

GUB 구조를 갖는 0-1 프로그래밍에 의한 시스템 신뢰성의 최적화

(Optimization of a System Reliability by Zero-One Programming with GUB Structure)

李 在 旭*, 田 龜 濟**, 玄 光 男***

(Jae Uk Lee, Koo Chae Chun and Mitsuo Gen)

要 約

본 논문은 각 서브시스템에 여러 종류의 고장모드를 갖는 시스템 신뢰성 최적화 문제를 본 논문에서 제안한 알고리즘으로 처리된 것을 나타내었다. 이 알고리즘은 시스템 신뢰성 최적화의 비선형 문제를 0-1정수계획 문제로 변환할 때 반드시 제약조건에 부가되는 GUB구조(generalized upper bounding structure)를 고려하여 개발되었으며 시스템 신뢰도 개선을 위한 용장 서브시스템 최적배분 문제를 제안한 알고리즘에 따라 계산한 결과 계산반복회수(컴퓨터 계산시간)와 사용된 변수(요구 메모리 용량) 등에서 종래의 알고리즘 보다 우수함을 알 수 있다.

Abstract

This paper showed that the system reliability optimization problem with several failure modes at each subsystem can be treated as efficient computational algorithm proposed here. This algorithm was developed by considering a generalized upper bounding which always exists in constraints when the system reliability optimization problem may be reformulated as 0-1 integer programming problem. We demonstrated the optimal allocation of redundant units for system reliability by using the proposed algorithm. Our algorithm is superior to others in terms of number of iterations and variables used.

*正會員, 慶南專門大學 電子工學科

(Dept. of Elec. Eng., Kyungnam College)

**正會員, 東亞大學校 電子工學科

(Dept. of Elec. Eng., Donga Univ.)

***正會員, 足利工業大學 經營工學科

(Dept. of Ind. & System Eng., Ashikaga Ins. of Tech.)

接受日字: 1988年 9月 2日

I. 서 론

시스템 신뢰성의 최적화(optimization of system reliability) 문제는 비선형 자원의 제약조건 아래서 비선형 신뢰도 함수를 최대화 하는 최적해(optimal solution), 혹은 정해진 신뢰도값의 제한에서 자원을 최소화 하는 최적해를 구하는 대단히 어려운 문제이다.

과거 수년 동안 시스템의 신뢰성 향상을 위한 많은 최적화 문제가 소개되었으며, 이들 문제 해결의 유용성, 정확성, 계산시간 등의 특성에 따라 많은 기법과 알고리즘(algorithm)이 개발되었다. 이들은 동적계획법(dynamic programming),^{11~13)} 정수계획법(integer programming),^{14~16)} 휴리스틱방법(heuristic method)^{17~19)} 등으로 대별된다. 최근에도 이들 연구는 계속되어 복구조 시스템 해석,^{12~14)} 대규모 최적화 문제 해석,^{10, 15)} 시스템 파라메타 해석¹¹⁾ 등의 새로운 알고리즘이 보고 되었으며 Tillman, Hwang, Kuo^{12, 18)}는 시스템 해석의 기법과 알고리즘을 훌륭하게 검토한 바 있었다.

그러나 지금까지의 시스템의 비선형 최적화 문제를 0-1정수계획 문제로 변환할 때에 반드시 수반되는 GUB제약(generalized upper bounding constraints)을 고려하지 않으므로 전자계산기의 계산시간, 요구 메모리 용량에 상당한 결점을 지닌다. 본 논문은 GUB제약을 고려한 효과적인 알고리즘을 제안하고 다수의 고장모드(failure mode)를 갖는 시스템의 최적 용장 서브시스템(subsystem) 배분 문제를 해석하였다.

시스템의 신뢰성을 향상시키는 설계상의 방법은

- (1) 보다 높은 신뢰성을 갖는 서브시스템의 선택
- (2) 용장 서브시스템 부가
- (3) 고장 대응모드 부가
- (4) 딜레이팅(derating)의 강화

등이 있다. 시스템의 하드웨어(hard ware)의 성질을 갖는 (4)를 제외한 (1~3)은 시스템의 비용(cost), 중량(weight), 용적(volume), 전력(power) 등의 제한된 자원조건에서 시스템 신뢰도를 최대로 하기 위한 서브시스템의 최적 배분 문제로, (1)과 (2)를 함께 고려한 대규모 최적화 문제를 GUB 구조를 갖는 0-1정수계획법으로 해결하고 그 우수성을 보고한 바 있다.^{18, 20)}

한편, 시스템의 고장모드에 관한 신뢰성 문제는 Barlow, et al²¹⁾은 2개의 고장모드를 갖는 용장배분 문제를 해석하였으며 Kolesar²²⁾는 같은 문제를 선형 계획법으로 Henin²³⁾도 동일문제를 분지한정법으로 최적해를 구했다. Tillman²⁴⁾은 다종의 고장모드를 갖는 경우의 용장배분 문제를 평면절제법으로, Gen⁴⁾은 음적열거법으로 보다 효율적인 방법으로 최적해를 얻었다.

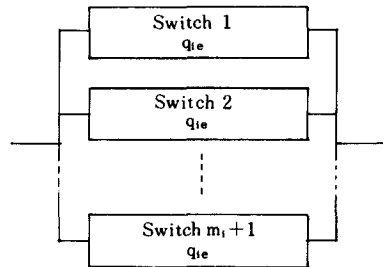
본 논문은 참고문헌 [19, 20]을 확장, 다수의 비선형 제약조건 아래서 여러 종류의 고장모드에 의한 비선형한 신뢰도를 최대로 하는 고장대응모드 최적 배분문제의 해결방법을 제시하였다.

우선 II절에서 일반적인 고장모드에 의한 시스템의 신뢰도를 도입하고 III절에서는 GUB 구조를 고려한 0-1정수계획 문제를 정식화하였으며, IV절에서 문제 해결을 위한 알고리즘을 제안, V절에서 제안한 알고리즘을 이용하여 문제를 풀고, 지금까지의 Tillman의 방법 Gen의 방법과 동일 문제를 비교 평가하므로써 제안한 알고리즘의 우수성을 증명하였다.

II. 고장모드에 따른 시스템의 신뢰성 모델

1. 용장 서브시스템 배분의 수학적 모델

서브시스템(subsystem) $i(i=1, 2, 3, \dots, N)$ 의 병렬스위치 회로의 블럭선도는 그림 1과 같다. 서브시스템 i 의 전체 용장 서브시스템(기본 서브시스템 1과 용장 서브시스템 m_i 개)는 어떤 임의의 시간에는 한 종류의 고장모드에 의해 고장이 일어난다고 한다. 서브시스템 i 의 S_i 종류의 고장모드와 상호간의 배타적인 두 형태의 고장모드 $O(e=1, 2, \dots, h_i)$ 와 $A(e=h_i+1, h_i+2, \dots, S_i)$ 로 분리된다. Q_{ie} 는 서브시스템 i 에 대한 고장모드 e 의 고장확률이다.



$e=1 \sim h_i$: "O" 형태의 고장 모드
 $e=h_i+1 \sim S_i$: "A" 형태의 고장 모드

그림 1. 서브시스템 i 의 병렬스위치 회로의 블럭선도

Fig. 1. Block diagram of parallel switching circuit at the i -th subsystem.

O형태의 고장모드의 예로서는 서브시스템 내에서 하나의 스위치가 닫히지 않거나 어느 하나의 서브시스템이 고장난 경우이다. 이 O형태의 고장모드에 따라 동작하는 서브시스템 i 의 논리적 블럭선도는 그림 2와 같다. 구체적으로 O형태의 고장모드가 일어나는 경우는 (1)서브시스템의 진동, (2)서브시스템의 전압 또는 전류의 파동에 의해 일어난다.²⁴⁾

Q_{ier} 은 서브시스템 i 에 의한 고장모드 e 의 용장 서

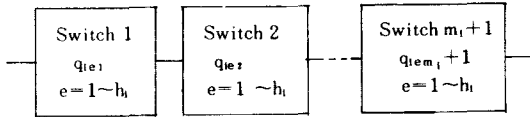


그림 2. O형태의 고장모드에 의해 고장을 일으키는 서브시스템 i의 블록 다이어그램
 Fig. 2. Logic diagram of the i-th subsystem to operate when subject to the "O" type failure modes.

서브시스템 r (r=1, 2, ..., mi+1)에 따라 동작하는 서브시스템 i의 고장확률이며 다음으로 나타낼 수 있다.

$$Q_e^o(m_i) = 1 - \prod_{r=1}^{m_i+1} (1 - q_{ier}) \quad (e=1, 2, \dots, h_i) \quad (1)$$

상기의 i, e에 대하여 같은 용장 서브시스템의 고장확률 $q_{ie} = q_{ier}$ (r=1, 2, ..., mi+1)로 되어 다음 식으로 나타낼 수 있다.

$$Q_e^o(m_i) = 1 - (1 - q_{ie})^{m_i+1} \quad (e=1, 2, \dots, h_i) \quad (2)$$

그러므로 O형태의 고장모드에 따른 전체 서브시스템 i의 고장확률(불신뢰도)는 다음 식이 된다.

$$Q_o(m_i) = \sum_{e=1}^{h_i} Q_e^o(m_i) = h_i - \sum_{e=1}^{h_i} (1 - q_{ie})^{m_i+1} \quad (3)$$

A형태의 고장모드의 예로서는 서브시스템의 전체 스위치가 고장이거나 전체 서브시스템이 고장난 경우이며, 이 A의 고장모드에 따라서 동작하는 서브시스템 i의 논리적 불확신도는 그림 3과 같다. 구체적으로 (1)감지할 수 없을 정도의 약한 신호를 스위치가 받는 경우, (2)습한 환경에서 산화에 따른 접촉이 나쁜 경우이다.^[24]

A형태의 고장모드 e에 따라 동작하는 서브시스템 i의 고장확률은 다음과 같다.

$$Q_e^A(m_i) = \prod_{r=1}^{m_i+1} q_{ier} \quad (e=h_i+1, h_i+2, \dots, S_i) \quad (4)$$

상기의 i, e에 대하여 같은 용장 서브시스템의 경우는 식 (5)와 같이 되며, A형태의 고장모드에 의한 전체 서브시스템 i의 고장확률(불신뢰도)는 식 (6)으로 나타낼 수 있다.

$$Q_e^A(m_i) = (q_{ie})^{m_i+1} \quad (e=h_i+1, h_i+2, \dots, S_i) \quad (5)$$

$$Q_A(m_i) = \sum_{e=h_i+1}^{S_i} Q_e^A(m_i) = \sum_{e=h_i+1}^{S_i} (q_{ie})^{m_i+1} \quad (6)$$

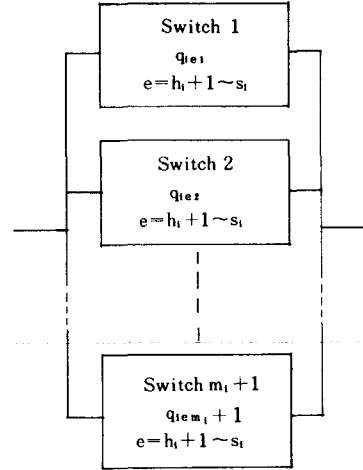


그림 3. A형태의 고장모드에 의해 고장을 일으키는 서브시스템 i의 블록 다이어그램
 Fig. 3. Logic diagram of the i-th subsystem to operate when subject to the "A" type failure modes.

O와 A형태의 고장모드로 분리된 서브시스템의 불신뢰도는 다음과 같다.

$$Q_i(m_i) = Q_o(m_i) + Q_A(m_i) = h_i - \sum_{e=1}^{h_i} (1 - q_{ie})^{m_i+1} + \sum_{e=h_i+1}^{S_i} (q_{ie})^{m_i+1} \quad (7)$$

따라서 시스템의 고장모드에 의한 신뢰도는 다음 식으로 표시된다.

$$R_i(\mathbf{m}) = \sum_{i=1}^N \{1 - Q_i(m_i)\} = \prod_{i=1}^N \{1 - h_i + \sum_{e=1}^{h_i} (1 - q_{ie})^{m_i+1} - \sum_{e=h_i+1}^{S_i} (q_{ie})^{m_i+1}\} \quad (8)$$

이 신뢰도 함수는 각 서브시스템의 용장 서브시스템의 수 m_i (미지의 양의 정수)에 관한 비선형이며 $\mathbf{m} = (m_1, m_2, \dots, m_N)$ 가 된다. 시스템에는 사용 가능한 T종류의 자원과 그 이용 제한량 b_t (t=1, 2, ..., T)가 있으며, 서브시스템 i에 관한 비선형인 자원의 필요량 $g_{it}(m_i)$ 가 주어져 있다면 시스템 자원의 제약조건은 다음 식으로 쓸 수 있다.

$$G_t(\mathbf{m}) = \sum_{i=1}^N g_{it}(m_i) \leq b_t \quad (t=1, 2, \dots, T) \quad (9)$$

O와 A형태의 고장모드에 따른 시스템 신뢰성의 최적화 문제(system reliability optimization problem)는 비선형한 T개의 제약조건 하에서 시스템의 비선

형한 신뢰도 함수를 최대로 하기 위한 용장 서브시스템 $\mathbf{m} = (m_1, m_2, \dots, m_N)$ 를 배분하는 것이다.

2. 서브시스템의 선택과 용장 배분의 수학적 모델
고장모드 e 에 따라 동작하는 서브시스템 i 에 관해서 α_i 종류의 서로 다른 유니트가 있다고 하고, q_{iea_i} 를 서브시스템 i 에 있는 유니트 $a_i (a_i = 1, 2, \dots, \alpha_i)$ 의 고장모드 e 의 고장 확률로 나타낸다. 각 서브시스템의 유니트를 선택하고, 또 용장 서브시스템을 배분하는 시스템의 고장모드에 의한 신뢰도 함수는 식 (8)로부터 유니트 a_i 를 선택한 q_{iea_i} 로 대치하여 다음 식으로 나타낼 수 있다.

$$R_2(\mathbf{a}, \mathbf{m}) = \prod_{i=1}^N \left\{ 1 - h_i + \sum_{e=1}^{h_i} (1 - q_{iea_i})^{m_i+1} - \sum_{e=h_i+1}^{s_i} (q_{iea_i})^{m_i+1} \right\} \quad (10)$$

단, $\mathbf{a} = (a_1, a_2, \dots, a_N)$

시스템 자원의 제약조건에 관해서는 서브시스템 i 에서 유니트 a_i 를 선택하고 용장 서브시스템의 수 m_i 에 따른 비선형한 자원의 필요량 $g_{ti}(\mathbf{a}_i, m_i)$ 는 $g_{tia_i}(m_i)$ 로 나타내어 제약조건은 다음과 같이 된다.

$$G_t(\mathbf{a}, \mathbf{m}) = \sum_{i=1}^N g_{tia_i}(m_i) \leq b_t \quad (t=1, 2, \dots, T) \quad (11)$$

0형태와 A형태의 고장모드에 따른 시스템 신뢰성의 최적화 문제에서는 비선형한 T개의 제약조건(11)에서 시스템의 비선형한 신뢰도 (10)를 최대로 하기 위하여 서브시스템의 유니트 $\mathbf{a} = (a_1, a_2, \dots, a_N)$ 를 선택하고 용장 서브시스템 $\mathbf{m} = (m_1, m_2, \dots, m_N)$ 을 배분하는 것이다.

III. GUB 구조를 갖는 0-1 계획문제의 선형정식화

서브시스템의 유니트 선택과 용장 배분의 비선형인 최적화 문제를 선형정식화 하기 위하여 다음과 같이 0-1변수를 정의한다.

$$x_{ijk} = \begin{cases} 1; & \text{서브시스템 } i \text{에서 유니트 } j \text{를 선택하고} \\ & k \text{개의 용장 서브시스템을 배분하는 경우.} \\ 0; & \text{그렇지 않는 경우.} \end{cases}$$

비선형인 신뢰도 함수 (10)를 0-1변수로 다음의 선형화된 목적함수를 얻을 수 있다.

$$f_2(\mathbf{x}) = \sum_{i=1}^N \sum_{j=1}^{\alpha_i} \sum_{k=1}^{L_i} c_{ijk} x_{ijk} \quad (13)$$

$\mathbf{x} = (x_{ijk})$ 이며 계수 c_{ijk} 는 전체의 i, j, k 에 대하여 다음으로 된다.

$$c_{ijk} = \ell_n \left\{ 1 - h_i + \sum_{e=1}^{h_i} (1 - q_{ie_j})^{k+1} - \sum_{e=h_i+1}^{s_i} (q_{ie_j})^{k+1} \right\} \quad (14)$$

같은 방법으로 비선형의 T개의 제약조건식도 다음과 같이 선형화된 T개의 제약조건식으로 된다.

$$g_t(\mathbf{x}) = \sum_{i=1}^N \sum_{j=1}^{\alpha_i} \sum_{k=1}^{L_i} a_{tijk} x_{ijk} \leq b_t \quad (t=1, 2, \dots, T) \quad (15)$$

계수 a_{tijk} 는 전체 t, i, j, k 에 대하여 다음과 같다.

$$a_{tijk} = g_{tji}(k) \quad (16)$$

또한 0-1 변수의 정의 (12)로부터 다음의 N개의 선형한 GUB(generalized upper bounding) 구조를 갖는 제약조건을 부가해야 한다.

$$g_{T+i}(\mathbf{x}) = \sum_{j=1}^{\alpha_i} \sum_{k=1}^{L_i} x_{ijk} = 1 \quad (i=1, 2, \dots, N) \quad (17)$$

그러므로 비선형 최적화 문제는 0-1 변수를 적용하여 선형화된 T개의 제약조건(15)와 N개의 GUB 제약조건 (17)하에서 선형화된 목적함수 (13)를 최대로 하는 0-1 선형계획 문제로 정식화된다.

각 서브시스템의 디자인이 정해진 경우는 상기의 식 (12~17)은 다음과 같이 나타낼 수 있다. (T+N)개의 제약조건식은

$$g_t(\mathbf{x}) = \sum_{i=1}^N \sum_{k=1}^{L_i} a_{tik} x_{ik} \leq b_t \quad (t=1, 2, \dots, T) \quad (18)$$

$$g_{T+i}(\mathbf{x}) = \sum_{k=1}^{L_i} x_{ik} = 1 \quad (i=1, 2, \dots, N) \quad (19)$$

이며 이 조건하에서 선형화된 목적함수는

$$f_1(\mathbf{x}) = \sum_{i=1}^N \sum_{k=1}^{L_i} c_{ik} x_{ik} \quad (20)$$

로 나타낼 수 있으며 목적함수를 최대로 하는 0-1 변수 $\mathbf{x} = (x_{ik})$ 를 구한다.

IV. GUB 구조를 고려한 알고리즘

여기서 제안한 GUB 제약을 갖는 0-1 정수계획문제를 풀기 위한 알고리즘은 다음의 두가지 패이저(phasor)로 구성되어 있다.

- (1) 효율지표(effective gradient)에 따라 실행가능해를 찾는다. (스텝 1~스텝 5)
- (2) 각 GUB 제약 가운데 실행가능해를 개선한다. (스텝 6~스텝 7)

GUB 제약을 갖는 0-1 정수계획문제를 풀기 위한 알고리즘을 다음 스텝(step)으로 나타낸다.

스텝 0; 결정변수 전체를 0으로 하고($x_{ik} = 0, \forall i, k$)

각 GUB 집합에 대하여 목적함수 최대의 계수를 갖는 변수값을 1로 한다.

$$k_i^* = \text{ind max } \{c_{ik} | k = l_i, l_i+1, \dots, L_i\} \quad \forall i \quad (21)$$

$$x_{ik_i^*} = 1, \quad \forall i$$

전체의 시스템의 제약조건을 만족하는 경우에는 스텝 8로 간다. 만족하지 않으면 스텝 1로 간다.

스텝 1 : 다음 식을 이용하여 각 GUB 제약의 변수들을 내림차순으로 정렬한다.

$$r_{ik} = \frac{c_{ik}}{\sum_{t=1}^T \frac{a_{tik}}{b_t}}, \quad \forall i, k \quad (22)$$

$$\tilde{k}_i = [\tilde{k}_{i1}, \tilde{k}_{i2}, \dots, \tilde{k}_{iL_i}]$$

$$= \text{ind sort-d } \{r_{ik} | k = l_i, \dots, L_i\}, \quad \forall i \quad (23)$$

스텝 2 : 전체의 결정변수를 0으로 하고 ($x_{ik} = 0$) 각 GUB제약의 집합에 관하여 최고순위의 변수값을 1로 한다.

$$x_{ik_{i1}} = 1, \quad \forall i$$

$$P = [P_1, P_2, \dots, P_N] = [1, 1, \dots, 1]$$

이들 변수의 집합을 초기해 x^1 이라고 한다.

스텝 3 : 전체의 시스템의 제약조건을 만족하면 스텝 6으로, 그렇지 않으면 스텝 4에 간다.

$$d_t = g_t(x^1) - b_t, \quad t = 1, 2, \dots, T \quad (24)$$

스텝 4 : 해집합(값 1을 갖는 변수)의 효율지표를 다음식에 따라 계산한다.

$$e_i = \frac{c_{ik_i P_i}}{\sum_{t=1}^T a_{tik_i P_i} d_t}, \quad \forall i \quad (25)$$

다만 $d_t < 0$ 의 경우에는 $d_t = 0$ 로 하며 이 효율지표를 기초로 대응변수를 오름차순으로 정렬하고 다음식에 따라 최소의 효율지표를 갖는 변수를 포함하는 GUB제약에 관하여 낮은 순위의 변수를 찾는다.

$$i^* = \text{ind min } \{e_i | i = 1, 2, \dots, N\} \quad (26)$$

이러한 변수가 존재하면 스텝 5로 가고 없다면 다음의 낮은 효율지표를 갖는 변수를 포함한 GUB 제약에 대하여 동일하게 처리한다.

스텝 5 : 낮은 변수의 값을 1로 하고 지금까지의 낮은 효율지표 값(1)을 0으로 바꾸고 스텝 3으로 되돌아 간다.

$$X_i^* = \tilde{k}_{i^* P_i^* + 1} = 1$$

$$X_i^* = \tilde{k}_{i^* P_i^*} = 0$$

$$P_i^* = P_i^* + 1$$

스텝 6 : 각 GUB제약에서 해에 포함된 변수보다 높은 목적함수의 계수를 갖는 실행가능 변수를 찾는다. 존재하면 스텝 7로 가고, 없다면 스텝 8로 간다.

$$S_t = -d_t, \quad \forall t$$

$$K_s^* = \text{ind } \{c_{ik} | c_{ik} \geq c_{i\tilde{k}_{i P_i}}, (s_t - a_{tik_{i P_i}} + a_{iik} \geq 0, \forall t), k = l_i, l_i+1, \dots, L_i\}, \quad \forall i \quad (27)$$

스텝 7 : 각 GUB 제약에서 실행가능 변수에 따른 목적함수의 계수차를 구하여 가장 큰 계수차를 갖는 변수 조합의 값을 1로 교환하고, 스텝 6으로 되돌아 간다.

스텝 8 : 최적해 또는 최적근사해를 얻고 종료한다.

V. 수치예 및 평가

GUB 제약을 고려한 대규모 최적화 문제⁽¹⁹⁾와 서브시스템의 디자인을 고려한 시스템의 최적화 문제⁽²⁰⁾는 그 효율성을 보고한 바 있다.

본 논문에서는 비선형 다수의 제약조건 하에서 서로 다른 종류의 고장모드를 갖는 비선형 시스템의 최적 배분 문제를 여기서 제안한 알고리즘에 따라 유닉스 시스템 [Sanyo/Icon MPS 020] 로 풀고, 정량적 평가를 위해 Tillman의 방법⁽²⁴⁾과 Gen의 방법⁽⁴⁾을 비교 검토하였다.

예로서 Tillman이 소개한 다음의 3개의 비선형 제약조건

$$G_1(m) = (m_1 + 3)^2 + (m_2)^2 + (m_3 + 2)^2 \leq 51$$

$$G_2(m) = 20(m_1 + \exp(-m_1)) + 20(m_2 + \exp(-m_2)) + 20(m_3 + \exp(-m_3)) \geq 120$$

$$G_3(m) = 20(m_1 \exp(-m_1/4)) + 20(m_2 \exp(-m_2/4)) + 20(m_3 \exp(-m_3/4)) \geq 65$$

$$m = (m_1, m_2, m_3), \quad m_i \text{는 } i=1, 2, 3 \text{에 대한 양의 정수.}$$

과 비선형 신뢰도 함수

$$R_1(m) = \prod_{i=1}^3 \left[1 - \sum_{e=1}^{h_i} \{1 - (1 - q_{ie})^{m_i+1}\} - \sum_{e=h_i+1}^{s_i} (q_{ie})^{m_i+1} \right]$$

을 최대로 하기 위한 용장 유니트 배분 $m = (m_1, m_2, m_3)$ 의 최적화 문제는 표 1에서 각 서브시스템의 고장모드 및 고장확률을 구체적으로 나타내었다.

선형화된 0-1변수와 계산기에서 사용된 변수단계는 기본 유니트를 제외한 $k=1, 2, 3, 4$ 를 사용하여, $X_{10} = X_{3,2}$ 를 나타낸다. (Tillman은 기본 유니트를 포함 $k=1, 2, 3, 4, 5$ 로 표시되었음).

표 2에서 최적해는 $m_1^* = 2, m_2^* = 1, m_3^* = 3$ 이며, 이

표 1. 각 서브시스템 유니트에 대한 고장형태와 고장확률

Table 1. The type of failure and its failure probability for each element.

SUBSYSTEM	TYPE OF FAILURE	FAILURE PROBABILITY
i	e	q_{ie}
1	O	0.01
	A	0.05
	A	0.10
2	A	0.18
	O	0.08
	A	0.02
3	A	0.15
	A	0.12
	O	0.04
3	A	0.05
	A	0.20
	A	0.10

결과는 Tillman의 평면절제법(cutting plane method), Gen의 음적열거법(implicit enumeration method) 과 동일한 최적해를 얻었으며, 물론 나타난 반복회수는 계산알고리즘에 따라 다르게 체크될 수도 있다.

본 논문에서 제안한 알고리즘은 지금까지의 방법에 비하여 계산시간과 계산반복회수가 적으며 Tillman의 방법과 Gen의 방법을 표 3에서 비교, 변수와 제약조건수에 따른 필요 메모리 용량의 면에서도 양호함을 알 수 있다.

최근에 보고된 대규모 시스템 최적화 문제를 위한 알고리즘^[10, 15]도 GUB 제약을 고려하지 않으므로 계산시간, 메모리 용량의 문제점과 한정된 제약조건(2

표 3. Tillman의 방법, Gen의 방법, 제안한 방법의 비교

Table 3. Comparison of Tillman's method, Gen's method and our method.

compared factors	Tillman's method	Gen's method	Our method
no. of variables	n_T (15)	n_G (12)	n_X (12)
no. of constraints	m_T (18)	m_G (6)	m_X (3)
storage requirement	$n_T * m_T$ (170)	$n_G * m_G$ (72)	$n_X * m_X$ (36)

() : The number of variables and constraints in example.

개)에 적합하게 개발된 알고리즘^[10, 15]의 적용성의 문제점을 지적할 수 있다.

VI. 결 론

본 논문에서는 자원의 제약조건 하에서 여러 형태의 고장을 갖는 시스템의 신뢰도를 최대로 하기 위한 용장 배분의 최적화 문제를 제안된 알고리즘으로 처리하였다. 이 알고리즘은 비선형 최적화 문제를 0-1선형 최적화 문제로 변환시 부가되는 GUB 제약을 효율지수를 이용하여 효과적으로 다루었다.

수치예로서 다수의 비선형 제약조건을 갖고 여러 형태의 고장확률에 의한 비선형인 시스템 용장 최적화 문제를 풀고 동일 문제를 해결한 종래의 방법과 비교 평가한 결과, 계산시간, 계산반복회수를 줄일 수 있음은 물론 결정변수와 제약조건식에 따른 필요 메모리 용량을 줄이므로써 퍼스날 컴퓨터로서도 대규모 최적화 문제를 처리할 수 있을 것으로 사료된다.

표 2. 각 방법의 최적해

Table 2. The optimal solutions.

step 53	T 1 * 1	T 2 1	T 3 1	T 4 0	T 5 0	T 6 * 1	T 7 1	T 8 0	T 9 0	T 10 0	T 11 * 1	T 12 1	T 13 1	T 14 1	T 15 0
step 91	*	G 1 0	G 2 1	G 3 0	G 4 0	*	G 5 1	G 6 0	G 7 0	G 8 0	*	G 9 0	G 10 0	G 11 1	G 12 0
step 4	*	X 1 0	X 2 1	X 3 0	X 4 0	*	X 5 1	X 6 0	X 7 0	X 8 0	*	X 9 0	X 10 0	X 11 1	X 12 0

T : The variables of Tillman's method

G : The variables of Gen's method

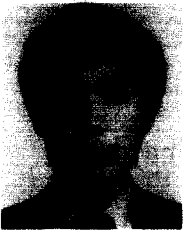
X : The variables of our method

* : Basic units

參 考 文 獻

- [1] D.E., Fyffe, W.W. Hines & N.K. Lee, System reliability allocation and a computational algorithm, *IEEE Trans. Reliability*, vol. R-17, no. 2, pp. 64-69, 1968.
- [2] K.B. Misra, Reliability/optimization of a series-parallel system Part I; Lagrange multiplier approach; Part II; Maximum principle approach, *IEEE Trans. Reliability*, vol. R-21, pp. 230-238, 1972.
- [3] C.F. Woodhouse II, Optimal redundancy allocation by dynamic programming, *IEEE Trans. Reliability*, vol. R-21, pp. 60-62, 1972.
- [4] M. Gen, Reliability optimization by 0-1 programming for a system with several failure modes, *IEEE Trans. Reliability*, vol. R-24, no. 3, pp. 206-210, 1975
- [5] R. Luus, Optimization of system reliability by a new nonlinear integer procedure, *IEEE Trans. Reliability*, vol. R-24, pp. 14-16, 1975.
- [6] M.S. Chern & R.H. Jan, Reliability optimization problems with multiple constraints, *IEEE Trans. Reliability*, vol. R-35, pp. 431-436, 1986.
- [7] W. Kuo, H.H. Lin, Z. Xu & W. Zhang, Reliability optimization with the lagrange multiplier and branch-and-bound technique, *IEEE Trans. Reliability*, vol. R-36, no. 5, pp. 624-630, 1987.
- [8] Y. Nakagawa & K. Nakashima, A heuristic method for determining optimal reliability allocation, *IEEE Trans.* vol. R-26, pp. 156-161, 1977.
- [9] P. Jedrzjowicz, Allocation of resources to maximize quality measure of a system, *IEEE Trans. Reliability*, vol. R-31, pp. 105-108, 1982.
- [10] R.L. Buffin & C.Y. Liu, Optimal allocation of redundant components for large systems, *IEEE Trans. Reliability*, vol. R-34, no. 3, pp. 241-247, 1985.
- [11] J.B. White, JBW redundancy-allocation algorithm, *IEEE Trans. Reliability*, vol. R-34, pp. 510-513, 1985.
- [12] C.L. Hwang, F.A. Tillman & W. Kuo, Reliability optimization by generalized lagrangian-function and reduced-gradient method, *IEEE Trans. Reliability*, vol. R-28, no. 4, pp. 316-319, 1979.
- [13] J.Y. Kim & L.C. Frair, Optimal reliability design for complex systems, *IEEE Trans. Reliability*, vol. R-30, pp. 300-302, 1981.
- [14] S. Dinghua, A new heuristic algorithm for constrained redundancy-optimization in complex systems, *IEEE Trans. Reliability*, vol. R-36, no. 5, pp. 621-623, 1987.
- [15] Y. Nakagawa & S. Miyazaki, Surrogate constraints algorithm for optimization problems with two constraints, *IEEE Trans. Reliability*, vol. R-30, pp. 175-180, 1981.
- [16] M.S. Chern & R.H. Jan, Parametric programming applied to reliability optimization problems, *IEEE Trans. Reliability*, vol. R-34, no. 2, pp. 165-170, 1985.
- [17] F.A. Tillman, C.L. Hwang & W. Kuo, Optimization techniques for system reliability with redundancy-A review, *IEEE Trans. Reliability*, vol. R-26, pp. 148-155, 1977.
- [18] F.A. Tillman, C.L. Hwang & W. Kuo, *Optimization of systems reliability*, Marcel Dekker, Inc., NY, 1980.
- [19] K. Ida, J.U. Lee & M. Gen, Software package for large-scale 0-1 linear programming; GUB technique, *Computers & IE*, vol. 15, 1988.
- [20] M. Gen, J.U. Lee & K. Ida, Reliability optimization by 0-1 integer programming technique, *IEEE Trans. Reliability*, 1988.
- [21] R.E. Barlow, L.C. Hunter & F. Proschan, Optimum redundancy when components are subject to two kinds of failure, *J. SAIM*, 11, 1963.
- [22] P.J. Kolesar, linear programming and the reliability of multicomponent system, *Naval. Res. Log. Quart.*, 14, 1967.
- [23] C. Henin, Double failure and other related problems in stand by redundancy, *IEEE Trans. Reliability*, vol. R-21, no. 1, 1972.
- [24] F.A. Tillman, Optimization by integer programming of constrained reliability problems with several modes of failure, *IEEE Trans. Reliability*, vol. R-18, no. 2, pp. 47-53, 1969.*

著 者 紹 介



李 在 旭 (正會員)

1949年 6月 5日生. 1975年 2月
경북대학교 전자공학과 졸업. 1987
年 8月 동아대학교 대학원 전자
공학과 박사과정 수료. 1987年 9
月~1988年 9月 일본 아시카가공
대 컴퓨터시스템 연구실 객원교수.

1979年~현재 경남전문대 전자과 부교수. 주관심분
야는 시스템 신뢰성의 최적화, 목표계획등의 알고
리즘 개발 등임.



玄 光 男 (正會員)

1944年 3月 13日生. 1969年 일본
공학원대학 전자공학과 졸업. 1975
年 일본 공학원대학 공학박사 학
위 취득. 1981年~1982年 미국
Nebraska대학교 객원교수. 1973年
~현재 일본 아시카가공대 시스템

및 산업공학과 교수. 주관심분야는 시스템 최적화기
법, 수치계산법, 목표계획법등의 소프트웨어 개발등
임.



田 龜 濟 (正會員) 1925年 3月 19日生. 1956年 3月
동아대학교 물리학과 졸업. 1959
年 9月 동아대학교 대학원 물리
학과 졸업. 1959年~현재 동아대
학교 전자공학과 교수