Search for MPs in Modified Networks

Wei-Chang Yeh
Feng Chia University, Taichung, Taiwan

Abstract — An efficient and effective method is proposed to search for all MPs in a network (modified networks) obtained by modifying the original network. This process can be used for reliability analysis of various modifications in an existing network for network expansion or reinforcement evaluation and planning. Our method is based upon the reformation of all MPs in the original network. Without researching for all MPs in the modified network, the proposed algorithm is more efficient and effective to implement. In this presentation, first we develop an intuitive algorithm to search for all MPs in a modified network. Next the computational complexity of the proposed algorithm is analyzed and compared with the existing methods. Finally, examples are illustrated to show how all MPs are generated in a modified network based upon the reformation of all of the MPs in the corresponding original network.

1. INTRODUCTION

In recent years, in order to validate, verify the designs and evaluate the performance, many real world systems, such as computer and communication systems [1,2,3], power transmission and distribution systems [4,5], transportation systems [6, 7], oil/gas production system [7, 8] etc. are first modeled as networks. Reliability is then usually selected to be one of the most important indexes of this network.

To evaluate reliability, all MCs or MPs of the system must be known in advance. However, both the problems in locating all MCs/MPs and computing the system reliability in terms of the known MCs/MPs are NP-hard [4,5,9-25]. There is generally a need to evaluate only a few of the modifications in an existing network for expansion or reinforcement without locating all MPs or MCs.

In many cases, modification of an existing network for network expansion or reinforcement planning is often necessary [23,24]. Such problems have been posed and solved recently for MCs [23] and MPs [24], separately. Nevertheless, for the MP problem, the best known method proposed by [24] was a straightforward approach that required extensive comparison and verification. This method failed to solve some special but important cases in a modified network. The need for a more efficient, intuitive and generalized method to search for all MPs without an extensive research procedure thus arose. The main purpose of this article is to present an efficient method to search for all of the MPs in a modified network. This proposed algorithm is very effective and efficient when compared to the best known method.

This paper is organized as follows. Section 2 describes the notation, nomenclature and assumptions required. Some important lemmas and theorems are discussed in Section 3. Section 4 discusses the best known algorithm, then presents the proposed method in detail, together with a discussion of the time complexity and a comparison of the efficiency between our method and the best-known algorithm. Two illustrative examples are included in Section 5 to show how to generate all of the MPs in a modified network through our algorithm. Concluding remarks are given in Section 6.

2. NOTATION, NOMENCLATURE AND ASSUMPTIONS

Notation:

- $G(V, E)$: An original network with the set of nodes V and the set of arcs E.
- n, m, δ: The number of nodes, arcs and MPs in $G(V, E)$, respectively.
- s, t: $s, t\in V$ is the specified source node and sink node, respectively.
- $|\bullet|$: The number of elements of $\bullet$.
- $\pi_{ab}(\bullet)$: The path from node a to node b in $\bullet$.
- B_{xy}: A branch string, i.e. a path, which connects node x to node y.
- $P_{a\backslash b}$: The set of all MPs which all pass through node a but not node b.
- $P_{a\to b}$: The set of all MPs which all pass through node a before passing through node b.
- $S\otimes T$: $S\otimes T=\cup_{i,j}\{S_i\cup T_j\}$, if $S\neq\varnothing$ and $T\neq\varnothing$, where for all $S_i\in S$ and $T_j\in T$. Otherwise $S\otimes T=\varnothing$.
- $S\otimes_{ab}T$: $S\otimes_{ab}T=S\otimes T\otimes B_{ab}$.
- $G(V, E)\cup B_{xy}$: The modified network after inserting a branch string B_{xy} in $G(V, E)$, where $B_{xy}\cap G(V, E)=\{x, y\}$.

Nomenclature

MP/MC: It is a path (cut) set such that if any arc is removed from the set, then the remaining set is no longer a path (cut) set.

Branch String: A branch string is a connected path with at least two arcs between two original network nodes, for example x and y [23,24]. All nodes, except nodes x and y, contained in the branch string are added with only two degrees. For example, $B_{13}=\{e_{10}, e_{03}\}$ is a branch string of Fig 2 and is added into the network in Fig 1.

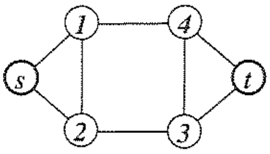

Fig 1.

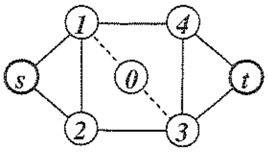

Fig 2.

Modified Network: A modified network is an updated network after inserting a branch string between two nodes in the original network [23,24]. For example, Fig 2 is the modified network of Fig 1 by adding B_{13}, i.e. Fig 1 is the original network of Fig 2.

Assumptions

The network must satisfy the following assumptions [23,24]:

(1) Each node is perfectly reliable [5,25].
(2) The network graph is connected [25].
(3) There are no common-cause outages [20,21].
(4) There are no parallel branches.
(5) Degrees of all network nodes are at least 2, except for the source node s and sink node t.

3. PRELIMINARIES

A special property is considered first, which is simply proved, discusses network modification by open removal of branches as follows:

Theorem 1: Any MP p in $G(V, E)$ is also a MP in the modified network after modifying $G(V, E)$ by removing a branch string B_{xy}, if and only if $p\cap B_{xy}\cap\subseteq\{x, y\}$.

From Theorem 1, the modifications to MPs are trivially accounted for by just discarding the MPs associated with the removed branch [24]. Therefore, inserting a branch string connecting two distinct nodes will only focus on the modification of the original network by inserting a branch string connecting two distinct nodes in the remainder of this study.

Theorem 2: Any MP in $G(V, E)$ is also an MP after modifying $G(V, E)$ by inserting a branch string between two nodes.

From Theorem 2, to determine all new MPs after inserting a branch string connecting two distinct nodes is the key part of enumerating the MPs of a modified network. Furthermore, all new MPs in modified network are related to the MPs passing through node x and/or y in $G(V, E)$. Thus, all of the MPs in $G(V, E)$ that passed through node x and/or y are separated into four disjoint subsets as in Lemma 1.

Lemma 1: All MPs that passed through node x and/or node y is $P_{x\backslash y}\cup P_{y\backslash x}\cup P_{x\to y}\cup P_{y\to x}$, and the intersection of any two of $P_{x\backslash y}$, $P_{y\backslash x}$, $P_{x\to y}$, and $P_{y\to x}$ are empty.

Proof: It is trivial.

The relationship between a new branch string and any MP is discussed as follows.

Lemma 2: If MP p_1 passes node a and MP p_2 passes node b in $G(V, E)\cup B_{ab}$, individually, then $B_{ab}\cap\pi_{sa}(p_1)=\{a\}$ and $B_{ab}\cap\pi_{bt}(p_2)=\{b\}$.

Proof: All of the arcs in B_{xy} are new to all of the arcs in E. Therefore, this proof follows.

Any MP, say p, can be separated into two paths $\pi_{sa}(p)$ and $\pi_{at}(p)$, where node $a\in p$, $\pi_{sa}(p)$ is from node s to node a, and $\pi_{at}(p)$ is from node a to node t. Moreover, $\pi_{sa}(p)$ passes (not) through node b, if and only if $p\in P_{b\to a}$ $(p\in P_{b\backslash a})$, and $\pi_{at}(p)$ passes (not) through node b, if and only if $p\in P_{a\to b}$ $(p\in P_{a\backslash b})$. Greater detail is stated as follows.

Lemma 3: Let p be an MP. Then $p=\pi_{sa}(p)\cup\pi_{at}(p)$, if and only if $\pi_{sa}(p)\cap\pi_{at}(p)=\{a\}$, where $a\in V-\{s, t\}$.

Proof: If $p=\pi_{sa}(p)\cup\pi_{at}(p)$, i.e. passes node a, then it is trivial that p can be separated into two paths $\pi_{sa}(p)$ and $\pi_{at}(p)$ with $\pi_{sa}(p)\cap\pi_{at}(p)=\{a\}$, where $a\in V-\{s, t\}$. Suppose $\pi_{sa}(p)\cap\pi_{at}(p)=\{a\}$, then $p-[\pi_{sa}(p)\cap\pi_{at}(p)]$ still connects nodes s and t, i.e. this contradicts that p is an MP. This is concludes the proof.

Theorem 3 is a generalized of Lemma 3, and plays an important role in the following theorems and lemmas.

Theorem 3: Let p be an MP. Then $p=\pi_{sa}(p)\cup\pi_{ab}(p)\cup\pi_{at}(p)$, if and only if $\pi_{sa}(p)\cap\pi_{ab}(p)=\{a\}$, $\pi_{sa}(p)\cap\pi_{bt}(p)=\varnothing$ and $\pi_{ab}(p)\cap\pi_{bt}(p)=\{b\}$.

A special situation of the subpaths of two MPs is also an MP is described as follows.

Theorem 4: Let p and p^* be two MPs and both pass node a. Then $\pi_{sa}(p)\cup\pi_{at}(p^*)$ is an MP if and only if $\pi_{sa}(p)\cap\pi_{at}(p^*)=\{a\}$.

Proof: If $p^\#=\pi_{sa}(p)\cup\pi_{at}(p^*)$ is an MP, then $\pi_{sa}(p^\#)=\pi_{sa}(p)$ and $\pi_{at}(p^\#)=\pi_{at}(p^*)$. From Lemma 3, $\pi_{sa}(p^\#)\cap\pi_{at}(p^\#)=\{a\}$, i.e. $\pi_{sa}(p)\cap\pi_{at}(p^*)=\{a\}$. On the other hand, $\pi_{sa}(p)$ and $\pi_{at}(p^*)$ are two disjoint paths. The former is from node s to node a in p and the rear is from node a to node t in p^*. Thus, $\pi_{sa}(p)\cup\pi_{at}(p^*)$ is a path from node s to node t, if $\pi_{sa}(p)\cap\pi_{at}(p^*)=\{a\}$.

The following theorem is proposed first, without the condition $(P_{a\backslash b}\cup P_{a\to b})\neq\varnothing$ and $(P_{b\backslash a}\cup P_{a\to b})\neq\varnothing$ in [24] to find all of MPs in the modified network.

Theorem 5: If $(P_{a\backslash b}\cup P_{a\to b})\neq\varnothing$ and $(P_{b\backslash a}\cup P_{a\to b})\neq\varnothing$, then the new generated MPs in $G(V, E)\cup B_{xy}$ are $[\pi_{sx}(P_{x\backslash y}\cup P_{x\to y})\otimes_{xy}\pi_{yt}(P_{y\backslash x}\cup P_{x\to y})]\cup[\pi_{sy}(P_{y\backslash x}\cup P_{y\to x})\otimes_{yx}\pi_{xt}(P_{x\backslash y}\cup P_{y\to x})]$.

Theorem 5 is a straightforward method but less efficient because the intersections among MPs [24] must be verified. In addition, this method also fails to search for all new MPs in the modified network while $(P_{a\backslash b}\cup P_{a\to b})\neq\varnothing$ or $(P_{b\backslash a}\cup P_{a\to b})\neq\varnothing$. For example, consider a special but practical problem of the network in the following graph:

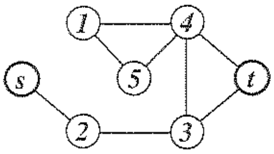

Fig 3.

No MPs pass node 1 in the original graph. But, after inserting e_{13}, $\{e_{s2}, e_{23}, e_{31}, e_{14}, e_{4t}\}$ the new MP passes node 1. Therefore, if without condition $(P_{a\backslash b}\cup P_{a\to b})\neq\emptyset$ and $(P_{b\backslash a}\cup P_{a\to b})\neq\emptyset$, Theorem 5 is not true. Hence, the best known algorithm proposed in [24] works only under the condition of $(P_{a\backslash b}\cup P_{a\to b})\neq\emptyset$ and $(P_{b\backslash a}\cup P_{a\to b})\neq\emptyset$.

To overcome the above two problems, some more important rules are presented to exploit the MP structure.

Lemma 4: The following statements hold:

Rule 1: $P_{a\to b}=(P_{a\to b}-P_{a\backslash b})\cup(P_{a\to b}\cap P_{a\backslash b})$.

Rule 2: $\pi_{sa}(S\cup T)=\pi_{sa}(S)\cup\pi_{sa}(T)$.

Rule 3: $\pi_{sa}(S\cup T)\otimes_{ab}\pi_{bt}(U)=[\pi_{sa}(S)\otimes_{ab}\pi_{bt}(U)]\cup[\pi_{sa}(T)\otimes_{ab}\pi_{bt}(U)]$.

Rule 4: $\pi_{sa}(S\cup T)\otimes_{ab}\pi_{bt}(U\cup W)=[\pi_{sa}(S)\otimes_{ab}\pi_{bt}(U)]\cup[\pi_{sa}(T)\otimes_{ab}\pi_{bt}(U)]\cup[\pi_{sa}(S)\otimes_{ab}\pi_{bt}(W)]\cup[\pi_{sa}(T)\otimes_{ab}\pi_{bt}(W)]$.

The formulation described in Theorem 5 is rewritten as follows:

Theorem 6 : $\pi_{sa}(P_{a\backslash b}\cup P_{a\to b})\otimes_{ab}\pi_{bt}(P_{b\backslash a}\cup P_{a\to b})=[\pi_{sa}(P_{a\to b}-P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{a\to b}-P_{b\backslash a})]\cup[\pi_{sa}(P_{a\to b}-P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})]\cup[\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{a\to b}-P_{b\backslash a})]\cup[\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})]$.

Proof: Since

$$\pi_{sa}(P_{a\backslash b}\cup P_{a\to b})=\pi_{sa}((P_{a\to b}-P_{a\backslash b})\cup P_{a\backslash b}) \quad \text{(rule 1 \& 2)}$$
$$=\pi_{sa}(P_{a\to b}-P_{a\backslash b})\cup\pi_{sa}(P_{a\backslash b}), \quad \text{(rule 3)}$$

And in the same way, we have

$$\pi_{sa}(P_{a\backslash b}\cup P_{a\to b})=\pi_{sa}(P_{a\backslash b})\cup\pi_{sa}(P_{a\to b}-P_{a\backslash b}).$$

Therefore, by rule 4, this concludes the proof.

The following theorem shows that if $p\in P_{a\to b}$ and $\pi_{ab}(p)$ is replaced in p by B_{ab}, then this new path is an MP in $G(V, E)\cup B_{ab}$.

Theorem 7 : If $p=\pi_{sa}(p)\cup\pi_{ab}(p)\cup\pi_{bt}(p)\in P_{a\to b}$, then $\pi_{sa}(p)\cup B_{ab}\cup\pi_{bt}(p)$ is an MP in $G(V, E)\cup B_{ab}$.

Proof: If $p=\pi_{sa}(p)\cup\pi_{ab}(p)\cup\pi_{bt}(p)$ is an MP, then $\pi_{sa}(p)\cap\pi_{bt}(p)=\emptyset$ by Theorem 3. Besides, $\pi_{sa}(p)\cap B_{ab}=\{a\}$ and $\pi_{bt}(p)\cap B_{ab}=\{b\}$. Hence, $\pi_{sa}(p)\cup B_{ab}\cup\pi_{bt}(p)$ is an MP in $G(V, E)\cup B_{ab}$.

The theorems and lemmas in the remainder of this section are presented to simplify in order the formulation in Theorem 7.

Lemma 5 : Let p_1 and p_2 be two MPs in $G(V, E)$. If $\pi_{sa}(p_1)\in\pi_{sa}(P_{a\to b})$, $\pi_{bt}(p_2)\in\pi_{bt}(P_{a\to b})$, and $\pi_{sa}(p_1)\cap\pi_{bt}(p_2)=\emptyset$, then either $\pi_{sa}(p_1)\cup\pi_{ab}(p_2)\cup\pi_{bt}(p_2)\in P_{a\to b}$ or $\pi_{bt}(p_2)\in\pi_{bt}(P_{b\backslash a})$.

Proof: Since $\pi_{sa}(p_1)\cap\pi_{bt}(p_2)=\emptyset$, if $\pi_{sa}(p_1)\cap\pi_{ab}(p_2)=\{a\}$, then $\pi_{sa}(p_1)\cup\pi_{ab}(p_2)\cup\pi_{bt}(p_2)\in P_{a\to b}$ (from Theorem 3). Otherwise, there is a node α ($\neq a$) in $\pi_{sa}(p_1)\cap\pi_{ab}(p_2)$, such that $\pi_{s\alpha}(p_1)\cap\pi_{\alpha b}(p_2)=\{\alpha\}$ and $\pi_{s\alpha}(p_1)\cup\pi_{\alpha b}(p_2)\cup\pi_{bt}(p_2)\in P_{b\backslash a}$, i.e. $p_2\in\pi_{bt}(P_{b\backslash a})$.

Lemma 6 : If $p\in P_{a\to b}$, $p^*\in P_{b\backslash a}$ and $\pi_{sa}(p)\cap\pi_{bt}(p^*)=\emptyset$, then $\pi_{sa}(p)\cup\pi_{bt}(p^*)\cup B_{ab}\in\pi_{sa}(P_{a\to b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})$. Furthermore, if $\pi_{sa}(p)\notin\pi_{sa}(P_{a\backslash b})$, then $\pi_{ab}(p)\cap\pi_{bt}(p^*)=\emptyset$, $\pi_{sa}(p)\cup\pi_{ab}(p)\cup\pi_{bt}(p^*)\in P_{a\to b}$, $\pi_{bt}(p^*)\in\pi_{bt}(P_{a\to b})$, and $\pi_{sa}(p)\cup\pi_{bt}(p^*)\cup B_{ab}\in\pi_{sa}(P_{a\to b}-P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})$.

Proof: If $p\in P_{a\to b}$, $p^*\in P_{b\backslash a}$ and $\pi_{sa}(p)\cap\pi_{bt}(p^*)=\emptyset$, then directly from the definition of MP, we have $\pi_{sa}(p)\cup\pi_{bt}(p^*)\cup B_{ab}\in\pi_{sa}(P_{a\to b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})$. Next, since $\pi_{sa}(p)\cap\pi_{bt}(p^*)=\emptyset$, $\pi_{sa}(p)\cap\pi_{ab}(p)=\{a\}$, if $\pi_{ab}(p)\cap\pi_{bt}(p^*)\neq\{b\}$, then there is a node c ($\neq b$) in $\pi_{ab}(p)\cap\pi_{bt}(p^*)$ such that $\pi_{ac}(p)\cup\pi_{ct}(p^*)$ is a path from node a to node t, and $\pi_{sa}(p)\cup\pi_{ac}(p)\cup\pi_{ct}(p^*)\in P_{a\backslash b}$. Therefore, if $\pi_{sa}(p)\notin\pi_{sa}(P_{a\backslash b})$, then $\pi_{ab}(p)\cap\pi_{bt}(p^*)=\{b\}$. If $\pi_{ab}(p)\cap\pi_{bt}(p^*)=\{b\}$, then it is obviously $\pi_{sa}(p)\cup\pi_{ab}(p)\cup\pi_{bt}(p^*)\in P_{a\to b}$ and $\pi_{sa}(p)\cup\pi_{bt}(p^*)\cup B_{ab}\in\pi_{sa}(P_{a\to b}-P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{a\to b}\cap P_{b\backslash a})$. Hence, the theorem follows.

In the same way Lemma 6 is proven. Lemma 7 follows immediately.

Lemma 7 : If $p\in P_{a\to b}$, $p^*\in P_{a\backslash b}$ and $\pi_{sa}(p^*)\cap\pi_{bt}(p)=\emptyset$, then $\pi_{sa}(p^*)\cup\pi_{bt}(p)\cup B_{ab}\in\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{a\to b})$. Besides, if $\pi_{bt}(p)\notin\pi_{bt}(P_{b\backslash a})$, then $\pi_{sa}(p^*)\cap\pi_{ab}(p)=\emptyset$, $\pi_{sa}(p^*)\cup\pi_{ab}(p)\cup\pi_{bt}(p)\in P_{a\to b}$, $\pi_{sa}(p^*)\in\pi_{sa}(P_{a\backslash b})$, and $\pi_{sa}(p^*)\cup\pi_{ab}(p)\cup B_{ab}\in\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{a\to b}-P_{b\backslash a})$.

The following theorem is a simplified form of Theorems 6 and 7. This theorem provides the basis for our algorithm and is generalized in the next theorem.

Theorem 8: If $P_{a\backslash b}\cup P_{a\to b}\neq\emptyset$ and $P_{b\backslash a}\cup P_{a\to b}\neq\emptyset$, then $\pi_{sa}(P_{a\backslash b}\cup P_{a\to b})\otimes_{ab}\pi_{bt}(P_{b\backslash a}\cup P_{a\to b})=[\cup_i\{\pi_{sa}(p_i)\cup\pi_{bt}(p_i)\cup B_{ab}\}]\cup[\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})]$, where for all $p_i\in P_{a\to b}$ with $\pi_{sa}(p_i)\notin P_{a\backslash b}$ or $\pi_{bt}(p_i)\notin P_{b\backslash a}$.

Proof: Since $\pi_{sa}(P_{a\backslash b}\cup P_{a\to b})\otimes_{ab}\pi_{bt}(P_{b\backslash a}\cup P_{a\to b})=\pi_{sa}(P_{a\to b}-P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{a\to b}-P_{b\backslash a})\cup\pi_{sa}(P_{a\to b}-P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})\cup\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{a\to b}-P_{b\backslash a})\cup\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})$, $\pi_{sa}(P_{a\to b}-P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{a\to b}-P_{b\backslash a})\subseteq\pi_{sa}(P_{a\to b})\otimes_{ab}\pi_{bt}(P_{a\to b})]\subseteq\cup_i\{\pi_{sa}(p_i)\cup\pi_{bt}(p_i)\cup B_{ab}\}\cup\pi_{sa}(P_{a\to b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})$, $\pi_{sa}(P_{a\to b}-P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})\subseteq\cup_i\{\pi_{sa}(p_i)\cup\pi_{bt}(p_i)\cup B_{ab}\}$, and $\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{a\to b}-P_{b\backslash a})\subseteq[\cup_i\{\pi_{sa}(p_i)\cup\pi_{bt}(p_i)\cup B_{ab}\}]$, we have $\pi_{sa}(P_{a\backslash b}\cup P_{a\to b})\otimes_{ab}\pi_{bt}(P_{b\backslash a}\cup P_{a\to b})\subseteq[\cup_i\{\pi_{sa}(p_i)\cup\pi_{bt}(p_i)\cup B_{ab}\}]\cup[\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})]$.

Besides, if $p_i\in[\cup_i\{\pi_{sa}(p_i)\cup\pi_{bt}(p_i)\cup B_{ab}\}]$ or $p_i\in[\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})]$, then p_i is a new MP. Hence, $[\cup_i\{\pi_{sa}(p_i)\cup\pi_{bt}(p_i)\cup B_{ab}\}]\cup[\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})]\subseteq\pi_{sa}(P_{a\backslash b}\cup P_{a\to b})\otimes_{ab}\pi_{bt}(P_{b\backslash a}\cup P_{a\to b})$. Thus, $\pi_{sa}(P_{a\backslash b}\cup P_{a\to b})\otimes_{ab}\pi_{bt}(P_{b\backslash a}\cup P_{a\to b})=[\cup_i\{\pi_{sa}(p_i)\cup\pi_{bt}(p_i)\cup B_{ab}\}]\cup[\pi_{sa}(P_{a\backslash b})\otimes_{ab}\pi_{bt}(P_{b\backslash a})]$.

The following theorem is the criteria implemented in our algorithm for building the associated MPs of the modified network. It is very simple but more efficient than the other best known algorithm for generating new MPs in a modified network. Our algorithm has none of the limitations listed in Theorem 9.

Theorem 9: If $(P_{x\backslash y}\cup P_{x\to y})\neq\varnothing$, then let node x to be node X. Otherwise, there is a path p_x from node s to node x (or from node x to node t) with node $X\in p_x$ such that there is a path from node s to node X to node t. If $(P_{x\to y}\cup P_{y\backslash x})\neq\varnothing$, then let node y to be node Y. Otherwise, there is a path p_y from node s to node y (or from node y to node t) with node $Y\in p_y$ such that there is a path from node s to node Y to node t. Let $B_{XY}=B_{xy}$ if $X=x$ and $Y=y$, otherwise $B_{XY}=\{p$: $p^*\cup B_{xy}$, where for all paths p^* from node X to node Y, and $B_{YX}=B_{yx}$ if $X=x$ and $Y=y$, otherwise $B_{YX}=\{p$: $p^*\cup B_{xy}$, where for all paths p^* from node Y to node X. Then the set of all new MPs, after inserting B_{xy} into $G(V, E)$ are $[\{\cup_i\{\pi_{sX}(p_i)\cup\pi_{Yt}(p_i)\}\otimes B_{XY}]\cup[\pi_{sX}(P_{X\backslash Y})\otimes\pi_{Yt}(P_{Y\backslash X})\otimes B_{XY}]\cup[\{\cup_j\{\pi_{sY}(p_j)\cup\pi_{Xt}(p_j)\}\otimes B_{YX}]\cup[\pi_{sY}(P_{Y\backslash X})\otimes\pi_{Xt}(P_{X\backslash Y})\otimes B_{YX}]$, where $\forall$ $p_i\in P_{x\to y}$ with $\pi_{sx}(p_i)\notin P_{x\backslash y}$ or $\pi_{yt}(p_i)\notin P_{y\backslash x}$, $\forall$ $p_j\in P_{y\to x}$ with $\pi_{sy}(p_j)\notin P_{y\backslash x}$ or $\pi_{yt}(p_j)\notin P_{x\backslash y}$.

Proof: Follows directly from **Theorem 11**.

4. THE PROPOSED ALGORITHM

To compare our algorithm to the best-known method presented in [24], the calculation procedure of the best-known method is listed first by transforming it into our notations:

Algorithm 0:

STEP 1 : Separate all MPs that pass node x and/or node y into $P_{x\backslash y}$, $P_{y\backslash x}$, $P_{x\to y}$, or $P_{y\to x}$.

STEP 2 : New set of MPs generated from $P_{x\to y}$, $P_{x\backslash y}$, and $P_{y\backslash x}$ are $\pi_{sx}(P_{x\to y}\cup P_{x\backslash y})\otimes_{xy}\pi_{yt}(P_{x\to y}\cup P_{y\backslash x})$.

STEP 3 : New set of MPs generated from $P_{x\to y}$, $P_{x\backslash y}$, and $P_{y\backslash x}$ are $\pi_{sy}(P_{y\to x}\cup P_{y\backslash x})\otimes_{yx}\pi_{xt}(P_{y\to x}\cup P_{x\backslash y})$.

STEP 4 : All MPs in $G(V, E)$ and New MPs that generated from STEPs 2 and 3 are all of the MPs in $G(V, E)\cup B_{xy}$.

In the above procedure, this algorithm fails to find new MPs, while there are no MPs from node s to node t passing through node x or node y (see the example in Section 3).

Next, our algorithm (Algorithm 1) that based on Theorem 9 will be proposed to enumerate of all MPs after inserting a branch string to a network.

Algorithm 1:

STEP 1 : If there is a path from node s to node t through node x, then $X\leftarrow x$ and go to STEP 3. Otherwise, find a shortest path from node s to node x (if it does not exist, then find a shortest path from node x to node t instead). Let node X be the node nearest to node x in this shortest path such that there is a path from node s to node t through node X.

STEP 2 : If there is a path from node s to node t through node y, then $Y\leftarrow y$ and go to STEP 3. Otherwise, find a shortest path from node s to node y (if it does not exist, then find a shortest path from node y to node t instead). Let node Y be the node nearest to node y in this shortest path such that there is a path from node s to node t through node Y.

STEP 3 : Find $P_{X\backslash Y}$, $P_{Y\backslash X}$, $P_{X\to Y}$ and $P_{Y\to X}$ using the definition in Section 2.

STEP 4 : Find $[\cup_i\{\pi_{sX}(p_i)\cup\pi_{Yt}(p_i)\}\otimes B_{XY}]\cup[\{\pi_{sX}(p_i)\otimes\pi_{Yt}(p_i)\}\otimes B_{XY}]$, where for all $p_i\in P_{X\to Y}$, $\pi_{sX}(p_i)\notin P_{X\backslash Y}$ or $\pi_{Yt}(p_i)\notin P_{Y\backslash X}$, and $B_{XY}=B_{xy}$ if $X=x$ and $Y=y$, otherwise $B_{XY}=\{p$: $p^*\cup B_{xy}$, where for all path p^* from node X to node Y.

STEP 5 : Find $[\cup_i\{\pi_{sX}(p_i)\cup\pi_{Yt}(p_i)\}\otimes B_{YX}]\cup[\{\pi_{sY}(p_j)\otimes\pi_{Xt}(p_j)\}\otimes B_{YX}]$, where for all $p_i\in P_{Y\to X}$, $\pi_{sY}(p_j)\notin P_{Y\backslash X}$ or $\pi_{Xt}(p_j)\notin P_{X\backslash Y}$, and $B_{YX}=B_{yx}$ if $X=x$ and $Y=y$, otherwise $B_{YX}=\{p$: $p^*\cup B_{xy}$, where for all path p^* from node Y to node X.

STEP 6 : All MPs in $G(V, E)$ and new MPs that generated from STEPs 4 and 5 are all of the MPs in $G(V, E)\cup B_{xy}$.

Theorem 10 : The above algorithm locates all of the MPs in a modified network, if the MPs are all known in the corresponding original network in advance.

Proof: Follows directly from Theorem 9.

Theorem 11 : Algorithm 1 is more efficient and effective than Algorithm 0.

Proof: Since Algorithm 0 works only under $(P_{x\backslash y}\cup P_{x\to y})\neq\varnothing$ and $(P_{x\to y}\cup P_{y\backslash x})\neq\varnothing$,. There are no limitations for Algorithm 1. Algorithm 1 is more effective than Algorithm 0.

Under the condition $(P_{x\backslash y}\cup P_{x\to y})\neq\varnothing$ and $(P_{x\to y}\cup P_{y\backslash x})\neq\varnothing$, the major time complexity in finding new MPs for Algorithms 0 and 1 are from $\pi_{sx}(P_{x\to y}\cup P_{x\backslash y})\otimes_{xy}\pi_{yt}(P_{x\to y}\cup P_{y\backslash x})$ and $\pi_{sy}(P_{y\to x}\cup P_{y\backslash x})\otimes_{yx}\pi_{xt}(P_{y\to x}\cup P_{x\backslash y})$, and $\pi_{sx}(P_{x\backslash y})\otimes_{xy}\pi_{yt}(P_{y\backslash x})$ and $\pi_{sy}(P_{y\backslash x})\otimes_{xy}\pi_{xt}(P_{x\backslash y})$, respectively. Therefore, it is trivial that the major time complexity of $\pi_{sx}(P_{x\backslash y})\otimes_{xy}\pi_{yt}(P_{y\backslash x})$ and $\pi_{sy}(P_{y\backslash x})\otimes_{xy}\pi_{xt}(P_{x\backslash y})$ is simpler than that of $\pi_{sx}(P_{x\to y}\cup P_{x\backslash y})\otimes_{xy}\pi_{yt}(P_{x\to y}\cup P_{y\backslash x})$ and $\pi_{sy}(P_{y\to x}\cup P_{y\backslash x})\otimes_{yx}\pi_{xt}(P_{y\to x}\cup P_{x\backslash y})$.

5. EXAMPLES

The following example, which is originally from [24], illustrates our algorithm for the MP problem in a modified network under the condition that $(P_{x\backslash y}\cup P_{x\to y})\neq\varnothing$ and $(P_{x\to y}\cup P_{y\backslash x})\neq\varnothing$.

Example 1: Consider the network in Fig. 2 , which was modified by inserting a branch string $\{e_{10}, e_{03}\}$ into the example network in Fig. 1. If nodes s and t are the source and sink nodes, respectively, then all of the MPs in Fig. 1 are as follows:

$p_1=\{e_{s1}, e_{14}, e_{4t}\}$, $p_2=\{e_{s1}, e_{14}, e_{34}, e_{3t}\}$
$p_3=\{e_{s1}, e_{12}, e_{23}, e_{3t}\}$, $p_4=\{e_{s1}, e_{12}, e_{23}, e_{34}, e_{4t}\}$,

$p_5=\{e_{s2}, e_{23}, e_{3t}\}$, $p_6=\{e_{s2}, e_{23}, e_{34}, e_{4t}\}$,
$p_7=\{e_{s2}, e_{21}, e_{14}, e_{4t}\}$, $p_8=\{e_{s2}, e_{21}, e_{14}, e_{34}, e_{3t}\}$.

Use Algorithm 1 to find all of the MPs in the modified network shown in Fig. 2.

Solution : According to the definition, classify all MPs into the following groups: $P_{1\backslash 3}=\{p_1, p_7\}$, $P_{3\backslash 1}=\{p_5, p_6\}$, $P_{1\to 3}=\{p_2, p_3, p_4, p_8\}$ and $P_{3\to 1}=\varnothing$.

Since there is a path from node s to node t through node x and node y, respectively, and $\pi_{s1}(P_{1\backslash 3})=\{\{e_{s1}\}, \{e_{s2}, e_{21}\}\}$, $\pi_{1t}(P_{1\backslash 3})=\{\{e_{14}, e_{4t}\}\}$, $\pi_{s3}(P_{3\backslash 1})=\{\{e_{s2}, e_{23}\}\}$, $\pi_{3t}(P_{3\backslash 1})=\{\{e_{3t}\}, \{e_{34}, e_{3t}\}\}$, $\pi_{s1}(P_{1\to 3})=\{\{e_{s1}\}, \{e_{s2}, e_{21}\}\}$. $\pi_{3t}(P_{1\to 3})=\{\{e_{3t}\}\ \{e_{34}, e_{3t}\}\}$, and $\pi_{s3}(P_{3\to 1})=\pi_{1t}(P_{3\to 1})=\varnothing$. We have $\pi_{s1}(P_{1\backslash 3})\otimes_{13}\pi_{3t}(P_{3\backslash 1})=\{\{e_{s1}, e_{3t}, e_{13}\}, \{e_{s1}, e_{34}, e_{3t}, e_{13}\}, \{e_{s2}, e_{21}, e_{3t}, e_{13}\}, \{e_{s2}, e_{21}, e_{34}, e_{3t}, e_{13}\}\}$ and $\pi_{s3}(P_{3\backslash 1})\otimes_{31}\pi_{1t}(P_{1\backslash 3})=\{\{e_{s2}, e_{23}, e_{14}, e_{4t}, e_{31}\}\}$.

Besides, $[\cup_i\{\pi_{s1}(p_i)\cup\pi_{3t}(p_i)\cup B_{ab}\}]=[\cup_j\{\pi_{s3}(p_j)\cup\pi_{1t}(p_j)\cup B_{ab}\}]=\varnothing$, where for all $p_i\in P_{1\to 3}$, $\pi_{s1}(p_i)\notin P_{1\backslash 3}$, $\pi_{s1}(p_i)\notin P_{3\backslash 1}$ and $p_j\in P_{3\to 1}$. Therefore, the set of all MPs in Fig. 2 is $\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, \{e_{s1}, e_{3t}, e_{13}\}, \{e_{s1}, e_{34}, e_{3t}, e_{13}\}, \{e_{s2}, e_{21}, e_{3t}, e_{13}\}, \{e_{s2}, e_{21}, e_{34}, e_{3t}, e_{13}\}, \{e_{s2}, e_{23}, e_{14}, e_{4t}, e_{31}\}\}$.

In the next example, we will show how Algorithm 1 works if $P_{x\backslash y}\cup P_{x\to y}=\varnothing$ and/or $P_{x\to y}\cup P_{y\backslash x}=\varnothing$.

Example 2: Consider the example network in Fig. 3, if nodes s and t are the source and sink nodes, respectively, and all of the MPs in Fig. 3 are as follows: $p_1=\{e_{s2}, e_{23}, e_{3t}\}$, $p_2=\{e_{s2}, e_{23}, e_{34}, e_{4t}\}$. Use Algorithm 1 to find all MPs in the modified network, which was modified by inserting a branch string $\{e_{13}\}$.

Solution :

STEP 1 : Since there is no path from node s to node t through node 1, and the nearest node to node 1 in the shortest path from node s to node 1 $\{e_{s2}, e_{23}, e_{34}, e_{41}\}$ is node 4.

STEP 2 : Since there is a path from node s to node t through node 3, let $Y\leftarrow 3$.

STEP 3 : By the definition in Section 2, we have $P_{Y\backslash X}=\{p_1\}$, $P_{Y\to X}=\{p_2\}$, $P_{X\backslash Y}=P_{X\to Y}=\varnothing$, $P_{Y\backslash X}=\{p_1\}$, $P_{Y\to X}=\{p_2\}$, and $P_{X\backslash Y}=P_{X\to Y}=\varnothing$. $P_{1X}=\{\{e_{1X}\}, \{e_{15}, e_{5X}\}\}$

STEP 4 : Since $\pi_{s4}(P_{4\backslash 3})=\pi_{4t}(P_{4\backslash 3})=\pi_{s4}(P_{4\to 3})=\pi_{3t}(P_{4\to 3})=\varnothing$, $\pi_{s3}(P_{3\backslash 4})=\{\{e_{s2}, e_{23}\}\}$, $\pi_{3t}(P_{3\backslash 4})=\{\{e_{3t}\}\}$, $\pi_{s3}(P_{3\to 4})=\{\{e_{s2}, e_{23}\}\}$, and $\pi_{4t}(P_{3\to 4})=\{\{e_{4t}\}\}$. Thus, $\pi_{s4}(P_{4\backslash 3})\otimes\pi_{3t}(P_{3\backslash 4})=\varnothing$, and $\{[\pi_{sa}(p_1)\cup\pi_{bt}(p_1), \pi_{sa}(p_2)\cup\pi_{bt}(p_2)]\otimes B_{XY}\}=\{\{e_{s2}, e_{23}, e_{4t}\}\}\otimes\{\{e_{14}, e_{31}\}, \{e_{15}, e_{54}, e_{31}\}\}=\{\{e_{s2}, e_{23}, e_{31}, e_{14}, e_{4t}\}, \{e_{s2}, e_{23}, e_{31}, e_{15}, e_{54}, e_{4t}\}\}$, where $\pi_{sX}(p_i)\notin P_{X\backslash Y}$, $\pi_{Yt}(p_i)\notin P_{Y\backslash X}$, and $B_{XY}=\{\{e_{15}, e_{54}, e_{31}\}\}$.

STEP 5 : We have $[\cup_i\{\pi_{sX}(p_i)\cup\pi_{Yt}(p_i)\}\otimes B_{YX}]\cup[\{\pi_{sY}(p_j)\otimes\pi_{Xt}(p_j)\}\otimes B_{YX}]=\varnothing$, where for all $p_i\in P_{Y\to X}$, $\pi_{sY}(p_j)\notin P_{Y\backslash X}$ and $\pi_{Xt}(p_j)\notin P_{X\backslash Y}$, and $B_{YX}=\{\{e_{15}, e_{54}, e_{31}\}\}$.

STEP 6 : The set of all MPs after inserting branch string $\{e_{13}\}$ in Fig. 3 is $p_1=\{e_{s2}, e_{23}, e_{3t}\}$, $p_2=\{e_{s2}, e_{23}, e_{34}, e_{4t}\}$, $\{e_{s2}, e_{23}, e_{31}, e_{14}, e_{4t}\}$ and $\{e_{s2}, e_{23}, e_{31}, e_{15}, e_{54}, e_{4t}\}$.

6. CONCLUSIONS

Network reliability theory has been applied extensively in many real-world systems. Thus, system reliability plays important roles in our modern society [8]. The reliability evaluation of a network is *NP*-hard but practical [4,5,9-19]. Most of these methods are formulated in terms of either MCs or MPs [9-14]. However, to search for all MCs or MPs is also a *NP*-hard problem [9-14]. Moreover, it is a very cumbersome and time-consuming task if all of the MPs in the modified network must be sought .

The existing approaches for the enumeration of MPs in modified networks are practical but tedious. Basically, these systems locate all new MPs in a straightforward enumeration procedure that works only under some special cases. Therefore, the main purpose of this article is to present a more efficient and effective algorithm to solve such a problem.

By exploiting the MP structure in the original network, the proposed algorithm not only requires fewer calculations to generate new MPs in the modified network, but is also more effective in every situation in the modified network. Hence, our method is very effective and efficient when compared to the existing methods.

7. ACKNOWLEDGEMENT

This research was supported in part by the National Science Council of Taiwan, R.O.C. under grant NSC-88-2213-E-035-017.

REFERENCE

1. K.K. Aggarwal et al., "A simple method for reliability evaluation of a communication system", *IEEE Trans. Commun.* COM-23, 1975, pp 563-565.
2. P. Kubat, "Estimation of reliability for communication/computer networks simulation/ analytical approach", *IEEE Trans. Communications*, 37, 1989, pp.927–933.
3. M. A. Samad, "An efficient algorithm for simultaneously deducing MPs as well as cuts of a communication network", *Microelectron. Reliab.*, vol 27, 1987, pp 437-441.
4. C.C. Jane, J.S. Lin, and J. Yuan, "Reliability evaluation of a limited-flow network in terms of MC sets", *IEEE Trans. Reliability*, vol R-42, 1993, pp 354-361.
5. W.C. Yeh, "A Revised Layered-Network Algorithm to Search for All d-Minpaths of a Limited-Flow Acyclic Network", *IEEE Trans. Reliability*, vol 46, 1998, pp 436-442.
6. Doulliez, E. Jalnoulle, "Transportation network with random arc capacities", *RAIRO, Rech. Operations Research*, vol 3, 1972, pp 45-60.

7. T. Aven, "Availability evaluation of oil/gas production and transportation systems", *Reliability Engineering*. vol 18, 1987, pp 35-44.
8. T. Aven, "Some considerations on reliability theory and its applications", *Reliability Engineering and System Safety*, vol 21, 1988, pp 215-223.
9. M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-hardness*, 1979; E.H. Freeman.
10. L.G. Valiant, "The complexity of enumeration and reliability problems", *SIAM J. Computing*, vol 8(3), 1979, pp 410-421.
11. J.S. Provan, M.O. Ball, "The complexity of counting cuts and of computing the probability that a graph is connected", *SIAM J. Computing*, vol 12(4), 1983, pp 777-788.
12. A. Aggarwal and R.E. Barlow, "A survey of network reliability and domination theory", *Operations Research*, **22**, 1984, pp.478–492.
13. M.O. Ball, "Computational complexity of network reliability analysis: an overview", *IEEE. Trans. Reliability*, **35**, 1986, pp.230–239.
14. S.H. Lee, "Reliability evaluation of a flow network", *IEEE Trans. Reliability*, vol R-29, 1980, pp 21-26.
15. J. Xue, "On multistate system analysis", *IEEE Trans. Reliability*, vol R-34, 1985, pp 329-337.
16. T. Aven, "Reliability evaluation of multistate systems with multistate components", *IEEE Trans. Reliability*, vol R-34, 1985, pp 473-479.
17. C.C. Jane, J.S. Lin, J. Yuan, "Reliability evaluation of a limited-flow network in terms of minimal cutsets", *IEEE Trans. Reliability*, vol R-42, 1993, pp 354-361.
18. J.C. Hudson, K.C. Kapur, "Reliability bounds for multistate systems with multistate components", *Operations Research*, vol 33, 1985, pp 153-160.
19. R.E. Barlow, A.S. Wu, "Coherent systems with multi-state components", *Math. Operations Research*, vol 3, 1978, pp 275-281.
20. R. N. Allan, I. L. Rondiris, and D. M. Fryer, "An efficient computational technique for evaluating the cut/tie and common cause failures of complex systems", *IEEE Trans. Reliability*, vol R-30, 1981, pp 101-109.
21. J. M. Nahman, "MPs and cuts or networks exposed to common-cause failures", *IEEE Trans. Reliability*, R-41, 1982, pp 76-80, 84.
22. R. N. Allan, R. Billinton, and M. F. De Oliveira, "An efficient algorithm for deducing the MCs and reliability indices of a general network configuration", *IEEE Trans. Reliability*, vol R-25, 1976, pp 226-233.
23. J. M. Nahman, "Enumeration of minimal cuts of modified networks", Microelectron. Reliab., vol 37, 1997, pp 483-485.
24. J. M. Nahman, "Enumeration of MPs of modified networks", *Microelectron. Reliab.*, vol 34, 1994, pp 475-484.
25. R.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Netwrok Flows—Theory, Algorithms, and Applications*, 1993; Prentice-Hall International.