

# Algorithms for Reliability Calculation of Complex Bridge System

S.C. Lee, H.Y. Lee, E.S. Lee, H.K. Kwon

Dept. of industrial & Management Engineering, Dept. of Health & Exercise Science, Dept. of Tourism Management  
, Namseoul University  
Chonan, South Korea

**Abstract**—In this paper, we present an algebraic technique for computing the system reliability of a complex system. And we also studied the problem of inverting minimal path sets to obtain minimal cut sets of the complex system. We described efficiency of inversion algorithm by the use of Boolean algebra and we developed an inclusion-exclusion algorithm and a pivotal decomposition algorithm for the reliability calculation of the complex system. We verified that inclusion-exclusion algorithm is more useful than pivotal decomposition algorithm in series structure. Otherwise, pivotal decomposition algorithm is more useful than inclusion-exclusion algorithm in parallel structure. Several examples are illustrated and the computation speeds between the two algorithms are undertaken.

**Keywords**—system reliability; boolean algebra; inversion algorithm; inclusion-exclusion algorithm; pivotal decomposition algorithm

## I. INTRODUCTION

The reliability literature of the past 10 years contains many papers with the reliability calculation of coherent structure. This paper extended and discussed some algorithms for the reliability calculation. Problems related to the coherent structure of the system are based on [1]. The path set and cut set method for determining the system reliability is used. In section 2, we faintly investigate an inversion algorithm by Heidtmann into the case of inverting paths and cuts of 2-state systems [2]. The method in section 2 was based on symbolic manipulation of Boolean functions by applying two De Morgan's laws [3, 4]. To invert paths from cuts and vice versa, Boolean structure function must be derived, inverted, and reduced. Then the sets can be deduced. We also propose an inclusion-exclusion algorithm and a pivotal decomposition algorithm by the use of an inclusion-exclusion formula and a decomposition rule. These algorithms are easy to program and expedient with the automated computation and their results had been described and verified on the last section.

### Notation

$N$ : set of integers from 1 to  $n$

$I$ : Subset of  $N$

$2^N$ : Power set of  $N$

$N\{A\}$ : Number of element of  $A$

$A'$ : Inverse of  $A$

$A^*$ : Star function of  $A$

$x_i$ : Boolean variable of  $i$ -th component

$\bar{x}_i$ : Complement Boolean variable of  $i$ -th component

$S$ : Success expression of the system

$R$ : Reliability expression of the system

$\psi(x)$ : System structure function

$R(x)$ : System reliability function

## II. INVERSION ALGORITHM AND ALGEBRAIC TECHNIQUE

K. D. Heidtmann suggested an inversion algorithm in his paper [2]. Now we will try to describe his algorithm in simpler terms. Each component of a system of  $n$  component is uniquely represented by its index  $i \in N$ , and any assembly of components by a subset  $I$  of  $N$ . So any path or cut is a subset of  $N$ , and set of all paths, or the set of all cuts, is a subset of the power set  $2^N$ . The concept of inverse combines two complements to invert paths and cuts as subsets  $A$  and  $A'$  of  $2^N$ ; that is, if  $A$  is set of paths, then  $A'$  is set of cuts; and vice versa.

Let  $A \subset 2^N$ ,  $K \subset N$ . From the complement  $\bar{I}$  of every  $I \in A$ . Eliminate from  $2^N$  all such  $\bar{I}$ . The result is  $A'$  which is the inverse of  $A$ . More formally, let  $A \subset 2^N$ ,  $A' \subset 2^N$ , and  $K \subset N$ . Then  $A'$  is the inverse of  $A$  if and only if for every possible  $I$ , either  $I \in A$  or  $\bar{I} \in A'$ . The inverse property is reciprocal:  $(A')' = A$ .  $n\{A\} + n\{A'\} = 2^n$ . There exists one and only one inverse of  $A$ .

Let  $A \subset 2^N$ ,  $K \subset N$ . From the complement  $\bar{I}$  of every for  $I \in A$ ,  $A^*$  is the set of all those  $\bar{I}$ .  $A$  and  $(A')^*$  are a partition of  $2^N$ .  $A' = (A^c)^* = (A^c)^*$ . These inverse and star relationships were derived and proved in Lee(1996) [5].

In applying the inverse concept by hand calculation, many sets must be considered because  $2^N$  contains  $2^n$  elements. Thus it is helpful to verify the correctness of a computed

$A'$  by the following test which is based on the facts that  $A$  and  $(A')^*$  form a partition of  $2^N$ , and  $n[2^N] = 2^*$ .

### III. INVERSION ALGORITHM

Input:  $A, N$

Step1. Compute  $A^*$ . The  $I$  is the elements of  $2^N$ .

Step2. Set  $I \leftarrow \emptyset$  and  $A' \leftarrow \emptyset$ .

Step3. If  $I \notin A^*$  then  $A' \leftarrow A' \cup \{I\}$ .

Step4. If  $I \neq N$  then replace  $I$  by its successor and go to 3.

Step5. Stop ( $A'$  is inverse of  $A$ ).

Now, we present an algebraic technique computing the system reliability. The system success expression  $S$  is given by the union of all the system minimal path sets: that is,

$$S = \bigcup_{i=1}^m P_i = P_1 + \overline{P_1}P_2 + \overline{P_1}\overline{P_2}P_3 + \dots + \overline{P_1}\overline{P_2}\dots\overline{P_{m-1}}P_m \quad (1)$$

Each  $P_i$  in (1) is a product term of the form

$$P_i = \prod_{j=1}^{n_i} x_{ij} \quad i=1, 2, \dots, m \quad (2)$$

When (2) is substituted in (1), and all  $P_i$  are expanded using DeMorgan's Law, the product terms in the resulting sum of product expression lose their disjointness. By above two formulas, we reduce the following algorithm which gives a minimum reliability expression.

#### A. Boolean Algorithm

Step1. Enumerate all the  $m$  paths of the system and arrange these in a sequence  $P_1, P_2, \dots, P_m$ .

Step2. Write  $S$  in a more convenient form given by (1)

$$S = P_1 + \overline{P_1}(P_2 + \overline{P_2}(P_3 + \overline{P_3}(\dots(P_{m-1} + \overline{P_{m-1}}(P_m)))))) \quad (3)$$

Step3. Substitute expression of  $P_m$  in (3); let  $j = m - 1$ .

Step4. In the resulting expression substitute expression of  $P_j$  let  $j = j - 1$ .

Step5. Repeat step 4 if  $j \geq 1$ .

Step6. Replace logical variables by their reliabilities to get the required reliability expression.

#### B. Inclusion-Exclusion Algorithm

The inclusion-exclusion rule came from the additive law of probability. Reliability analysis by the original method of the inclusion-exclusion algorithm assumes the knowledge of all minimal path or cut sets [4, 6].

Let  $E_i(\overline{E_i})$  be the event that  $i$ -th component  $x_i$  is functioning (failed) with probability  $p_i(1 - p_i)$ .

Let  $A_r(\overline{A_r})$  be the event that all components in  $r$ -th minimal path set  $P_r$  is functioning (failed) i.e.

$$A_r = \bigcap_{x_i \in P_r} E_i, \quad \overline{A_r} = \bigcap_{x_i \in K_r} \overline{E_i} \quad (4)$$

Where  $K_r$  is  $r$ -th minimal cut set.

Then

$$P(A_r) = P\left(\bigcap_{x_i \in P_r} E_i\right), \quad P(\overline{A_r}) = P\left(\bigcap_{x_i \in K_r} \overline{E_i}\right) \quad (5)$$

System success corresponds to event  $\bigcup_{r=1}^n A_r$  if the system has  $n$  minimal path sets.

The system reliability function is

$$h(p) = P\left[\bigcup_{r=1}^n A_r\right] \quad (6)$$

Let

$$S_k = \sum P[A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_k}] \quad 1 \leq i_1 < i_2 < \dots < i_k \leq n$$

by the inclusion-exclusion principal [2]

$$h(p) = \sum_{k=1}^n (-1)^{k-1} S_k \quad (7)$$

and

$$\begin{aligned} h(p) &\leq S_1 \\ h(p) &\geq S_1 - S_2 \\ h(p) &\leq S_1 - S_2 + S_3 \end{aligned} \quad (8)$$

and so on.

Now we define the approximation to system reliability function  $h(p)$  of step  $m$  by

$$h^{(m)}(p) = \sum_{k=1}^m (-1)^{k-1} S_k \quad (9)$$

For  $m > 1$ ,

$$h^{(m)}(p) = h^{(m-1)}(p) + (-1)^{m-1} S_m \quad (10)$$

$$h^{(n)}(p) = h(p) \quad (11)$$

Although it is not true in general that the upper bounds decrease and the lower bounds increase, in practice it may be necessary to calculate only a few  $S_k$ 's to obtain a close approximation. Of course, similar formulas for computing system unreliability  $\overline{h}(p)$  in terms of minimal cut sets and component unreliability  $1 - p_i$  can be given. Now we state the algorithm in detail.

1) *Inclusion-exclusion algorithm* :Input:

$n, p_1, p_2, \dots, p_n, N_0$  or  $\bar{p}$

Output:  $h(p)$

Step1. Set  $i = 1$

Step2. While  $i \leq N$  do step 3 ~ 5

$$S_i = \sum_{1 \leq i_1 < \dots < i_n \leq n} P_{i_1} \dots P_{i_n} \quad (\text{compute } S_i)$$

Step3. Set

$$h(p) = \sum_{i=1}^n (-1)^{i-1} S_i$$

Step4. If  $|S_i - S_{i-1}| < \epsilon$

Output:  $h(p)$

Stop:

Step5. Set  $i = i + 1$

$$S_i = S_{i+1}$$

Step6. Output (method failed after  $N_0$  iterations,  $N_0$  or  $h(p)$ )

2) *Pivotal decomposition algorithm*: This section gives an algorithm for the pivotal decomposition rule. The following identity holds for any structure function  $\Phi$  of order  $n$ :

$$\Phi(x) = x_i \cdot \Phi(1_i, x) + (1 - x_i) \cdot \Phi(0_i, x) \dots \dots \dots (12)$$

We immediately obtain the corresponding pivotal decomposition of the reliability function.

$$h(p) = E[\Phi(X)] = p_i \cdot h(1_i, p) + (1 - p_i) \cdot h(0_i, p) \quad i = 1, \dots, n \dots (13)$$

Now we proposed the following algorithm.

3) *Algorithm for Series System*: Input:

$$n, p_1, p_2, \dots, p_n$$

Output:  $h(p)$

Step1.  $i = 1$

Step2. While  $i \leq n$  do step 3 ~ 4

Step3. Set  $h(0_i, p) = 0$

$$h(1_i, p) = p_{i+1} \cdot h(1_{i+1}, p) + (1 - p_{i+1}) \cdot h(0_{i+1}, p)$$

$$h(p) = p_i \cdot h(1_i, p) + (1 - p_i) \cdot h(0_i, p) \quad (\text{compute } h(p))$$

Step4.  $i = i + 1$

Step5. Output  $h(p)$

4) *Algorithm for Parallel System*: Input:

$$n, p_1, p_2, \dots, p_n$$

Output:  $h(p)$

Step1.  $i = 1$

Step2. While  $i \leq n$  do step 3 ~ 4

Step3. Set  $h(1_i, p) = 1$

$$h(0_i, p) = p_{i+1} \cdot h(1_{i+1}, p) + (1 - p_{i+1}) \cdot h(0_{i+1}, p)$$

$$h(p) = p_i \cdot h(1_i, p) + (1 - p_i) \cdot h(0_i, p) \quad (\text{compute } h(p))$$

Step4.  $i = i + 1$

Step5. Output  $h(p)$

#### IV. COMPUTATIONAL RESULTS

This section discusses the results obtained when algorithms inclusion-exclusion and pivotal decomposition were programmed and run on a variety of complex system reliability calculation problems. We verify that inclusion-exclusion algorithm is more useful than pivotal decomposition algorithm in series structure described on table 1. Otherwise, pivotal decomposition algorithm is more useful than inclusion-exclusion algorithm in parallel structure described on table 2.

TABLE I .SERIES STRUCTURE.

	<b>inclusion exclusion</b>	<b>pivotal decomposition</b>
n=2	$p_1 p_2$	$p_1 p_2 + (1 - p_1) 0$
n=3	$p_1 p_2 p_3$	$p_1 \{p_2 p_3 + (1 - p_2) 0\} + (1 - p_1) 0$
n=4	$p_1 p_2 p_3 p_4$	$p_1 \{p_2 \{p_3 p_4 + (1 - p_3) 0\} + (1 - p_2) 0\} + (1 - p_1) 0$

TABLE II .PARALLEL STRUCTURE.

	<b>inclusion - exclusion</b>	<b>pivotal decomposition</b>
n=2	$p_1 + p_2 - p_1 p_2$	$p_1 + (1 - p_1) p_2$
n=3	$p_1 + p_2 + p_3 - (p_1 p_2 + p_1 p_3 + p_2 p_3) + p_1 p_2 p_3$	$p_1 + (1 - p_1) \{p_2 + (1 - p_2) p_3\}$
n=4	$p_1 + p_2 + p_3 + p_4 - (p_1 p_2 + p_1 p_3 + p_1 p_4 + p_2 p_3 + p_2 p_4 + p_3 p_4) + (p_1 p_2 p_3 + p_1 p_2 p_4 + p_1 p_3 p_4 + p_2 p_3 p_4) - p_1 p_2 p_3 p_4$	$p_1 + (1 - p_1) [p_2 + (1 - p_2) \{p_3 + (1 - p_3) p_4\}]$

As described on table 1 and 2, we show that in case of series structure, the inclusion-exclusion algorithm is proper in computational complexity, but pivotal decomposition algorithm is proper in computational complexity in case of parallel structure. Now we compare the CPU time between two algorithms in complex system. Table 3 displays elapsed computation time of CPU for each algorithm.

TABLE III .CPU TIME FOR THE ALGORITHMS.

system(Algorithm)	# Min Paths	1	2	3	4	5	mean	CPU sec
6-component parallel (inclusion-exclusion)	6	53.45s	53.63s	53.55s	53.27s	53.59s	53.50s	5.35ms
4-component parallel (pivotal decomposition)	4	3.69s	3.56s	3.62s	3.56s	3.66s	3.62s	0.36ms
2-out-of-4:G system (inclusion-exclusion)	6	59.54s	59.49s	59.05s	59.15s	59.19s	59.28s	5.93ms
5-component bridge system (inclusion-exclusion)	4	30.63s	30.67s	30.36s	30.54s	30.41s	30.52s	3.05ms
5-component bridge system (pivotal decomposition)	4	4.29s	3.99s	4.07s	4.12s	3.94s	4.08s	0.41ms
7-component bridge system (inclusion-exclusion)	7	59.25s	59.60s	59.53s	59.69s	59.61s	59.54s	5.95ms
7-component bridge system (pivotal decomposition)	7	4.30s	4.13s	4.23s	4.20s	4.25s	4.22s	0.42ms
8-component bridge system (inclusion-exclusion)	8	67.02s	66.56s	66.99s	66.80s	66.88s	66.85s	6.69ms
8-component bridge system (pivotal decomposition)	8	3.90s	3.78s	3.94s	3.98s	3.86s	3.89s	0.39ms

\* Operation Environment:

PC - IBM 586; CPU - Pentium III; RAM - 128MB  
(Synchronous DRAM 100MHz); OS - Windows 2000;  
Compiler - Microsoft Visual C++ 6.0; 0.08s/10000(iteration) =  
8 $\mu$ s = 0.008ms.

As described on table 3, we also suggest that pivotal decomposition algorithm provides an efficient method for complex system. We expect that our method applied in this paper is further extended to the case when components of the system are given multi-states.

#### ACKNOWLEDGEMENT

Funding for this paper was provided by Namseoul University.

#### REFERENCES

- [1] S. C. Lee, Determination of Probability of Component or Subsystem Failure, Journal of Korean Society for Quality Control, 2-2(1993) 121-130.
- [2] K. D. Heidtmann, Inverting Paths and Cuts of 2-State Systems, IEEE Transactions on Reliability, 32-5 (1983) 469-471.
- [3] J.Tang, Mechanical System Reliability Analysis Using a Combination of Graph Theory and Boolean Function, Reliability Engineering and System Safety, 72 (2001) 21-30.
- [4] W. Feller, An Introduction to Probability Theory and Its Applications I. II. John Wiley & Sons Inc., New York, 1968.
- [5] S. C. Lee, Reliability Expression for Complex System, Journal of Korean Communications in Statistics, 3-3(1996) 125-133.
- [6] R. E. Barlow, F. Proschan, Statistical Theory of Reliability and Life testing, Holt, Reinhardt & Winston Inc., 1975.