Journal of the KSQC
Vol. 9, No. 2, 1981

非線型計劃法을 이용한 信頼性의 最適化
Reliability Optimization By using a Nonlinear Programming

李 昌 鎬*

요 약

점증되고 있는 고신뢰성 제품의 설계에 있어 주어진 선형제약조건 内에서 체계의 신뢰성을 최대화 하는 방법을 소개하고 이를 해결하는 비선형계획법을 반복단계로 하여 Computer Programming을 하였다. 단, 본 논문에서 다루는 체계는 병렬중복구조를 갖는 직렬다단계 구조이다. 타당성 검토를 위한 예제를 해결하였으며 Computer Program은 지면관계로 생략하였다.

ABSTRACT

This paper deals with the reliability optimization of parallel - in - series system subject to several linear constraints.

The model of nonlinear constrained optimization is transformed to a saddle point problem by using Lagrange multipliers. Then Newton - Raphson method is used to solve the resulting problem and these step - by - step solution procedures are programmed in Basic Level II of micro - computer TRS-80.

An example which has two linear constraints is solved and the results are analyzed.

I. 序 論

現在에 와서는 信頼性이라는 말이 많이 쓰이게 되었지만 20여년 전만해도 흔하게 사용되지 않았다. 그만큼 모든 제품에 대해서 高信頼性을 요구하게 되었고, 그러한 고신뢰성을 달성하기 위해서는 정밀한 제품설계는 물론 제조과정의 고급화가 필요하다. 결국 고신뢰도를 달성하기 위해서는 상당한 비용과 시간이 필요하게 되었다. 본 논문에서는 이러한 점을 고려하여 한정된 자원, 체계에 부과되는 제한등을 감안한 체계의 신뢰성을 최적화하였다. 또한, 신뢰성의 최적화를 달성하기 위해 비선형계획법의 기법을 이용하였으며, 알고리즘 접근식의 방법으로 하여 컴퓨터 프로그램을 하였다. 타당성 검토에 대한 예로서 논문 [1]의 2개의 선형제약조건을 갖는 체계에 대한 최적 신뢰성을 구하고 비교·분석하였다.

II. 역사적 고찰

Mann [2]은 하부체계 (Subsystem)가 이항분포를 하는 직렬과 병렬체계의 신뢰성에 대한 상한·하한을 계산하였다. Tillman, Hwang, 그리고 Kuo [3]는 체계의 신뢰성을 극대화하기 위해서 비선형 혼합정수계획법을 사용하였다. 이밖에 직렬·병렬 체계가 아닌 체계의 신뢰성을 얻는 방법으로 최소경로기법 (Minimal path method)을 사용한 논문도 있다. [4], [5], [6] 또한, 이러한 여러 종류 체계의 신뢰성을 얻으려는 많은 논문은 Kumar, Agarwal [7], Tillman, Hwang, 그리고 Kuo[8]에서 보여주

*仁荷大學校 産業工學科 專任講士

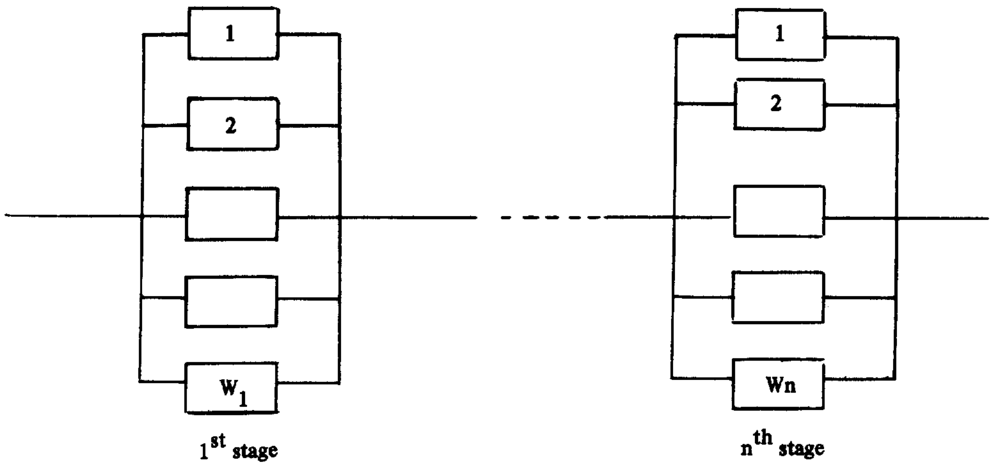


Fig. 1. System with n series stages and parallel redundancy

고 있다. 본 논문은 직렬다단계, 각 단계는 병렬중복구조를 갖는 체계의 신뢰성을 최적화하기 위해 비선형계획법을 사용하였다.

Ⅲ. 비선형 신뢰성 문제

1. 체계 설명과 가정

본 논문에서 다루는 체계는 〔그림 1〕과 같은 구조를 갖는다.

또한, 이 체계에 대한 가정은 다음과 같다.

1) 직렬의 n개 구조는 상호독립이며, 병렬중복구조도 상호독립이다. 병렬중복에서의 모든 개체들(units)은 같은 조건하(in active state)에 있다.

2) 체계에는 m개의 선형제약조건이 있다. 예를들면 비용, 체계의 무게, 부피 등이다. 이러한 제약조건은 병렬중복수와 선형관계에 있다.

3) 각 개체나 단계의 상태는 2등분 된다. 하나만의 개체 상태가 좋으면 그 단계의 상태도 좋다.

체계의 신뢰성을 극대화하려는 것은 그의 대수(logarithm)를 극대화하려는 것과 같다. 이는 체계신뢰성의 대수는 병렬중복수에 대해서 단조증가 오목함수이기 때문이다.

2. 수학적 모형

체계의 신뢰성 R_s는

$R_s = \Pi_j R_j = \Pi_j (1-Q_j)$ ......(1)

여기서 $R_j = 1-Q_j$은 j번째 단계신뢰성.

선형제약조건은

$\sum_j a_{ij} w_j \le b_i,\ i=1,2,\cdots\cdots m$ ......(2)

여기서 a_{ij}은 j번째 단계에서 병렬중복 개체당 i번째 자원의 필요량

w_j는 j번째 단계의 병렬 중복 개체수,

b_i는 i번째 자원의 이용가능량

여기서 각 단계의 비신뢰성 Q_j는

$Q_j = q_j^{w_j}$ ......(3)

$q_j = 1-p_j$는 j번째 단계에서 개체 하나의 비신뢰성

(3)식에서 $w_j = lnQ_j/lnq_j$ ......(4)

(4)를 (2)에 대입하면

$\sum_j a_{ij} \cdot lnQ_j/lnq_j \le b_i,\ i=1,2,\cdots\cdots m$ ......(5)

(5)식에서 $\alpha_{ij} = a_{ij}/lnq_j$라 놓으면 체계신뢰성에 대한 식은

극대화 $S = ln\, R_s = \sum_j ln(1-Q_j)$ ......(6)

제약조건 $\sum_j \alpha_{ij}\, lnQ_j \le b_i,\ i=1,2,\cdots m$ ......(7)

3. Lagrange 방법의 적용성 검토

극대화 $f(x)$ ......(8)

제약조건 $g_i(x),\ i=1,2,\cdots,m$ ......(9)

(9)식을 만족하는 $f(x)$의 극대화 조건은 다음과 같다.[8]

1) Lagrange 함수의 편미분 값이 x^*에서 0이 되면 $f(x^*)$는 x^*에서 최대값을 갖는다.(필요조건)

2) $f(x)$는 단조오목함수, $g_i(x)$는 볼록 함수이면

1)의 조건은 필요충분조건이다.

위의 관계를 (6)식과 (7)식에 적용시키면 목적함수 S는 Q_j의 단조오목함수, 제약조건은 Q_j의 볼록함수이므로 Lagrange 방법을 적용시킬 수 있다.

4. Lagrange 方法의 적용

(6), (7)식에서 Lagrange 함수를 정의하면

$$F(R, \lambda)=\sum_j lnR_j+\sum_i \lambda_i(\sum_j a_{ij}\, lnQ_j-b_i) \cdots\cdots (10)$$

(6)과 (7)식을 최적화한다는 것은 3의 1)과 2) 에 의해서 (10)의 편미분함수를 0으로 하는 변수값을 찾는 것과 같다.

즉, $$\frac{\partial F(R,\lambda)}{\partial R_j}=\frac{1}{R_j}-\sum_i \lambda_i a_{ij}/Q_j=0,$$

$$j=1, 2, \cdots, n \cdots\cdots (11-a)$$

또는, $$f_j=\frac{Q_j}{R_j}-\sum_i \lambda_i \cdot a_{ij}=0, \quad j=1,2,\cdots,n \cdots\cdots (11-b)$$

같은 방법으로

$$f_{n+i}=\lambda_i\cdot\frac{\partial F}{\partial \lambda_i}=\lambda_i[\sum_j a_{ij}\, lnQ_j-b_i]=0,$$

$$i=1, 2, \cdots, m \cdots\cdots (12)$$

앞으로는 표현을 다음과 같이 한다.

$$f_R=(f_1, f_2, \cdots, f_n), \quad f_\lambda=(f_{n+1}, f_{n+2}, \cdots, f_{n+m})$$

$$\lambda_i^* \le 0, \quad i=1, 2, \cdots, m \cdots\cdots (13)$$

$$\sum_j a_{ij}\, lnQ_j \le b_i, \quad i=1, 2, \cdots, m \cdots\cdots (7)$$

앞으로의 신뢰성문제 해결 과정은 (7)과 (13)의 기준을 만족하면서 (11-b)와 (12)를 성립시키는 λ_i, Q_j를 찾는 것이다.

5. 해를 얻기위한 Newton-Raphson 방법[10]

(11-b)식과 (12)식을 성립시키는 λ_i, Q_j값은 Newton-Raphson 방법에 의한 연속적인 개선에 의해서 찾을 수 있다. 체계 신뢰성의 최적화를 위한 해의 연속적인 개선 과정은 다음과 같다.

1) Newton-Raphson 방법의 초기치를 결정한다. Newton-Raphson 방법에 있어서의 초기치는 중요한 위치를 차지하나 신뢰성 문제에서의 초기치로는 $R_j^0=p_j$, $\lambda_i^0=-0.01$로 하는 것이 편리하다고 인정되고 있다. [1] R_j^0, λ_i^0 초기치에서 (11-b), (12)함수값을 계산한다.

2) R, λ에 대한 개선치 계산에는 일반적인 Newton-Raphson 방법을 사용하는데 계산편리상 행렬식으로 표현한다.

$$\lambda^t=\lambda^{t-1}-\Delta\lambda \cdots\cdots (14-a)$$

$$R^t=R^{t-1}-\Delta R \cdots\cdots (14-b)$$

여기서 $$\begin{pmatrix}\Delta\lambda\\ \Delta R\end{pmatrix}=\nu\cdot[\text{Jacobian}]^{-1}\begin{bmatrix}f_\lambda\\ f_R\end{bmatrix} \cdots\cdots (15)$$

(15)식에서 오른쪽 항에서의 Jacobian과 f_λ, f_R값은 $(t-1)$번째 반복단계에서 계산된다. ν는 반복단계에서 수정항의 크기를 결정한다. (일반적으로 Newton-Raphson 방법에서는 $\nu=1$이나, $\nu<1$을 택하므로서 좀 더 자세한 해를 얻을 수 있다)

Jacobian 행렬식의 구성에 대해서 설명하면 다음과 같다. Jacobian 행렬식은 (11-b), (12)식의 $(n+m)$개의 함수에 대한 편미분으로 구성된다. $(n+m)\times(n+m)$ 행렬식인 Jacobian은 (15)식과 연관해서 하부 행렬식으로 구분하여 설명하면 다음과 같다.

$$(\text{Jacobian})=\begin{pmatrix}J_1 & J_2\\ J_3 & J_4\end{pmatrix} \qquad \text{여기서}$$

$$J_1=\frac{\partial f_{n+i}}{\partial \lambda_i}=\sum_j a_{ij}\, lnQ_j-b_i, \quad m\times m \text{ 행렬식}$$

$$J_2=\frac{\partial f_{n+i}}{\partial R_j}=-a_{ij}\lambda_i/Q_j \quad , \ m\times n \text{ 행렬식}$$

$$J_3=\frac{\partial f_j}{\partial \lambda_i}=-a_{ij} \quad , \ n\times m \text{ 행렬식}$$

$$J_4=\frac{\partial f_j}{\partial R_j}=-1/R_j^2 \quad , \ n\times n \text{ 행렬식}$$

이와같은 계산 과정을 거쳐서 t번째 반복단계에서의 R_j, λ_i값이 계산되면 1)로 되돌아 간다.

체계 신뢰성문제 (7), (11-b), (12), (13)식에 대한 해를 개선하는 과정에서 반복단계를 어떻게 중단할 것인가?

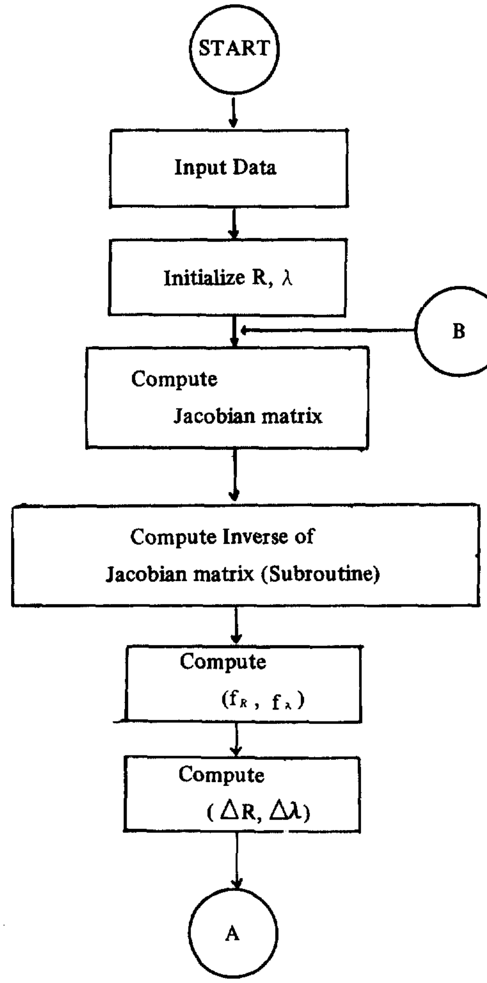
다음과 같은 3가지 방법을 사용할 수 있다.

1) 식(13)은 해를 구해가는 과정에 만족되고 있으므로 식(7)을 만족시키는 해를 구하도록 해야 한다. 그래서 Jacobian의 하부행렬식 J_1중 한 원소가 양의 값에서 음의 값으로 변화되는 반복단계(하나의 제약조건이 위반되는 경우)에서 중단한다.

2) $|\Delta R_j|$에 대한 기준에 의해서 반복단계를 중단한다.

3) 2)와 같은 방법으로 임의의 반복단계에서 (f_λ, f_R)의 절대값에 대한 기준에 의해서 중단한다.

이상의 3가지 방법중 본 논문에서는 3)의 방법을 채택하였으며 1)과 2)의 기준과 비교하여 보았다.



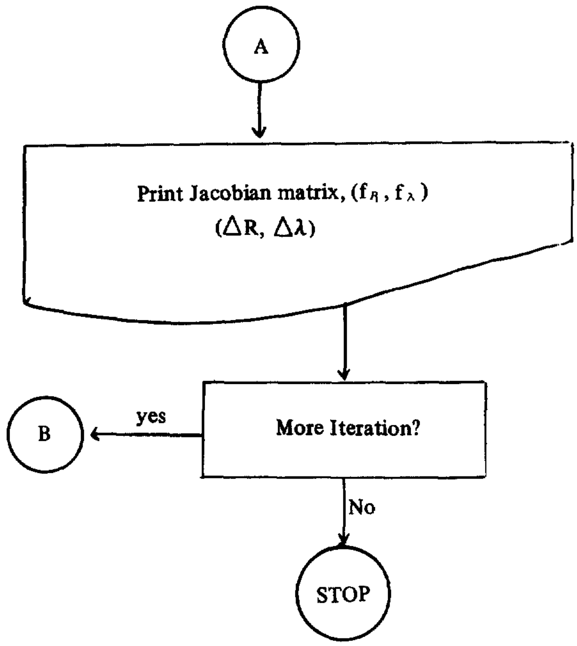


Fig. 2 Flow Chart

Ⅳ. Computer Program과 예제결과 분석

다음 〔그림 2〕는 Newton-Raphson 방법에 대한 반복과정을 Computer Programming 하기위한 흐름도표이다. 〔그림 2〕에 대한 Computer Program 은 Basic Level Ⅱ 언어로 쓰여졌다.

1. 예제

타당성 검토를 위해서 다음 예제를 사용했다. 직렬 4 단계 구조를 갖는 체계에 대해서 각 단계당 병렬중복수가 어떨때 2 개의 선형제약 조건内에서 체계의 신뢰성이 최대화 되겠는가?

단계 / 제약조건	1	2	3	4	허용가능량
비 용($)	1.2	2.3	3.4	4.5	56
무 게 (g)	5.0	4.0	8.0	7.0	120
P_j	0.80	0.70	0.75	0.85	

2. 결과분석

다음의 분석은 반복단계에서 수정항의 크기를 결정하는 $\nu=1$에 대한 것이다.

1) 첫번째 반복단계에서의 Jacobian 행렬식은

$$
(\text{Jacobian}) = \left[\begin{array}{cc|cccc}
-44.6 & 0 & -0.03728 & -0.06367 & -0.09810 & -0.15813 \\
0 & -96 & -0.15533 & -0.11074 & -0.23083 & -0.24598 \\
\hline
0.74560 & 3.10668 & -1.5625 & 0 & 0 & 0 \\
1.91034 & 3.32233 & 0 & -2.04082 & 0 & 0 \\
2.45258 & 5.77078 & 0 & 0 & -1.77778 & 0 \\
2.37202 & 3.6898 & 0 & 0 & 0 & -1.38408
\end{array}\right]
$$

위 행렬식의 $J(1, 1)=-44.6$, $J(2, 2)=-96$은 첫번째 반복단계에서 제약조건의 허용 가능량 중에서 여유분을 뜻한다.

마지막 반복단계에서(7번째)의 Jacobian행렬식은

$$
(\text{Jacobian}) = \left[\begin{array}{cc|cccc}
-9.79\text{E-}3 & 0 & -0.564 & -0.756 & -0.694 & -0.776 \\
0 & -0.014 & -0.436 & -0.244 & -0.304 & -0.224 \\
\hline
0.745 & 3.106 & -1.0005 & 0 & 0 & 0 \\
1.910 & 3.322 & 0 & -1.001 & 0 & 0 \\
2.452 & 5.770 & 0 & 0 & -1.001 & 0 \\
2.372 & 3.689 & 0 & 0 & 0 & -1.001
\end{array}\right]
$$

$J(1, 1)=-0.00979$, $J(2, 2)=-0.014$가 되어 여유분이 거의 없고 이용 가능량을 벗어나지 않는다.

2) 반복단계의 중단에 대한 고찰

$|\Delta R_i|$나 함수 (f_R, f_λ)에 대한 기준에 의한 반복단계의 중단은 〔표1〕과 〔표2〕에서 볼 수 있는 바와 같이 같은 반복단계에서 끝난다. 그렇지만, 7번째 반복단계 이전에서 좀 더 정확한 해를 구하려면 (f_R, f_λ)에 의한 반복중단 기준이 유리하다는 것을 알 수 있다.

Table 1. Correction to R_i and λ_i at each iteration

Iteration	ΔR_1	ΔR_2	ΔR_3	ΔR_4
1	−0.156	−0.206	−0.181	−0.122
2	−0.040	−0.083	−0.061	−0.025
3	−2.34 E-3	−9.15 E-3	−5.59 E-3	−1.60 E-3
4	−4.83 E-5	−1.23 E-4	−1.03 E-4	−4.30 E-5
5	−9.05 E-5	5.30 E-5	−2.43 E-5	9.26 E-5
6	2.60 E-5	−1.85 E-5	3.93 E-6	−3.09 E-5
7	4.41 E-7	−9.90 E-7	−5.55 E-6	−1.36 E-6

Table 2. (f_R, f_λ) evaluated at the beginning of the interations

Iteration	FR(1)	FR(2)	FR(3)	FR(4)	FL(1)	FL(2)
1	0.211	0.376	0.251	0.115	0.446	0.96
2	0.040	0.096	0.063	0.021	0.042	0.091
3	1.78E-3	8.55E-3	4.43E-3	7.04E-4	3.99E-3	8.45E-3
4	5.55E-6	8.55E-5	3.16E-5	2.58E-6	9.09E-5	2.36E-4
5	2.67E-8	3.40E-8	2.17E-8	1.44E-8	−1.55E-5	1.83E-5
6	3.68E-8	2.45E-8	1.99E-8	2.12E-8	2.53E-5	−7.23E-7
7	4.01E-9	2.66E-8	1.04E-9	−2.83E-8	1.96E-6	5.18E-7

Table 3. Jacobian submatrix J_1, value of system reliability and slacks at the each iteration

Iteration	System Reliability	Slack on resources		Jacobian(1, 1)	Jacobian(2, 2)
		cost	weight		
1	.357	44.6	96	−44.6	−96.
2	.785891	34.05	73.65	−34.05	−73.65
3	.979051	15.85	34.55	−15.85	−34.55
4	.997608	.907	2.475	− 0.907	− 2.475
5	.997926	−0.144	.229	0.144	− 0.229
6	.997895	0.104	− 0.038	− 0.104	0.038
7	.997914	9.79E-3	0.0139	− 9.79 E-3	− 0.0139

하부행렬식 J_1원소의 부호 변화를 기준으로 하는 반복중단 방법은 위의 두 방법보다는 일관성이 없다. 다음 〔표3〕에서 볼 수 있는 바와 같이 J_1의 부호가 변화되는 반복단계나 전후의 반복단계에서 체계의 신뢰성이 증가되거나 제약조건의 이용 가능량이 초과됨을 알 수 있다.

3) 최종 결과치는 다음과 같다.

i) 각 직렬단계에서 병렬중복수는 (5.11, 6.30, 5.23, 3.90)이다. 즉, 병렬중복수는 (5, 6, 5, 4)라는 결론을 내린다. 이때의 체계의 신뢰성은 0.9974702, 비용에 대한 여유분은 1.2, 무게에 대한 여유분은 3이다.

ii) 반복단계 기준으로 본 논문에서는 (f_R, f_λ)를 사용했지만 하부 행렬식 J_1이나 $|\Delta R_i|$를 기준으로 사용할 수 있다. 그러나 위의 예에서 볼 수 있었던 바와 같이 (f_R, f_λ) 기준을 사용하므로서 일관성 있게 빨리 끝낼 수 있다.

iii) 마지막 단계에서 λ값이 $(-2.002$ E-4,

−3.7177 E-5)이 되어 가능한 해의 영역內에 있다.

iv) 반복 단계에서 수정항의 크기를 결정하는 ν 값이 다를 때에도 체계의 신뢰성은 비슷하며 이에서 얻어진 병렬중복수도 (5, 6, 5, 4)로 같다. 그렇지만 ν값이 작아질수록 반복단계의 수가 커져서 $\nu=0.9$일때 8단계, $\nu=0.8$일때 10단계, $\nu=0.7$일때 12단계로 되었다. 또한 ν값이 작아질수록 $|\Delta R_j|$에 의한 반복중단 기준보다는 (f_R, f_λ)에 대한 반복단계 기준이 유리한 것으로 나타났다.

V. 결언

역사적 고찰에서도 볼 수 있었던 바와 같이 체계의 신뢰성을 극대화하려는 시도는 많이 있어 왔지만, 최소 경로 방법을 이용한 Network 체계에 대한 논문이 많았던 반면에 비교적 간단한 직렬 다단계, 병렬중복을 포함하는 체계에 대해서는 많지 않았다. 이에 본 논문에서는 이에 대한 Computer Programming을 작성하여 신뢰성 설계에 도움이 되고자 하였다.

그러나, 다음 연구과제로는 병렬중복에서 본 논문과 달리 대기중복으로 처리할 경우의 체계 신뢰성 극대화 문제이다.

《參考文獻》

(1) Krishma B. Misra, *"Reliability optimization of a series parallel system"*, IEEE Trans. on Rel., VOL. R−21, NO. 4, pp 230−238, November 1972

(2) Nancy R. Mann, *"Approximately optimum confidence bounds on series - and parallel system reliability for systems with binomial subsystem data,* "IEEE Trans. on Rel., VOL. R-23, NO. 5,pp 295−304, December 1974

(3) Frank A. Tillman, Ching - Lai Hwang, Way Kuo, *"Determining component reliability and redundancy for optimum system reliability"*, IEEE Trans. on Rel., VOL. R-26, NO. 3, pp 162−165, August 1977

(4) P. M. Lin, B. J. Leon, T. C. Huang,*"A new algorithm for symbolic system reliability analysis,"* IEEE Trans. on Rel., VOL. R-25, No. 1, pp 2−15, April 1976

(5) J. deMercardo, N. Spyratos, B. A. Bowen, *"A method for calculation of network reliability,"* IEEE Trans. on. Rel., VOL. R-25, NO. 2, pp 71−76, June 1976

(6) A. Satyanarayana, A. Prabhakar, *"New topological formula and rapid algorithm for reliability analysis of complex network."* IEEE Trans. on Rel., VOL. R-27, NO. 2, pp 82−100, June 1978

(7) Ashok Kumar, Manju Agarwal, *"A review of stanby redundant system"*, IEEE Trans. on Rel., VOL. R-29, NO. 4, pp 290−294, October 1980

(8) Frank A. Tillman, Ching-Lai Hwang, Way Kuo, *"Optimization techniques for system reliability with redundancy-A review,"* IEEE Trans. on Rel., VOL. R-26, NO. 3,pp 148−155, August 1977

(9) William I. Zangwill, *Nonlinear Programming: A unified approach*, pp 44−50, 1969, Prentice-Hall Inc.

(10) Thomas L. Saaty, Joseph Bram, *Nonlinear mathematics,* pp 56−70, 1964, McGraw-Hill Book Co.