Olympiads in Informatics, 2014, Vol. 8, 133–144

On Maximal Level Minimal Path Vectors of a Two-Terminal Network

Marija MIHOVA[1], Natasha STOJKOVIKJ[2],
Mile JOVANOV[1], Emil STANKOV[1]

[1] *Faculty of Computer Science and Engineering, University Ss.Cyril and Methodius Rugjer Boshkovikj 16, Skopje, Republic of Macedonia*
[2] *Faculty of Computer Science, University Goce Delchev Krste Misirkov bb, Shtip, Republic of Macedonia*
e-mail: {marija.mihova, mile.jovanov, emil.stankov}@finki.ukim.mk natasa.maksimova@ugd.edu.mk

Abstract. The reliability of a two-terminal flow network with a discrete set of possible capacities for its arcs is usually computed in terms of minimal path or minimal cut vectors. This paper analyzes the connection between minimal path vectors and flow functions, which supports the development of an efficient algorithm that solves the problem of finding the set of all such vectors.

Keywords: two-terminal flow network, minimal path vector, minimal cut vector.

1. Introduction

Many real-life industrial systems, such as telecommunication, electric power generation and transmission, transportation and manufacturing systems may be viewed as networks whose arcs have discrete set of possible capacities. Such systems can be regarded as multi-state systems with multi-state components, where the arcs are the system's components, whereas the demand levels of the system are all possible netflow values. Analyzing the reliability of such systems has become attractive to many researchers in recent decades. The reliability of a multi-state system can be computed in terms of minimal path vectors to demand level d, called d-MinPaths (*d-MPs*) (Lin, 2001; Ramirez-Marquez and Coit, 2003; Mihova and Maksimova, 2011), or minimal cut vectors to demand level d, called d-MinCuts (*d-MCs*) (Ramirez-Marquez *et al.*, 2003; Jane *et al.*, 1993). Both strategies extract candidates that are not minimal cut vectors by mutually comparing all pairs of vectors and removing the smaller one, if such exists. The problem of computing reliability of a multi-state system is NP-hard, but solvable (Wilson *et al.*, 2005; Provan and Balls, 1983), and commonly the inclusion-exclusion approach is used for this purpose (Provan and Balls, 1983).

Thus, the problem of searching for all *d-MCs* or *d-MPs* is one of the most important problems in multi-state network reliability, and several algorithms have been proposed as a solution to this problem. Jane *et al.* (Jane *et al.*, 1993) propose a methodology that solves the problem of generating all multi-state *MCs* for the multi-state two-terminal network, obtainning a set of candidates, while Ramirez-Marquez *et al.* (Ramirez-Marquez *et al.*, 2003) optimize this procedure in such a way that their set of candidates has significantly lower cardinality.

In (Lin *et al.*, 1995), Lin *et al.* give an algorithm that finds a set of candidates for *d-MPs* and extracts all *d-MPs* by comparing all pairs of candidates and eliminating vectors that are not minimal. An approach for elimination of nonminimal candidates without comparison is given by Forghani-elahabad *et al.* (Forghani-elahabad *et al.*, 2013). Given a *d-MP* candidate, they form m smaller vectors (where m is the number of nonzero coordinates) in such a way that each of these vectors differs from the *d-MP* candidate in unit vector. If all appropriate graphs have maximum flow equal to d, then that vector is not a *d-MP*. The method for computing all *d-MPs* proposed in (Mihova and Maksimova, 2011) uses additional calculations that help in avoiding to obtain vectors which are not minimal.

In this paper we analyze some properties of *d-MPs* that will show the connection between *d-MPs* and flow functions to level d on a given two-terminal network. This helps to develop a strategy for checking whether some candidate is a *d-MP* with time complexity $O(|E|)$, which is significantly better than $O(|V|^2|E|^{3/2})$, the complexity of the strategy given in (Forghani-elahabad *et al.*, 2013). Moreover, using further analysis we give the relationship between two *d-MPs* and we propose another algorithm that directly finds all *d-MPs*. At the end, we explain the advantage of this approach, especially in the case when d is a maximum flow.

2. Basic Assumptions

A *two-terminal flow network* is a directed graph $G(V,E)$ with two special vertices, a source s and a sink $t (s \neq t)$, in which each edge $(u,v) \in E$ has a nonnegative capacity $c(u,v) \geq 0$. The function c is called *capacity function*. Shortly, we will denote such a capacity network by $G(V,E,c)$.

A *flow* in $G(V,E,c)$ is a function $f : E \to R^+ \cup \{0\}$ that satisfies the following two constraints:

1. *Capacity constraint*: $0 \leq f(u,v) \leq c(u,v)$, for each $(u,v) \in E$, i.e., the flow of an edge cannot exceed its capacity.
2. Flow conservation:

$$f(V,v) - f(v,V) = \sum_{u \in V} f(u,v) - \sum_{w \in V} f(v,w) = \begin{cases} 0, & \{s,t\} \\ |f|, & v = s \\ -|f|, & v = t \end{cases},$$

where $|f|$ is the value of the flow.

In other words, the total flow in a node v, $f(V,v)$, must equal the total flow out the node v, $f(v,V)$, for all vertices $v \in V \setminus \{s,t\}$; the flow leaving s and the flow entering t is equal to the value of the flow.

It is assumed that if there is no edge (u, v), i.e. $(u, v) \notin E$, then $f(u, v) = 0$.

A flow is a *maximum flow* if it has the largest possible value among all flows from s to t in a given capacity network (Erickson, J., 2009).

A *pseudoflow* is a function $f: E \rightarrow R^{+} \cup \{0\}$ defined on arcs that satisfy only the capacity constraints; it need not satisfy flow conservations (Ahuja and Orlin, 1993). Note that each flow function is also a pseudoflow function.

Let us assume that the set of edges in the flow network is ordered, i.e., $E = \{e_1, e_2, \ldots, e_{|E|}\}$. Considering the edges as components, the network represents a multi-component system. It can be assumed that each component, the edge e_i, can operate in some demand level $x_i \leq c(e_i)$. The vector $\vec{x}$ is called *state vector*. In the multi-state reliability theory (Wilson *et al.*, 2005), the vector $\vec{x}$ is called *path vector to level d* if and only if the system in state $\vec{x}$ works with level equal or greater than d.

Below we introduce a few definitions that give a connection between systems and two-terminal networks.

DEFINITION 1. Let $G(V, E, c)$ be a two-terminal flow capacity network. For a pseudoflow l_c, we define *state vector* $\overrightarrow{x_{l_c}}$ *induced by* l_c by

$$x_i = l_c(e_i).$$

For each state vector $\vec{x}$, with $x_i \leq c(e_i)$, we define *pseudoflow function* $l_{\vec{x}}$ *induced by* $\vec{x}$, by

$$l_{\vec{x}}(e_i) = x_i.$$

The state vector $\vec{x}$ is called a *flow vector*, whenever $l_{\vec{x}}$ is a flow function.

Aggarwal *et al.* (Aggarwal *et al.*, 1982) defines two-terminal reliability as the probability that the network can adequately deliver a demanded flow from the source to the sink. In other words, the system is in a working state if and only if it is possible to successfully transmit the required flow from the source to the sink node. The next definition explains this more precisely.

DEFINITION 2. Let $G(V, E, c)$ be a two-terminal flow network and $\overrightarrow{x_{l_c}}$ is a state vector induced by the pseudoflow l_c. We will say that $\overrightarrow{x_{l_c}}$ is a *path vector to level d, d-P,* if and only if a flow d may be delivered in the two-terminal network $G(V, E, l_c)$. The state vector $\vec{x}$ is a *minimal path vector to level* d, *d-MP*, if and only if the two-terminal flow network $G(V, E, l_{\vec{x}})$ has a maximum flow d, and for each $\overrightarrow{x'} \leq \vec{x}$, the two-terminal flow network $G(V, E, l_{\overrightarrow{x'}})$ has a maximum flow less than d.

Next we give some known facts from the two-terminal network theory (Cormen *et al.*, 2009). Suppose that we have a two-terminal flow network $G(V, E, c)$. Let f be a flow in G, and consider a pair of vertices $u, v \in V$. We define the *residual capacity* $c_f(u, v)$ by

$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v), & (u, v) \in E \\ f(u, v), & (v, u) \in E \\ 0, & otherwise \end{cases}.$$

For the flow network $G(V,E)$ and a flow f, the residual network of G induced by f is $G_f(V,E_f)$ where

$$E_f = \{(u,v) \in E \colon c_f(u,v) > 0\}.$$

Note that each flow network $G(V,E)$ can be regarded as a residual network induced by a function f, where $f(u,v) = 0$ for all (u,v).

Given a flow network $G(V,E)$ and a flow f, the *augmenting path* is defined as a simple path $\mathcal{P}$ from s to t in the residual network G_f. Similarly, we will define an *augmenting cycle* as a simple cycle $\mathcal{C}$ from some node v to v in the residual network G_f. For each augmenting cycle $\mathcal{C}$ in residual network G_f, we define *augmented vector of level d' for a cycle $\mathcal{C}$*, $\vec{y}^{\mathcal{C},d'}$, by

$$y_i^{\mathcal{C},d'} = \begin{cases} d', & \text{if } e_i = (u,v) \in E \text{ and } (u,v) \text{ is on } \mathcal{C} \\ -d', & \text{if } e_i = (u,v) \in E \text{ and } (v,u) \text{ is on } \mathcal{C} \\ 0, & \text{otherwice} \end{cases}$$

for some $d' \leq \min\{c_f(e_i) | e_i \in \mathcal{C}\}$.

A *cut* (S,T) of the flow network $G(V,E)$ is a partition of V into S and $T = V \backslash S$ such that $s \in S$ and $t \in T$. The *capacity* of the cut (S,T) is

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v).$$

A *minimum cut* of a network is a cut whose capacity is minimum over all cuts of the network, i.e., (S,T) is a minimum cut if for all other cuts (S',T'), $c(S,T) \leq c(S',T')$.

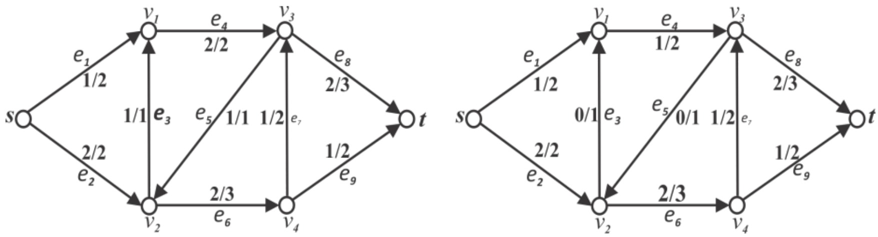
3. The Connection between *d-MPs* and Flow Functions to Level *d* in a Two-Terminal Network

In this section we present an approach for checking if a given flow function corresponds to a *d-MP*.

Given a flow function f, let E^f denote the set of all vertices with a positive flow, i.e. $E^f = \{e \in E \mid f(e) > 0\}$. We will refer to the unweighted graph $G(V,E^f)$ as the *graph induced by f*.

The next theorem states that a flow f corresponds to a minimal path vector if and only if E^f is acyclic. This is illustrated in Fig. 1. Namely, the flow in Fig. 1 a) is a flow function to level 3, but the state vector induced by it is not a 3-*MP* since the state vector induced by the flow of level 3 in Fig. 1 b) has induced a state vector lower then it. Note that the flow in Fig. 1 a) has additional flow through the cycle $<v_1, v_3, v_2, v_1>$, while the flow in Fig. 1 b) has no cycle.

Theorem 1. The state vector $\vec{x}$ is a *d-MP* for the two-terminal flow network $G(V,E,c)$ iff the pseudoflow function $l_{\vec{x}}$ is a flow function with $|l_{\vec{x}}| = d$, and the corresponding graph $G(V,E^{l_{\vec{x}}})$ induced by $l_{\vec{x}}$, has no cycles.



a) b)

Fig. 1. a) Flow function to level 3 with (1, 2, 1, 2, 1, 2, 1, 2, 1) as a vector induced by it, which is not 3-*MP*;
b) Flow function to level 3 with (1, 2, 0, 1, 0, 2, 1, 2, 1) as a vector induced by it, which is 3-*MP*.

Proof. Assume that $\vec{x}$ is a *d-MP*. First, we will prove that $l_{\vec{x}}$ is a flow function with $|l_{\vec{x}}| = d$. Since $\vec{x}$ is a *d-P*, the maximum flow of $G(V, E, l_{\vec{x}})$ is equal to *d*. Then there is a flow *f* to level d for $G(V, E, l_{\vec{x}})$. Let $\vec{y}_f$ be the vector induced by *f*. It is clear that $\vec{y}_f$ is a *d-P* and $\vec{y}_f \leq \vec{x}$. Since there is no lower path vector to level d than $\vec{x}$, we have $\vec{y}_f = \vec{x}$, which implies that $l_{\vec{x}} = f$. This proves that $l_{\vec{x}}$ is a flow function with $|l_{\vec{x}}| = d$.

Next, let us suppose that $l_{\vec{x}}$ is a flow function and suppose the network $G(V, E^{l_{\vec{x}}})$ has a cycle. Then, $G(V, E^{l_{\vec{x}}})$ has a simple cycle, and let $e_{i_1}, e_{i_2}, \ldots, e_{i_r}$, are the edges from that cycle. By $A = \{i_1, \ldots, i_r\}$ we will denote the set of indices of the cycle's edges.

Let $m = \min\{x_j | j \in A\}$ and $\vec{y}$ is defined as $y_j = \begin{cases} m, & j \in A \\ 0, & j \notin A \end{cases}$.

We will show that the state vector $\vec{z} = \vec{x} - \vec{y}$ is also *d-P*, which, having in mind that $0 \leq \vec{x} - \vec{y} < \vec{x}$, since $c(e_i) > x_i - y_i > 0$, contradicts with the assumption that $\vec{x}$ is *d-MP*.

First we will show that the total flow for each vertex u remains the same. Clearly, if u does not belong to the cycle, the total flow in and the total flow out have no changes. If u belongs to the cycle, we have:

$$\sum_{v \in V} l_{\vec{z}}(u, v) - \sum_{v \in V} l_{\vec{z}}(v, u) = \sum_{v \in V} l_x(u, v) - m - \sum_{v \in V} l_{\vec{x}}(v, u) + m$$

$$= \sum_{v \in V} l_x(u, v) - \sum_{v \in V} l_{\vec{x}}(v, u)$$

This implies that the flow conservation constraints are satisfied and $|l_{\vec{z}}| = |l_{\vec{x}}| = d$. So $\vec{z}$ is a *d-P*.

In opposite, assume that the state vector $\vec{x}$ is such that $l_{\vec{x}}$ is a flow function with $|l_{\vec{x}}| = d$ and $G(V, E^{l_{\vec{x}}})$ is acyclic. We will prove that for each state vector $\overrightarrow{x'} < \vec{x}$, $|l_{\overrightarrow{x'}}| < d$.

Let us suppose that there is a path $\overrightarrow{x'}$ to level d such that $\overrightarrow{x'} < \vec{x}$. Without any loss of generality, we can suppose that $(\exists! i)\ x'_i < x_i$. Let $e_i = (w, w_1)$.

We have that

$$x_i = l_{\vec{x}}(w, w_1) > l_{\overrightarrow{x'}}(w, w_1) = x'_i,$$

and for all other vertices u and v,

$$l_{\vec{x}}(u,v) = l_{\overrightarrow{x'}}(u,v).$$

Since $G(V, E^{l_{\vec{x}}})$ is acyclic, we are able to sort its vertices topologically. The same topological sort may be applied to $G(V, E^{l_{\vec{x}'}})$. Taking S to be the set of all vertices between s and w, inclusively, and $T = V \setminus S$, we will obtain a cut in $G(V, E^{l_{\vec{x}'}})$ with flow $d - (x_i - x_i') < d$. This proves that $G(V, E^{l_{\vec{x}'}})$ has maximal flow lower than d, which is in contradiction with our assumption that $\overrightarrow{x'}$ is a *d-P*.

Using this Theorem and Lin's algorithm (Lin *et al.*, 1995) for calculating *d-MP* candidates, the family of all *d-MPs* can be generated by the following steps:

Algorithm 1.

Step 1. Using Lin's Algorithm, find the set Q of all flow functions for which the induced vectors are candidates for *d-MP*.

Step 2. For each candidate $\vec{x}$ check for cycle in $G(V, E^{l_{\vec{x}}})$, and, if there is a cycle, remove it from Q.

Checking for a cycle in a graph may be simply done using DFS (Kamil, 2003), so this takes time $O(|E|)$. As a result, the time complexity of our algorithm is $O(|E|\lambda)$, where λ is an upper bound for the number of obtained candidates by Lin's algorithm. This is a significantly lower complexity than the complexity of the algorithm given in (Forghani-elahabad *et al.*, 2013), $O(|V|^2|E|^{3/2})$.

4. The Correlation between Two Minimal Path Vectors

The Ford-Fulkerson algorithm gives us a way to compute only one flow function for maximal flow in a given two-terminal network, as well as its corresponding residual network, with time complexity $O(|V||E|^2)$. The same approach may be used for computation of a flow function to level d. Using Theorem 1 we are able to find one *d-MP*. Here we give the connection between the two flow functions which may contribute in developing another algorithm for computing all *d-MP*s.

Theorem 2. Let f be a flow with $|f| = d$ in a two-terminal flow network $G(V,E,c)$ with source s and sink t, and $\mathcal{C}$ be an augmenting cycle in the residual graph $G_f(V,E_f)$. For $d' \le \min\{c_f(e_i) | e_i \in \mathcal{C}\}$, the function f_1 defined as

$$f_1(u,v) = \begin{cases} f(u,v) + d', & (u,v) \in \mathcal{C} \cap E \\ f(u,v) - d', & (v,u) \in \mathcal{C} \text{ and } (u,v) \in E \\ f(u,v), & (u,v) \notin \mathcal{C} \text{ and } (v,u) \notin \mathcal{C} \end{cases} \tag{4.1}$$

is a flow in $G(V,E,c)$ with $|f_1| = |f|$.

Proof. To prove that f_1 is a flow function we need to show that capacity constraints and flow conservations are satisfied.

Capacity constraints.

- If $(u,v) \notin \mathcal{C}$ and $(v,u) \notin \mathcal{C}$,
  $f_1(u,v) = f(u,v)$, so
  $0 \le f_1(u,v) \le c(u,v)$.

- If $(u,v) \in \mathcal{C} \cap E$,
  $f_1(u,v) = f(u,v) + d' \ge f(u,v) \ge 0$ and
  $$f_1(u,v) = f(u,v) + d' \le f(u,v) + c_f(u,v) = f(u,v) + c(u,v) - f(u,v) = c(u,v)$$
- If $(v,u) \in \mathcal{C}$ and $(u,v) \in E$,
  $f_1(u,v) = f(u,v) - d' \le f(u,v) \le c(u,v)$ and
  $f_1(u,v) = f(u,v) - d' \ge f(u,v) - c_f(u,v) = f(u,v) - f(u,v) \ge 0$

Flow conservation. To show that the flow conservation conditions are satisfied, it is sufficient to prove that

$$\sum_{u \in V} f_1(u,v) - \sum_{w \in V} f_1(v,w) = \sum_{u \in V} f(u,v) - \sum_{w \in V} f(v,w).$$

Clearly, the last equation holds for $v \notin \mathcal{C}$, since in that case $f(u,v) = f_1(u,v)$ and $f(v,w) = f_1(v,w)$, for all u and w.

For $u \in \mathcal{C}$, since $\mathcal{C}$ is simple, u appears exactly once in $\mathcal{C}$. Let u_1 and w_1 are nodes such that $(u_1,v) \in \mathcal{C}$ and $(v,w_1) \in \mathcal{C}$. There are four possibilities:

- For $(u_1,v) \in E$ and $(v,w_1) \in E$

$$\begin{aligned} f_1(V,v) - f_1(v,V) &= f_1\left(\frac{V}{\{u_1\}},v\right) + f_1(u_1,v) - f_1\left(v,\frac{V}{\{w_1\}}\right) - f_1(v,w_1) \\ &= f\left(\frac{V}{\{u_1\}},v\right) + f(u_1,v) + d' - f\left(v,\frac{V}{\{w_1\}}\right) - (f(v,w_1) + d') \\ &= f(V,v) - f(v,V). \end{aligned}$$

- For $(u_1,v) \in E$ and $(w_1,v) \in E$

$$\begin{aligned} f_1(V,v) - f_1(v,V) &= f_1\left(\frac{V}{\{u_1,w_1\}},v\right) + f_1(u_1,v) + f_1(w_1,v) - f_1(v,V) \\ &= f\left(\frac{V}{\{u_1,w_1\}},v\right) + f(u_1,v) + d + f(w_1,v) - d' - f(v,V) \\ &= f(V,v) - f(v,V) \end{aligned}$$

- For $(v,u_1) \in E$ and $(v,w_1) \in E$

$$f_1(V,v) - f_1(v,V) = f_1(V,v) - f_1\left(v, \frac{V}{\{u_1, w_1\}}\right) - f_1(v,u_1) - f_1(v,w_1)$$

$$= f(V,v) - f\left(v, \frac{V}{\{u_1, w_1\}}\right) - (f(v,u_1) - d') - (f(v,w_1) + d')$$

$$= f(V,v) - f(v,V)$$

- For $(v,u_1) \in E$ and $(w_1,v) \in E$

$$f_1(V,v) - f_1(v,V) = f_1\left(\frac{V}{\{w_1\}}, v\right) + f_1(w_1,v) - f_1\left(v, \frac{V}{\{u_1\}}\right) - f_1(v,u_1)$$

$$= f\left(\frac{V}{\{w_1\}}, v\right) + f(w_1,v) - d' - f\left(v, \frac{V}{\{u_1\}}\right) - (f(v,u_1) - d')$$

$$= f(V,v) - f(v,V)$$

The proof is completed with

$$|f_1| = \sum_{u\in V} f_1(u,s) - \sum_{w\in V} f_1(s,w) = \sum_{u\in V} f(u,s) - \sum_{w\in V} f(s,w) = |f|.$$

Directly from the last Theorem we have the following corollary:

COROLLARY 1. Let $\vec{x}$ be a state vector for a two-terminal flow network $G(V,E,c)$ with source s and sink t such that the pseudoflow function $l_{\vec{x}}$ induced by $\vec{x}$ is a flow function with $|l_{\vec{x}}| = d$, and let $\vec{y}$ be an augmenting vector to level d' for a cycle C in the residual network $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$. Then $\vec{x} + \vec{y}$ is a state vector for $G(V,E,c)$ such that the pseudoflow function $l_{\vec{x}+\vec{y}}$ induced by $\vec{x} + \vec{y}$ is a flow function with $|l_{\vec{x}+\vec{v}}| = d$.

Lemma 1. Let f be a flow with $|f| = 0$ in the two-terminal network $G(V,E,c)$ such that there is an edge (u,v) for which $f(u,v) > 0$. Then the graph $G(V,E^f)$ induced by f contains a cycle.

Proof. Directly follows from two facts. The first one is that the indegree of each node in the graph $G(V,E^f)$ is strictly greater than 0 if and only if its outdegree is also strictly greater than 0. The second one is that there is a path passing through (u,v).

Lemma 2. Let $\vec{x}$ be a state vector for a two-terminal flow network $G(V,E,c)$ with source s and sink t such that $l_{\vec{x}}$ is a flow function with $|l_{\vec{x}}| = 0$. Then there are augmenting vectors $\vec{y}_k,\ k = 1, \dots, r$ to levels d'_k for cycles C_k in $G(V,E,c)$, such that $\vec{x} = \sum_{k=1}^{r} \vec{y}_k$.

Proof: Since $l_{\vec{x}}$ is a flow function with $|l_{\vec{x}}| = 0$, from Lemma 1 it follows that $G(V,E^{l_{\vec{x}}})$ contains a cycle C_1. Taking $d'_1 = \min\{x_i | e_i \in C_1\}$ we may construct an augmenting vector $\vec{y}_1$ to level d'_1. The vector $\vec{z}_1 = \vec{x} - \vec{y}_1$ is a state vector for the two-terminal flow network $G(V,E,c)$ with $|l_{\vec{z}_1}| = 0$, too, and moreover, the graph $G(V,E^{l_{\vec{z}_1}})$ has at least one positive edge less than $G(V,E^{l_{\vec{x}}})$. If $G(V,E^{l_{\vec{z}_1}})$ has no edge (u,v) such that $l_{\vec{z}_1}(u,v) > 0$, we are finished. In opposite, we continue with this procedure of constructing an augmenting vector $\vec{y}_k$ to level d'_k for $G(V, E^{l_{\vec{z}_{k-1}}})$ and a vector $\vec{z}_k = \vec{x} - \vec{y}_k$, until $\vec{z}_k = \vec{0}$. The procedure will finish in at least $|E|$ steps. Each cycle C_k is a cycle in $G(V, E, c)$ since the graph $G(V, E^{l_{\vec{z}_{k-1}}})$ is a subgraph of G.

The next theorem shows that given a state vector $\vec{x}$, every other flow vector can be obtained by adding cycles from $G(V, E^{l_{\vec{x}}})$ to $\vec{x}$.

Theorem 3. Let $\vec{x}$ and $\vec{y}$ be two state vectors for a flow network $G(V, E, c)$ with source s and sink t, such that $l_{\vec{x}}$ and $l_{\vec{y}}$ are flow functions with $|l_{\vec{x}}| = |l_{\vec{v}}| = d$. Then there are augmenting vectors $\vec{y}_k,\ k = 1, \dots, r$ of levels d'_k for cycles C_k in the residual network $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$ such that $\vec{y} = \vec{x} + \sum_{k=1}^{r} \vec{y}_k$.

Proof. The function $-l_{\vec{x}}$ is a flow function in $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$ from t to s with $|l_{\vec{v}}| = d$, and its residual graph is $G(V, E, c)$. Using this and the fact that $l_{\vec{v}}$ is a flow function from s to t in $G(V, E, c)$ with $|l_{\vec{v}}| = d$, we have that the function $l_{\vec{v}-\vec{x}}$ defined as $l_{\vec{v}-\vec{x}}(u, v) = l_{\vec{v}}(u, v) - l_{\vec{x}}(u, v)$ is a flow function in the residual graph $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$ with $|l_{\vec{v}-\vec{x}}(u, v)| = 0$. The state vector for $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$ induced by $l_{\vec{v}-\vec{x}}(u, v)$ is $\vec{y} - \vec{x}$. Since $|l_{\vec{v}-\vec{x}}(u, v)| = 0$, there are augmenting vectors $\vec{y}_k,\ k = 1, \dots, r$ of levels d'_k for cycles C_k in $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$, such that $\vec{y} - \vec{x} = \sum_{k=1}^{r} \vec{y}_k$ (by Lemma 2).

Theorem 1, Theorem 2 and Theorem 3 are sublimated in the next theorem.

Theorem 4. Given a two-terminal flow network $G(V, E, c)$ with source s and sink t, let $\vec{x}$ be a *d-MP*. Then $\vec{y}$ is a *d-MP* if and only if $G(V, E^{l_{\vec{y}}})$ is an acyclic graph and there are augmenting vectors $\vec{y}_k,\ k = 1, \dots, r$ of levels d'_k for cycles C_k in the residual network $G_{l_{\vec{x}}}(V, E_{l_{\vec{x}}})$, such that $\vec{y} = \vec{x} + \sum_{k=1}^{r} \vec{y}_k$.

5. Algorithm for Calculating all Minimal Path Vectors Using One Specified Path Vector and Cycles in their Corresponding Residual Network

Given a two-terminal flow network $G(V, E, c)$ with source s and sink t, let us suppose that the i-th component may operate in one of the levels from the set $\{0, 1, \dots, M_i\}$. Assuming that the maximal flow of the network is M, the set $\{0, 1, \dots, M\}$ is the set of all possible flows. Using the results from the previous section, we give an approach that can help us design an algorithm for computing all *d-MPs* for a given level $d \leq M$. The pseudocode for the main algorithm is the following:

Algorithm 2.

Step 1. Using Ford-Fulkerson algorithm, find one flow function f to level d.
Step 2. While $G(V, E^f)$ has a cycle, set $f = f_1$ using (4.1).
Step 3. Find all cycles in the f's residual graph, and construct their augmenting vectors $\vec{y}_k$.
Step 4. Check each $\vec{x} + \sum_{k=1}^{r} \vec{y}_k$ for cycle and print it if there is no cycle.

We can do some optimizations in order to minimize the repetition of *d-MPs* as well as to obtain vectors that are not *d-MPs*. Here, we do not discuss the strategy for enumeration of cycles.

This algorithm is useful for finding all *d-MP*s near the maximal one, since instead of adding d 1-*MP* vectors as in the algorithms proposed in (Mihova and Maksimova, 2011; Forghani-elahabad *et al.*, 2013; Lin *et al.*, 1995), here we obtain a candidate for *d-MP* only by one vector addition. The approach is especially useful for level M, since in this case the residual graph can be divided into a few connected components; each cycle must lie into exactly one of those components.

The residual graph obtained using the Ford-Fulkerson algorithm for maximum flow M, can be divided into strongly connected components, as explained in (Picard and Maurice, 1980; Bezakova and Friedlander, 2010). These components are used to obtain all minimum cuts. The edges lying on some minimum cut must be used with their full capacity. Moreover, each cycle must lie into exactly one of those components. This reduces the length, as well as the number of cycles. Furthermore, each *M-MP* vector can be obtained by joining the sub-vectors corresponding to each of the strongly connected components, as well as the subvector of the edges connecting a pair of strongly connected components.

Let $G(V,E,c)$ be a two-terminal network with maximum flow M and $G_{l_{\vec{x}}}(V,E_{l_{\vec{x}}})$ be the residual network for the *M-MP* $\vec{x}$. Assume that $G(V_k,E_k)$ are subgraphs of $G(V,E)$ such that V_k is the set of all nodes in the k-th connected component of $G_{l_{\vec{x}}}(V,E_{l_{\vec{x}}})$, and $E_k = \{(u,v) \mid u,v \in V_k\}$. Let $E' = \bigcap_k\{(u,v) \notin Vk\}$, i.e. the set of all arcs that are not in a connected component. Then each $e_i \in E'$ lies on some minimum cut and $l_{\vec{x}}(e_i) = x_i$. Moreover, for each other *M-MP* $\vec{y}$, $l_{\vec{y}}(e_i) = x_i$. On the other hand, the flow in $G(V_k,E_k)$ is equal to the flow out $G(V_k,E_k)$, i.e. $l_{\vec{y}}(V\backslash V_k,V) = l_{\vec{y}}(V,V\backslash V_k) = d_k$. Now we may propose an algorithm for *M-MP*.

Algorithm 3.

Step 1. Use Ford-Fulkerson algorithm to find a flow function f for maximum level M.

Step 2. Find strongly connected components of the residual graph and set $x_i = f(e_i)$ for all $e_i \in E'$.

Step 3. For each strongly connected component, use Algorithm 2 to compute the set of all subvectors D_k.

Step 4. Find all *M-MP*s $\vec{y}$ for which

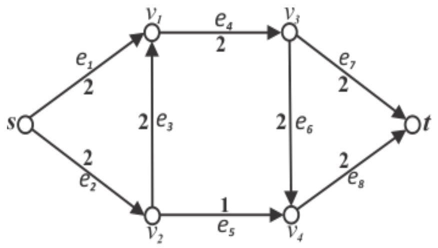
$$y_i = \begin{cases} f(e_i), & e_i \in E' \\ l_{\vec{x}^k}(e_i), & e_i \in V_k \text{ and } \vec{x}^k \in D_k \end{cases}$$

The algorithm is illustrated in the following example.

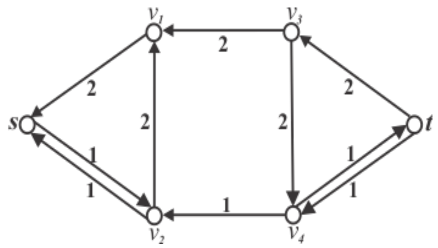
EXAMPLE 1. Given the network in Fig. 2 a), the residual network for a maximal level 3 for the flow vector (2, 1, 0, 2, 1, 0, 2, 1) is shown in Fig. 2 b). The graph is divided into two subgraphs. Since the components in the cut must be used with their full capacity, each minimal 3-*MP* has the form $(x_1, x_2, x_3, 2, 1, x_6, x_7, x_8)$. There is only one augmenting cycle in the first strongly connected component, consisting of edges e_1, e_2 and e_3. This cycle is (–1, 1, 1).

The second strongly connected component, consisting of edges e_6, e_7 and e_8 also has one augmenting cycle: (1, –1, 1).

The subvectors corresponding to the first strongly connected component for which the



a)



b)

Fig. 2.: a) A flow network. b) Residual network for the flow vector (2, 1, 0, 2, 1, 0, 2, 1).

graph is acyclic are $\{(2,1,0),\ (1,2,1)\}$, $\{(2-1, 1+1, 0+1) = (1,2,1)\}$, while the subvectors corresponding to the second strictly connected component are $\{(0,2,1), (1,1,2)\}$. So all 3-*MPs* are all vectors obtained by joining the subvectors from this two sets together with the values of the components on the cut, i.e. $\{(2,1,0,2,1,0,2,1), (1,1,2,2,1,0,2,1), (2,1,0,2,1,1,1,2), (1,1,2,2,1,1,1,2)\}$.

6. Conclusion

Known algorithms for computing the set of all *d-MP*s commonly use network flows. In this paper we proved that a flow function corresponds to a *d-MP* if and only if the graph appropriate to that flow is acyclic. This property helped us to design an algorithm for calculating the set of all *d-MP*s, which is more efficient than the known algorithms for solving this problem. Moreover, by further analyses on the connection between two *d-MP*s, we proposed a strategy for calculating all *d-MP*s, given only one *d-MP* obtained using the Ford-Fulkerson algorithm. The proposed strategy is especially efficient for large levels. The strategy for enumeration of cycles addressed in Section 5 is one topic for our future work.

References

Aggarwal, K., Chopra, Y., Bajwa, J. (1982). Capacity consideration in reliability analysis of communication systems. *IEEE Transactions on Reliability*, 31(2), 177–180.

Ahuja, R.K., Orlin, J.B. (1993). A capacity scaling algorithm for the constrained maximum flow problem. *Networks*, 25, 89–98.

Bezakova, I., Friedlander, A.J. (2010). Counting minimum (s, t)-Cuts in weighted planar graphs in polynomial time. In: *Proceedings of 35th International Symposium, MFCS 2010, Brno, Czech Republic, August 23–27, 2010*.

Cormen, H.T., Leiserson, E.C., Rivest, R.L., Stein, C. (2009). *Introduction to Algorithms*, 3rd edition. The MIT Press

Erickson, J. (2009). *Algorithms*. University of Illinois at Urbana-Champaign.

Forghani-elahabad, M., Mahdavi-Amiri, N. (2013). 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*.

Jane, C.C., Lin, J.S., Yuan, J. (1993). Reliability evaluation of a limited-flow network in terms of minimal cut sets. *IEEE Transactions on Reliability*, 42, 354–361.

Kamil, A. (2003). "*Graph Algorithms*", *CS61B*. University of California, Berkeley

Lin, J.S., Jane C.C., Yuan, J. (1995). On reliability evaluation of a capacitated – flow network in terms of minimal path sets. *Networks*, 3, 131–138.

Lin, Y.K. (2001). A simple algorithm for reliability evaluation of a stochastic – flow network with node failure. *Computers and Operations Research*, 28, 1277–1285.

Mihova, M., Maksimova, N. (2011). Estimation of minimal path vectors of multi state two terminal networks with cycles control. *Mathematica Balkanica*, 25(4), 437–447.

Picard, J.C., Maurice, Q. (1980). On the structure of all minimum cuts in a network and application. *Mathematical Programming Study*, 13, 8–16.

Provan, J.S., Balls, M.O. (1983). The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4), 777–788.

Ramirez-Marquez, J.E, Coit, D. (2003). Alternative approach for analyzing multistate network reliability. In: *Proceedings of the Industrial Engineering Research Conference (IERC), Portland, OR, May 2003*.

Ramirez-Marquez, J.E., Coit, D., Tortorella, M., (2003). Multi-statetwo-terminal reliability: a generalized cut-set approach. *Rutgers University IE Working Paper*, 03–135.

Wilson, A., Limnios, N., Keller-Mc-Nulty, S., Armijo, Y. (2005). *Modern Statistical and Mathematical Methods in Reliability*. World Scientific Publishing, Series on Quality, Reliability, and Engineering Statistics, vol. 10.

M. Mihova is an associate professor at the Faculty of Computer Science and Engineering, University "Ss. Cyril and Methodius", in Skopje. She is a member of the board of the Computer Society of Macedonia. Her research interest is in the field of applied mathematics, more specifically applied probability and statistics, with focus on mathematical models in reliability, especially reliability of multi-state systems.

N. Stojkovikj is a teaching and research assistant of the Faculty of Computer Science, University "Goce Delchev" in Shtip. Currently, she is preparing her PhD thesis on the topic of reliability of multi-state two-terminal network.

M. Jovanov is an ass. professor at the Faculty of Computer Science and Engineering, University "Ss. Cyril and Methodius", in Skopje. He is the President of the Computer Society of Macedonia and has actively participated in the organization and realization of the Macedonian national competitions and Olympiads in informatics since 2001. He has been a team leader for the Macedonian team since 2006.

E. Stankov is a teaching and research assistant at the Faculty of Computer Science and Engineering, University "Ss. Cyril and Methodius", in Skopje. He is a member of the Executive Board of the Computer Society of Macedonia and has actively participated in the organization and realization of the Macedonian national competitions and Olympiads in informatics since 2009. Currently he is PhD student at the Faculty of Computer Science and Engineering.