

유전자 알고리즘을 이용한 (m,n) 중-연속 (r,s) :고장 격자 시스템의 정비 모형

윤원영* · 김귀래 · 정철훈

부산대학교 산업공학과

A Maintenance Design of Connected- (r,s) -out-of- (m,n) :F System Using Genetic Algorithm

Won-Young Yun · Gui-Rae Kim · Cheol-Hun Jeong

Department of Industrial Engineering, Pusan National University, Busan, 609-735

This study considers a linear connected- (r,s) -out-of- (m,n) :F lattice system whose components are ordered like the elements of a linear (m,n) -matrix. We assume that all components are in the state 1 (operating) or 0 (failed) and identical and s -independent. The system fails whenever at least one connected (r,s) -submatrix of failed components occurs. The purpose of this paper is to present an optimization scheme that aims at minimizing the expected cost per unit time. To find the optimal threshold of maintenance intervention, we use a genetic algorithm for the cost optimization procedure. The expected cost per unit time is obtained by Monte Carlo simulation. The sensitivity analysis to the different cost parameters has also been made.

Keywords: connected- (r,s) -out-of- (m,n) :F lattice system, age replacement, genetic algorithm, Monte Carlo simulation

1. 서론

Kontoleon(1980)이 Consecutive- k -out-of- n :F 시스템을 처음으로 제안한 후부터 이 특수한 시스템에 많은 관심이 집중되어 왔다. 이 시스템은 n 개의 부품으로 이루어진 선형 혹은 원형 시스템으로 연속적인 k 개의 부품이 고장나면 시스템이 고장나게 되는 형태를 말한다. 여기서 모든 부품들은 작동과 고장 두 가지의 상태를 가지며 k 가 1일 경우 직렬 시스템이 되고, k 가 n 일 경우 병렬 시스템이 된다. 대표적인 예로 통신 네트워크, 파이프라인 시스템, 거리의 가로등 등을 들 수 있다.

(m,n) 중-연속 (r,s) :고장 격자 시스템(Connected- (r,s) -out-of- (m,n) :F lattice system)은 Consecutive- k -out-of- n :F 시스템의 2차원 형태의 시스템이다. (m,n) 행렬과 같이 배열되어 있는 부품들 중 (r,s) 의 부분 행렬에 속한 부품들이 모두 고장일

경우 시스템의 고장으로 정의된다. 이러한 2차원의 Consecutive- k -out-of- n :F 시스템은 근래 들어 많은 분야에서 활용되고 있다. 안전을 담당하는 모니터 시스템, 전자장치를 연결하는 연결 커넥터 그리고 각종 질환을 진단하는 X-ray 등이 가장 대표적인 예이다.

선형/원형 (m,n) 중-연속 (r,s) :고장 격자 시스템은 Salvia and Lasher(1990)에 의해 처음으로 소개되었는데, 그들은 m 과 n 이 동일하고 r 과 s 가 동일하다고 가정하였다. 즉, (n,n) 중-연속 (r,r) :고장 격자 시스템에 대하여 시스템의 신뢰도를 연구하였다. 이 시스템을 Boehme, Kossow and Preuss(1992)가 선형/원형 (m,n) 중-연속 (r,s) :고장 격자 시스템으로 새로이 정의하여 일반화시키고 시스템 신뢰도를 계산하였다.

그 후 시스템 신뢰도를 계산하는 효율적인 알고리즘을 위한 연구들이 수행되었다. Zuo(1993)는 Abraham(1979), Locks(1987),

본 연구는 한국과학재단 목적기초연구(R05-2002-000-00995-0) 지원으로 수행되었음.

* 연락저자 : 윤원영 교수, 609-735 부산광역시 금정구 장전동 산 30번지 부산대학교 산업공학과, Fax : 051-512-7603,

E-mail : wonyun@pusan.ac.kr

2003년 11월 접수; 2004년 3월, 2004년 4월 수정본 접수; 2004년 7월 게재 확정.

Wilson(1990)에 의해 연구되었던 SDP(Sum of disjoint products) 방법론을 이용하여 시스템의 신뢰도를 계산하였다

Yamamoto and Miyakawa(1995)는 YM 알고리즘이라 불리는 시스템 신뢰도에 대한 순환식(Recursive formula)을 제안하였다. 또한 Noguchi, Sasaki, Yanagi and Yuge(1996)가 시스템 신뢰도 계산을 위한 또 다른 순환식(Recursive formula)을 제안하였다.

(m,n) 중-연속 (r,s) :고장 격자 시스템의 신뢰도에 대한 상한치(Upper bound)와 하한치(Lower bound)에 대한 연구 역시 많이 수행되었다. 예를 들면, Salvia and Lasher(1990), Ksir(1992), Fu and Koutras(1994), Yamamoto and Miyakawa(1995,1996), Makri and Psillakis(1996), Boutsikas and Koutras(2000), Koutras, Papadopoulos and Papastavridis(1993, 1997) 그리고 Malinowski and Preuss(1996)등의 논문들이 있다.

이러한 다부품 시스템(Multi-component system)에 대한 예방정비는 시스템 신뢰도 및 운영비용 관점에서 볼 때 아주 중요하다. 만약 예방정비를 하지 않으면 시스템의 신뢰도를 떨어뜨리게 됨으로써 시스템이 자주 고장나게 되어 시스템 정지로 인한 큰 손실을 불러올 수 있다. 반면, 빈번한 정비는 고신뢰도 시스템을 유지하게 하지만 예방정비 자체에 과다하게 비용이 지출되어 오히려 비용증대가 이루어질 수 있다. 따라서 가장 경제적인 예방정비 시점을 결정하는 것은 매우 중요한 문제이다. Yun and Kim (2000)이 경제적 설계문제를 Consecutive- k -out-of- n :F 시스템에 처음으로 적용하였다.

본 논문에서는 부품 간의 고장이 독립이고 동일한 부품으로 구성된 (m,n) 중-연속 (r,s) :고장 격자 시스템에서 단위시간당 기대비용을 최소로 하는 최적 예방정비 기간을 결정하고자 한다. 목적함수인 단위시간당 기대비용은 해석적으로 계산하기 어려우므로 시뮬레이션을 통하여 추정하고 이를 유전자알고리즘(Genetic algorithm)을 통하여 최적화한다. 또한 수치실험들을 통하여 다양한 모수들의 영향을 평가한다

우선 2장에서는 대표적인 예제 시스템을 통하여 (m,n) 중-연속 (r,s) : 고장 격자 시스템을 소개한다. 이어 3장에서는 이러한 시스템의 단위시간당 기대비용 모형을 제시하고 이를 최

적화하기 위한 유전자 알고리즘을 소개하며 4장에서는 다양한 수치실험들의 결과를 통하여 다양한 모수들의 영향을 분석한다. 마지막으로 5장에서 결론을 다룬다.

2. 시스템 모형 및 정비 정책

2.1 (m, n) 중-연속 (r, s) :고장 격자 시스템

선형 (m, n) 중-연속 (r, s) :고장 격자 시스템은 가로 m , 세로 n 의 부품들이 행렬 (m, n) 의 요소처럼 구성되어 있는 시스템을 의미한다. 즉, 각 요소들이 m 개의 행이 각각 n 개의 요소들을 포함하고 있는 시스템이다. 이러한 시스템의 대표적인 형태로 감시 카메라 시스템(Supervision system)을 들 수 있다. 감시 카메라 시스템은 <Figure 1-a>에 잘 나타나 있다.

각각의 작은 원($C_1 \sim C_9$)은 감시 카메라를 나타낸다. 각 감시 카메라는 반지름이 c 인 원형 지역을 감시할 수 있으며, 이것은 카메라의 성능을 의미한다.

모든 카메라는 동일한 성능을 가진 것으로 가정하며 각 카메라간의 거리는 d 로 동일하다. 이 감시 카메라 시스템의 고장은 몇몇 카메라들의 고장으로 인하여 감시되지 않는 영역이 생기는 것으로 정의한다. 만약 각 감시 카메라 성능이 $\sqrt{2}d$ 라고 할 경우 <Figure 1-b>와 같이 가로와 세로가 각각 2대씩 연속적으로 카메라가 고장났을 경우 시스템 고장이 발생하며 이는 선형(3,3)중-연속(2,2):고장 격자 시스템이 된다.

2.2 고장 발생 및 정비 정책

선형 (m,n) 중-연속 (r,s) :고장 격자 시스템의 고장은 (m,n) 개의 부품들 중에서 (r,s) 의 격자 내의 모든 부품들이 고장나면 시스템의 고장으로 정의한다. 각 시스템을 구성하는 부품들은 지수분포(Exponential distribution)를 따라 고장이 발생하고 모두 동일한 부품으로 구성되어 있다고 가정한다

시스템을 운용함에 따라 각 부품들은 고장이 나기 마련이며

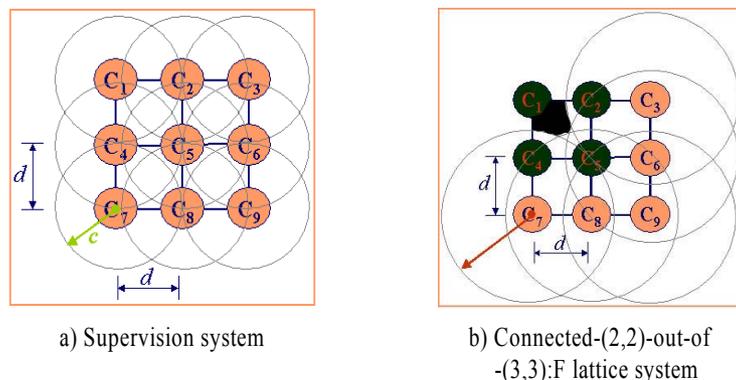


Figure 1. Supervision system.

따라서 정비가 필요하다. 시스템 정비정책은 시기와 방법에 따라 여러 가지로 나뉘 볼 수 있으며 본 연구에서는 <Figure 2> 와 같이 시스템이 고장나거나 혹은 일정 수명이 되면 정비를 실시하는 수명교체 정비정책(Age replacement policy)을 사용한다.

또한 정비방법은 시스템 내의 고장난 부품을 찾아 고장난 부품들만 새것으로 교체한다. 이는 각 부품들의 수명이 지수 분포를 따르므로, 정비 시 고장나지 않은 부품은 새것과 같은 상태이므로 예방교체를 할 필요가 없기 때문이다 비용 측면에서 사후 정비비용이 예방 정비비용보다 큰 것으로 가정한다

본 논문에서의 기본 가정은 다음과 같다

기본 가정

1. 시스템의 모든 부품의 수명은 서로 독립이고 동일한 지수분포를 따른다.
2. 시스템 및 부품은 작동과 고장의 2가지 상태를 가진다.
3. 시스템은 수명교체 정책에 따라 예방정비되며 정비 시 고장난 부품들만 교체한다.
4. 교체시간은 무시할 수 있다.

기호

- T_R : 수명교체 주기
- $R(t), F(t)$: 시스템 신뢰도, 분포함수
- $L(T_R)$: 한 주기의 기대길이
- $N(T_R)$: 한 주기당 기대 고장부품 수
- $CT(T_R)$: 한 주기의 총 기대비용
- $C(T_R)$: 단위시간당 기대비용
- C_0 : 시스템 고장 시 소요되는 고정비용
- C_1 : 고장난 부품을 교체하는 데 드는 비용
- C_2 : 예방정비 시 소요되는 고정비용

3. 시스템 최적 설계

3.1 비용함수

만약 계획된 예방정비 기간까지 시스템이 고장나지 않으면 예방정비가 실시되고, 이 때 예방정비에 드는 비용은 다음의

비용모수에 의해 발생된다.

C_1 = 한 부품당 교체비용

C_2 = 회당 예방정비비용

만약 예방정비 주기 이전에 시스템이 고장나면 사후정비가 실시되고 정비비용은 아래의 비용모수들에 의해 발생된다

C_1 = 한 부품당 교체비용

C_0 = 회당 시스템 고장비용

예방정비에서나 사후정비에서나 고장난 부품의 한 단위당 교체비용은 C_1 으로 동일하며 $C_0 > C_2$ 라고 가정한다.

부품의 작동비용은 시간과 독립되므로 단위시간당 기대비용은 Blanks and Tordan (1986) 그리고 Jardine(1973)에 의해 단위시간당 기대비용은 식(1)과 같이 주어진다.

$$C(T_R) = \frac{\text{한 주기에서의 총 기대비용}}{\text{한 주기의 기대길이}} \quad (1)$$

한 주기에서의 총 기대비용은 다음과 같다

$$CT(T_R) = C_1N(T_R) + C_0F(T_R) + C_2R(T_R) \quad (2)$$

또한 한 주기의 기대길이는 일반적으로

$$L(T_R) = \int_0^{T_R} R(t)dt \quad (3)$$

주어지므로, 따라서 단위시간당 기대비용은 식(4)와 같다.

$$C(T_R) = \frac{C_1N(T_R) + C_0(1 - R(T_R)) + C_2R(T_R)}{L(T_R)} \quad (4)$$

3.2 시뮬레이션

선형(m, n)중-연속(r, s):고장 격자 시스템의 신뢰도를 정확하게 계산하는 것에 대한 연구가 많이 수행되었지만 큰 시스템인 경우 많은 시간과 노력을 요한다. 따라서 한 주기의 기대 길이나 주기당 기대 고장부품 수를 해석적으로 계산하는 것은

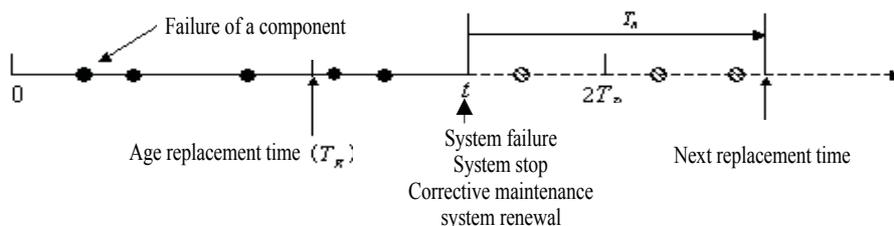


Figure 2. Age replacement policy.

매우 힘든 일이다. 따라서 본 연구에서는 몬테 카를로 시뮬레이션(Monte Carlo simulation)을 통하여 단위시간당 기대비용을 추정한다. 시뮬레이션의 간략한 절차는 아래와 같다.

- 단계 1: 의사결정변수와 관련 모수들을 초기화한다
- 단계 2: 작동중인 부품들의 고장을 발생시킨다.
- 단계 3: 시스템 고장을 확인한다. 만약 고장이면 단계 4로 간다. 만약 시스템 고장이 아니면 단계 5로 간다.
- 단계 4: 통계량을 갱신한 후 단계 6으로 간다.
- 단계 5: 예방정비 실행을 확인한다. 만약 예방정비가 실행되면 단계 4로 가고, 만약 예방정비가 실행되지 않으면 단계 2로 간다.
- 단계 6: 시뮬레이션 종료조건을 확인한다. 만약 시뮬레이션이 종료조건에 해당하면 통계량을 출력하고 종료한다 만약 그렇지 않으면 시스템을 재생(Renewal)시키고 단계 2로 간다.

상세한 절차는 부록(Appendix)에 나타내었다.

시뮬레이션을 이용하여 여러 상황에서 단위시간당 기대비용을 추정하여 보면 <Figure 3>과 같이 정비시간에 대한 단봉(Unimodal)함수 형태를 띠는 경우들이 다수 있으나 이를 증명할 수는 없고 또한 목적함수의 값을 시뮬레이션으로 추정하는 상황이므로 본 연구에서는 국소해에 빠지는 위험을 줄일 수 있으며 효율적으로 해를 찾는 유전자 알고리즘을 사용하여 정비주기를 결정하고자 한다.

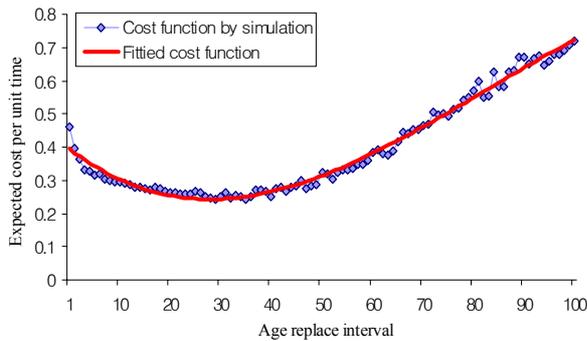


Figure 3. Cost function.

3.3 유전자 알고리즘

3.3.1 유전자 알고리즘 설계

단위시간당 기대비용을 최소로 하는 최적 교체주기 T_R 을 구하고자 한다. 그러나 단위시간당 기대비용이 복잡함에 따라 이를 해석적으로 최적화하기는 힘들다. 따라서 본 연구에서는 비록 결정변수가 최적 교체주기 하나이지만 그 효율이 증명되어 있고 또한 확장과 재사용이 유용한 메타 휴리스틱(Meta heuristic)의 일종인 유전자 알고리즘(Genetic algorithm)

을 사용한다. 본 논문에 사용 된 유전자 알고리즘은 아래와 같은 세 가지 요소로 구성된다(Malanie(1996)).

- **선별(Selection)** : 선별은 적자생존의 자연법칙에 근거하여 목적함수 값에 의한 적합도(Fitness)평가에 의해 다음 세대에 생존할 개체를 찾는 과정이다. 적합도함수(Fitness function)는 본 연구의 목적함수인 단위시간당 기대비용 식(4)를 그대로 사용하였다. 선별방법은 엘리트(Elite) 방식을 적용하였다.
- **교차(Crossover)** : 교차는 변이와 함께 유전 연산자의 하나로써 두 부모가 갖는 유전자를 조합하여 자손을 생산하는 과정이다. 본 논문에서는 일점균등교차(One-point uniform crossover)방법을 사용하였다.
- **변이(Mutation)** : 이 연산자는 확률적으로 염색체의 한 유전자를 변화시킨다. 본 논문에서는 돌연변이점을 임의로 선정하여 그 지점의 염색체를 새로운 값으로 대체하는 균등돌연변이(Uniform mutation)방법을 사용하였다.

본 논문에서 사용된 염색체(Chromosome)의 구성은 <Figure 4>와 같다. 한 염색체는 5개의 유전자로 구성되어 있다. 즉, 0-9까지의 정수 값을 가진 5비트(bit)의 문자열(string)이다.

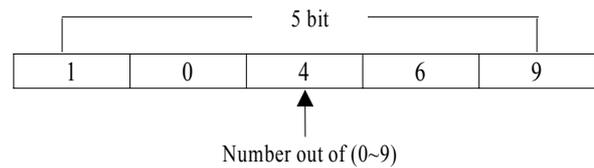


Figure 4. An example of decoding.

하나의 염색체는 소수점 세 자리를 가진 유리수로 환산된다. 예를 들어, 위 염색체 '10469'의 수명 교체주기는 <Figure 5>와 같이 계산된다.

1	0	4	6	9
---	---	---	---	---

$$\begin{aligned}
 T_R &= 1 \times 10^2 + 0 \times 10^1 + \\
 &\quad 4 \times 10^{-1} + 6 \times 10^{-2} + 9 \times 10^{-3} \\
 &= 10.469
 \end{aligned}$$

Figure 5. An example of decoding.

새로이 생성된 염색체는 '00000'인 경우가 발생할 수 있다. 이 경우 교체주기가 연속이라는 의미인데 본 연구에서는 이러한 경우는 고려하지 않는다. 따라서 새로운 염색체를 생성할 때 이 경우가 발생하지 않도록 조정하였다. 유전자 알고리즘의 개략적인 절차는 다음과 같고 자세한 절차는 <Figure 6>과 같다.

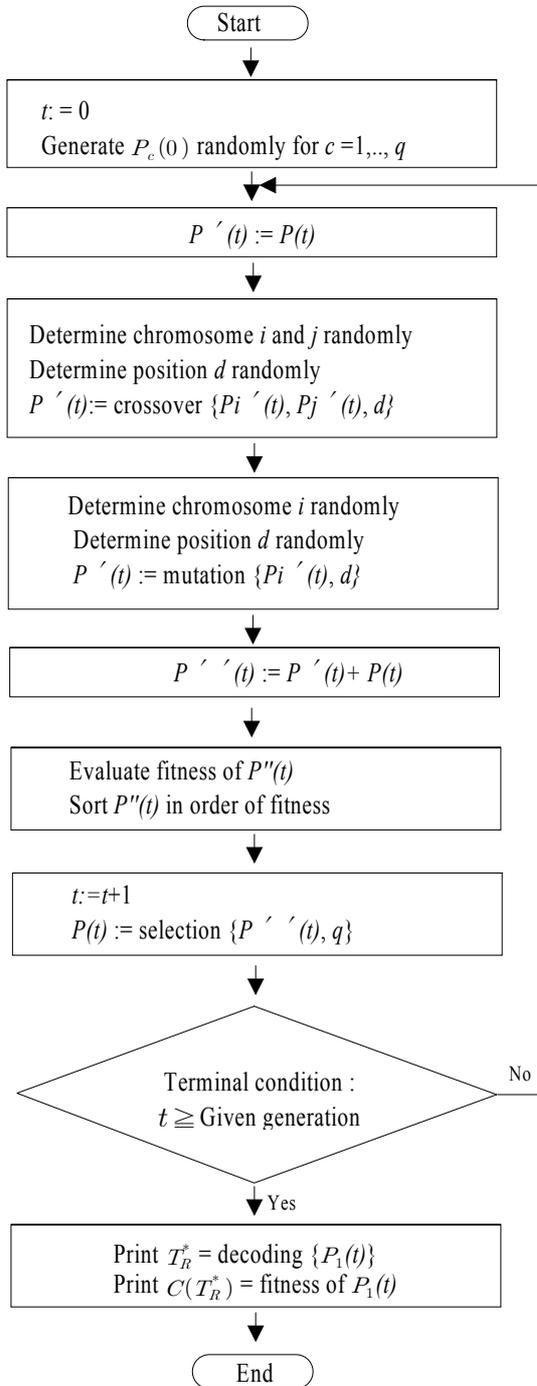


Figure 6. Procedure of genetic algorithm.

- 단계 1: 랜덤하게 생성된 x 개 염색체의 개체집단을 가지고 시작한다. 여기서 x 개의 염색체는 각각 5개의 비트, 즉 유전자로 구성되어 있다. 총 세대 수는 100으로 하였다.
- 단계 2: 개체집단에서 각 염색체의 적합도를 계산한다. 이러한 적합도는 몬테 카를로 시뮬레이션을 이용하여 추정한다
- 단계 3: x 개의 자손이 생길 때까지 다음의 과정을 반복한다.
- 현재 개체집단으로부터 개체의 적응도에 비례하는 개체

- 를 선별한다.
 - 한 쌍의 부모에서 교차율을 가지고 임의로(Random) 선택된 지점에서 일점균등교차를 실시한다.
 - 변이확률을 가지고 각 위치에서 균등 돌연변이를 실시한 후 개체집단에 포함시킨다.
 - 현재 개체집단을 새로운 개체집단으로 교체한다(오름차순으로 정렬된 $2r$ 개의 개체 중 x 순위까지 선별).
- 단계 4: 단계 2로 돌아간다.

3.3.2 유전자 알고리즘 모수 설계

본 연구의 시뮬레이션에서 입력받는 모수의 값은 시스템 모수인 시뮬레이션 실험횟수, 시스템 모수인 m, n, r, s , 그리고 비용모수들 C_0, C_1, C_2 과 유전자 알고리즘 모수인 염색체 수, 세대 수, 교차율, 돌연변이율로 모두 12가지이다. 이러한 모수들 중에서 시뮬레이션 시험횟수와 유전자 알고리즘의 모수들은 수치실험에서의 해의 질과 계산시간에 영향을 미친다. 따라서 이 모수들을 어떻게 결정할 것인가가 중요한 문제가 된다

다양한 모수들에 대하여 실험을 해 본 결과 교차율과 돌연변이율은 별 영향을 주지 않았다. 따라서 본 연구에서는 각각 0.3, 0.05로 고정하였다. 목적함수 값에 가장 큰 영향을 미치는 모수는 '염색체 수'였으며 또한 시뮬레이션에서의 실험횟수와 유전자 알고리즘의 세대 수도 영향을 주었다

시뮬레이션 실험횟수 결정

실험의 횟수가 많아질수록 실험결과의 정밀도는 높아지지만 실험시간이 많이 늘어나는 것은 당연하므로 우선 가장 적절한 실험횟수를 결정하였다. 교차율은 0.3, 돌연변이율은 0.05이며 실험횟수는 10회부터 500회까지 실험을 실시하였고, 그 결과는 아래의 <Table 1>과 같다. <Table 1>에서 초반에 실험 횟수가 증가할수록 $C(T_R)$ 값이 변화하다가 약 100회부터 $C(T_R)$ 값이 큰 변화가 없음을 알 수 있다. 따라서 적정 실험 횟수를 100회로 결정하였다.

Table 1. $C(T_R)$ versus simulation replication number

$(m,n)(r,s) : (4,4)(3,3)$				
$C_0, C_1, C_2 : 100, 1, 0.1$				
	Population	Generation	Population	Generation
	100	100	50	100
Number	$C(T_R)$		$C(T_R)$	
10	0.1646		0.1740	
30	0.1997		0.1686	
50	0.1894		0.1625	
70	0.1926		0.2002	
100	0.2202		0.2136	
150	0.2211		0.2111	
200	0.2269		0.2298	
300	0.2182		0.2197	
500	0.2295		0.2123	

염색체 수 결정

염색체 수는 10개부터 200개까지 실험을 실시하였으며 결과를 <Table 2>에 정리하였다. <Figure 7>은 염색체 수에 따른 $C(T_R)$ 과 T_R 의 값을 타점한 결과이다. 염색체 수가 50개 이상의 평균값은 $C(T_R)$ 과 T_R 이 각각 34.792와 0.2020이며, 이 값은 염색체 수가 50개 이후부터 차이가 거의 없다는 것을 보여준다. 따라서 적정 염색체 수를 50개로 결정하였다.

Table 2. $C(T_R)$ and T_R versus size of population

$(m,n)(r,s) : (4,4)(3,3)$		
$C_0, C_1, C_2 : 100, 1, 0.1$		
Generation : 100		
size of population	T_R	$C(T_R)$
10	28.765	0.2260
30	34.287	0.2170
50	34.624	0.2034
70	34.196	0.1989
100	34.557	0.2040
150	35.616	0.2036
200	34.967	0.2000

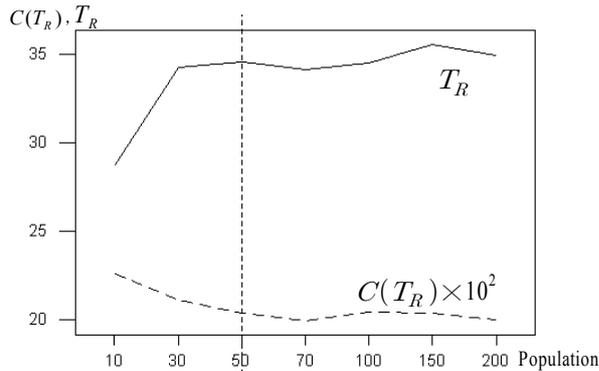


Figure 7. $C(T_R)$ and T_R versus size of population.

세대 수 결정

<Figure 8>은 모집단의 크기가 50개와 100개일 때 세대 수에 따른 단위시간당 기대비용의 변화를 보여준다. 모집단의 크기가 50개인 경우 100세대에서 안정화되었으며 그 이후는 변화가 없음을 알 수 있다. 따라서 세대 수는 100세대로 결정하였다.

3.3.3 예제

3.3.1절에서의 절차를 이용하여 간단한 예를 보이코자 한다. 선별은 엘리트 방식을 사용하고 일점균등교차와 균등돌연변이를 사용하고 필요한 모수들은 <Table 3>과 같다. 간략하게 절차를 설명하기 위하여 작은 모집단으로 2세대까지만 전개를 설명한다. 본 연구에서는 이와 같은 과정을 100세대까지 전개하였다.

$C(T_R)$

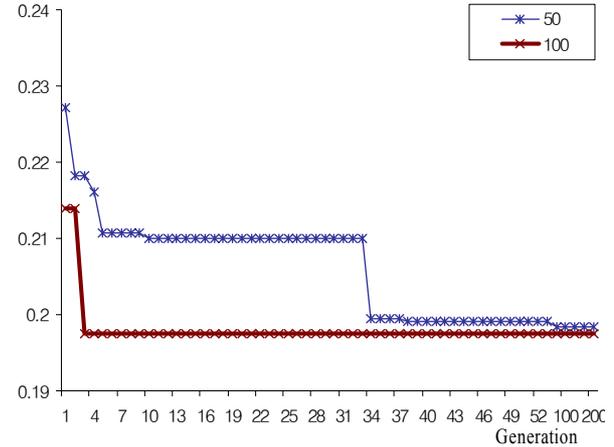


Figure 8. $C(T_R)$ versus generation number.

Table 3. Parameters of an example

System : consecutive-(3,3)-out-of-(4,4):F lattice system	
Failure distribution : Exp(0.02)	
Cost parameters : C_0 100	
C_1 1	
C_2 0.1	
Genetic algorithm parameters :	
size of population	5
crossover rate	0.3
mutation rate	0.05

0세대

단계 1 : (초기화)

$v1=(75523), v2=(70787), v3=(04873)$
 $v4=(98093), v5=(57054)$

단계 2 : (적합도 평가)

$C(v1)=0.590, C(v2)=0.427, C(v3)=0.353$
 $C(v4)=0.694, C(v5)=0.394$

1세대

단계 3-a : (선별)

모집단 중 적합도가 가장 작은 순서부터 5개를 선택하면 다음과 같이 선택된다.

$v1'=(04873), v2'=(57054), v3'=(70787)$
 $v4'=(75523), v5'=(98093)$

단계 3-b : (교차)

교차 대상자를 정하기 위해 각 염색체마다 난수를 발생하면

1 : 0.1558 2 : 0.4813 3 : 0.7903
 4 : 0.8374 5 : 0.0288

1번과 5번이 교차율보다 작으므로 교차 대상자로 선택된다. [1,4]사이의 정수난수를 발생시켜 절단점을 결정하면 3이 되어 새로운 개체는 다음과 같다.

$$\begin{matrix} v1'=(04873) & \rightarrow & v6'=(98073) \\ v5'=(98093) & & v7'=(04893) \end{matrix}$$

따라서 전체 모집단은 다음과 같다.

$$\begin{matrix} v1'=(04873), v2'=(57054), v3'=(70787) \\ v4'=(75523), v5'=(98093), v6'=(98073) \\ v7'=(04893) \end{matrix}$$

단계 3-c : (돌연변이)

위 7개의 염색체를 가지고 돌연변이 연산을 실시한다. 돌연변이가 일어날 개체를 선정하기 위해 난수를 발생하면 다음과 같다.

$$\begin{matrix} 1 : 0.3078 & 2 : 0.9001 & 3 : 0.3630 \\ 4 : 0.2187 & 5 : 0.1225 & 6 : 0.8253 \\ 7 : 0.5007 \end{matrix}$$

돌연변이율보다 적은 난수가 하나도 발생하지 않았으므로 돌연변이가 발생하지 않는다. 이 예제에서는 모집단의 수가 5개로 작아 돌연변이가 발생하지 않았다. 모집단의 수가 50개라면 평균 2.5개의 돌연변이가 발생한다. 예제에서는 진행과정을 설명하기 위하여 7번 염색체가 돌연변이를 위해 선택되었다고 가정하면 돌연변이가 일어날 유전자를 선택하기 위하여 각 비트에 대해 난수를 발생하여 돌연변이율보다 작은 비트를 선택한다. 그 결과 1번 비트와 4번 비트가 선택되었다면 각 비트에서 랜덤하게 [0-9] 사이의 값을 선택한다. 그 결과는 다음과 같다.

$$v2'=(04893) \rightarrow v8'=(34813)$$

따라서 전체 모집단은 다음과 같다.

$$\begin{matrix} v1'=(04873), v2'=(57054), v3'=(70787) \\ v4'=(75523), v5'=(98093), v6'=(98073) \\ v7'=(04893), v8'=(34813) \end{matrix}$$

단계 2 : (적합도 평가)

$$\begin{matrix} C(v1')=0.353, C(v2')=0.394, C(v3')=0.427 \\ C(v4')=0.590, C(v5')=0.694, C(v6')=0.689 \\ C(v7')=0.361, C(v8')=0.263 \end{matrix}$$

2세대

단계 3-a : (선별)

모집단 중 적합도가 가장 작은 순서부터 5개를 선택하면 다음과 같이 선택된다.

$$\begin{matrix} v1''=(34813), v2''=(04873), v3''=(04893) \\ v4''=(57054), v5''=(70787) \end{matrix}$$

4. 실험결과

시스템에 있는 모든 부품의 고장은 서로 독립이며 각 부품의 고장분포는 $\lambda=0.02$ 인 지수분포를 따른다. 정비정책은 수명교체 정책을 이용하며 단위시간당 기대비용이 최소가 되는 정비주기를 결정하였다. 시뮬레이션 프로그램의 입력 값은 유전자 알고리즘(Genetic algorithm) 모수(Parameter)와 시스템 모수이며 출력 값은 기간 정비주기(T_R)와 기대비용($C(T_R)$)이다. (m, n)은 (5,5)와 (10,10)의 두 가지 경우로 실험을 실시하였으며 (r, s)는 모든 실험에 (3,3)으로 고정하였다. C_1 은 1로 고정하였으며 $C_0 = 0.1, 0.5, 1, C_2 = 10, 50, 100, 1000$ 의 12가지 경우에 대해 실험을 시행하였다. 유전자 알고리즘의 모든 모수들은 3.2절에서 설명한 값과 같다.

각 비용모수에 따른 선행(5,5)중-연속(3,3):고장 격자 시스템의 최적 정비주기와 단위시간당 기대비용을 <Table 4>에 나타내었다.

Table 4. $C(T_R)$ and T_R versus cost parameters for connected-(3,3)-out-of-(5,5):F lattice system

$(m,n)(r,s) : (5,5)(3,3)$			
C_2	C_0	T_R	$C(T_R)$
0.1	10	61.043	0.2851
	50	36.524	0.3001
	100	33.921	0.3084
	1000	29.486	0.3206
0.5	10	68.836	0.2893
	50	36.756	0.3219
	100	34.472	0.3223
	1000	30.475	0.3452
1	10	80.093	0.2899
	50	37.876	0.3376
	100	34.822	0.3426
	1000	30.277	0.3640

비용모수에 다른 최적 정비주기와 단위시간당 기대비용의 변화를 살펴보자. <Figure 9>에서 알 수 있듯이 C_0 의 비용이 증가할수록 $C(T_R)$ 의 값이 증가한다. 또한 C_2 의 값이 커질수록 대체로 $C(T_R)$ 의 값이 커짐을 알 수 있다. <Figure 10>은 C_0 와 C_2 의 변화에 따른 기간 정비주기(T_R)를 타점한 결과이다. C_0 가 증가할수록 T_R 의 값은 작아지며 또한 C_2 의 값이 클수록 T_R 이 큰 값을 보임을 알 수 있다.

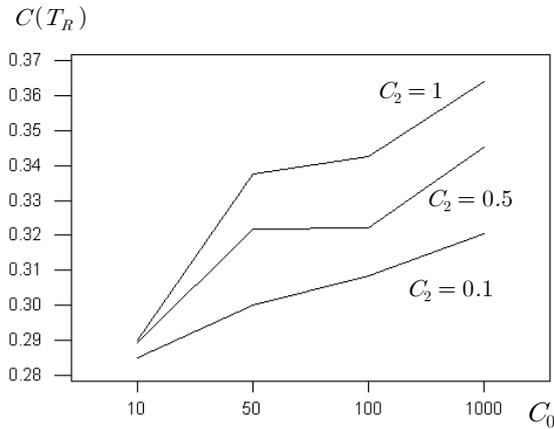


Figure 9. $C(T_R)$ versus cost parameters for onected-(3,3)-out-of-(5,5):F lattice system.

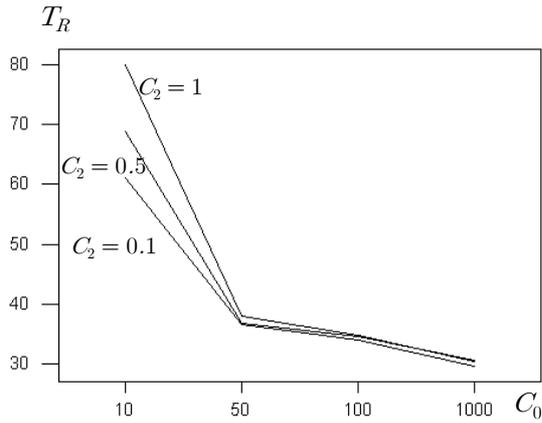


Figure 10. T_R versus cost parameters for connected-(3,3)-out-of-(5,5):F lattice system.

<Table 5>는 이 실험을 10번 반복하여 최적 정비주기와 단위 시간당 기대비용의 평균 및 분산 값을 정리한 것이다. 각 수치들의 분산이 다소 크게 나온 이유는 컴퓨터 간의 난수발생 차이에 의한 것으로 사료되며 동일한 컴퓨터 내에서의 분산은 아주 작은 값이 나왔다. $C(T_R)$ 의 값에 비해 T_R 값의 분산이 크게 나온 것은 T_R 에 대한 단위시간당 기대비용함수가 아주 완만하기 때문인 것으로 판단된다.

<Table 6>은 선형(10,10)중-연속(3,3):고장 격자 시스템에서 실험한 결과를 정리한 것이다. 선형(5,5)중-연속(3,3):고장 격자 시스템보다 시스템의 크기가 훨씬 크므로 최적 정비주기가 작아지는 당연한 결과를 보였다.

또한 앞선 실험과 마찬가지로 C_2 에 대한 C_0 의 값이 증가할수록 $C(T_R)$ 의 값이 증가하는 추세를 보이고 있다. <Figure 11>은 C_0 와 C_2 에 따른 $C(T_R)$ 의 결과를 타점한 그래프이며 <Figure 12>는 C_0 와 C_2 의 변화에 따른 기간 정비주기(T_R)의 변화를 나타낸다. 선형(5,5)중-연속(3,3):고장 격자 시스템과 비슷한 추세를 알 수 있다.

Table 5. Means and variances of the optimal maintenance interval and the expected cost rate for connected-(3,3)-out-of-(5,5):F lattice system

C_2	C_0	T_R		$C(T_R)$	
		Mean	Variance	Mean	Variance
0.1	10	61.043	31.34736	0.285	0.000348
	50	36.523	0.88436	0.300	0.000157
	100	33.921	2.94691	0.308	0.000898
	1000	29.486	9.39136	0.320	0.000686
0.5	10	68.835	40.91234	0.289	0.000189
	50	36.755	2.64172	0.321	0.000300
	100	34.471	0.66195	0.322	0.000291
	1000	30.474	2.06622	0.345	0.000434
1	10	80.093	17.27322	0.289	0.000299
	50	37.877	5.49340	0.337	0.000167
	100	34.821	0.41521	0.342	0.000185
	1000	30.276	0.48542	0.364	0.000217

Table 6. $C(T_R)$ and T_R versus cost parameters for connected-(3,3)-out-of-(10,10):F lattice system

$(m,n)(r,s) : (10,10)(3,3)$			
C_2	C_0	T_R	$C(T_R)$
0.1	10	55.057	1.3207
	50	26.550	1.4489
	100	21.371	1.5212
	1000	16.777	1.6048
0.5	10	56.214	1.3522
	50	29.430	1.5043
	100	26.486	1.5352
	1000	19.844	1.6954
1	10	57.482	1.4104
	50	30.145	1.5205
	100	26.625	1.5793
	1000	21.112	1.7138

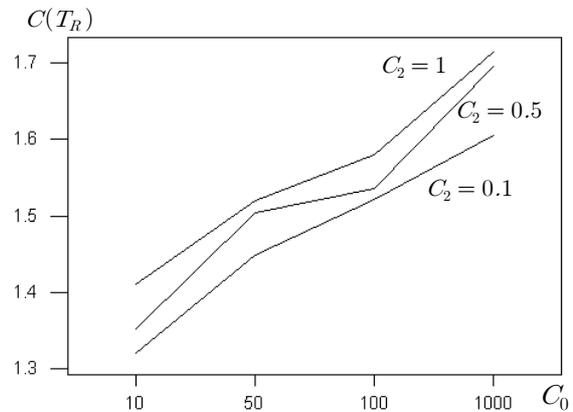


Figure 11. $C(T_R)$ versus cost parameters for connected-(3,3)-out-of-(10,10):F lattice system.

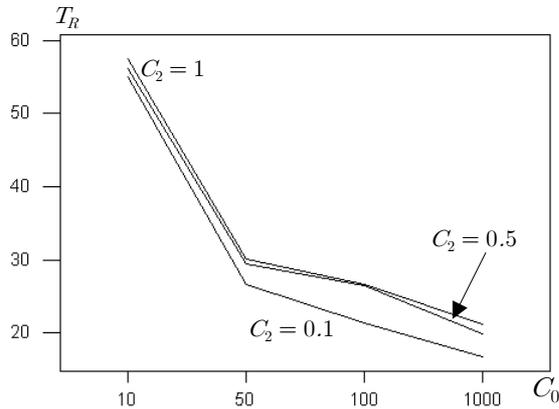


Figure 12. T_R versus cost parameters for connected-(3,3)-out-of-(10,10):F lattice system.

본 절에서는 또한 선형 (m,n) 중-연속 (r,s) :고장 격자 시스템에서 예방정비를 실시하지 않은 경우, 즉 사후정비만을 실시하였을 경우에 대해 단위시간당 기대비용을 예측하여 예방정비를 하는 경우와 비교해 보았다.

사후정비만을 수행하는 경우 시스템의 기대수명을 EL 이라 하고 평균 고장개수를 EN 으로 두면 단위시간당 기대비용함수는 아래의 식(5)와 같다.

$$C_F = \frac{C_1 EN + C_0}{EL} \quad (5)$$

앞 실험에서의 비용모수들에 대하여 선형(5,5)중-연속(3,3):고장 격자 시스템과 선형(10,10)중-연속(3,3):고장 격자 시스템에 대하여 실험을 실시한 결과는 <Table 7>에 나타내었다. <Table 4>와 <Table 6>의 결과와 비교해 보면 C_0 의 값이 클수록 예방정비가 비용 측면에서 효율적이며 예방정비가 효과가 있기 위해서는 적어도 C_0/C_2 가 50 정도는 되어야 함을 알 수 있다.

Table 7. Expected cost per unit time of the system without preventive maintenance

C_0	Expected cost per unit time	
	$(m,n)(r,s) : (5,5)(3,3)$	$(m,n)(r,s) : (10,10)(3,3)$
10	0.3333	1.5416
50	0.7803	2.4039
100	1.3557	3.3816
1000	11.4498	22.1396

5. 결론

본 연구에서는 선형 (m,n) 중-연속 (r,s) :고장 격자 시스템의 정

비모형을 제안하였다. 시스템의 정비기간을 최적화하는 것은 고신뢰도 및 총 정비비용 관점에서 매우 중요하다. 본 연구에서는 시스템의 단위시간당 기대정비비용을 최소화 하는 기간 정비주기를 결정하였다. 이러한 문제를 풀기 위해 본 연구에서는 몬테 카를로 시뮬레이션을 이용하여 신뢰도 및 평균 고장개수, 평균 고장시간 등을 결정하였고 유전자 알고리즘을 이용하여 부품 간의 고장이 독립일 경우의 선형 (m,n) 중-연속 (r,s) :고장 격자 시스템의 효율적인 정비기간을 제안하였다.

본 연구는 정비주기 하나만을 결정하였지만 향후 시스템 모수인 (r,s) 를 동시에 결정하는 연구 또한 유전자 알고리즘에서 교차나 돌연변이 연산자의 수정 없이 <Figure 13>과 같이 간단한 염색체 수정만으로도 가능할 것으로 판단된다.

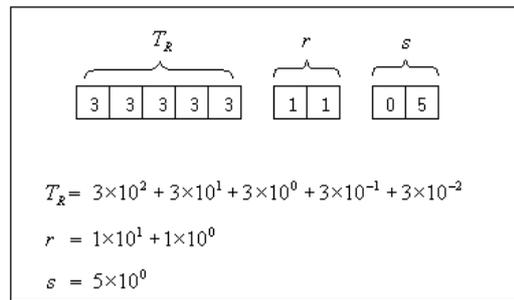


Figure 13. Extended chromosome.

또한 본 연구에서 개발된 절차들은 원형 (m,n) 중-연속 (r,s) :고장 격자 시스템이나 이외의 2차원 시스템 등의 다양한 시스템이나 고장분포가 와이블 분포를 따르는 경우 또는 부품들이 종속성을 가지는 경우 등 다양한 다른 경우에 대해서도 큰 수정 없이 이용될 수 있다.

부록

A. 시뮬레이션 절차

기호

- SN : 시뮬레이션 번호
- SRN : 시뮬레이션 반복 횟수(입력모수)
- (i,j) : (m,n) 행렬에서의 i 열과 j 행에 있는 부품의 번호
- $s(i,j)$: (i,j) 부품의 상태 및 시스템(1:작동, 0:고장)
- $t(i,j)$: (i,j) 부품의 고장시간
- NF : 고장난 부품의 고장시간
- FN : 한 주기 동안 고장난 부품의 개수
- CL : 한 주기의 길이
- TCL : 주기 CL 의 총 길이
- TFN : 고장난 부품의 총 개수
- NS : 한 주기의 끝 부분에서 시스템이 작동할 횟수

절차

0 단계 : (시작)

$SN=0$

$TCL=0, TFN=0, NSW=0$

1 단계 : (통계량 초기화)

For $i=1$ To m

For $j=1$ To n

$s(i, j) = 1, t(i, j) = \infty$

Next j

Next i

$FN = 0, CL = 0, S_{sys} = 1$

2 단계 : (고장 발생)

For $i=1$ To m

For $j=1$ To n

If $s(i, j) = 1$ Then

$t(i, j) = F^{-1}(Random(0,1))$

/* 고장 발생*/

Else

$t(i, j) = \infty$

Next j

Next i

$NF = \{(i, j) | \text{Min}(t(i, j), \text{for } i = 1, \dots, m, j = 1, \dots, n)\}$

$CL = CL + t(NF)$

3 단계 : (예방정비 및 고장 발생 확인)

If $CL > T_R$ Then

$CL = T_R, S_{sys} = 1, \text{Go to step 4}$ /* 예방정비 발생 */

Else

$s(NF) = 0, FN = FN + 1$

If $FN > r \times s$ Then

For $ii = \text{Max}(1, NF_i - r + 1)$

To $\text{Min}(m - r + 1, NF_i)$

For $jj = \text{Max}(1, NF_i - s + 1)$

To $\text{Min}(m - s + 1, NF_j)$

If $\sum_{i=ii}^{ii+1} \sum_{j=jj}^{jj+s-1} s(i, j) = 0$ Then

$S_{sys} = 0, \text{Go to step 4}$

/* 시스템 고장 발생 */

Next jj

Next ii

$S_{sys} = 1, \text{Go to step 2}$ /* 시스템 작동 중 */

4 단계 : (시뮬레이션 종료)

$TCL = TCL + CL$

$TFN = TFN + FN$

$NSW = NSW + S_{sys}, SN = SN + 1$

If $SN < SRN$ Then

Go to Step 1

Else /* 시뮬레이션 종료*/

$L(T_R) = TCL/SRN,$

$N(T_R) = TFN/SRN$

$R(T_R) = NSW/SRN$

$C(T_R) = (C_1 N(T_R) + C_0(1-R(T_R)) + C_2 R(T_R)) / L(T_R)$

참고문헌

Abraham, J. A.(1979), An improved method for network reliability, *IEEE Trans. Reliability* **R-28**(1), 58-61.

Blanks, H. S. and Tordan, M. J.(1986), Optimum replacement of deteriorating and inadequate equipment, *Quality and Reliability Engineering* **2**, 183-197.

Boehme, T. K., Kossow, A. and Preuss, W.(1992), A generalization of consecutive-k-out-of-n:F systems, *IEEE Trans. Reliability* **41**(3), 451-457.

Boutsikas, M. V. and Koutras, M. V.(2000), Generalized reliability bounds for coherent structures, *Journal of Applied Probability* **37**, 778-794.

Fu, J. C. and Koutras, M. V.(1994), Poisson approximation for 2-dimensional patterns, *Annals of the Institute of Statistical Mathematics* **46**(1), 179-192.

Jardine, A. K. S.(1973), Maintenance, replacement, and reliability, New York, John Wiley.

Kontoleon, J. M.(1980), Reliability determination of a r-successive-out-of-n:F system, *IEEE Trans. Reliability* **R-29**, 437.

Koutras, M. V., Papadopoulos, G. K. and Papastavridis, S. G.(1993), Reliability of 2-dimensional consecutive-k-out-of-n:F systems, *IEEE Trans. Reliability* **42**(4), 658-661.

Koutras, M. V., Papadopoulos, G. K. and Papastavridis, S. G.(1997), A reliability bound for 2-dimensional consecutive-k-out-of-n:F systems, *Nonlinear Analysis, Theory, Methods and Applications* **30**(6), 3345-3348.

Ksir, B.(1992), Comment on: 2-dimensional consecutive k-out-of-n:F models, *IEEE Trans. Reliability* **41**(4), 575.

Locks, M. O.(1987), A minimizing algorithm for sum of disjoint products, *IEEE Trans. Reliability* **R-36**, 120-127.

Makri, F. S. and Psillakis, Z. M.(1996), Bounds for reliability of k-within two-dimensional consecutive-r-out-of-n failure systems, *Microelectronics and Reliability* **36**(3), 341-345.

Malanie, M.(1996), An introduction to genetic algorithms, Massachusetts, Massachusetts Institute of Technology.

Malinowski, J. and Preuss, W.(1996), Lower & upper bounds for the reliability of connected-(r,s)-out-of-(m,n):F lattice systems, *IEEE Trans. Reliability* **45**(1), 156-160.

Noguchi, K., Sasaki, M., Yanagi, S. and Yuge, T.(1996), Reliability of connected-(r,s)-out-of-(m,n):F lattice system, *IEICE Transac-*

- tions Fundamentals* **J79-A(4)**, 954-963.
- Salvia, A. A. and Lasher, W. C.(1990), 2-Dimensional consecutive- k -out-of- n :F models, *IEEE Trans. Reliability* **39(3)**, 333-336.
- Wilson, J. M.(1990), An improved minimizing algorithm for sum of disjoint products, *IEEE Trans. Reliability* **39(1)**, 42-45.
- Yamamoto, H. and Miyakawa, M.(1995), Reliability of a linear connected- (r,s) -out-of- (m,n) :F lattice systems, *IEEE Trans. Reliability* **44(2)**, 333-336.
- Yamamoto, H. and Miyakawa, M.(1996), Reliability of circular connected- (r,s) -out-of- (m,n) :F lattice systems, *Journal of the Operations Research Society of Japan* **39(3)**, 389-406.
- Yun, W. Y. and Kim, G. R.(2000), Economic design of consecutive k -out-of- n :F system, *Journal of the Korean Institute of Industrial Engineers* **26(2)**, 128-135 (In Korean).
- Zuo, M. G.(1993), Reliability & design of 2-dimensional consecutive- k -out-of- n :F systems, *IEEE Trans. Reliability* **42(4)**, 488-490.