論文97-34S-9-2

시스템 신뢰도 최적화를 위한 중복 설계

(The Redundancy for System Reliability Optimization)

金 振 哲 *, 吳 英 煥 **, 趙 龍 九 ***

(Jin Cheol Kim, Young Hwan Oh, and Yong-Gu Cho)

요 약

본 논문에서는 제약조건들 하의 중복 설계의 수를 할당하는 것을 0-1배낭 문제(0-1Knapsack problem)로 보고 운영 기간 안에 시스템의 불신뢰도를 최소화하는 문제로 정식화하였다. 정식화한 문제에 라그랑쥐 완화법을 적용하고 변형된 분지 탐색법을 이용하여 문제의 크기를 줄였다. 분지 탐색법을 적용한 실행가능한 해들에 부분경사법을 이용하여 최적 해를 얻었다. 제안한 알고리즘을 검증하기 위하여 본 논문은 2개의 부품 군, 10개의 부품 군으로 이루어진 각각의 시스템들에 대한 중복 설계 할당의 수를 구하고 백분율 오차를 구하여 최적해임을 입증하였다. 또한 본 논문의 알고리즘에 대한 계산량을 기존 논문의 계산량과 비교, 검토하였다.

Abstract

In this paper, we supposed allocating the number of redundancies as the model of 0-1 knapsack problem and formulated the problem to maximize the systems reliability for a mission length. The formulated problem reduced the problem size using the modified branch and bound algorithm by Lagrangian relaxation. The subgradient method can optimize the set of solution. To verify the proposed method, we presented the improved results of the systems composed of two and ten component groups as the comparison of those in other papers.

I. 서 론

현대의 많은 시스템에서 신뢰도를 개선시키는 방법은 기본적으로 두 가지의 방법이 있다. 첫째, 각 부품(component)의 신뢰도를 증가시키는 방법과 둘째, 시스템에 중복 설계(redundancy)를 하는 방법이다[1].

* 正會員, 韓電情報네트웍株式會社

(Korea Electric Power Data Network Co., LTD.)

** 正會員, 光云大學校 電子通信工學科, 新技術硏究所

(Kwangwoon Univ., Dept. of Electronic Communication Eng., Institute of New Tech.)

*** 正會員, 永同工科大學校 電子工學部

(Youngdong Institute of Technology Faculity of Electronic Engineering)

接受日字: 1996年11月26日, 수정완료일: 1997年8月1日

부품의 신뢰도를 증가시키는 방법은 재료 부문에서 연구되고 있는 분야이고, 중복 설계를 하여 신뢰도를 증가시키는 방법은 시스템 신뢰도 부문에서 연구되고 있다.

시스템의 중복 설계를 하는데 있어서는 무게, 부피와 같은 각각의 부품 군(component group)에 사용되는 중복 설계를 제한하는 그 시스템 고유의 제한조건들(constraints)이 있다. 그러므로 이들 제한조건들에 대해 만족하면서 각각의 부품 군에 해당하는 중복 설계의 수를 할당한다[2]. 부품 군에 중복 설계를 할당하는 문제(Redundancy Allocation Problem, *RAP*)는 NP-complete 문제에 속하는 것으로 시스템의 크기가 증가하면 그 계산량도 지수 함수적으로 증가하는 형태의 문제인 것이다[3].

1950년대 이후로 이 *RAP*에 관한 많은 모델들과 알고리즘들이 연구되어 왔다. 이러한 알고리즘들에는

근사적인(approximate) 알고리즘, 정확한(exact) 알고리즘, 발견적인(heuristic) 알고리즘이 있다[4].

첫째, 근사적인 알고리즘들은 NP-complete 문제에 계산량을 줄이는 장점이 있지만, 일반적으로 결정 변수(decision variable)의 최적 값이 정수가 아니기 때문에 결정 변수의 최적 할당을 위해 다시 정수로 바꿔 주어야 하는 문제점이 있다.

둘째, 정확한 알고리즘들은 최적해를 보장하는 장점이 있지만 정확한 해 과정은 시스템의 크기가 증가하면 계산량이 지수 함수적으로 증가한다는 문제점이 있다. 여기에는 정수 계획법(integer programming), 동적 계획법(dynamic programming)과 같은 알고리즘들이 있다.

세째, 발견적인 알고리즘들은 실행하기 쉽고 계산량이 비교적 작은 장점이 있지만 최적 해를 보장하지 않는 문제점이 있다. 현존하는 발견적인 알고리즘들은 제한조건의 수가 증가하면 해의 질(quality)이 감소하므로 부품 군이 많은 대형 시스템에 대해서는 적당하지 못하다.

본 논문에서는 라그랑쥐 완화법을 이용한 *RAP*에 대한 최적해를 구하는 근사적인 알고리즘을 제안하였다.

먼저 운영 기간 동안 각각 일정한 고장율이 주어진 부품 군들로 이루어진 시스템의 *RAP*을 0-1배낭 문제로 정식화하고, 라그랑쥐 완화법을 이용하여 정식화한 *RAP*에 관한 식을 완화했다. 그리고, 완화한 문제에 변형된 분지 한계법(branch and bound algorithm)으로 *RAP*의 해집합을 축소하고 부분경사법(subgradient method)으로 최적 해를 구하였다.

본 논문에서도 시스템의 신뢰도를 개선하기 위한 *RAP*을 하는데 있어서 일반적으로 채택하고 있는 다음의 가정을 두기로 한다.

1) 시스템 신뢰도 구조는 대체 불가능한 부품들로 이루어진 고정적인 부분과 대체 가능한 부품들로 이루어진 부분들로 이루어져 있다. 이때 부품 간에 고장이 독립이라고 가정하면 시스템의 신뢰도는 고정적인 부분의 신뢰도와 대체 가능한 부분의 신뢰도의 곱이 된다.
2) 대체 가능한 부분의 구조는 독립적인 부품 군으로 직렬로 연결되어 시스템의 신뢰도는 부품 군들의 신뢰도의 곱으로써 나타낸다. 본 논문은 이 대체 가능한 부분에 대해서 고려한다.
3) 부품의 고장 분포를 지수 함수로 가정하며, 각각의 부품들의 고장율은 일정한 것으로 한다.
4) 운영 기간 중에는 부품에 대한 고장 검파가 완벽하게 이루어지고 중복 설계된 부품들은 무고장으로 대기 상태에 있다.
5) 대체 가능 부분에 병렬로 중복 설계를 할당하며 부품 군에 대해 적어도 하나 이상의 중복 설계를 할당한다.

Ⅱ. 시스템 신뢰도 최적화를 위한 정식화

1. 문제 설정

시스템의 신뢰도는 시스템의 다양한 부품에 중복 설계를 할당하여 증가될 수 있다. 하지만 시스템에는 항상 비용, 무게, 부피 등과 같이 각각의 부품 군에 대해서 할당되는 중복 설계의 양에 제한을 주는 고유의 제한조건들이 있다. 따라서, 일반적으로 *RAP*에 대해서 특별한 제한조건들 하에서 최적 해를 구한다[2].

그림 1과 같은 시스템의 구조에서 운영 기간이 인 시스템의 신뢰도를 최대화하기 위해서 다음과 같은 관계식이 있다.

$$Max \prod_{i=1}^{n} R_i(t', k_i) \quad (1)$$

그리고, 제한조건식들은

$$\sum_{i=1}^{n} k_i w_i \le W \quad (2)$$

$$\sum_{i=1}^{n} k_i c_i \le C \quad (3)$$

이다.

여기서 W, C는 중복 설계에 대한 최대 무게와 비용이며 k_i는 할당하는 중복 설계의 갯수이다.

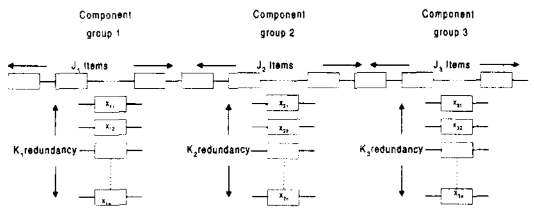


그림 1. 세 개의 대체 가능한 부품 군들로 구성된 시스템 구조의 예

Fig. 1. The example of the replaceable system structure consisting of three component group.

가정에 의하여 부품의 고장 분포가 지수 분포로 가정하면 부품의 고장율은 일정하므로 i 번째 부품 군의 고장율을 λ_i라고 하면

$$\lambda_i = j_i \times \text{(부품 군의 고장율)} \tag{4}$$

이다. 여기서는 j_i각각의 부품 군에 있는 부품들의 갯수이다[5].

그러나, 부품의 고장율이 그 응용(application)이나 환경에 영향을 받는다면 i번째 부품 군의 고장율은 λ_i = i번째 부품 군에 있는 부품들의 고장율 합

(5)

이 된다[2].

$N_i(t)$을 t시간 동안의 번 부품 군에서 부품의 고장 갯수라고 한다면 고장 분포가 지수 분포로서 고장율이 λ_i이고 $N_i(t)$는 평균이 $\lambda_i t$ 인 포아송 분포이다.

그러므로 i 번째 부품 군에서 t 시간 동안 l 개의 부품들이 고장날 확률은

$$\Pr\{N_i(t) = l\} = (\lambda_i t)^l e^{-\lambda_i t} / l! \tag{6}$$

이다.

또한, 각각의 부품 군에 k_i로 중복 설계를 할당한 시스템의 신뢰도는 $N_i(t)$에 대해서 다음과 같이 된다.

$$R_i(t, k_i) = \Pr\{N_i(t) \le l\} = e^{-\lambda_i t} \frac{(\lambda_i t)^l}{l!} \tag{7}$$

이때 n 개의 대체 가능한 부품 군으로 이루어진 시스템의 불신뢰도는

$$\begin{aligned}\prod_{i=1}^{n} Q_i(t, \sum_{j=1}^{m} x_{ij}) &= 1 - \prod_{i=1}^{n} R_i(t, \sum_{j=1}^{m} x_{ij}) \\ &= 1 - \prod_{i=1}^{n}\left[1 - \left\{e^{-\lambda_i t} \frac{(\lambda_i t)^{\sum_{j=1}^{m} x_{ij}}}{\left(\sum_{j=1}^{m} x_{ij}\right)!}\right\}\right]\end{aligned} \tag{8}$$

이다. 여기서 $\sum_{j=1}^{m} x_{ij} = k_i$이다.

그러므로, 무게와 비용의 제한조건하에 n 개의 대체 가능한 부품 군으로 이루어진 시스템의 불신뢰도에 대한 문제는 다음과 같은 식으로 설정할 수 있다.

문제 *(RAP)*

$$\begin{aligned}&Min \prod_{i=1}^{n} Q_i(t, \sum_{j=1}^{m} x_{ij}) = \\ &= Min\ 1 - \prod_{i=1}^{n}\left[1 - \left\{e^{-\lambda_i t} \frac{(\lambda_i t)^{\sum_{j=1}^{m} x_{ij}}}{\left(\sum_{j=1}^{m} x_{ij}\right)!}\right\}\right]\end{aligned} \tag{9}$$

에 대한 제한조건식들은

$$\sum_{i=i}^{n} \sum_{j=1}^{m} x_{ij} w_{ij} \le W \tag{10}$$

$$\sum_{i=i}^{n} \sum_{j=1}^{m} x_{ij} c_{ij} \le C \tag{11}$$

$$x_{ij} \ge x_{ij+1} \tag{12}$$

이다.

여기서 식(9)는 n 개의 대체 가능한 부품 군으로 이루어진 시스템의 불신뢰도를 최소화하는 식으로서 문제*(RAP)*의 목적 함수(objective function)이고, 식(10)은 시스템 전체의 무게에 관한 제한조건식이고, 식(11)은 시스템 전체의 비용에 관한 제한조건식이며, 식 (12)은 본 논문에서 새로 추가한 제한조건식으로 이 식은 각각의 부품 군에서 상위 중복 설계가 먼저 선택될 때만 하위 중복 설계가 선택될 수 있다는 것을 의미한다.

Ⅱ. 라그랑쥐 완화법

라그랑쥐 완화법은 라그랑쥐 승수 벡터를 이용하여 제한조건식들을 목적 함수식에 완화하는 방법이다. 그리고 원래의(original) 최적화 문제의 값에 대해 하한치(low bound)인 완화된 문제를 라그랑쥐 문제(Lagrangean problem)라고 한다.

일반적으로 라그랑쥐 문제에 대한 가능 해는 원래의 최적화 문제의 해에 대한 실행가능해(feasible solution)를 구하기 위해 사용된다. 또한 이 실행 가능 해와 하한치를 비교함으로써 이 실행 가능 해를 검증하는 과정은 라그랑쥐 완화법의 중요한 특징이다[6].

무게와 비용의 제한조건하에 n 개의 대체 가능한 부품 군으로 이루어진 시스템의 불신뢰도에 대한 문제가 설정된 후 문제에 대한 라그랑쥐 완화법을 적용한 식은 다음과 같다.

문제 *(RAPL)*

$$Min\ 1 - \prod_{i=1}^{n}\left[1 - \left\{e^{-\lambda_i t} \frac{(\lambda_i t)^{\sum_{j=1}^{m} x_{ij}}}{\left(\sum_{j=1}^{m} x_{ij}\right)!}\right\}\right] + \prod_{i=1}^{n} \alpha_i \sum_{j=1}^{m} (x_{ij} - x_{ij+1}) \tag{13}$$

에 대한 제한조건식들은

$$\sum_{i=i}^{n} \sum_{j=1}^{m} x_{ij} w_{ij} \le W \tag{14}$$

$$\sum_{i=i}^{n} \sum_{j=1}^{m} x_{ij} c_{ij} \leq C \tag{15}$$

여기서 α_i는 라그라쥐 승수 벡터이다. 또한 식 (13)은 라그랑쥐 완화법을 이용하여 문제(RAP)의 목적함수식인 (9)와 제한조건식 (12)에 라그랑쥐 승수 벡터 α_i를 완화한 문제($RAPL$)의 목적함수이다. 그리고 (14)와 식 (15)는 각각 무게와 비용에 대한 제한조건식들이다.

Ⅲ. 제안한 RAPL 알고리즘

본 장에서는 먼저 변형된 분지 탐색법으로 문제($RAPL$)의 해집합을 축소하고, 부분경사법으로 최적해를 구하는 $RAPL$ 알고리즘을 제안하였다.

1. 분지 한계법

분지 한계법은 각 단계에서 최적 해가 포함되어 있을 가능성이 가장 커 보이는 부분 집합을 설정하며, 이 부분집합에 계산량을 줄일 수 있는 어떤 전략을 적용하여 후보 문제라고 하는 두 개 이상의 새로운 부분집합으로 분할된다. 이를 분지(branch)라고 한다. 각 분지된 문제에서 그 부분집합에서의 해의 한계(bound)를 계산한다. 만약 해의 한계를 구하는 중에 최적 해를 구했으면 알고리즘을 끝낸다. 그렇지 않으면 후보 문제들의 한계를 사용하여 새로 분지할 부분집합을 선정하고 위의 과정을 반복한다. 이와 같이, 가능해의 영역을 분지하고 한계를 계산함으로써 해의 많은 부분집합을 고려할 필요가 없게 되어 헛된 탐색 시간을 절약하게 된다[3]. 본 논문의 해 과정에서 분지 전략은 제한 조건식들을 한계로 설정하여 가능 해의 영역을 분지함으로써 탐색 시간을 절약할 수 있도록 하였다.

본 논문의 분지 한계법은 다음의 과정들을 통해서 시행하였다

단계 1. [해집합을 구하고 정렬함]

전체 해집합 S을 구하고 중복 설계의 할당 수에 의하여 먼저 정렬을 하되 정의된 x_{ij}에 의하여 x_{11}이 1인 해부터 정렬을 한다.

단계 2. [제한조건식 식 (12)에 의한 해집합의 축소]

번호 순서로 정렬된 해집합에 대해 제한조건식 식 (12)에 의하여 적합한 부분 해집합 S_1을 구한다.

단계 3. [제한조건식 식 (14)에 의한 해집합의 축소]

부분 해집합 S_2에 대해 제한조건식 (14)에 의하여 부분 해집합 S_3를 구한다.

단계 4. [제한조건식 (15)에 의한 해집합의 축소]

부분 해집합 S_3에 대해 제한조건식 (15)에 의하여 부분 해집합 S_4를 구한다.

단계 5. [끝냄]

끝낸다.

2. 부분경사법

라그랑쥐 완화법을 사용하여 한계치를 구하는데 있어서 가장 중요한 것은 승수 집합을 계산하는 것이다. 승수 집합을 계산하는 방법에는 부분 경사법(subgradient method), 단체법(simplex method), 승수 조정법(multiplier adjustment method) 등의 방법들이 있다. 특별한 경우를 제외하면 이것은 매우 어려운 과정이다. 그리고, 이들 방법들을 사용하여 승수의 최적 집합을 계산하는 것은 반드시 필요하지 않다[8]. 본 논문의 알고리즘에서 승수 집합을 계산하기 위해서 부분 경사법을 사용하였다. 부분 경사법은 경사법을 응용한 것으로 부분 경사가 경사를 대신하여 사용되는 것이다[6],[7],[8].

문제(RAP)의 값을 Z라고 놓으면 그 식은 다음과 같다.

$$Z = Min\, 1 - \prod_{i=1}^{n}\left[1-\left\{e^{-\lambda_i t}\frac{(\lambda_i t)^{\sum_{j=1}^{m} x_{ij}}}{\left(\sum_{j=1}^{m} x_{ij}\right)!}\right\}\right] \tag{16}$$

또한, 문제($RAPL$)의 값을 $Z_L(\alpha_i)$라고 놓으면 그 식은 다음과 같다.

$$Z_L(\alpha_i) = Min\, 1 - \prod_{i=1}^{n}\left[1-\left\{e^{-\lambda_i t}\frac{(\lambda_i t)^{\sum_{j=1}^{m} x_{ij}}}{\left(\sum_{j=1}^{m} x_{ij}\right)!}\right\}\right] + \prod_{i=1}^{n} \alpha_i \sum_{j=1}^{m}(x_{ij} - x_{ij+1}) \tag{17}$$

Z 와 $Z_L(\alpha_i)$ 는 다음과 같은 관계가 항상 성립한다[6].

$$Z \geq Z_L(\alpha_i) \tag{18}$$

따라서, $Z_L(\alpha_i)$는 Z의 하한치(low bound)가 된다. Z와의 이러한 관계를 이용하여 최적해가 존재하는 한계치를 구할 수 있다.

문제(D)

$$Z_L(\alpha_i^*) = \underset{\alpha}{Max}\{Z_L(\alpha_i)\} \tag{19}$$

여기서 α_i^*는 $Z_L(\alpha_i)$에 대한 최적해를 구할 수 있는 승수 벡터이다.

단계 1. [초기화]

(1.1) 문제(D)의 실행 가능 한 해의 집합이 유한하다고 가정하여 문제(D)의 해집합을 분지 한계법의 **단계 1**에서의 해집합 S로 놓는다.

$S = \{S^r, = 1, 2, 3, \ldots R\}$

단계 2. [반복 과정의 한계를 설정하고 반복 과정을 시행함]

(2.1) $K(K = 1, 2, \ldots, k)$를 반복 과정횟수라고 정의하고 $K = 0$의 초기값을 둔다.

(2.2) 한 번의 반복 과정 후에도 최적 해를 얻지 못하면 $K = K + 1$로 하고 **단계 3**으로 간다.

(2.3) $K = 200$을 반복 과정의 한계로 설정한다. $K \le 200$이면 **단계 3**으로 간다. $K = 200$에서 α_i^*와 $Z_L(\alpha_i^*)$ 의 값을 최적값으로 정하고 **단계 6**으로 간다.

단계 3. [스텝의 크기를 정함]

(3.1) Z_j는 실행 가능한 해의 값이고 각각의 부품군에서 δ_i^k는 스칼라 값으로 $0 < \delta_i^k \le 2$의 값이다. δ_k의 초기값을 2로 놓고 20회의 반복과정을 통해서 한계치가 개선되지 않을 때 마다 반감한다.

(3.2) 초기 승수 벡터가 α_i^0으로 주어졌을 때, 스텝 크기를 t_i^k 라고 놓고 다음의 관계식으로 스텝 크기를 정의한다.

$$t_i^j = \delta_K(Z_f - Z_L(\alpha_i^k)) / \left\| \sum_{j=1}^{m}(x_{ij} - x_{ij+1}) \right\|^2$$

단계 4. [승수 벡터를 구함]

초기 승수 벡터가 α_i^0와 δ_0를 대입하고, 다음의 관계식을 이용하여 승수의 시퀀스를 발생시킨다.

$$\alpha_i^{k+1} = \alpha_i^j + {}_k\sum_{j=1}^{m}(x_{ij} - x_{ij+1})$$

여기서, $\sum_{j=1}^{m} x_{ij}^*$는 $Z_L(\alpha_i^*)$ 의 최적 해이고, $Z_L(\alpha_i^k)$는 $t_i^k \to 0$이고 $\sum_{e=0}^{\infty} t_i^e \to \infty$일 때 $Z_L(\alpha_i^*)$로 수렴한다[6],[7].

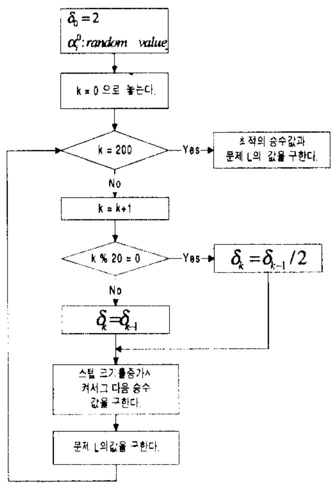
단계 5. [α_i^*와 $Z_L(\alpha_i^*)$를 구함]

(5.1) 각각의 부품군에 대해 $K = k(k \le 200)$에서의 α_i^k와 전체 시스템의 $Z_L(\alpha_i^k)$를 구한다.

(5.2) $Z \ge Z_L(\alpha_i)$의 조건 하에서 분지 한계법을 실행한 후의 해집합 S_4에 해당하고 $Z_L(\alpha_i^*) = \underset{\alpha}{Max}\{Z_L(\alpha_i)\}$를 만족하는 $\alpha_i^*, Z_L(\alpha_i^*)$ 구하면 **단계 6**으로 간다.

$Z_L(\alpha_i^*) = \underset{\alpha}{Max}\{Z_L(\alpha_i)\}$를 만족하지 않으면 **단계 2**의(2..2)로 간다.

단계 6. [끝냄]



끝낸다.

Ⅳ. RAPL 알고리즘을 이용한 중복 설계 할당

1. 두 개의 부품 군으로 이루어진 시스템

그림 2는 두 개의 부품 군으로 이루어진 시스템의 구조를 모형화한 것이다.

그림 2와 같은 시스템에서 표 1과 같은 각각의 부품 군에 대한 임의의 기간 동안 고장 갯수와 무게와 비용이 주어질 때, 무게에 대한 제한조건으로는 45이

며 비용에 대한 제한조건은 18000인 시스템에서 본 논문의 알고리즘에 의하여 불신뢰도를 최소로 하는 중복 설계를 할당한다.

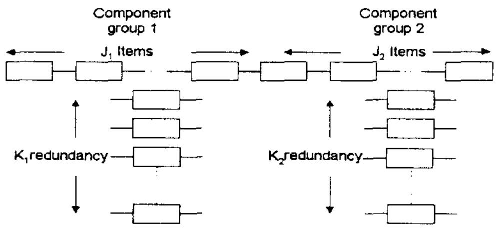


그림 2. 2개의 대체 가능한 부품 군으로 이루어진 시스템의 구조

Fig. 2. The example of the replaceable system structure consisting of two component group.

표 1. 이론적인 두 부품 군으로 이루어진 직렬 시스템

Table 1. Hypothetical two component group series system

부품 군	무게	비용	$\lambda_i t$
1	2.0	1000	1.0
2	3.0	1000	3.0

먼저 변형된 분지 한계법에 의하여

단계 1. 2^{87}개의 전체 해집합 S를 구하고 정의된 기준에 의해서 해집합을 정렬한다.

단계 2. 제한조건식 (2.12)에 의하여 87개의 부분해집합 S_1를 구한다.

단계 3. 제한조건식 (2.14)에 의하여 68개의 부분해집합 S_3를 구한다.

단계 4. 제한조건식 (2.15)에 의하여 34개의 부분해집합 S_4를 구한다.

단계 5. 분지 한계법 알고리즘의 과정을 마친다.

부분경사법에 의하여

단계 1. 해집합을 정의된 기준에 의하여 재정렬한다.

단계 2. **단계2**의 (2.1),(2.2)에 의해 $K=0$의 초기값을 둔다. **단계2** 의 (2.3)에 의해 $K=0$이므로 **단계 3**으로 간다.

단계 3. 각각의 부품군에 대해 초기 승수 벡터 a_i^0가 표 2와 같고 $\delta_i=2$일 때 를 **단계2**의 (2.2) 관계식에 의하여 표 4-2과 같이 구할 수 있다.

표 2. 2개의 부품 군으로 이루어진 시스템에서의 한 번의 반복 a_i^0와 t_i^0값

Table 2. The value of a_i^0 and t_i^0 hypothetical two component group series system after one iteration.

부품 군	a_i^0	t_i^0
1	-0.999854016	0.735642856
2	-0.999558539	0.447641996

단계 4. a_i^0와 t_i^0를 대입하면 단계3의 관계식에 의하여 a_1^1= -0.26421116의 값을 구할 수 있으며 a_2^1= -0.551916543의 값을 구할 수 있다.

단계 5.

(5.1) $Z \geq Z_L(a_i)$의 조건 하에서 $Z_L(a_i^*) = \underset{a}{Max}\{Z_L(a_i)\}$인 a_i^*와 $Z_L(a_i^*)$를 표 3과 같이 구한다.

표 3. 2개의 부품 군으로 이루어진 이론적 직렬 시스템에서의 한 번의 반복과정 후의 a_i^*와 $Z_L(a_i^*)$값

Table 3. The value of a_i^* and $Z_L(a_i^*)$ hypothetical in the two component group series system after one iteration.

부품 군	a_i^*	$Z_L(a_i^*)$
1	-0.26421116	0.367733458
2	-0.551916543	0.000463656

표 4. 2개의 부품군으로 이루어진 시스템에서의 최적의 a_i^*와 $Z_L(a_i^*)$ 값

Table 4. The value of a_i^* and $Z_L(a_i^*)$ hypothetical two component group series system.

부품 군	a_i^*	$Z_L(a_i^*)$
1	-3.31706×10^{-10}	0.000072992
2	-6.78327×10^{-10}	0.000022051

(5.2) 현재의 a_i^*와 $Z_L(a_i^*)$ 가 최적이 아니면 (2.2)로 다시 간다. 현재의 a_i^*와 $Z_L(a_i^*)$ 가 최적이면 **단계 6**으로 다시 간다. 200번의 반복과정을 통해 표 4와 같은 a_i^*와 $Z_L(a_i^*)$를 구

한다.

단계 6. 200번의 반복 과정을 마친 후 끝낸다.

본 논문의 알고리즘을 이용하여 세 개의 교체 가능한 부품 군으로 이루어진 시스템에 대해 중복 설계를 할당하여 구한 최적 해로 시스템을 설계하면 그림 3과 같다.

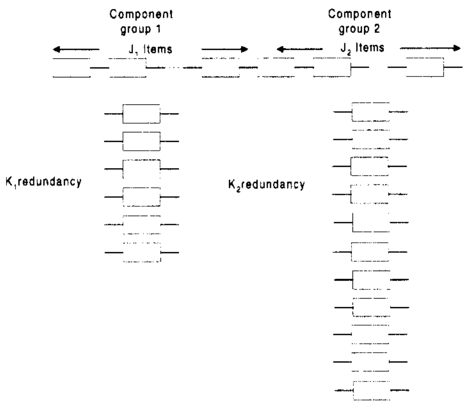


그림 3. 두 개의 대체 가능한 부품 군으로 이루어진 시스템에서의 최적의 중복 설계를 할당한 구조

Fig. 3. Replaceable system structure consisting of two component group with optimal redudacy allocation.

또한, 알고리즘의 모든 과정을 수행한 후에 시스템의 전체 불신뢰도는 0.731783×10^{-4}이다.

2. 열 개의 부품 군으로 이루어진 시스템

열 개의 부품 군으로 이루어진 시스템에서 표 5와 같은 각각의 부품 군에 대한 임의의 기간 동안 고장율과 무게와 비용이 주어질 때, 무게에 대한 제한조건이 65kg이고 비용에 대한 제한조건이 315만원 이내이라고 할 때 중복 설계를 할당한다.

먼저 변형된 분지 한계법에 의하여

단계 1. 2^{544}개의 전체 해집합 S를 구하고 정의된 기준에 의해서 해집합을 정렬한다.

단계 2. 제한조건식 (12)에 의하여 544개의 부분 해집합 S_1를 구한다.

단계 3. 제한조건식 (14)에 의하여 266개의 부분 해집합 S_3를 구한다.

단계 4. 제한조건식 (15)에 의하여 251개의 부분 해집합 S_4를 구한다.

단계 5. 분지 한계법 알고리즘의 과정을 마친다.

표 5. 열 개의 부품 군으로 이루어진 이론적인 직렬 시스템

Table 5. Hypothetical ten component group series system.

부품 군	1	2	3	4	5	6	7	8	9	10
$\lambda_i t$	0.202	0.018	0.004	0.182	0.109	0.036	0.004	0.328	0.026	0.026
무게	2.5	4.0	0.5	2.0	5.0	6.5	1.5	1.0	2.5	1.0
비용	12	15	8	11	3	10	20	14	13	18

부분경사법을 이용하여 열 개의 부품 군으로 이루어진 시스템에 대해 중복 설계를 할당하여 구한 승수 벡터 α_i의 값은 표 6과 같다.

표 6. 열 개의 부품 군으로 이루어진 이론적 직렬 시스템에서의 승수 벡터 α_i값

Table 6. Multiplier vector α_i values hypothetical ten component group series system.

부품 군	α_i
1	-3.7153×10^{-10}
2	-9.12757×10^{-10}
3	-4.26×10^{-11}
4	-6.0573×10^{-10}
5	-3.50694×10^{-10}
6	-8.9825×10^{-11}
7	-4.26×10^{-11}
8	-4.1727×10^{-10}
9	-7.757×10^{-10}
10	-7.39712×10^{-10}

표 7. 열 개의 부품 군으로 이루어진 이론적 직렬 시스템에서 중복 설계 할당 수

Table 7. Redundancy allocation numbers in hypothetical ten component group series system.

부품 군	1	2	3	4	5	6	7	8	9	10
할당된 중복 설계의 수	3	2	1	3	3	2	1	4	2	2

구한 승수 벡터 α_i의 값을 이용하여 열 개의 부품 군으로 이루어진 시스템에서 중복 설계를 할당하여 최적 해를 구하면 그 시스템의 할당된 중복 설계의 수는 표 7과 같다.

또한, 알고리즘의 모든 과정을 수행한 후에 시스템의 전체 불신뢰도는 0.731783×10^{-4}이다.

Ⅴ. 비교 및 고찰

이 장에서는 본 논문에서 제안한 라그랑쥐 완화법을 이용한 중복 설계 할당으로 시스템의 신뢰도를 최적화하는 *RAPL* 알고리즘과 종전의 논문에서 제안한 알고리즘들과 비교하였다.

시스템의 신뢰도를 최적화하기 위해서 어떠한 제한 조건 하에 중복 설계를 할당하는 알고리즘들은 크게 세 가지로 나뉜다. 근사적인 알고리즘, 정확한 알고리즘, 그리고 발견적인 알고리즘이다[4].

첫째, 근사적인 알고리즘은 결정 변수의 최적 값이 일반적으로 정수가 아니기 때문에 결정 변수의 최적 할당을 위해 다시 정수로 바꿔주어야 한다. 이러한 알고리즘에는 라그랑쥐 승수법, 기하 함수적인 계획법(geometric programming), 선형 계획법, 미분 동적 계획법(differential dynamic programming) 등이 있다.

둘째, 정확한 알고리즘은 동적 계획법, implicit search, cutting plane, genetic algorithm[9] 등이 있다. 이들 가운데 동적 계획법[2]은 이마 가장 널리 사용되고 있는 잘 알려진 알고리즘이다. 하지만 동적 계획법으로 최적해를 얻기 위해서는 계산량이 결정 변수의 수에 따라 지수 함수적으로 증가한다[4]. 이것은 라그랑쥐 승수를 사용하여 상당량을 감소 할 수 있다. 하지만 동적 계획법은 대형 시스템이나 2 개 이상의 제한조건들이 있는 시스템에 대해서 적당하지 못하다. 모든 정확한 방법들은 대형 시스템의 중복 설계를 할당하는 경우에는 널리 사용되지 않는다.

세째, 발견적인 알고리즘은 직관적인 방법으로 합리적인 시간 안에 효율적으로 최적에 가까운 해들을 구한다는 장점이 있지만 거의 모든 해 과정들의 최종해가 최적해라는 보장을 하지 못한다.

Messinger등[2]은 동적 계획법, 파운드 당 신뢰도 증가법 (incremental reliability per pound) 그리고, 일반화된 라그랑쥐 승수법의 세 가지 알고리즘으로 2 개의 부품군과 10 개의 부품군으로 이루어진 시스템의 신뢰도를 최적화하고 그 결과값을 제시하여 세 가지 알고리즘을 비교하였다. 하지만 목적 함수를 유도하는 과정 중에서 각각의 부품 군에 대한 신뢰도를 구하는 식의 오류로 인해 너무나 많은 중복 설계를 할당하였다. 예를 들어 10 개의 부품 군으로 이루어진 시스템에 동적 계획법을 이용하여 제 1 부품 군에 34개의 중복 설계를 할당하고도 전체 시스템의 신뢰도는 0.827904이었다.

Sharma등[10]은 고장율이 가장 높은 부품 군에 우선 중복 설계를 할당하는 과정을 반복하여 시스템의 신뢰도 최적화하는 알고리즘을 제안하였다. 하지만, 시스템의 신뢰도를 최대화하기 위해 불신뢰도를 이용한 목적 함수를 설정하는 과정 중에서 오류를 범했다. 그리고, 고장율이 최대인 부품 군에 우선하여 중복 설계를 하는 알고리즘에 대해 직관과 경험을 통해서라는 것은 다른 논문의 중복 설계 할당 과정과 비교했을 때 타당성이 없다.

본 논문은 시스템의 신뢰도를 최적화하기 위해 라그랑쥐 완화법을 이용하여 중복 설계를 할당하는 *RAPL* 알고리즘을 제안하였다.

첫째, 본 논문의 *RAPL* 알고리즘은 근사적인 알고리즘이며, *(RAP)*을 0-1배낭 문제로 설정함으로써 결정 변수가 반드시 0 또는 1이므로 다시 최적 해를 구하기 위해 결정 변수를 정수로 바꾸는 과정이 없다.

둘째, 2장에서 문제 설정시 시스템의 신뢰도를 최대로 하는 목적 함수로 설정하지 않고 불신뢰도를 최소로 하는 목적 함수로 설정하였다. 그리고, 새로운 제한 조건식을 추가하여 정식화하였고 이를 라그랑쥐 완화법을 적용할 시 목적 함수에 완화하였다. 이러한 문제 설정으로 라그랑쥐 완화법의 적용을 쉽게 할 수 있었다.

세째, 종전의 많은 논문이 선형 목적 함수를 사용하였으나 본 논문에서는 임의의 기간 중 부품의 고장 분포와 부품의 고장 갯수 등을 고려하여 비선형 목적 함수를 제안하였다.

표 8. 제안한 RAPL 알고리즘을 이용한 계산 결과에 대한 검증
Table 8. Verification of computational result using presented RAPL algorithm.

부품 군의 수	라그랑쥐 한계치	최상의 실행 가능 해
2	0.00073178	0.00073178
10	0.00498876	0.00498876

넷째, 시스템의 신뢰도를 최적화하기 위해서 중복설계의 수를 할당하는 문제에 대해 라그랑쥐 완화법을

이용한 본 논문이 제안한 *RAPL* 알고리즘으로 구한 해가 최적 해인가를 표 8에 검증하였다.

여기서, 라그랑쥐 한계치는 라그랑쥐 완화법에 의한 하한치의 최대값으로 즉 문제 (D)의 최적값인 $Z_L(\alpha^*)$이고 최상의 실행 가능 해는 $Z_L(\alpha^*)$를 원래의 문제에 대입하여 얻은 값이다. 그리고 여기서 백분율 오차는 라그랑쥐 한계치와 최상의 실행 가능 해의 차에 최상의 실행 가능 해로 나눈 값에 100을 곱하여 얻은 값이다. 본 논문의 백분율 오차의 값은 소수점 여덟 번째 자리 수까지 구했을 때 0.00000000으로 본 논문이 제안한 해 과정으로 얻은 해들이 최적해임을 나타낸다.

다섯째, 라그라쥐 완화법, 분지 한계법, 그리고 부분 경사법을 이용한 알고리즘으로 계산량을 줄일 수 있었다. 원래의 (RAP) 은 배낭 문제이므로 m을 할당된 중복 설계의 수라고 하면 계산량은 $O(2^m)$이다. 그리고 RAP에 동적 계획법을 이용하여 중복 설계를 할당하면 마찬가지로 계산량은 $O(2^m)$이고[2], q를 분지 전략의 수라고 하면 RAP에 분지 한계법만을 이용하여 중복 설계를 할당하면 일반적으로 $O(2^{m-q})$이다[7]. 그러나 n을 시스템의 부품 군의 갯수이라 하면, 분지 한계법에서의 계산량은 **단계 2**에 $O(m)$의하여 이고 부분경사법에서의 계산량은 $O(n)$이다[11].

Ⅵ. 결 론

본 논문에서는 라그랑쥐 완화법을 이용하여 교체 가능한 부품 군들로 이루어진 시스템의 구조에 중복 설계의 수를 할당하는 알고리즘을 제안하였다.

첫째, 임의의 기간 동안 각각 일정한 고장율이 주어진 부품 군들로 이루어진 시스템의 (RAP)을 0-1배낭 문제로 설정하였으며, 기존의 논문들에서는 선형 목적 함수를 사용하였으나 본 논문에서는 운영 기간 중 부품의 고장 분포와 부품의 고장 갯수 등을 고려하여 비선형 목적 함수를 정식화하였다

둘째, 목적 함수 설정시 라그랑쥐 완화법의 적용을 쉽게 할 수 있도록 시스템의 신뢰도를 최대로 하는 목적 함수로 설정하지 않고 불신뢰도를 최소로 하는 목적 함수로 설정하였다.

세째, 대체 가능한 부품 군들로 이루어진 시스템의 구조에 중복 설계의 수를 할당하는 문제에 대하여 새로운 제한조건식을 추가하였고, 이를 라그랑쥐 완화법 적용시 목적 함수에 완화하여 라그랑쥐 문제로 정식화하였다.

넷째, 라그랑쥐 완화법과 분지 한계법, 부분 경사법을 사용하는 알고리즘을 제안하였고, 이 근사적인 해 과정으로 계산량을 줄이고 최적 해를 **얻었다. 그리고,** 이를 검증하는 결과 값을 제시함으로써 **최적해임을** 입증하였다.

참 고 문 헌

[1] Y. nakagawa, K. Nakashma, "A Heuristic method for determining optimal reliability allocation," IEEE Trans. on Rel., vol. R-26, pp 156 ~ 161, 1977 Aug.

[2] M. Messinger and M. L. Shooman, Techniques for optimal spares allocation : A tutorial review," IEEE Trans. on Rel., vol. R-19, pp 156 ~ 166, 1970 Nov.

[3] 강맹규, 네트워크와 알고리듬, 박영사, 1991

[4] K. B. Misra, U. Sharma, "An efficient algorithm to solve integer-programming problems arising in system-reliability design," IEEE Trans. on Rel., vol.40, pp 8 1~91, 1991 Apr.

[5] Martin L. Shooman, "Probabilistic Reliability : An Engineering Approach," McGraw-Hill, 1968.

[6] M. L. Fisher, "The lagrangian relaxation for solving integer programming Problems," Management Science, vol.27, pp 1 ~ 18, Jan. 1981.

[7] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, "Network flows," Prentice Hall, 1993.

[8] S. Narasimhan, H.Pirkul, P.De, "Route selection in backbone data networks," Computer Netwok and ISDN System, vol.15, pp 121 ~ 133, 1988.

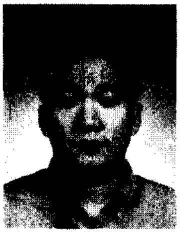
[9] D. W. Coit, A. E. Smith, "Reliability optimization of series and parallel systems using a genetic algorithm," IEEE Trans. on Rel., vol. 45, 1996 June.

[10] J. Sharma, K. V. Venkateswaran, "A direct method for maximizing the system reliability," IEEE Trans. on Rel., vol.R-20,

pp 256 ~ 259, 1971 Nov.

[11] 강맹규, 데이타구조론, 홍릉과학사, 1988

저 자 소 개



金 振 哲(正會員)
1971년 8월 17일생. 1995년 2월 광운대학교 전자통신공학과 (공학사). 1997년 2월 광운대학교 대학원 전자통신공학과 (공학석사). 1997년 2월 ~ 현재 한전정보네트웍주식회사 근무. 주관심분야 : 통신망 설계 및 성능평가

趙 龍 九(正會員)
1989~1995년 국방정보체계 연구소
현재 영동공과대학 저자공학부 교수

吳 英 煥(正會員) 第 32卷 A編 第 6號 參照
광운대학교 전자통신공학과 교수