

시스템의 信賴性 豫測을 위한 컴퓨터 프로그램

(A Computer Program for System Reliability Prediction)

金 永 輝*
崔 文 基*

Abstract

A computer program for computing complex system reliability is described. The program is composed of three phases: Phase I program reduces all series, parallel and series-parallel components and subsequently obtains an irreducible non-series-parallel system. Phase II program enumerates all the possible paths from the source to the sink of the graph. Phase III program then computes system reliability based on the information obtained by the Phase II program. The program is based on a modified version of the algorithm published in [6]. An example of the use of the computer program is given.

1. 緒 論

시스템의 信賴性(Reliability)을 豫測하는 문제는

(a) 設計圖面上에 나타난 시스템이 完成될 경우 豫定된 기간 동안에 만족할 만한 信賴性을 유지 할 수 있는가를 豫測하고

(b) 시스템의 建設 또는 生産을 實現할 수 있는 여러가지 設計方案을 비교 선택할 수 있는 判斷基準을 提供하며

(c) 어떤 特定한 시스템 設計상에 존재하는 信賴性문제의 소재를 파악케 함으로서

信賴性이 높은 시스템을 경제적으로 生産할 수 있는 設計의 指針이 된다는 데 보다 큰 意義가 있다. 信賴性豫測에는 設計圖面上에 명시된 시스템 構成 部品の 신뢰성에 관한 情報를 이용한다. 部品の 신뢰성에 관한 정보가 현재로서 완벽하다고 할 수 없으나 특히 電子部品の 경우 外國에서는 상당한 量의 故障率(Hazard Rate)에 관한 데이터가 公開되고 있다. 따라서 信賴性의 豫測문제는 시스템의 신

뢰성 블록 다이어그램(Reliability Block Diagram)상에 나타난 部品 상호간의 관계를 고려하여 시스템의 信賴性을 계산해 내는 문제로 귀착된다.

시스템의 構成部品이 신뢰성 블록 다이어그램상에서 直列 또는 並列로 연결되어 있는 경우 시스템의 신뢰성은 간단히 계산된다. 그러나 시스템 또는 그 일부의 部品相互간의 관계가 非直並列(Non-series-parallel)관계에 있을 때에는 信賴性豫測問題는 간단히 해결되지 않는다.

非直並列 부품상호간의 관계를 포함하는 복잡한 시스템의 信賴性을 豫測할 수 있는 방법에는 여러가지가 있다. 시스템이나 部品이 成功 또는 失敗의 2가지 상태(State)만을 가지는 경우를 생각하자.

Event-space法은 시스템이 임의의 時點에서 놓여 있을 수 있는 모든 상태(사상)를 羅列하고 그 중에서 시스템이 성공적으로 기능을 다할 수 있는 상태만을 가려내고 그 確率의 總合 卽 시스템의 신뢰성을 구하는 方法이다. 이 方法은 가령 시스템을 구성하는 部品の 數

* 高麗大學校 産業工學科

가 10인 경우 그 상태는 무려 $2^{10}=1024$ 가지에 달한다. 따라서 시스템을 구성하는 部品の數가 많아지면 Event-space法은 그 計算量이 막중하여 사실상 사용 불가능해진다.

Tree-diagram法은 어떤 의미에서는 Event-space法의 개량형이라고도 할 수 있다. 이 方法은 Event-space法에서 이미 言及한 바 있는 가능한 모든 상태를 Tree-diagram을 이용하여 展開한다. 시스템의 信賴性은 Event-space法과 마찬가지로 시스템이 成功할 수 있는 상태가 일어나는 확률의 乘積이 된다. Tree-diagram法은 “가능한 모든상태”를 展開하는 과정에서 시스템이 失敗하리라는 것이 확정된 Tree의 줄기는 그 이상 존재할 필요가 없으므로 Event-space法에 비해 計算量을 감소시킬 수 있다. Tree-diagram法은 Chelson [3]에 의해 컴퓨터 프로그램이 작성된 것이 있다.

이 밖에도 Cut-set and Tie-set [1, 9]이라든가 信賴性의 概算値와 限界値(Approximation and Bounds)를 구하는 문제[2, 4, 5, 7, 8, 9]가 이미 고찰된 바 있다.

앞으로 考察하고자 하는 研究課題는 본 연구자의 한 사람이 이미 발표한 논문 [6]의 理論에 기초를 둔 시스템의 信賴性豫測 컴퓨터 프로그램을 개발하는 것이다. 이 프로그램은 다음 3단계로 구성된다.

- (1) 시스템의 直列 또는 並列관계에 있는 모든 部品을 가상적인 한 개의 等價部品으로 代替시킨다.
- (2) (1)단계에서 얻어진 縮小化한 신뢰성 블록 다이어그램상에서 入力端(Source)과 出力端(Sink)을 연결하는 모든 經路(Path)를 구한다.
- (3) (2)에서 구해지는 經路에 관한 情報를 이용하여 시스템의 신뢰성을 算出한다.

2. 信賴性豫測의 理論

컴퓨터 프로그램 개발에 이용된 信賴性豫測의 理論[6]은 다음과 같이 요약된다. 이 문제를 고찰함에 있어서 시스템은 하나의 Probabilistic Graph로 간주한다. 즉 이 Graph에는 시스템을 구성하는 部品(또는 Subassembly)

하나 하나에 대응하는 Branch가 존재하며 또 각 Branch에는 해당 部品の 壽命(Time-to-failure)을 나타내는 確率變數가 대응한다.

지금 $G=(N, M)$ 을 q 개의 Node $N=\{a_i\}$ 와 m 개의 Branch $M=\{b_{ij}\}$ 로 구성되는 방향을 가지는 Graph(Directed Graph)라고 정의하자. 각 Branch b_{ij} 에는 陽의 實數 값을 가지는 確率變數 T_{ij} 가 대응한다. 또 이 Graph에는 入力端 (Source) s 와 出力端 (Sink) z 를 연결하는 經路가 r 개 존재한다고 하고 그 l 번째의 經路를 $P_l(s, z)$ 로 표시한다고 하자. 그러면 시스템의 수명 T 는 다음과 같이 定義할 수 있다.

$$Y_i = \text{Min}_{b_{ij} \in P_i} \{T_{ij}\} \dots \dots \dots (1)$$

$$T = \text{Max}_{1 \leq l \leq r} \{Y_l\} \dots \dots \dots (2)$$

또한 시스템의 信賴性 $R_T(t)$ 는 다음과 같이 정의된다.

$$R_T(t) = 1 - \int_0^t f(\tau) d\tau$$

단 $f(t)$ 는 T 의 確率密度函數이다. 지금 T_{ij} 의 確率密度函數 $f_{ij}(t)$ 를 알고 있다고 가정하면

$$R_{ij}(t) = 1 - \int_0^t f_{ij}(\tau) d\tau$$

또 l 번째 經路 $P_l(s, z)$ ($l=1, 2, \dots, r$)상에 있는 모든 Branch b_{ij} 에 대응하는 T_{ij} 가 서로 獨立관계에 있다면

$$R_{Y_l}(t) = \prod_{b_{ij} \in P_l(s, z)} R_{ij}(t) \dots \dots \dots (3)$$

r 개 經路는 서로 並列관계에 있으므로 시스템의 신뢰성은 다음 관계식에서 구할 수 있다.

$$R_T(t) = \left(1 - \prod_{i=1}^r (1 - \prod_{b_{ij} \in P_i(s, z)} R_{ij}(t)) \right)^* \dots \dots (4)$$

여기서 演算子 *는 $[\prod_{ij} P_{ij}]^* = \prod_{ij} P_{ij}$ 즉 예를 들자면 $[P_1^3 P_2^2 P_3^4 P_4^2]^* = P_1 P_2 P_3 P_4$ 를 의미한다. 일반적으로 시스템의 Graph를 구성하는 Branch b_{ij} 가운데는 2개 이상의 經路에서 重複되어 나타나는 Branch가 있으므로 確率變數 Y_l , $l=1, 2, \dots, r$ 은 서로 獨立된 確率變數라고 할 수 없다. 연산자 *는 이러한 문제를 [6]를 해결해 준다.

3. 컴퓨터 프로그램을 위한 Algorithm

(a) Phase I Algorithm

이 프로그램의 計算能率은 入力端과 出力端을 연결하는 經路의 수와 각 經路上에 있는 Branch의 수에 따라 크게 좌우 된다. 따라서 計算能率을 向上시키기 위해서는 원래의 信賴性 블록 다이어그램을 그 特性을 유지하는 범위내에서 최소한으로 縮小시킬 필요가 있다 제 1 단계 프로그램은 이 目的의 달성을 위한 것으로 블록 다이어그램에 포함되어 있는 모든 直列 또는 並列관계에 있는 部品을 가상적인 한 개의 等價部品으로 代替시킨다.

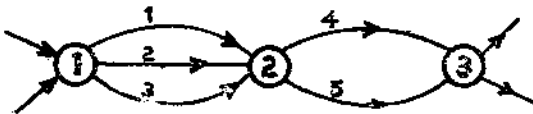


그림 1 가상적인 Subsystem

지금 그림 1에 圖示한 바와 같은 가상적인 Subsystem을 생각하여 보자. 분명히 Branch 1, 2, 3과 Branch 4, 5는 並列관계에 있다. 컴퓨터에서 並列관계에 있는 Branch를 식별하는 方法은 그 Starting Node와 End Node가 같은 가를 비교하는 것이다. 그림 1에서 Branch 1, 2, 3의 Starting Node와 End Node는 각각 1과 2를 共通으로 사용하고 있다. 일단 並列관계에 있는 Branch가 확인되면 관계식

$$R_p(t) = 1 - \prod_{i=1}^k [1 - R_i(t)]$$

을 적용하므로써 가상적인 部品에 해당하는 한 개의 Branch로 代替한다. 단 윗 식에서 $R_i(t)$ 는 병렬회로의 i 번째 部品의 신뢰성, k 는 병렬관계에 있는 部品의 수이다. 이 方法으로 그림 1은 그림 2로 축소된다.

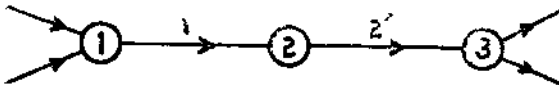


그림 2

그림 2에서 Branch 1'와 2'는 直列관계에 있다. 즉 Branch 1'의 End Node는 곧 Branch 2'의 Starting node가 되어 있다. 直列로 연결된 經路上에서는 그 經路의 Starting Node와 End Node를 除外한 모든 Node가 2개의 인

접 Branch의 End Node임과 동시에 Starting Node가 되며 또 다른 Branch가 연결되어 있지 않다. 直列관계에 있는 Branch는 관계식을 이

$$R_s = \prod_{i=1}^m R_i(t)$$

용하여 한개의 Branch로 代替된다. 단 여기서 $R_i(t)$ 는 直列로 된 경로상에서 i 번째 部品의 신뢰성이며 m 은 直列經路를 구성하는 部品의 수이다.

이와 같은 方法을 全시스템에 反復的으로 적용하여 並列 및 直列部品을 모두 代替시키면 최종적으로 그 이상 축소할 수 없는 시스템 블록 다이어그램이 얻어진다. 이것이 Phase I Algorithm의 出力이다.

(b) Phase II Algorithm

제 2 단계 프로그램은 제 1 단계 프로그램의 결과로 얻어진 縮小된 블록 다이어그램에서 제 3 단계의 信賴性 계산에 직접 필요한 入力端과 出力端은 연결하는 모든 경로를 찾아낸다. 참고문헌 [6]의 제 2 단계 Algorithm에서는 시스템의 블록 다이어그램을 나타내는 行列을 어떤 特定한 演算法則 [6]에 따라 조작하는 方法을 사용하였다. 그러나 이 行列의 조작에는 컴퓨터의 많은 Core-space를 필요로 하기 때문에 비능률적이다. 이 研究에서는 다음과 같은 새로운 Algorithm을 이용하였다.

(1) 入力端(Source)과 Starting Node가 일치하는 모든 Branch로서 Tree를 형성한다.

(2) Tree의 End Node가 出力端인가를 확인한다.

① 만일 이 조건이 만족되면 해당 經路를 적어내고 (Print out) GO TO STEP(4)

② 만일 이 조건이 만족되지 않으면 Tree의 End Node가 Starting Node가 되는 새 Branch를 찾아서 GO TO STEP (3)

(3) 새 Branch의 Node가 Tree의 入力端에서 해당 End Node에 이르는 經路上에서 重複(Cycle인가를 확인)되는 가를 확인하고

④ 새 Branch가 Cycle을 형성하면 GO TO (2)-①

⑤ 새 Branch가 Cycle을 형성하지 않으면

이 Branch를 해당 End Node에 추가하고 GO TO (2)

(4) Tree의 End Node가 모두 出力端과 일치하는가를 확인한다.

⑧ 이 조건이 만족되면 STOP

⑨ 이 조건이 만족되지 않으면 GO TO (2)

①

(c) Phase II Algorithm

제 3 단계 프로그램은 제 2 단계에서 얻어지는 入力端과 出力端을 연결하는 經路 $P_l(s, z), l=1, 2, \dots, r$ 에 관한 정보로부터 시스템의 信賴性을 산출한다.

지금 제 2 절의 식(4)에서 시간 t 가 주어졌다고 가정하고 $R_{Y_l} = P_l$, 시스템의 信賴性을 P_S 라 하면.

$$P_S = [1 - \prod_{i=1}^r (1 - \prod_{b_{ij} \in P_i(s, z)} P_{ij})]^*$$

이 식을 이용하여 信賴性을 구하고자 할 경

우 []안의 項 하나 하나가 決定되어야 *演算을 할 수가 있다. 따라서 이 方法을 이용하기 위해서는 임의의 시스템에 대해서 []안의 각 項을 展開하는 문제를 해결해야 한다.

이 보다도 다소 수월한 方法은 다음식

$$P_S = \sum_{l=1}^r P_l - \sum_{l_1, l_2} [P_{l_1} P_{l_2}]^* + \sum_{l_1, l_2, l_3} [P_{l_1} P_{l_2} P_{l_3}]^* - \dots$$

$$\text{단 } P_l = \prod_{b_{ij} \in P_l(s, z)} P_{ij}, l=1, 2, \dots, r$$

을 이용하는 것이다. 이 연구에 사용된 프로그램은 컴퓨터의 Core-space를 감안하여 $r=10$ 을 최대 허용 經路의 수로 하고 방금 설명한 方法을 채택하였다. Phase I에서 원래의 시스템 블록 다이어그램이 상당히 축소될 것이므로 최대경로의 수를 10으로 잡더라도 상당한 규모의 시스템을 다룰 수 있을 것으로 생각된다. 전체 프로그램의 Macro-flow Chart를 그림 3에 圖示하였다.

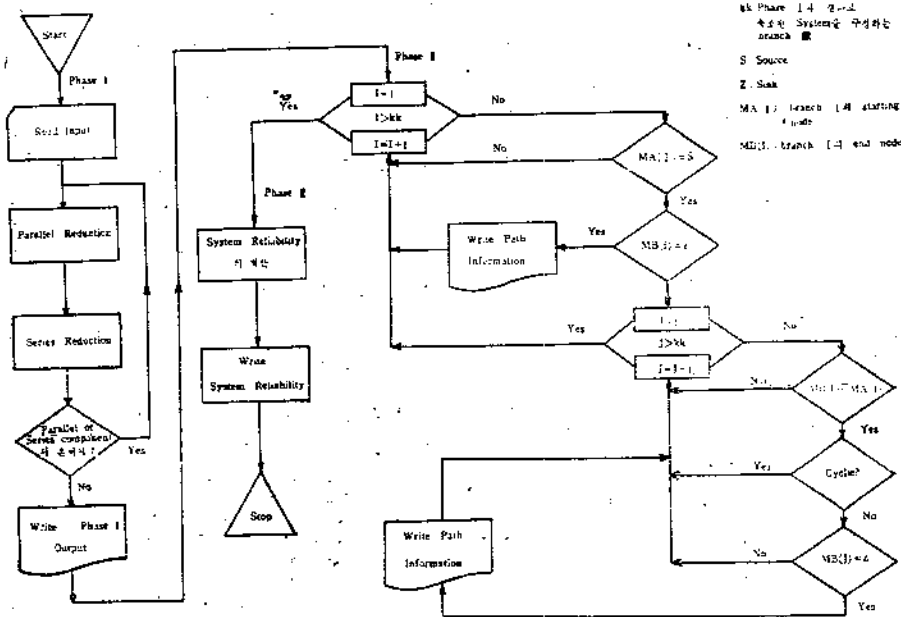


그림 3 Macro-flow chart

4. 컴퓨터 프로그램의 개요

이 프로그램은 高麗大學校 電子計算所에 설치된 IBM-1130(16K) 전자계산기에 맞도록 작성되었다. 프로그램의 구조상 시스템을 구성하는 부품의 수에는 제한이 없다. 다만 제 3 단계 프로그램에서 入力端과 出力端간의 경

로의 수가 10이하로 制限되어 있으므로 제 2 단계 프로그램의 出力으로 나오는 經路의 수가 10보다 많을 경우에는 시스템을 몇 개의 Subsystem으로 分解하여 각 Subsystem에 대해서 이 프로그램을 적용하면 될 것이다.

이 프로그램을 이용하기 위해서는 제 1 단계에서 다음과 같은 入力 카이드가 필요하다.

Type I card : JJ(시스템을 구성하는 부품의 수)

Type II card : 각 부품의 블록 다이어그램상의 Starting Node와 End Node 및 신뢰성

그림 4와 같이 그 신뢰성 블록 다이어그램이 나타나는 가상적인 시스템의 入力 데이터는 표 1과 같다.

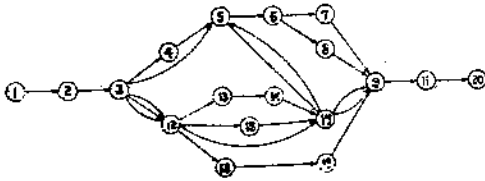


그림 4. 가상적인 시스템의 reliability graph

*** RAW DATA ***

STARTING NODE	END NODE	COMPONENT RELIABILITY (R(MA,MB))
1	2	0.900000
2	3	0.900000
3	4	0.800000
3	5	0.700000
3	12	0.600000
3	12	0.600000
3	12	0.700000
4	5	0.800000
5	6	0.850000
5	17	0.950000
6	7	0.700000
6	8	0.700000
7	9	0.750000
8	9	0.750000
9	11	0.950000
11	20	0.900000
12	13	0.500000
12	15	0.650000
12	17	0.550000
12	18	0.850000
13	14	0.600000
14	17	0.600000
15	17	0.550000
17	5	0.950000
17	9	0.650000

17	9	0.650000
18	19	0.750000
19	9	0.700000

표 1 그림 4의 시스템에 대한 入力 데이터 표 I과 같은 入力데이터에서 제 1 단계 프로그램의 出力으로 표 2와 같은 결과가 얻어진다.

*** A Non Series-Parallel System Which is Irreducible***

Starting Node	End Node	Component Reliability (R(MA,MB))
1	3	0.810000E 00
3	5	0.891999E 00
3	12	0.952000E 00
5	9	0.658218E 00
5	17	0.950000E 00
9	20	0.850000E 00
12	17	0.762917E 00
12	9	0.446249E 00
17	5	0.950000E 00
17	9	0.877500E 00

제1단계 프로그램의 Output

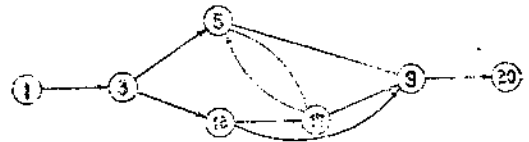


그림 5. 축소된 시스템의 reliability graph

표 2의 Output을 圖表化한 것이 그림 5이다. 즉 제 1 단계 프로그램의 결과로 28개의 Branch는 10개로 18개의 node는 7개로 축소된다는 것을 알 수 있다. 제 2 단계 프로그램은 표 2의 output을 入力데이터로 하여 표 3과 같은 經路에 관한 정보를 算出한다.

*** Path Information ***

3	3	3	3	3
5	5	12	12	12
9	17	17	17	9
20	9	5	9	20
	20	9	20	
		20		

표 3 제2단계 프로그램의 Output

이 시스템에는 (1, 3, 5, 9, 20), (1, 3, 5, 17, 9, 20) 등 5개의 경로가 존재한다. 표 3의 Output을 入力데이터로 제 3 단계 프로그램은 시스템의

신뢰성을 算出한다.

*** System Reliability is 0.701856E 00 ***

표 4. 제3단계 프로그램 Output

이 문제의 신뢰성을 算出하는 데는 제 1 단계와 2 단계에서 각각 2분씩 제 3 단계에서 6분 도합 10분 (I/O+Compilation+Execution time)의 시간이 소요되었다.

5. 結 言

이 프로그램에는 時間的인 要素를 직접 考慮대상에서 除外하고 있으나 프로그램의 論理는 시간 如何에 關係될 없이 항상 成立한다. 따라서 만일 시간의 變化에 따른 시스템의 信賴性의 變化에 관심이 있을 경우에는 일정한 시간 t 에 대해서 각 Branch b_{ij} 에 해당하는 部品の 신뢰성 $R_{ij}(t)$ 를 $P_{ij}=R_{ij}(t)$ 로 하고 이 프로그램을 적용한 다음 다시 시간 t 를 變化시키면서 프로그램을 反復하여 적용함으로써 문제를 해결할 수 있다.

本 研究는 財團法人 產學協同財團의 學術研究費의 支援을 받아 實施된 것이다. 同財團에 깊은 謝意를 表明하는 바이다.

References

[1] Aeronautical Radio Inc., *Reliability Engineering*. Englewood Cliffs, N.J.: Prentice-Hall, 1964

[2] R.E. Barlow and F. Proschan, *Mathematical Theory of Reliability*. New York: Wiley, 1965

[3] P.O. Chelson, "Reliability Math Modeling Using the Digital Computer," *NASA TR-32-1089*, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, Calif., Apr. 1967

[4] H. Frank, "Shortest Paths in Probabilistic Graph," *J. ORSA*, vol. 17, pp. 583-599, July-Aug. 1969

[5] P.A. Jensen and M. Bellmore, "An Algorithm to Determine the Reliability of a Complex System," *IEEE Trans. Rel.*, vol. R-18, pp. 169-174, Nov. 1969

[6] Y.H. Kim, K.E. Case and P.M. Ghare, "A Method for Computing Complex System Reliability," *IEEE Trans. Rel.*, vol. R-21, pp. 215-219, Nov. 1972

[7] M. Messinger and M. Shooman, "Reliability Approximations for Complex Structures," *Proc. 1967 Annu. Symp. Rel.* (New York), pp. 292-301.

[8] A.C. Nelson, J.R. Batts and R.L. Beadles, "A Computer Program for Approximating System Reliability," *IEEE Trans. Rel.*, vol. R-19, pp. 61-65, May 1970

[9] M.L. Shooman, *Probabilistic Reliability: An Engineering Approach*. New York: McGraw-Hill, 1968