海洋環境·安全學會誌 第 6 卷 第 1 號, pp. 23~33, 2000. 1.

論 文

직렬-병렬 시스템의 중복 설계 문제의 전역 최적화 해법에 관한 연구

김 재 환* · 유 동 훈*

A Study on A Global Optimization Method for Solving Redundancy Optimization Problems in Series-Parallel Systems

Jae-Hwan Kim · Dong-Hoon Yu**

〈목 차〉

Abstract

This paper is concerned with finding the global optimal solutions for the redundancy optimization problems in series-parallel systems related with system safety. This study transforms the difficult problem, which is classifed as a nonlinear integer problem, into a 0/1 IP(Integer Programming) by using binary integer variables. And the global optimal solution to this problem can be easily obtained by applying GAMS (General Algebraic Modeling System) to the transformed 0/1 IP. From computational results, we notice that GA(Genetic Algorithm) to this problem, which is, to our knowledge, known as a best algorithm, is poor in many cases.

* 한국해양대학교 공과대학 응용수학과

Ⅰ. 서 론

본 논문에서 다루고자 하는 중복 설계 문제는 선박, 비행기, 통신시스템 등의 신뢰도를 높이기 위해 비용, 무게 등을 고려한 다양한 제약 하에서 하부 시스템의 중복 설계를 최적화 하는 것이다. 특히 이 문제는 만족스러운 서비스 제공, 경제적 효율성뿐만 아니라 인간의 생명, 안전과도 밀접한 관계가 있다.

중복 설계 문제는 크게 다음과 같은 5가지의 구조로 분류된다(Tillman[1]).

1) 직렬 시스템 (Series System, 그림 1)
2) 병렬 시스템 (Parallel System, 그림 2)
3) 직렬-병렬 시스템 (Series-Parallel System, 그림 3)
4) 병렬-직렬 시스템 (Parallel-Series System, 그림 4)
5) 콤플렉스 시스템 (nonseries, nonparallel, 그림 5)

그림 1. 직렬 시스템

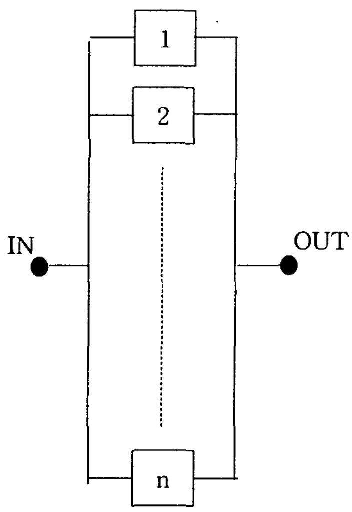


그림 2. 병렬 시스템

본 논문에서는 위의 5가지 모형 중 직렬-병렬 시스템에 대한 중복 설계 문제를 다루고자 하며, 모형 설명을 위한 기호는 다음과 같다.

1.1 기 호

R_s : 전체 시스템 신뢰도

R_s^* : 전체 시스템 신뢰도의 최적해

q_i : i번째 부품 비신뢰도

q_{ij} : i번째 하부시스템의 j번째 대안 부품의 비신뢰도

x_i : i번째 하부시스템의 부품의 개수

a_{ij} : i번째 하부시스템에 소요되는 j번째 자원의 양

d_j : j번째 자원의 최대 사용 가능량

n : 하부시스템의 개수

m : 제약식의 개수

m_i : i번째 하부시스템의 대안 부품의 개수

w_{ijk} : k번째 제약식의 i번째 하부시스템에 소요되는 j번째 자원의 양

W_k : k번째 제약식 자원의 최대 사용 가능량

x_{ij} : i번째 하부시스템의 j번째 대안 부품의 중복 개수

U : 부품 중복 개수의 상한치

$y_{i,k_1,k_2,\cdots,k_{m_i}}$: i번째 하부시스템의 0/1 정수 변수

1.2 연구배경 및 목적

본 논문에서 다루고자 하는 직렬-병렬 시스템에 대한 이해를 돕기 위해, 먼저 단순한 구조인 직렬 시스템에 대해 살펴 보고자 한다.

그림 1의 직렬 시스템에 대한 중복 설계 문제의 모형은 다음과 같다.

$$\text{Maximize} \quad R_s = \prod_{i=1}^{n} (1 - q_i^{x_i})$$

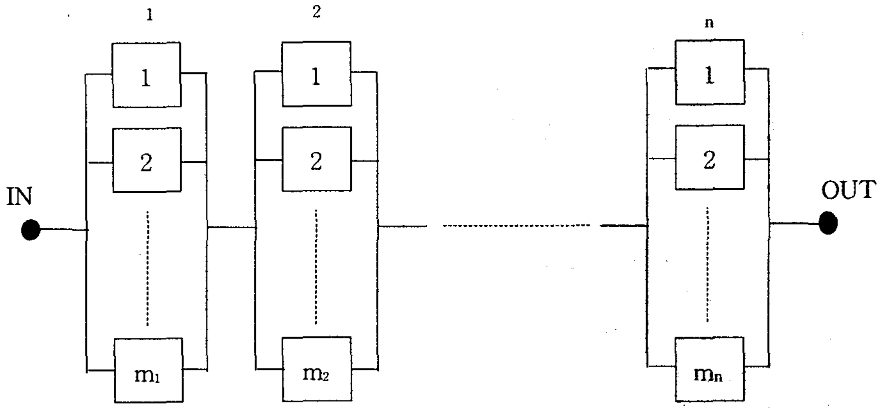


그림 3. 직렬-병렬 시스템

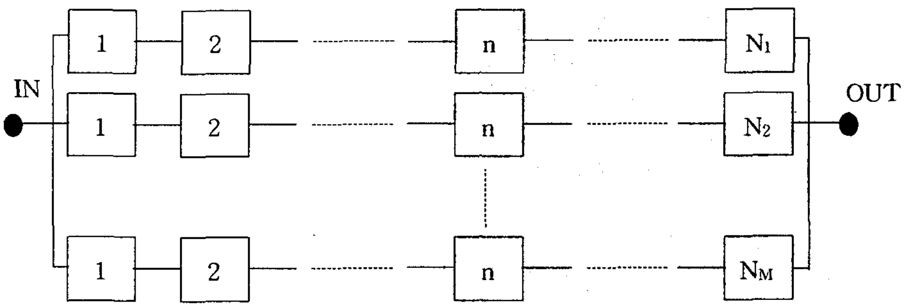


그림 4. 병렬-직렬 시스템

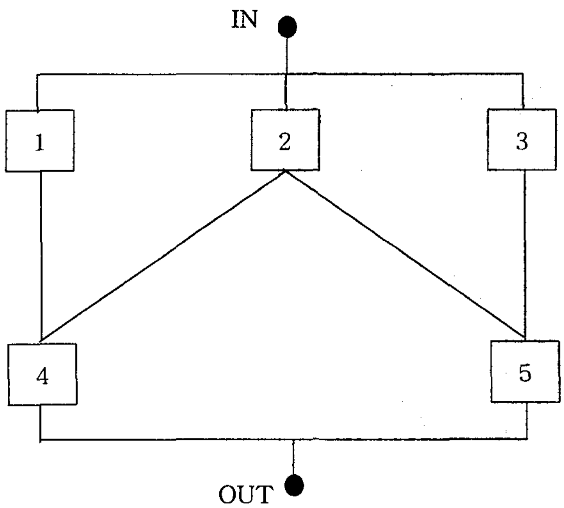


그림 5. 콤플렉스 시스템

subject to

$$\sum_{i=1}^{n} a_{ij} x_i \le d_j, \qquad j = 1, \cdots, m;$$

$$x_i \ge 1; \qquad x_i : \text{integers}$$

위의 모형은 Bellman[7], Bellman/Dreyfus[8,9] 등이 처음으로 제시 하였으며, 동적 프로그램(dynamic programming)을 사용하여 최적해를 구하였다. Ghare/Taylor[2]는 0/1 변수 변환을 이용하여 위의 모형을 풀기 쉬운 0/1 정수계획 문제로 변환한 다음, 분지 한계법(branch-and-bound method)을 이용하여 최적해를 구하였다. 또한, Nakagawa/Nakashima[3]는 효율적인 분지 한계법을 개발하여 Ghare/Taylor[2]의 방법과의 성능을 비교 하였고, Bulfin[6]은 보다 큰 규모의 문제에 대해 0/1 정수계획 문제로 변환한 후, 분지 한계법을 이용하여 최적해를 구하였다.

본 논문에서 다루고자 하는 직렬-병렬 시스템의 중복 설계 문제는 그림 3에서 처럼 직렬 시스템의 구조에 병렬 시스템을 하부구조로 복합시킨 형태이며, Fyffe[4]는 다음과 같은 직렬-병렬 시스템의 모형을 제시하였고, 동적 프로그램(dynamic programming)을 이용하여 최적해를 구하였다.

$$\text{Maximize} \quad R_s = \prod_{i=1}^{n} \left[\sum_{j=1}^{m_i} (1 - q_{ij})^{x_{ij}} \right]$$

subject to

$$\sum_{i=1}^{n} \sum_{i=1}^{m_i} w_{ijk} x_{ij} \le W_k, \qquad \forall k = 1, 2, \cdots, m$$

$$\sum_{i=1}^{m_i} x_{ij} \ge 1$$

$$0 \le x_{ij} \le U, \quad \text{integers}$$

또한 Nakagawa/Miyazaki[5]는 surrogate 제약식을 이용하여 위의 문제에 대한 최적해를 개선시켰다.

이에 반해, Smith[10]는 그림 6에서 처럼 Fyffe[4]의 모형의 단계 7, 8에서 서로 다른 하부시스템 1, 3을 중복 설계하여 신뢰도를 증가시키는 다음과 같은 모형을 제시 하였고, 발견적 해법인 GA(Genetic Algorithm)을 이용하여 최적해를 구하였다.

$$\text{Maximize} \quad R_s = \prod_{i=1}^{n} \left[\left(1 - \prod_{j=1}^{m_i} q_{ij}\right)^{x_{ij}} \right]$$

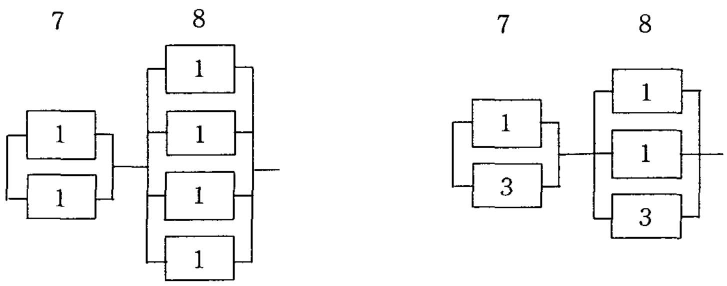
subject to

$$\sum_{i=1}^{n} \sum_{i=1}^{m_i} w_{ijk} x_{ij} \le W_k, \qquad \forall k = 1, 2, \cdots, m$$

$$\sum_{i=1}^{m_i} x_{ij} \ge 1$$

$$0 \le x_{ij} \le U, \quad \text{integers}$$

그러나, Smith[10]가 GA을 이용하여 구한 해는 전역 최적해(global optimal solution)가 아니므로, 본 논문에서는 이 문제에 대해 0/1 변수 변환을



(Fyffe) (Smith)

그림 6. Smith의 하부시스템의 중복 설계

이용하여 전역 최적해를 구하고자 한다.

1.3 연구 범위

본 논문에서는 직렬-병렬 시스템의 중복설계 문제에 대한 전역 최적해를 구하기 위해, 먼저 0/1 변수 변환을 이용하여 풀기 쉬운 0/1 정수계획 문제로 변환한 다음 GAMS(General Algebraic Modeling System)을 이용하여 최적해를 구하였다. 또한, 2장에서는 널리 알려진 Fyffe[4]의 33개의 문제에 대해 전역 최적해를 구하여 Nakagawa/ Miyazaki[5], Smith[10] 등의 해와 비교 분석 하였다.

Ⅱ. 0/1 변수 변환을 이용한 해법

본 장에서는 Smith[10]가 제시한 다루기 힘든 비선형 정수계획 문제인 직렬-병렬 시스템의 중복 설계 문제를 풀기 위해, 이와 유사한 직렬 시스템의 0/1 변수 변환에 의한 해법을 살펴보고, 이에 기초를 둔 직렬-병렬 시스템에 대한 0/1 변수 변환을 이용한 해법을 제시하고자 한다.

2.1 직렬 시스템에 대한 0/1 변수 변환

Ghare/Taylor[2]는 서론에서 언급한 다음 직렬 시스템 모형 (P_0)에서

(P_0)

Maximize $\quad R_s = \prod_{i=1}^{n}(1-q_i^{x_i})$

subject to

$$\sum_{i=1}^{n} a_{ij}x_i \le d_j, \qquad j=1,\cdots,m;$$

$$x_i \ge 1; \qquad x_i : \text{integers}$$

$C_{ik} = \log(1-q_i^{k+1}) - \log(1-q_i^{k})$,

$b_j = d_j - \sum_{i=1}^{n} a_{ij}$ 로 치환하여 다음과 같은 0/1 정수계획 문제 (P_0')로 변환시켰다.

(P_0')

Maximize $\quad R_s' = \sum_{i=1}^{n}\sum_{k=1}^{k=\infty} c_{ik}x_{ik}$

subject to

$$\sum_{i=1}^{n}\sum_{k=1}^{\infty} a_{ij}x_{ik} \le b_j, \qquad \forall j=1,2,\cdots,m$$

$x_{ik} = 0$ or 1

여기서, $x_{ik}=0$ 은 $x_{il}=0 \ (l>k)$을 의미한다.

Ghare/Taylor[2]는 모형 (P_0)와 (P_0')는 동일하다는 것을 증명하였으며, (P_0')에 대해 분지한계법을 이용하여 최적해를 구하였다.

2.2 직렬-병렬구조 시스템에 대한 0/1 변수 변환

직렬-병렬 시스템의 다음 모형 (P_1)의 목적함수 R_s는 직렬 시스템과는

(P_1)

Maximize $\quad R_s = \prod_{i=1}^{n}\left(1-\prod_{j=1}^{m_i} q_{ij}^{x_{ij}}\right)$

subject to

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i} w_{ijk}\cdot x_{ij} \le W_k, \qquad \forall k=1,2,\cdots,m$$

$$\sum_{j=1}^{m_i} x_{ij} \ge 1$$

$$0 \le x_{ij} \le U, \quad \text{integers}$$

다른 복잡한 형태이므로, 직렬 시스템에서 언급한 Ghare/Taylor[2]의 0/1 변수 변환은 가능하지 않다. 따라서, 비선형 정수계획 문제인 (P_1)을 풀기 쉬운 0/1 정수계획 문제로 변환하기 위해 Ghare/Taylor[2]와는 달리, $0 \le x_{ij} \le U$ 의 모든

경우의 정수해를 0/1 변수를 이용하여 나열하였고, 변환한 모형은 다음과 같다.

Maximize

$$R_s' = \sum_{i=1}^{n} \Big[\sum_{k_1=0}^{U} \sum_{k_2=0}^{U} \cdots \sum_{k_{m_i}=0}^{U} (r_{i,k_1,k_2,\cdots,k_{m_i}} \times y_{i,k_1,k_2,\cdots,k_{m_i}}) \Big] \quad ,(k_1,k_2,\cdots,k_{m_i}) \neq 0$$

subject to

$$\sum_{i=1}^{n} \Big[\sum_{k_1=0}^{U} \sum_{k_2=0}^{U} \cdots \sum_{k_{m_i}=0}^{U} \Big[\Big(\sum_{j=1}^{m_i} k_j w_{ijk} \Big) \times y_{i,k_1,k_2,\cdots,k_{m_i}} \Big) \Big] \le W_k$$

$$\forall k = 1, 2, \cdots, m \quad ,(k_1,k_2,\cdots,k_{m_i}) \neq 0$$

$$\sum_{k_1=0}^{U} \sum_{k_2=0}^{U} \cdots \sum_{k_{m_i}=0}^{U} y_{i,k_1,k_2,\cdots,k_{m_i}} = 1$$

$$\forall i = 1, 2, \cdots, n \quad ,(k_1,k_2,\cdots,k_{m_i}) \neq 0$$

$$y_{i,k_1,k_2,\cdots,k_{m_i}} = 0, 1 \quad \forall i = 1, 2, \cdots, n$$

$$\forall k_1 = 0, 1, 2, \cdots, U$$

$$\forall k_2 = 0, 1, 2, \cdots, U$$

$$\forall k_{m_i} = 0, 1, 2, \cdots, U$$

여기서,

$$r_{i,k_1,k_2,\cdots,k_{m_i}} = \log\left(1 - q_{i_1}^{\ k_1} q_{i_2}^{\ k_2} \cdots q_{i_{m_i}}^{\ k_{m_i}}\right)$$

이다.

위의 문제는 풀기 쉬운 0/1 정수계획 문제이므로, GAMS를 이용하여 전역 최적해를 구할 수 있다.

2.3 예 제

본 절에서는 2.2 절에서 언급한 모형을 Fyffe[4]의 예제를 통해 구체적으로 제시하고자 한다.

Fyffe[4]의 예제

$$\text{Maximize} \quad R_s = \prod_{i=1}^{14} \Big(1 - \prod_{j=1}^{m_i} q_{ij}^{\ x_{ij}}\Big)$$

subject to

$$\sum_{i=1}^{14} \sum_{j=1}^{m_i} w_{ijk} \cdot x_{ij} \le W_k, \qquad \forall k = 1, 2$$

$$\sum_{j=1}^{m_i} x_{ij} \ge 1$$

$$0 \le x_{ij} \le 5, \quad \text{integers}$$

여기서, n=14, W_1=130, W_2=191 이며, m_i 계수는 표 2.1에, q_{ij} 및 w_{ijk}의 계수는 표 2.2에 각각 주어져 있다.

위의 비선형 정수계획 문제는 2.2절에서 언급한 0/1 변수 변환의 모형에 대입하면, 다음과 같은 0/1 정수계획 문제로 표현된다.

$$\text{Maximize} \quad R_s' = -0.10536 \times y_{1,1,0,0,0} + \cdots + 0 \times y_{14,5,5,5,5}$$

subject to

$$1 \cdot y_{1,1,0,0,0} + \cdots + 95 \cdot y_{14,5,5,5,5} \le 130$$

표 2.1 m_i의 계수

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
m_i	4	3	4	3	3	4	3	3	4	3	3	4	3	4

표 2.2 q_{ij}, w_{ijk} 의 계수

i	j (Design Alternative)											
	1			2			3			4		
	q_{ij}	w_{ij1}	w_{ij2}	q_{ij}	w_{ij1}	w_{ij2}	q_{ij}	w_{ij1}	w_{ij2}	q_{ij}	w_{ij1}	w_{ij2}
1	0.10	1	3	0.07	1	4	0.09	2	2	0.05	2	5
2	0.05	2	8	0.06	1	10	0.07	1	9	*	*	*
3	0.15	2	7	0.10	3	5	0.13	1	6	0.08	4	4
4	0.17	3	5	0.13	4	6	0.15	5	4	*	*	*
5	0.06	2	4	0.07	2	3	0.05	3	5	*	*	*
6	0.01	3	5	0.02	3	4	0.03	2	5	0.96	2	4
7	0.09	4	7	0.08	4	8	0.06	5	9	*	*	*
8	0.19	3	4	0.10	5	7	0.09	6	6	*	*	*
9	0.03	2	8	0.01	3	9	0.04	4	7	0.91	3	8
10	0.17	4	6	0.15	4	5	0.10	5	6	*	*	*
11	0.06	3	5	0.05	4	6	0.04	5	6	*	*	*
12	0.21	2	4	0.18	3	5	0.15	4	6	0.90	5	7
13	0.02	2	5	0.01	3	5	0.03	2	6	*	*	*
14	0.10	4	6	0.08	4	7	0.05	5	6	0.99	6	9

$$3 \cdot y_{1,\,1,0,0,0} + \cdots + 140 \cdot y_{14,\,5,5,5,5} \le 191$$

$$\begin{aligned}
y_{1,\,1,0,0,0} + \cdots + y_{1,\,5,5,5,5} &= 1 \\
y_{2,\,1,0,0,0} + \cdots + y_{2,\,5,5,5,5} &= 1 \\
&\vdots \\
y_{14,\,1,0,0,0} + \cdots + y_{14,\,5,5,5,5} &= 1
\end{aligned}$$

$$y_{1,\,1,0,0,0}\,,\ \cdots\,,\ y_{14,\,5,5,5,5} = 0\,,1$$

위의 문제는 0/1 정수계획 문제이므로, GAMS를 이용하여 전역 최적해를 구하였고, 구체적인 GAMS의 프로그램 list는 <부록>에 수록되어 있다. 이 GAMS 프로그램을 IBM RS / 6000P로 수행시킨 결과, 수행시간은 0.32초가 소요되었다. 이 문제에 대한 0/1 변수의 총 개수는 9490개 이며, 최적해 $R_s^* = 0.955$이다.

Ⅲ. 전산 실험 결과

기존의 발견적 해법(N&M, GA)의 성능을 분석하기 위해 Smith[10]가 분석한 33개의 문제에 대해 본 연구에서 제시한 해법에 의해 전역 최적해를 구하였고, 비교한 결과는 표 3.1과 표 3.2에 나타나 있다.

표 3.1의 17문제(175~191)에 대해서 N&M과 GA는 전역 최적해를 한 번도 얻지 못했음을 알 수 있고, 표 3.2의 16문제(159~174)에 대해서는 N&M과 GA는 각각 1번, 11번의 전역 최적해를 구하는 것을 알 수 있었다.

위의 결과로부터 현재 가장 좋은 해를 찾아준다고 알려진 Smith[10]의 GA는 표 3.1에서 처럼 가능해의 영역이 점점 확대됨에 따라, 전역 최적해를 한 번도 찾지 못함을 본 연구를 통해 알 수 있었다.

표 3.1 기존의 발견적 해법과의 비교

	N&M			GA	0/1변수변환 해법
	최적해 R_s^*	Cost	Weight	최적해 R_s^*	전역 최적해 R_s^*
191	.9864	130	191	.9867 *	.9868
190	.9854	132	189	.9857 *	.9864
189	.9850	131	188	.9856 *	.9859
188	.9847	129	188	.9850 *	.9854
187	.9840	133	186	.9844 *	.9847
186	.9831	129	186	.9836 *	.9841
185	.9829	129	185	.9831 *	.9835
184	.9822	126	184	.9823 *	.9829
183	.9815	130	182	.9819 *	.9823
182	.9815	130	182	.9811 *	.9815
181	.9800	128	181	.9802 *	.9810
180	.9796	126	180	.9797 *	.9803
179	.9792	127	179	.9791 *	.9795
178	.9772	123	177	.9783 *	.9784
177	.9772	123	177	.9772 *	.9776
176	.9764	125	176	.9764 *	.9767
175	.9744	121	174	.9753 *	.9757

표 3.2 기존의 발견적 해법과의 비교

	N&M			GA	0/1변수변환 해법
	최적해 R_s^*	Cost	Weight	최적해 R_s^*	전역 최적해 R_s^*
174	.9744	121	174	.9744 *	.9749
173	.9723	122	173	.9738	.9738
172	.9720	123	172	.9727 *	.9731
171	.9700	119	170	.9719	.9719
170	.9700	119	170	.9708	.9708
169	.9675	121	169	.9692 *	.9693
168	.9666	120	168	.9681	.9681
167	.9656	117	167	.9663 *	.9664
166	.9646	116	166	.9650	.9650
165	.9621	118	165	.9637	.9637
164	.9609	116	164	.9624	.9624
163	.9602	114	163	.9606	.9606
162	.9589	112	162	.9591 *	.9592
161	.9565	111	161	.9580	.9580
160	.9546	110	159	.9557	.9557
159	.9546	110	159	.9546	.9546

Ⅳ. 결 론

본 논문에서는 Smith[10]가 제시한 다루기 힘든 비선형 정수계획 문제인 직렬-병렬 시스템의 중복 설계 문제의 전역 최적해를 구하기 위해, 먼저 0/1 변수 변환을 이용하여 풀기 쉬운 0/1 정수계획 문제로 변환한 다음, GAMS (Genetic Algebraic Modeling System)을 이용하여 최적해를 구하였다.

2.3절의 예제에서 총 9490개의 0/1 변수가 사용된 문제를 IBM RS/6000P 기종에 의해 GAMS 프로그램을 수행한 결과, 0.32초가 소요됨을 알 수 있었다. 또한, Fyffe[4]의 33개 문제에 대해 전산 실험한 결과, 현재까지 가장 좋은해를 구할 수 있다고 알려진 Smith[10]의 GA는 가능해의 영역이 점점 확대됨에 따라 전역 최적해를 한 번도 찾지 못함을 본 연구를 통해 알 수 있었다.

본 연구에서 제시한 0/1 변수변환 해법은 상당히 큰 규모의 문제에 대해 변수의 개수가 지수증가적으로 늘어 날 수 있으므로, 앞으로 필요한 연구과제는 변수의 개수를 효율적으로 줄이는 변수변환 방법등의 개발이다.

〈부 록〉 GAMS의 프로그램 list

```
*option lp=minos5, limrow=0, limcol=0,
 solprint=off;
option limrow=0, limcol=0;
option solprint=off;
*option mip=zoom;
option optcr=0;
parameter omega;
omega = 191 ;
sets    i /1*14/
        j /1*6/
        k /1*6/
        l /1*6/
        m /1*6/
        s /1*4/ ;
```

```
parameter i3(i) /
 2    1
 4    1
 5    1
 7    1
 8    1
10    1
11    1
13    1 /;
parameter i4(i) /
1     1
3     1
6     1
9     1
12    1
14    1 /;

table q(i,s)
       1      2      3      4
1     0.1    0.07   0.09   0.05
2     0.05   0.06   0.07
3     0.15   0.1    0.13   0.08
4     0.17   0.13   0.15
5     0.06   0.07   0.05
6     0.01   0.02   0.03   0.04
7     0.09   0.08   0.06
8     0.19   0.1    0.09
9     0.03   0.01   0.04   0.09
10    0.17   0.15   0.1
11    0.06   0.05   0.04
12    0.21   0.18   0.15    0.1
13    0.02   0.01   0.03
14    0.1    0.08   0.05    0.01 ;

table c(i,s)
       1      2      3      4
1      1      1      2      2
2      2      1      1
3      2      3      1      4
4      3      4      5
```

```
5     2     2     3
6     3     3     2     2
7     4     4     5
8     3     5     6
9     2     3     4     3
10    4     4     5
11    3     4     5
12    2     3     4     5
13    2     3     2
14    4     4     5     6 ;

table w(i,s)
      1     2     3     4
1     3     4     2     5
2     8     10    9
3     7     5     6     4
4     5     6     4
5     4     3     5
6     5     4     5     4
7     7     8     9
8     4     7     6
9     8     9     7     8
10    6     5     6
11    5     6     6
12    4     5     6     7
13    5     5     6
14    6     7     6     9 ;

parameter r4(i,j,k,l,m), r3(i,j,k,l);
parameters jexp, kexp, lexp, mexp;
loop((j,k,l) $ (ord(j) gt 1 or ord(k) gt 1 or
      ord(l) gt 1),
         jexp = ord(j)-1;
         kexp = ord(k)-1;
         lexp = ord(l)-1;
         r3(i,j,k,l) $ (i3(i) eq 1) = log(1-
            q(i,'1')**jexp *

q(i,'2')**kexp * q(i,'3')**lexp) ;
);
loop((j,k,l,m) $ (ord(j) gt 1 or ord(k) gt 1 or
      ord(l) gt 1 or ord(m) gt 1),
         jexp = ord(j)-1;
         kexp = ord(k)-1;
         lexp = ord(l)-1;
         mexp = ord(m)-1;
         r4(i,j,k,l,m)$(i4(i)  eq  1)=log(1-q
            (i,'1')**jexp * q(i,'2')**kexp
             * q(i,'3')**lexp * q(i,'4')**mexp) ;
);
*display r3, r4;
binary variables x3(i,j,k,l), x4(i,j,k,l,m);
variable rs;
equations obj, eq1, eq2, eq3, eq4 ;
obj .. rs =e= sum(i $ i3(i), sum ((j,k,l) $
         (ord(j) gt 1 or ord(k) gt 1 or ord(l) gt
           1),r3(i,j,k,l)*x3(i,j,k,l)))
+ sum(i $ i4(i), sum ((j,k,l,m)
         $ (ord(j) gt 1 or ord(k) gt 1 or ord(l)
            gt 1 or ord(m) gt 1),
         r4(i,j,k,l,m)*x4(i,j,k,l,m))) ;
eq1 .. sum(i $ i3(i), sum( (j,k,l)
         $ (ord(j) gt 1 or ord(k) gt 1 or ord(l)
            gt 1),
(c(i,'1')*(ord(j)-1)+c(i,'2')*(ord(k)-1)+
         c(i,'3')*(ord(l)-1))*x3(i,j,k,l))   )+sum(i
           $ i4(i), sum( (j,k,l,m)
         $ (ord(j) gt 1 or ord(k) gt 1 or ord(l)
            gt 1 or ord(m) gt 1),
(c(i,'1')*(ord(j)-1)+c(i,'2')*(ord(k)-1)+
         c(i,'3')*(ord(l)-1)+c(i,'4')*(ord(m)-1))*
           x4(i,j,k,l,m)) ) =l= 130 ;
eq2 .. sum(i $ i3(i), sum( (j,k,l)
         $ (ord(j) gt 1 or ord(k) gt 1 or ord(l)
            gt 1),
(w(i,'1')*(ord(j)-1)+w(i,'2')*(ord(k)-1)+
         w(i,'3')*(ord(l)-1))*x3(i,j,k,l)) ) +
         sum(i $ i4(i), sum( (j,k,l,m)
         $ (ord(j) gt 1 or ord(k) gt 1 or ord(l)
            gt 1 or ord(m) gt 1),
```

```
(w(i,'1')*(ord(j)-1)+w(i,'2')*(ord(k)-1)+

        w(i,'3')*(ord(l)-1)+w(i,'4')*(ord(m)-1))*
          x4(i,j,k,l,m)) ) =l= omega ;
eq3(i) $ i3(i) .. sum((j,k,l)
          $ (ord(j) gt 1 or ord(k) gt 1 or ord(l)
            gt 1),
          x3(i,j,k,l)) =e= 1 ;
eq4(i) $ i4(i) .. sum((j,k,l,m)
          $ (ord(j) gt 1 or ord(k) gt 1 or ord(l)
            gt 1 or ord(m) gt 1),
          x4(i,j,k,l,m)) =e= 1 ;
model reliable /all/;
solve reliable using mip maximizing rs ;
display x3.l, x4.l, rs.l;
```

참고문헌

[1] F.A Tillman, C.L.Hwang, & W.Kuo, "", IEEE Trans. on Rel. vol. R-26, No.3, 1977, pp. 148-155.

[2] P.M. Ghare & R.E. Taylor, "Optimal Redundancy for Reliability in Series System", O.R vol. 17, 1969, pp. 838-847.

[3] Nakagawa, Y. , Nakashima, K & Hattori, Y., "Optimal reliability allocation by Branch and Bound techniques", IEEE Trans. on Rel. vol. 27, 1978, pp. 31-38.

[4] D.E. Fyffe, W.W. Hines, & N.K.Lee, "System Reliability Allocation and a Computational Algorithm", IEEE Trans. on Rel., vol. R-17, 1968, pp. 64-69.

[5] Y. Nakagawa, & S. Miyazaki, "Surrogate constraint algorithm for reliability optimization problem with two constraints", IEEE Trans. on Rel., vol. R-30, 1981, pp. 175-179.

[6] R.L. Bulfin, & C.Y. Liu, "Optimal Allocation of Redundant Components for Large Systems", IEEE. Trans. on Rel., vol. R-34, 1985, pp. 241-247.

[7] R. Bellman, Dynamic Programming, Princeton, N. J. : Princeton University Press, 1957

[8] R. E. Bellman, E. Dreyfus, "Dynamic programming and reliability of multicomponent devices", Operations Research, vol 6, 1958, pp. 200-206.

[9] R. E. Bellman, E. Dreyfus, Applied Dynamic Programming, 1962, Princeton Univ. Press.

[10] A. E. Smith, D.M. Tate, "Genetic optimization using a penalty function", Proc. Genetic Algorithms, 1993, pp. 499-505.