x}\in E$ then $\mathbf{x}$ is a minimal path vector to level j. Next proposition gives the relation between MP_j and MPP_j.
Next we define an equivalent of MDP_j:

Definition 3 Let **x** be a minimal performance path vector of level j, then **minimal performance direction path vector of level *j*** $MDPP_j$ is obtained from **x** by getting $-x_i$ always when the i-th link is bidirectional and it is used in negative direction (transport true this link is in the negative direction)

Proposition 3: For each $\mathbf{x}\in MP_j$ there is vector $\mathbf{y}\in MPP_j$ such that $\mathbf{y}<\mathbf{x}$. Moreover, **x** is the smallest vector in E such that $\mathbf{y}<\mathbf{x}$.

So, for these types of systems we will use the same algorithm explain in Section 2, but instead looking for minimal path vectors to level j we will obtain MPP_j. Then for each element from $MPPj$ we will find the smallest vector $\mathbf{y}\in E$ such that $\mathbf{x}\leq\mathbf{y}$. This vector is a path vector for level j, but it is not necessary to be a minimal path vector. In fact, by this procedure we will get all minimal path vectors, but also we will get some vectors which are not minimal paths. So, to obtain the set of minimal path vectors we must dispose all no minimal vectors obtained by the pervious procedure.
Next we give the algorithm. Note that this algorithm can also be used for systems regarded in the Section 2.

Step 1: Find all minimal path vectors of binary system. This set is equal to the set MPP_1.
Step 2: Find all $MDPP_1$
Step 3: Using the set MPP_1 constructs the set MP_1 on the following way: for each element $\mathbf{x}\in MPP_1$ fined the smallest elements in $\mathbf{y}\in$E which is larger then **x**.
Step 4: Construct the set $MDPP'_{j+1}=\{x+y|x\in MDP_j, y\in MDP_1\}$
Step 5: The set $MDPP_j$ is obtain from the set $MDPP'_j$ by deleting duplicate elements and elimination all elements **x** such that Abs(**x**) is not smaller or equal to ***M***.
Step 6: The set MPP_j is obtained from $\{Abs(\mathbf{x})|\ \mathbf{x}\in MDP_j\}$ by elimination the elements that appear more then once.
Step 7: Using the set MPP_j constructs the set MP_j on the following way: for each element $\mathbf{x}\in MPP_j$ fined the smallest elements in $\mathbf{y}\in$E which is larger then **x**.
Step 8: Repeat steps 4, 5, 6 and 7 for all $j\leq M$.

Example 2 Again we regard the network with structure given by Figure 1, but now suppose that $b_1=\{0,1,3\}$, $b_2=\{0,3\}$, $b_3=\{0,2\}$, $b_4=\{0,2\}$ and $b_5=\{0,1,2,3\}$. The vector $\boldsymbol{M}=(3,3,2,2,3)$. The probabilities of the components are (p_{ij} means probality that the i-th component is in state j):$p_{10}=0.1$, $p_{11}=0.1$, $p_{13}=0.8$, $p_{20}=0.1$, $p_{23}=0.9$, $p_{30}=0.2$, $p_{22}=0.8$, $p_{40}=0.1$, $p_{42}=0.9$, $p_{50}=0.1$, $p_{51}=0.05$, $p_{52}=0.05$ and $p_{53}=0.8$.

j	MPP_j	MP smaller then MPP_j	MP_j	M2TRj
1	1 1 0 0 0	1 3 0 0 0	1 3 0 0 0	0.97848
	0 0 0 1 1	0 0 0 2 1	0 0 0 2 1	
	1 0 1 0 1	1 0 2 0 1	1 0 2 0 1	
	0 1 1 1 0	0 3 0 2 0	0 3 0 2 0	
2	0 0 0 2 2	0 0 0 2 2	0 0 0 2 2	0.95989
	0 1 1 2 1	0 3 2 2 1	0 3 2 2 0	
	0 2 2 2 0	0 3 2 2 0	1 3 0 2 1	
	1 0 1 1 2	1 0 2 2 2	3 0 2 0 2	
	1 1 0 1 1	1 3 0 2 1	3 3 0 0 0	
	1 2 1 1 0	1 3 2 2 0		
	2 0 2 0 2	3 0 2 0 2		
	2 1 1 0 1	3 3 2 0 1		
	2 2 0 0 0	3 3 0 0 0		
3	1 0 1 2 3	1 0 2 2 3	1 0 2 2 3	0.85041
	1 1 0 2 2	1 3 0 2 2	1 3 2 2 1	
	1 2 1 2 1	1 3 2 2 1	1 3 2 2 0	
	1 3 2 2 0	1 3 2 2 0	3 3 0 0 0	
	2 0 2 1 3	3 0 2 2 3		
	2 1 1 1 2	3 3 2 2 2		
	2 2 0 1 1	3 3 0 2 1		
	2 3 1 1 0	3 3 2 2 0		
	3 1 2 0 2	3 3 2 0 2		
	3 2 1 0 1	3 3 2 0 1		
	3 3 0 0 0	3 3 0 0 0		
4	2 1 1 2 3	3 3 2 2 3	3 3 0 2 1	0.5832
	2 2 0 2 2	3 3 2 2 1		
	2 3 1 2 1	3 3 2 2 1		
	3 1 2 1 3	3 3 2 2 3		
	3 2 1 1 2	3 2 2 2 2		
	3 3 0 1 1	3 3 0 2 1		
5	3 2 1 2 3	3 3 2 2 3	3 3 0 2 2	0.5508
	3 3 0 2 2	3 3 0 2 2		

Table 1: Minimal path vectors and reliability

VII. Conclusion

In this paper we give an algorithm for calculation of the minimal path vectors of a multi-state two terminal network. This algorithm needs only the minimal path set for binary network with the same structure as the multi-state network. That means that after the minimal path set for binary network is found, to obtain minimal path vectors for all other levels we do not need to know the adjacency matrix. In fact, all vectors are obtained by addition of these vectors. From the other side, the complexness of the algorithm is not exponential. So we have a quick algorithm.

References

[1] B. Natvig, S. Sørmo, A. Holen, and G. Høgåsen, "Multi-state Reliability Theory – A Case Study." Advances in Applied Probability, 1986, Vol 18, pp. 921-932.
[2] J.E.Ramirez-Marquez, and D. Coit, "Alternative Approach for Analyzing Multistate Network Reliability", *IERC Conference Proceedings* 2003
[3] J.E.Ramirez-Marquez., D.Coit and M. Tortorella, "Multi-state Two-terminal Reliability: A Generalized Cut-Set Approach", Rutgers University IE Working Paper
[4] J.E.Ramirez-Marquez, and D. Coit, "Composite Importance Measures for Multi-State Systems with Multi-state Components" *IEE Transactios on Reliability, Vol 5, No3, September* 2005.
[5] J.E.Ramirez-Marquez, and D. Coit, "Multi-state Component Criticality Analysis for Reliability Improvement in Multi-state Systems", *Reliability Engineering & System Safety*, Vol. 92, No. 10, October 2007.