International Journal of Computer Applications (0975 – 8887)
Volume 45– No.16, May 2012

Cutset Enumerating and Network Reliability Computing by a new Recursive Algorithm and Inclusion Exclusion Principle

Mohamed Benaddy, Mohamed Wakrim
Department of Mathematics and Computer Sciences
Mathematical Modelling and Simulation Laboratory (EMMS),
Faculty of Sciences Ibn Zohr University,Agadir, Morocco

ABSTRACT
In this work we present a new and efficient recursive algorithm that enumerate all the *s-t* minimal cut sets (MCs) separating nodes *s* (source) and *t* (terminal) in a network system. The networks studied here are considered as the undirected graphs. Later enumerating all the MCs, the inclusion-exclusion principle is used to compute the network reliability based on the probabilities of the links.

General Terms
Networked System Reliability, Algorithms, Optimisation and Modelling.

Keywords
Network Reliability, Minimal Cut Set, Undirected Graph, Inclusion Exclusion Principle, Algorithm.

1. INTRODUCTION
In recent years, network reliability theory has been applied in many real world systems such as computer and communication systems, power transmission and distribution systems. System reliability plays important role in our modern society [9]. Many studies has been realized to evaluate the network reliability [1,2,4,5,7], most of them use network-based algorithms founded in terms of minimal cut sets (MCs) or minimal path sets (MPs). Unfortunately, to search for all MCs or MPs is also an NP-hard problem but enumerating all the minimal cut sets may be a feasible to evaluate the reliability of the network if the number of paths is too huge to enumerate. One example of this kind is the 2x100 lattice network which has 299 paths and 10000 cuts [1,5] it's impractical to enumerate all these paths, but if the number of minimal cut sets is far less, then it may be a feasible way to evaluate the reliability through cut sets. Another approach is to obtain the minimal cut sets by inverting minimal path sets [6], but it may impractical to generate the path sets of a graph. Tsukiyama [8] presented algorithms and mathematical background to enumerate all the minimal cut sets of a graph. Based on some definitions founded in this interesting research, we propose an efficient algorithm to enumerate all the minimal cut sets in a network without duplication. To prove that we have implemented the procedure called BACKTRACK#1 in [8]. In this work we present the developed algorithm to enumerate all the minimal cut sets based on the link failure in a network which is considered as an undirected graph *G=[V,E]* with set of nodes *V* and set of links *E*.

This paper is organized as follows. Section 2 presents some preliminaries, notations and assumptions required in the rest of the paper. In section 3 we present the developed algorithm. The Section 4 is devoted to the method used to calculate the network reliability, using the minimal cut sets identified by our proposed algorithm. The last section presents the obtained results and discussion.

2. PRELIMINARIES, NOTATION AND ASSUMPTIONS
G=(V,E) : A network with set of nodes *V* and set of links *E.*

s : A source node in the network *G.*

t : A target node in the network *G.*

$e_{v,w}$: link connecting source node *v* to target node *w*

S : A set of nodes that contains *s.*

T : A set nodes that contains *t.*

C_i : A minimal cut set number *i*.

C_v : A minimal cut set *associated with the node v*.

MC : Minimal cut set.

MCs : Minimal cut sets.

MP : Minimal path.

MPs : Minimal path set.

Reliability: the probability that a signal can be transmitted from the source node (*s*) to the target node (*t*).

2.1 Preliminaries
The networks studied here are *s-t* networks with two special terminals: one source node called *s* and one target node called *t*. The success or failure of the networks depends on whether or not there is a connection between the source node and destination node. Meanwhile, as links or nodes are failure free, the networks can be classified into three categories:

1. Networks with node failures only : the obtained minimal cut sets are node minimal cut sets,
2. Networks with link failures only : the minimal cut sets are link minimal cut sets,
3. Networks with both node and link failures: in this case both node and link failures are considered.

In this paper, only we consider networks in the second category, where each link is represented by an unordered pair $<v,w>$ of end nodes. For an link $<v,w>$, *v* and *w* are said to be adjacent to each other, and the link is said to be incident with *v* and *w*.

A path between v_0 and v_n in *G* is a sequence of distinct links $<v_0 , v_1>,< v_1 , v_2>,...,<v_{n-1} , v_n>$, and *n* is the length of the path.

A graph is connected if there is a path between each pair of nodes in a graph. Henceforth we assume that all graphs considered here are connected and contain neither multiple links nor self-loop (i.e., links of the form $<v, v>$). A cut set of a graph is a set of links which interrupts all connections between source node and target node when removed from the graph. The minimal cut sets are a group of distinct cut sets containing a minimum number of terms (e.g. a minimal cut set is a minimal set of components whose failure ensures the failure of the system [8]). All system failures can be represented by the removal of at least one minimal cut set from the graph [7].

When two distinct nodes *s* and *t* are specified in *G = (V, E)*, let two disjoint sets X and Y $(X, Y \subset V)$ such that $s \in X$ and $t \in Y$, and let

$$\boldsymbol{\omega(X,Y)} = \{\langle \boldsymbol{v}, \boldsymbol{w}\rangle : \boldsymbol{v} \in \mathbf{X}, \boldsymbol{w} \in \boldsymbol{Y}\} \qquad (1)$$

be designated as an *s-t* cut *C*. If an *s-t* cut does not contain any other *s-t* cut, then it is referred to as a minimal *s-t* cut set. And for any node *v* and *A* $(A \subset V)$ we introduce the following sets:

$$\boldsymbol{\Gamma(v)} = \{\boldsymbol{w} \in \boldsymbol{V} : \langle \boldsymbol{v}, \boldsymbol{w}\rangle \in \boldsymbol{E}\} \qquad (2)$$

$$\boldsymbol{\Gamma(A)} = \cup_{\boldsymbol{v}\in \boldsymbol{A}} \boldsymbol{\Gamma(v)} \qquad (\mathbf{3})$$

$$\boldsymbol{\Gamma^+(v)} = \boldsymbol{\Gamma(v)} - \{\boldsymbol{v}\} \qquad (\mathbf{4})$$

$$\boldsymbol{\Gamma^+(A)} = \boldsymbol{\Gamma(A)} - \boldsymbol{A} \qquad (\mathbf{5})$$

Definition 1: for a given graph *G=(V,E)*, a subset of links $C \subset E$ is a minimal cutset if and only if deleting all links in *C* will divide *G* into two connected components. An isolated node is considered as a component. A minimal cutset divides the nodes of *G* into two disjoint subsets *X* and *Y*, each of which induce a connected subgraph.

Definition 2 : A cut set C_s is a minimal cut set that separate a graph into two disjoint subset *X* and *Y*, with *X={s}* and *Y=V-X*

Observation: From the above preliminaries, each *MC* of a given graph can be deduced from another *MC*, except C_s.

2.2 Assumptions

The network satisfies the following assumptions:

1. Each node is perfectly reliable.
2. The graph is connected.
3. Each link has two states: working or failed.
4. The states of links are statistically independent.

3. THE ALGORITHM

Using the cited assumptions and definitions we develop an algorithm "*MCsAlgorithm*" (see figure 1) to enumerate all the minimal cut sets in a network. Thus it can be readily verified that algorithm "*MCsAlgorithm*" enumerates all the s-t cut sets without duplication.

Procedure_all_cuts(G, V, E, s, t)
S={s}
$C_s = \{\langle v, w\rangle \in E : v \in S, w \in \Gamma^+(S)\}$
$MCsAlgorithm(S, C_s)$
  $\forall v \in \Gamma^+(S) \setminus \{t\} \neq \emptyset$
  If$(G = (\bar{S} \setminus \{v\}, E))$ *is connected*
  begin
    $C_v = C_s + \{\langle v, w\rangle \in E : w \in \Gamma^+(S)\}$
      $+\{\langle v, w\rangle \in E : w \in \Gamma^+(v)\}$
      $\setminus \{\langle w, v\rangle \in E : w \in S\}$
    output C_v
    $S = S \cup \Gamma^+(v)$
    $MCsAlgorithm(S, C_v)$
  endif
end_MCsAlgorithm
end_Procedure_all_cuts

Figure 1: Algorithm for enumerating all minimal cut sets in a graph

The proposed algorithm needs a set of parameters; the network (graph *G*) with a set of nodes *V* and a set of links *E*, the source node (*s*) and the terminal node (*t*). The first step is to compute the cut set that separate the singleton source node to other nodes in the network using the equations from (1) to (5), the second step is to determine the next cut from the first one recursively by giving the last cut and its associated nodes to the *"MCsAlgorithm"*. The graph $G = (\bar{S} \setminus \{v\}, E)$ is the subgraph with a set of nodes $(V \setminus S) \setminus \{v\}$.

4. RELIABILITY EVALUATION

The above algorithm enumerates all the MCs in the given network. If all MCs are listed, the last step is to evaluate the exact reliability in terms of these MCs, using the inclusion-exclusion principle or the sum of disjoints products method.

The inclusion-exclusion principle is used here to compute the network reliability, in terms of MCs according to the following theorem,

Theorem: Assume $C_1, C_2, \ldots, C_n$ are the MCs between nodes *s* and *t*, then the reliability R_{st} using the inclusion-exclusion principle in terms of these MCs is given by.

$$R_{st} = 1 - P(C_1 \cup C_2 \cup \ldots \cup C_n)$$

$$R_{st} = 1 - \sum_{M\subset[\![1,n]\!], M\neq\emptyset} (-1)^{-1+|M|} P(\cap_{i\in M} C_i)$$

5. EXPERIMENTAL RESULTS

Our algorithm has been implemented on a Linux Ubuntu operating system with Java language programming. We have used a JGraphT[3] library, which is a free Java class library that provides mathematical graph-theory objects and algorithms. In the evaluation we used 6 benchmark networks collected in [5,10] as shown in figure 2. The success probabilities of the links for networks from 1 to 5 are given in figure 3, for the network 6 the success probabilities are fixed to 0.9.

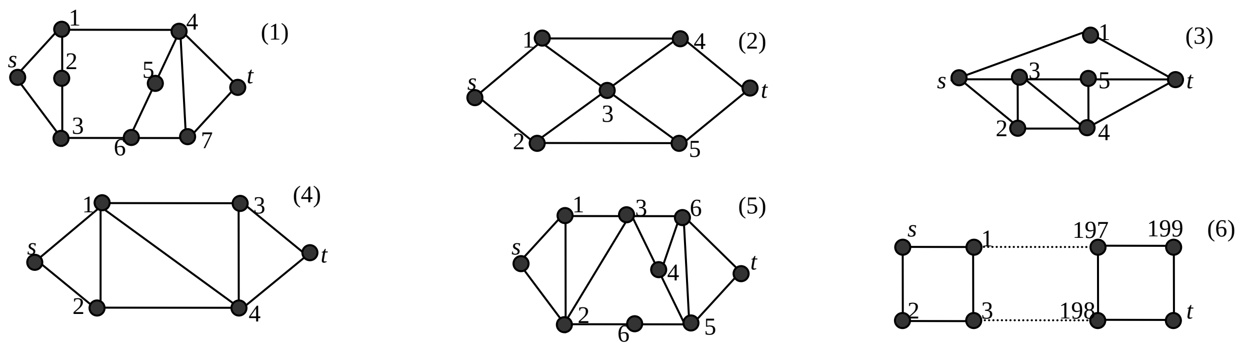

Figure 2: Benchmark networks

		1	2	3	4	5	6	7	*t*
(1)	*s*	0,97	-	0,9	-	-	-	-	-
	1		0,87	-	0,98	-	-	-	-
	2			0,86	-	-	-	-	-
	3						0,67	-	-
	4					0,89	-	0,9	0,9
	5						0,78	-	-
	6							0,9	-
	7								0,57
(2)	*s*	0,78	0,98	-	-	-	-	-	-
	1			0,95	0,65	-	-	-	-
	2			0,78	-	0,98	-	-	-
	3				0,69	0,61	-	-	-
	4								0,85
	5								0,9
(3)	*s*	0,9	0,87	0,56	-	-	-	-	-
	1					0,76	-	-	0,97
	2			0,87	0,59	-	-	-	-
	3					0,98	-	-	-
	4					0,79	-	-	0,55
	5								0,69
(4)	*s*	0,6	0,7	-	-	-	-	-	-
	1		0,9	0,8	0,75	-	-	-	-
	2				0,85	-	-	-	-
	3				0,95	-	-	-	0,7
	4								0,55
(5)	*s*	0,87	0,9	-	-	-	-	-	-
	1		0,88	0,68	-	-	-	-	-
	2			0,9	-	0,94	-	-	-
	3				0,87	-	0,98	-	-
	4					0,59	0,64	-	-
	5								0,88
	6								0,9

Figure 3: The success probabilities of the networks

Table 1 gives the number of cut sets and the reliability obtained for the benchmark networks.

Table 1: Obtained Number of MCs and Reliabilities

Net	#of MCs	R_{st}	Execution Time (second)	
			MCsAlgorithm	RM_BDD[5]
1	26	0.943995	0	0
2	16	0.966967	0	0
3	19	0.983068	0	0
4	9	0.749283	0	0
5	19	0.965378	0	0
6	10000	0.304317	12,5	96,71

The network 6 has 2^{99} [5] paths, it contains only 10000 MCs, so, it is impractical to enumerate all MPs since there is a huge number of paths. To verify that our proposed algorithm enumerate all the MCs without duplication, we have implemented the algorithm BACKTRACK#1 [8], both produce the same results.

6. CONCLUSION

In this paper we have proposed a new and efficient algorithm to enumerate all the s-t minimal cut sets of a network to use in computing the concerned network reliability. This algorithm was developed using the idea that any cut set in the network can be deduced from an other, except the first one that connect the source node and its adjacent nodes in the network. The developed algorithm has been implemented with Java programming language using the JGraphT library. The Inclusion-exclusion principle was applied to evaluate the network reliability, based on the listed cut sets. The execution time for scanning 10000 is only 12.5 seconds compared with other algorithm that needs more time to enumerate the same cuts in the same network.

7. REFERENCES

[1] Chang, Y.-R., Lin, H.-Y., Chen, I.-Y. & Kuo, S.-Y., A Cut-Based Algorithm for Reliability Analysis of Terminal-Pair Network Using OBDD, Proceedings of the 27th Annual International Conference on Computer Software and Applications, 2003.

[2] Fard, N. S. & Lee, T.-H., Cutset enumeration of network systems with link and node failures, Reliability Engineering and System Safety 1999, pp. 141 - 146.

[3] JGraphT, http://jgrapht.org/

[4] Khachiyan, L., Boros, E., Elbassioni, K., Gurvich, V. & Makino, K., Enumerating disjunctions and conjunctions of paths and cuts in reliability theory, Discrete Appl. Math 2007, pp. 137-149.

[5] Lin, H.-Y., Kuo, S.-Y. & Yeh, F.-M., Minimal Cutset Enumeration and Network Reliability Evaluation by Recursive Merge and BDD, in ISCC '03: Proceedings of the 8th IEEE international Symposium on Computers and Communications, 2003, pp. 1341-1346.

[6] Shier, D. R. & Whited, D. E., Algorithms for Generating Minimal Cutsets by Inversion, Reliability, IEEE Transactions vol R-34, 1985, pp. 314 -319.

[7] Tan, Z., Minimal cut sets of s–t networks with k-out-of-n nodes, Reliability Engineering & System Safety 2003, pp. 49 - 54.

[8] Tsukiyama, S., Shirakawa, I., Ozaki, H. & Ariyoshi, H., An Algorithm to Enumerate All Cutsets of a Graph in Linear Time per Cutset, J. ACM 27 1980, pp. 619-632.

[9] Yeh, W.-C., A simple algorithm for evaluating the k-out-of-n network reliability, Reliability Engineering and System Safety 2004, pp. 93 - 101.

[10] Yeh, W.-C., A new algorithm for generating minimal cut sets in k-out-of-n networks, Reliability Engineering and System Safety 2006, pp. 36 - 43.