

하이브리드 병렬 유전자 알고리즘을 이용한 최적 신뢰도-중복 할당 문제

김기태 · 전건욱[†]

국방대학교 운영분석학과

An Optimal Reliability-Redundancy Allocation Problem by using Hybrid Parallel Genetic Algorithm

Ki Tae Kim · Geonwook Jeon

Department of Operations Research, Korea National Defense University, Seoul 122-875

Reliability allocation is defined as a problem of determination of the reliability for subsystems and components to achieve target system reliability. The determination of both optimal component reliability and the number of component redundancy allowing mixed components to maximize the system reliability under resource constraints is called reliability-redundancy allocation problem(RAP).

The main objective of this study is to suggest a mathematical programming model and a hybrid parallel genetic algorithm(HPGA) for reliability-redundancy allocation problem that decides both optimal component reliability and the number of component redundancy to maximize the system reliability under cost and weight constraints. The global optimal solutions of each example are obtained by using CPLEX 11.1. The component structure, reliability, cost, and weight were computed by using HPGA and compared the results of existing meta-heuristic such as Genetic Algorithm(GA), Tabu Search(TS), Ant Colony Optimization(ACO), Immune Algorithm(IA) and also evaluated performance of HPGA.

The result of suggested algorithm gives the same or better solutions when compared with existing algorithms, because the suggested algorithm could paratactically evolved by operating several sub-populations and improve solution through swap, 2-opt, and interchange processes. In order to calculate the improvement of reliability for existing studies and suggested algorithm, a maximum possible improvement(MPI) was applied in this study.

Keyword: reliability-redundancy allocation problem, reliability optimization, hybrid parallel genetic algorithm

1. 서론

오늘날 과학과 기술의 발달로 인해 산업사회가 고도화되고, 생산되는 시스템의 기능과 구조가 점차 복잡화되면서 고장 발생의 우려가 높아지고 있는 반면 사용자는 고품질 및 고 신뢰

도를 갖는 시스템을 요구하고 있다. 이에 생산자는 품질향상을 통한 경쟁력 강화와 신뢰도 향상을 필수적으로 고려하게 되었다. 신뢰도는 일정기간 동안 주어진 조건하에서 사용자의 요구조건에 부합하는 기능을 원활하게 수행할 수 있는 확률을 말하며, 신뢰도가 높은 시스템이 품질 또한 우수한 시스템이

[†]연락처 : 전건욱 교수, 122-875 서울특별시 은평구 수색동 205번지 국방대학교 운영분석학과, Fax : 02-309-6233,

E-mail : g0jeon01@yahoo.co.kr

투고일(2009년 10월 29일), 심사일(1차 : 2010년 01월 06일, 2차 : 2010년 01월 25일), 게재확정일(2010년 01월 27일).

라 할 수 있다.

시스템 신뢰도의 향상을 위한 업무를 제조나 판매 이후에 수행하게 되면 막대한 비용이 소요되므로 설계단계에서부터 수행하는 것이 바람직하다. 설계단계에서의 시스템 신뢰도 향상을 위한 방법으로는 시스템을 구성하고 있는 구성품 자체의 신뢰도를 증가시키는 방법, 신뢰도가 낮은 부분을 중복(Redundancy)으로 설계하는 방법, 구성품의 배치를 변경하는 방법 등이 있다. 이러한 시스템 구성품의 신뢰도 향상과 중복 설계를 동시에 결정하는 문제를 신뢰도-중복 할당 문제(Reliability-Redundancy Allocation Problem, RAP)라 한다.

신뢰도-중복 할당 문제는 주어진 자원 제약 하에서 전체 시스템의 신뢰도를 최대화할 수 있도록 구성품의 최적 신뢰도와 중복 수를 결정하는 문제이다. 신뢰도-중복 할당 문제에서 구성품의 신뢰도는 0과 1사이의 실수 형태로 표현되고, 중복 수는 정수 형태로 표현되므로 혼합 정수 최적화 문제(Mixed Integer Optimization Problem)가 되며, NP-hard인 조합 최적화 문제(Combinatorial Optimization Problems)로 정의될 수 있다.

신뢰도-중복 할당 문제의 최적해를 구하기 위하여 다양한 연구들이 수행되었다. 중복 설계를 고려하지만 구성품의 혼합 사용이 허용되지 않는 시스템의 최적 신뢰도-중복 할당하는 문제에 대하여 Fyffe *et al.*(1968)은 동적계획법(Dynamic Programming)을 활용한 휴리스틱 방법을 제시하였으며, Nakagawa and Miyazaki(1981)는 Fyffe *et al.*(1968)의 예제를 구성품의 혼합 사용이 허용되는 경우로 확장하고, 최적해 탐색을 위해 대리제약(Surrogate Constraints) 기법을 이용한 휴리스틱 방법을 제시하였다. Chern and Jan(1986)은 신뢰도-중복 할당 문제에 0-1 결정 변수를 이용하여 동등하게 대수 변환한 0-1배낭문제(Knapsack Problem)를 제시하였고, 분지한계법(Branch and Bound algorithm)을 이용하여 해결하였다. Hsieh(2002)는 동일한 예제를 활용하여 최적해 탐색을 위한 효율적인 선형 근사해법(Linear Approximation)을 제시하였다. 이러한 휴리스틱 방법은 국부 최적해(Local Optima)에 도달하였을 경우 더 이상 해를 개선할 방법이 없으며, 해결하고자 하는 문제마다 각기 그 특성에 맞추어 개발해야 한다는 단점을 가지고 있다. 따라서 최근의 조합 최적화 문제의 최적해 탐색에 관한 연구에서는 이러한 휴리스틱 방법의 단점을 보완하고, 해결하고자 하는 문제가 갖는 특성에 크게 구속되지 않으며, 다양한 문제에 적용이 가능한 Genetic Algorithm (GA), Tabu Search(TS), Ant Colony Optimization(ACO), Immune Algorithm(IA) 등의 메타 휴리스틱(Meta-Heuristic) 기법을 이용한 연구가 주로 수행되고 있다. Yokota *et al.*(1996)은 중복 설계를 고려하지만 구성품의 혼합사용을 허용되지 않는 시스템의 최적 신뢰도-중복 할당 문제에 대하여 GA를 제안하였으며, 구성품의 혼합사용이 허용되는 경우에 대하여 GA(Coitt and Smith, 1996), TS(Kulturel-Konak *et al.*, 2003), ACO(Liang and Smith, 2004), IA(Chen and You, 2005)를 Fyffe *et al.*(1968)의 예제 문제에 적용하여 연구하였다.

본 연구에서는 최적해를 구하기 위하여 하나의 모집단을 운

용하거나 국부탐색을 하는 기존의 메타 휴리스틱 기법과는 달리 여러 개의 부분 모집단을 운용하여 서로 독립적이고 병렬적으로 진화함으로써 알고리즘의 효율성을 높일 수 있는 병렬 유전자 알고리즘을 적용하고, 이후 지역 탐색 알고리즘인 Swap, 2-Opt, Interchange 휴리스틱 과정을 통하여 해를 개선하는 하이브리드 병렬 유전자 알고리즘(Hybrid Parallel Genetic Algorithm, HPGA)을 제안하였다. Fyffe *et al.*(1968)의 예제를 Nakagawa and Miyazaki(1981)가 확장한 구성품의 혼합사용이 허용되는 신뢰도-중복 할당 문제에 하이브리드 병렬 유전자 알고리즘을 이용하여 구성품의 구조와 신뢰도, 비용, 중량을 산출한 후 산출 결과를 CPLEX 11.1로 구한 최적해와 비교하여 제안한 알고리즘의 성능을 평가하였으며, 메타 휴리스틱 기법을 적용한 이전의 연구들과 비교한 후 Maximum Possible Improvement(MPI)를 이용하여 신뢰도의 개선 정도를 확인하였다.

2. 수리모형의 구축

본 연구에서 다루고자 하는 신뢰도-중복 할당 문제는 동일한 기능을 갖지만 신뢰도와 비용, 중량이 서로 다른 구성품들이 비용과 중량의 제약을 만족하면서 시스템의 신뢰도를 최대화하기 위한 최적의 구성품과 중복 수를 결정하는 문제이며, <Figure 1>과 같은 직렬-병렬(Series-Parallel) 구조의 수리모형을 구축하였다. 본 모형에서 사용되는 용어와 결정변수는 다음과 같다.

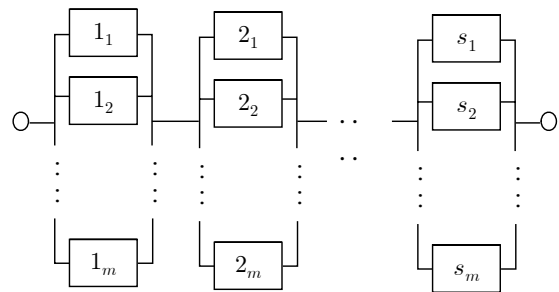


Figure 1. Series-Parallel Structure

• 용어정의

- i : 서브시스템($i = 1, 2, \dots, s$)
- j : 구성품($j = 1, 2, \dots, m$)
- R_S : 전체 시스템의 신뢰도
- R_i : 서브 시스템 i 의 신뢰도
- C_S : 전체 시스템의 가용한 비용
- W_S : 전체 시스템의 가용한 중량
- r_{ij} : 서브 시스템 i 에 사용된 구성품 j 의 신뢰도
- c_{ij} : 서브 시스템 i 에 사용된 구성품 j 의 비용
- w_{ij} : 서브 시스템 i 에 사용된 구성품 j 의 중량

u_i : 서버 시스템 i 에 사용된 구성품 수량의 상한

• 결정변수

x_{ij} : 서버 시스템 i 에 사용된 구성품 j 의 수량

직렬 구조에서 전체 시스템의 신뢰도 R_S 와 서버 시스템 i 의 신뢰도 R_i 와의 관계는 식 (1)과 같다.

$$R_S = \prod_{i=1}^s R_i \quad (1)$$

병렬 구조에서 서브시스템 i 의 신뢰도 R_i 와 서브시스템 i 에 사용된 구성품 j 의 신뢰도 r_{ij} 와의 관계는 식 (2)와 같다.

$$R_i = 1 - \prod_{j=1}^m [1 - r_{ij}]^{x_{ij}} \quad (2)$$

식 (1)과 식 (2)를 이용하여 직렬-병렬 구조를 갖는 시스템의 신뢰도를 최대화하고, 비용과 중량의 제약을 만족하는 신뢰도-중복 할당 문제의 수리모형은 다음과 같다.

$$\text{Maximize } R_S = \prod_{i=1}^s R_i = \prod_{i=1}^s \left\{ 1 - \prod_{j=1}^m [1 - r_{ij}]^{x_{ij}} \right\} \quad (3)$$

Subject to

$$\sum_{i=1}^s \sum_{j=1}^m c_{ij} \cdot x_{ij} \leq C_S \quad (4)$$

$$\sum_{i=1}^s \sum_{j=1}^m w_{ij} \cdot x_{ij} \leq W_S \quad (5)$$

$$1 \leq \sum_{j=1}^m x_{ij} \leq u_i, i = 1, 2, \dots, s \quad (6)$$

$$x_{ij} \geq 0, i = 1, 2, \dots, s, j = 1, 2, \dots, m, \text{ Integer} \quad (7)$$

목적함수 식 (3)은 직렬-병렬 구조를 갖는 시스템의 전체 신뢰도를 최대화하는 것이고, 제약함수 식 (4)와 식 (5)는 시스템의 자원제약을 의미하는 것으로 제약함수 식 (4)는 전체 시스템을 구성하고 있는 구성품들의 비용 총합이 가용한 비용을 초과할 수 없음을 의미하며, 제약함수 식 (5)는 구성품들의 중량 총합이 가용한 중량을 초과할 수 없음을 의미한다. 제약함수 식 (6)은 각 서브시스템에 사용될 수 있는 구성품 수량의 하한과 상한을 나타내는 제약이며, 제약함수 식 (7)은 결정변수의 정수제약을 나타낸다.

3. 하이브리드 병렬 유전자 알고리즘

병렬 유전자 알고리즘은 일반적인 유전자 알고리즘의 단일 모집단을 여러 개의 부분 모집단으로 나누어 격리시키고, 격리된 각 부분 모집단이 서로 독립적이면서 병렬적으로 진화해

가는 유전자 알고리즘이다. 일정한 모집단 크기에서 하나의 모집단을 이용하여 해를 찾는 것 보다는 몇 개의 서로 다른 부분 모집단으로 나누어 해를 찾는 것이 우수한 해를 빠르게 찾는 것으로 알려져 있다.

본 연구에서는 구성품의 혼합사용이 허용되고, 비용과 중량의 제약을 만족시키며, 시스템의 신뢰도를 최대화하는 신뢰도-중복 할당 문제의 최적해를 구하기 위한 하이브리드 병렬 유전자 알고리즘을 구축하였으며, 수행절차는 <Figure 2>와 같다.

3.1 유전자 표현

본 연구의 유전자 표현은 <Table 1>과 같이 정수 문자열 (Integer-String)로 표현된 일차원 배열의 형태로 나타내었다. 서브 시스템은 각 서브 시스템의 번호와 구성품의 종류를 나타내지만 불필요한 부분으로 실제 구성은 하지 않으며, 각 서브

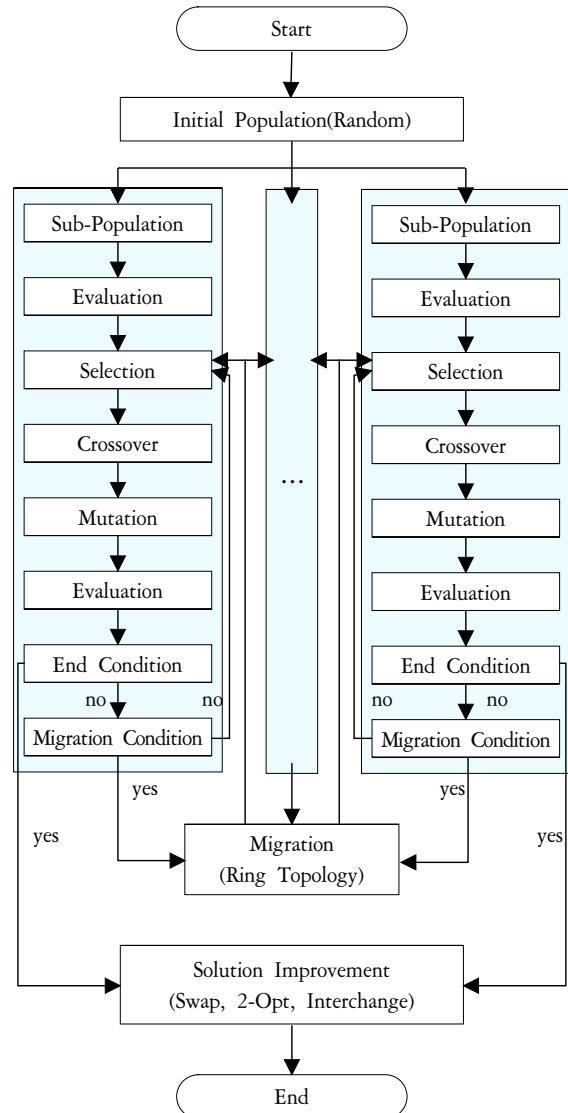


Figure 2. Hybrid Parallel Genetic Algorithm

Table 1. Gene Representation

Subsystems (Component Choices)	1(4)				2(3)			3(4)			
	2	1	1	0	1	0	2	0	1	3	0
Redundancy and Components											

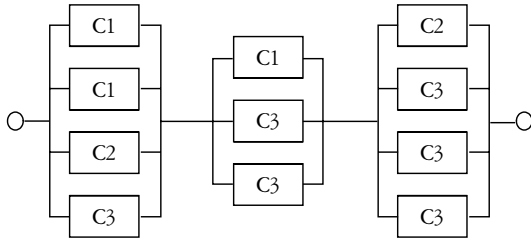


Figure 3. System Structure of Table 1

시스템을 유전 인자로 두고, 인자는 각 서브 시스템에서 가용할 수 있는 복수개의 구성품으로 표현하였다.

예를 들어, <Table 1>에서 서브 시스템 1은 C1 구성품 2개, C2 구성품 1개, C3 구성품 1개로 이루어지며, <Table 1>은 <Figure 3>과 같이 표현된다.

3.2 모집단

병렬 유전자 알고리즘의 모집단은 초기 모집단과 부분 모집단이 있다. 유전자 알고리즘은 개체들로 구성된 모집단을 통해 해의 개선을 유도하므로 초기에 모집단이 생성되어야 한다. 초기 모집단을 생성하는 방법은 문제의 특성을 이용한 발견적 기법과 임의생성 기법이 있다. 발견적 기법에 의해 생성된 해들은 초기에 수렴하여 다양한 해공간의 탐색을 방해하는 경향이 있기 때문에 본 연구에서는 임의생성 기법을 사용하였다. 100개의 해를 임의로 생성하여 초기 모집단을 구성하고, 각 부분 모집단에 20개씩 임의로 분배하였다.

3.3 적합도 평가

적합도 평가는 목적함수를 만족시키는 정도를 의미한다. 임의생성 기법에 의해 생성된 해는 실행 불가능 해를 포함할 수 있으므로 벌금함수(Penalty Function)를 사용하였으며, 식 (8)은 비용 벌금함수를 나타내고, 식 (9)는 중량 벌금함수를 나타낸다.

$$P_C = \begin{cases} 1, & \text{total cost} \leq C_s \\ \frac{1}{\text{total cost}}, & \text{otherwise} \end{cases} \quad (8)$$

$$P_W = \begin{cases} 1, & \text{total weight} \leq W_s \\ \frac{1}{\text{total weight}}, & \text{otherwise} \end{cases} \quad (9)$$

본 연구의 적합도 평가 함수는 비용과 중량의 제약을 만족

하면서 시스템의 신뢰도를 최대화하고, 실행 불가능 해가 선택되지 않도록 하기 위하여 시스템 신뢰도와 비용 및 중량 벌금함수의 곱으로 하였으며, 식 (10)과 같다.

$$fitness = R_s \cdot P_C \cdot P_W \quad (10)$$

3.4 선별

선별은 적자생존의 자연법칙에 기초하여 환경에 대한 적응도에 의해 현 세대의 모집단으로부터 다음 세대에 생존할 개체를 선택하는 과정이다. 세대별 우수한 해의 생존을 보장하고, 지역해로의 수렴을 방지하기 위하여 적합도가 작은 해의 생존을 확률적으로 보장해주는 룰렛 휠 방법을 적용하였으며, 집단 내에서 가장 강한 개체가 다음 세대로 변경되지 않고 전달되는 것을 보장하기 위하여 세대별 우수한 해는 반드시 생존할 수 있도록 하는 엘리트 보존전략을 적용하였다.

3.5 교차

교차 연산은 부모 세대의 개체를 조합하여 자손 세대를 형성하기 위한 과정으로 부모 세대보다 우수한 자손 세대를 만들어 낼 수 있다는 장점이 있다. 본 연구에서는 균등 교차(Uniform Crossover)를 적용하였으며, 그 절차는 <Figure 4>와 같다.

- 단계 1 : 각각의 개체들에게 난수를 발생시킨 후 교차율과 비교하여 교차대상자를 선정한다.
- 단계 2 : 교차대상자로 선정된 개체들에 대하여 짝을 지워준다.
- 단계 3 : 짝지어진 개체들에 대하여 비트 수 만큼 0과 1로 구성된 난수를 발생시킨다.
- 단계 4 : 난수가 1에 해당하는 두 부모 개체의 비트를 교차시켜 두 자손 개체를 생성한다.

P_1	3	5	5	1	1	3
P_2	4	2	2	3	3	5
RN	0	0	1	1	0	1
O_1	3	5	2	3	1	5
O_2	4	2	5	1	3	3

Figure 4. Uniform Crossover

3.6 돌연변이

돌연변이 연산은 의도적으로 부모 개체에 없는 형질을 갖게 하여 개체를 재생산하는 과정이며, 국부 최적해에 빠지거나 열등한 개체집단으로 수렴하는 것을 방지하는 유전 연산자로 본 연구에서는 균등 돌연변이(Uniform Mutation)를 적용하였고,

그 절차는 <Figure 5>와 같다.

단계 1: 모든 개체의 비트마다 0~1사이의 난수를 발생시킨 후 돌연변이율과 비교하여 돌연변이 대상 비트를 선정한다.

단계 2: 선정된 비트에 0과 서브 시스템 구성품 수량 상한의 여분 사이에서 난수를 발생시켜 새로운 값으로 대체한다.

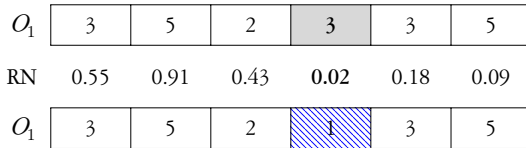


Figure 5. Uniform Mutation

3.7 이주

이주 연산은 이웃하는 부분 모집단 간에 서로 유용한 정보를 주고받기 위하여 일정 주기 또는 임의의 주기로 적용도가 높은 개체들을 교환하는 연산이다. 이웃 연산의 중요한 사항들로 이주가 일어나는 부분 모집단 간의 구성을 나타내는 이웃구조와 이주시킬 개체 및 교체될 개체의 선별, 부분 모집단의 수, 이주 주기, 이주율 등의 이주 파라미터가 있다. 본 연구에서의 이웃구조는 <Figure 6>과 같이 원형 이웃구조(Ring Topology)를 사용하였고, 이주시킬 개체 및 교체될 개체의 선별은 식 (10)의 적합도 평가 함수를 적용하였으며, 이주 파라미터는 <Table 2>와 같다.

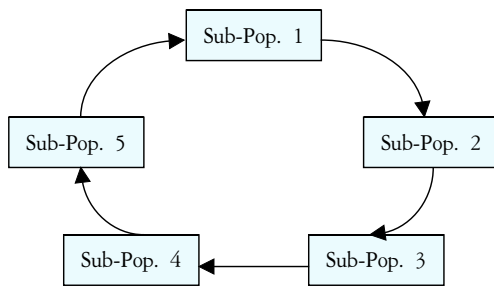


Figure 6. Neighborhood Structure(Ring Topology)

Table 2. Migration Parameters

Migration Parameters	Sub-Population Size	Migration Period	Migration Rate
Value	20	50	0.2

3.8 유전 파라미터

유전자 알고리즘이 확률과 파라미터에 의하여 다양한 결과를 발생시키기 때문에 유전자 알고리즘의 성능을 좌우하는 몇 가지 요인들을 고려해야 한다. 이러한 요인들로는 모집단 크

기(Population size), 교차 확률(P_c), 돌연변이 확률(P_m), 부호화(coding) 방법 등을 들 수 있다. 본 연구에서는 유전 파라미터들이 적합도에 미치는 영향을 반복 실험을 통해 비교한 후 알고리즘과 문제의 특성에 맞는 파라미터를 적용하였으며 본 연구의 유전 파라미터는 <Table 3>과 같다.

Table 3. Genetic Parameters

Genetic Parameters	Population Size	Crossover Rate(P_c)	Mutation Rate(P_m)	Number of Generations
Value	100	0.8	0.02	1,000~3,000

3.9 해 개선

병렬 유전자 알고리즘으로 생성된 해를 개선하고 최종해를 산출하기 위하여 Swap, 2-Opt, Interchange 기법을 적용하였다. Swap 기법은 병렬 유전자 알고리즘에서 생성된 5개의 해 중에서 2개의 해를 선택하여 하나의 비트를 상호 교환하는 방법이다. Swap 기법을 적용한 후 적합도가 가장 좋은 해를 병렬 유전자 알고리즘의 해로 선정하였다. 선정된 하나의 해를 대상으로 2개의 비트를 선택하여 상호 교환하는 2-Opt 기법을 적용하여 해를 개선하였으며, 개선된 해를 대상으로 각 서브시스템 내에서 순서를 교환하는 Interchange 기법을 적용하여 적합도가 가장 우수한 최종 해를 산출하였다.

4. 수치 실험 및 결과 분석

4.1 실험 예제

본 연구에서 제안한 하이브리드 병렬 유전자 알고리즘의 성능을 평가하기 위하여 수치 실험을 실시하였다. 실험 예제는 Fyffe *et al.*(1968)가 제안하고, Nakagawa and Miyazaki(1981)가 자원의 허용 한도를 다양하게 만든 구성품의 혼합사용이 허용되는 33개의 신뢰도-중복 할당 문제를 사용하였다. 실험 예제는 각각 병렬로 구성된 14개의 서브 시스템을 직렬로 연결한 직렬-병렬 구조로 비용제약을 130으로 고정하고, 중량제약을 191부터 159까지 문제번호가 증가될 때마다 1씩 차감하여 구성하였으며, 각 서브 시스템의 구성품 수량 상한은 6개로 하였다. 시스템 신뢰도를 구하기 위한 구성품의 데이터는 <Table 4>와 같다.

본 연구에서 비선형인 신뢰도-중복 할당 문제 수리모형의 목적함수를 선형의 0-1 배낭문제로 변환하였으며, 변환식은 식 (11)~식 (17)과 같다.

Maximize

$$\ln R_S = \sum_{i=1}^s \sum_{x_{i1}=0}^{u_i} \dots \sum_{x_{im}}^{u_i} r_{ix_{i1} \dots x_{im}} \cdot y_{ix_{i1} \dots x_{im}} \quad (11)$$

Table 4. Component Data for Testing Problems

Subsystem No.	Component Choices											
	Choice 1			Choice 2			Choice 3			Choice 4		
	R	C	W	R	C	W	R	C	W	R	C	W
1	0.90	1	3	0.93	1	4	0.91	2	2	0.95	2	5
2	0.95	2	8	0.94	1	10	0.93	1	9	*	*	*
3	0.85	2	7	0.90	3	5	0.87	1	6	0.92	4	4
4	0.83	3	5	0.87	4	6	0.85	5	4	*	*	*
5	0.94	2	4	0.93	2	3	0.95	3	5	*	*	*
6	0.99	3	5	0.98	3	4	0.97	2	5	0.96	2	4
7	0.91	4	7	0.92	4	8	0.94	5	9	*	*	*
8	0.81	3	4	0.90	5	7	0.91	6	6	*	*	*
9	0.97	2	8	0.99	3	9	0.96	4	7	0.91	3	8
10	0.83	4	6	0.85	4	5	0.90	5	6	*	*	*
11	0.94	3	5	0.95	4	6	0.96	5	6	*	*	*
12	0.79	2	4	0.82	3	5	0.85	4	6	0.90	5	7
13	0.98	2	5	0.99	3	5	0.97	2	6	*	*	*
14	0.90	4	6	0.92	4	7	0.95	5	6	0.99	6	9

* : Infeasible

Table 5. Numerical Results by using a Hybrid Parallel Genetic Algorithm(C=130)

No.	W	Components	Structure	Reliability	Cost	Weight
1	191	333, 11, 444, 3333,	222, 22, 111, 1111, 12, 233, 33, 1111, 11, 34	0.9868110	130	191
2	190	333, 11, 444, 3333,	222, 22, 111, 1111, 11, 233, 33, 1111, 12, 34	0.9864161	130	190
3	189	333, 11, 444, 3333,	222, 22, 111, 1111, 23, 233, 13, 1111, 11, 34	0.9859217	130	189
4	188	333, 11, 444, 3333,	222, 22, 111, 1111, 23, 223, 13, 1111, 12, 34	0.9853782	130	188
5	187	333, 11, 444, 3333,	222, 22, 111, 1111, 13, 223, 13, 1111, 22, 34	0.9846881	130	187
6	186	333, 11, 444, 333,	222, 22, 111, 1111, 23, 233, 33, 1111, 22, 34	0.9841755	129	186
7	185	333, 11, 444, 3333,	222, 22, 111, 1111, 23, 223, 13, 1111, 22, 33	0.9835049	130	185
8	184	333, 11, 444, 333,	222, 22, 111, 1111, 33, 233, 33, 1111, 22, 34	0.9829940	130	184
9	183	333, 11, 444, 333,	222, 22, 111, 1111, 33, 223, 33, 1111, 22, 34	0.9822557	129	183
10	182	333, 11, 444, 333,	222, 22, 111, 1111, 33, 333, 33, 1111, 22, 33	0.9815183	130	182
11	181	333, 11, 444, 333,	222, 22, 111, 1111, 33, 233, 33, 1111, 22, 33	0.9810271	129	181
12	180	333, 11, 444, 333,	222, 22, 111, 1111, 33, 223, 33, 1111, 22, 33	0.9802902	128	180
13	179	333, 11, 444, 333,	222, 22, 111, 1111, 33, 223, 13, 1111, 22, 33	0.9795047	126	179
14	178	333, 11, 444, 333,	222, 22, 111, 1111, 33, 222, 13, 1111, 22, 33	0.9784003	125	178
15	177	333, 11, 444, 333,	222, 22, 111, 113, 33, 223, 13, 1111, 22, 33	0.9775953	126	177
16	176	333, 11, 444, 333,	222, 22, 33, 1111, 33, 223, 13, 1111, 22, 33	0.9766905	124	176
17	175	333, 11, 444, 333,	222, 22, 13, 1111, 33, 223, 33, 1111, 22, 33	0.9757079	125	175
18	174	333, 11, 444, 333,	222, 22, 13, 1111, 33, 223, 13, 1111, 22, 33	0.9749261	123	174
19	173	333, 11, 444, 333,	222, 22, 13, 1111, 33, 222, 13, 1111, 22, 33	0.9738268	122	173
20	172	333, 11, 444, 333,	222, 22, 13, 113, 33, 223, 13, 1111, 22, 33	0.9730266	123	172
21	171	333, 11, 444, 333,	222, 22, 13, 113, 33, 222, 13, 1111, 22, 33	0.9719295	122	171
22	170	333, 11, 444, 333,	222, 22, 13, 113, 33, 222, 11, 1111, 22, 33	0.9707604	120	170
23	169	333, 11, 444, 333,	222, 22, 11, 113, 33, 222, 13, 1111, 22, 33	0.9692910	121	169
24	168	333, 11, 444, 333,	222, 22, 11, 113, 33, 222, 11, 1111, 22, 33	0.9681251	119	168
25	167	333, 11, 444, 333,	22, 22, 13, 113, 33, 222, 11, 1111, 22, 33	0.9663351	118	167
26	166	333, 11, 44, 333,	222, 22, 13, 113, 33, 222, 11, 1111, 22, 33	0.9650416	116	166
27	165	333, 11, 444, 333,	22, 22, 11, 113, 33, 222, 11, 1111, 22, 33	0.9637118	117	165
28	164	333, 11, 44, 333,	222, 22, 11, 113, 33, 222, 11, 1111, 22, 33	0.9624219	115	164
29	163	333, 11, 44, 333,	22, 22, 13, 113, 33, 222, 11, 1111, 22, 33	0.9606424	114	163
30	162	333, 11, 44, 333,	22, 22, 11, 113, 33, 222, 13, 1111, 22, 33	0.9591884	115	162
31	161	333, 11, 44, 333,	22, 22, 11, 113, 33, 222, 11, 1111, 22, 33	0.9580346	113	161
32	160	333, 11, 44, 333,	22, 22, 11, 111, 33, 222, 13, 1111, 22, 33	0.9557144	112	160
33	159	333, 11, 44, 333,	22, 22, 11, 111, 33, 222, 11, 1111, 22, 33	0.9545648	110	159

Subject to

$$\sum_{i=1}^s \sum_{x_{i1}=0}^{u_i} \dots \sum_{x_{im}=0}^{u_i} \left(\sum_{j=1}^m x_{ij} \cdot c_{ij} \right) \cdot y_{ix_{i1} \dots x_{im}} \leq C_S \quad (12)$$

$$\sum_{i=1}^s \sum_{x_{i1}=0}^{u_i} \dots \sum_{x_{im}=0}^{u_i} \left(\sum_{j=1}^m x_{ij} \cdot w_{ij} \right) \cdot y_{ix_{i1} \dots x_{im}} \leq W_S \quad (13)$$

$$\sum_{x_{i1}=0}^{u_i} \dots \sum_{x_{im}=0}^{u_i} y_{ix_{i1} \dots x_{im}} = 1, i = 1, 2, \dots, s \quad (14)$$

$$1 \leq \sum_{j=1}^m x_{ij} \leq u_i, i = 1, 2, \dots, s \quad (15)$$

$$x_{ij} \geq 0, i = 1, 2, \dots, s, j = 1, 2, \dots, m, \text{ Integer} \quad (16)$$

$$y_{ix_{i1} \dots x_{im}} = \begin{cases} 1, & \text{if } x_{ij} \text{ of the } j\text{th component} \\ & \text{are used for subsystem } i \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

$$\text{where, } r_{ix_{i1} \dots x_{im}} = \ln(1 - q_{i1}^{x_{i1}} q_{i2}^{x_{i2}} \dots q_{im}^{x_{im}}),$$

$$q_{im}^{x_{im}} = (1 - r_{im})^{x_{im}}, i = 1, 2, \dots, s$$

4.2 실험 결과 분석

<Table 4>의 실험 예제를 하이브리드 병렬 유전자 알고리즘에 적용하여 구성품의 구조, 신뢰도, 비용, 중량을 계산한 결과는 <Table 5>과 같다.

본 연구에서 제안한 하이브리드 병렬 유전자 알고리즘을 이용하여 얻은 신뢰도 값을 CPLEX, Coit and Smith(1996)의 GA, Kulturel-Konak *et al.*(2003)의 TS, Liang and Smith(2004)의 ACO, Chen and You(2005)의 IA 등 메타휴리스틱 기법을 이용한 연구 결과와 비교하였으며, 비교 결과는 <Table 6>과 같다.

Table 6. Comparison of CPLEX, Meta-Heuristics, and HPGA(C=130)

No.	W	Reliability					Number of Optimal by HPGA(10 runs)	
		CPLEX	GA (Coit and Smith)	TS (Kulturel-Konak <i>et al.</i>)	ACO (Liang and Smith)	IA (Chen and You)		HPGA (This Study)
1	191	0.9868110	0.9867	0.986811	0.9868	0.9868110	0.9868110	8 / 10
2	190	0.9864161	0.9857	0.986416	0.9859	0.9864161	0.9864161	7 / 10
3	189	0.9859217	0.9856	0.985922	0.9858	0.9859217	0.9859217	9 / 10
4	188	0.9853782	0.9850	0.985378	0.9853	0.9853297	0.9853782	8 / 10
5	187	0.9846881	0.9844	0.984688	0.9847	0.9844495	0.9846881	8 / 10
6	186	0.9841755	0.9836	0.984176	0.9838	0.9841755	0.9841755	9 / 10
7	185	0.9835049	0.9831	0.983505	0.9835	0.9834363	0.9835049	8 / 10
8	184	0.9829940	0.9823	0.982994	0.9830	0.9826980	0.9829940	9 / 10
9	183	0.9822557	0.9819	0.982256	0.9822	0.9822062	0.9822557	7 / 10
10	182	0.9815183	0.9811	0.981518	0.9815	0.9815183	0.9815183	9 / 10
11	181	0.9810271	0.9802	0.981027	0.9807	0.9810271	0.9810271	8 / 10
12	180	0.9802902	0.9797	0.980290	0.9803	0.9802902	0.9802902	9 / 10
13	179	0.9795047	0.9791	0.979505	0.9795	0.9795047	0.9795047	9 / 10
14	178	0.9784003	0.9783	0.978400	0.9784	0.9782085	0.9784003	7 / 10
15	177	0.9775953	0.9772	0.977474	0.9776	0.9772429	0.9775953	8 / 10
16	176	0.9766905	0.9764	0.976690	0.9765	0.9766905	0.9766905	7 / 10
17	175	0.9757079	0.9753	0.975708	0.9757	0.9757079	0.9757079	9 / 10
18	174	0.9749261	0.9744	0.974788	0.9749	0.9746901	0.9749261	6 / 10
19	173	0.9738268	0.9738	0.973827	0.9738	0.9737580	0.9738268	8 / 10
20	172	0.9730266	0.9727	0.973027	0.9730	0.9730266	0.9730266	9 / 10
21	171	0.9719295	0.9719	0.971929	0.9719	0.9719295	0.9719295	7 / 10
22	170	0.9707604	0.9708	0.970760	0.9708	0.9707604	0.9707604	9 / 10
23	169	0.9692910	0.9692	0.969291	0.9693	0.9692910	0.9692910	8 / 10
24	168	0.9681251	0.9681	0.968125	0.9681	0.9681251	0.9681251	8 / 10
25	167	0.9663351	0.9663	0.966335	0.9663	0.9663351	0.9663351	8 / 10
26	166	0.9650416	0.9650	0.965042	0.9650	0.9650416	0.9650416	9 / 10
27	165	0.9637118	0.9637	0.963712	0.9637	0.9637118	0.9637118	7 / 10
28	164	0.9624219	0.9624	0.962422	0.9624	0.9624219	0.9624219	7 / 10
29	163	0.9606424	0.9606	0.959980	0.9606	0.9606424	0.9606424	8 / 10
30	162	0.9591884	0.9591	0.958205	0.9592	0.9591884	0.9591884	6 / 10
31	161	0.9580346	0.9580	0.956922	0.9580	0.9580346	0.9580346	7 / 10
32	160	0.9557144	0.9557	0.955604	0.9557	0.9557144	0.9557144	6 / 10
33	159	0.9545648	0.9546	0.954325	0.9546	0.9545648	0.9545648	8 / 10

: Superior Solution.

본 연구에서 제안한 알고리즘이 모든 실험 예제에서 10회의 실험 중에 6~9회 CPLEX와 동일한 최적해를 제공하였으며, 메타휴리스틱 기법을 이용한 연구와 비교한 결과 본 연구가 모든 문제에서 동일하거나 우수한 해를 얻은 것으로 확인되었다. GA(1996)와 비교한 결과 본 연구가 25개의 문제에서 우수한 해를 8개의 문제에서는 동일한 결과를 얻었으며, TS(2003)에 비해 7개의 문제에서 우수한 해를 26개의 문제에서는 동일한 결과를 얻었다. ACO(2004)와 비교한 결과 본 연구가 7개의 문제에서 우수한 해를 26개의 문제에서는 동일한 결과를 얻었으며, IA(2005)에 비해 9개의 문제에서 우수한 해를 24개의 문제에서는 동일한 결과를 얻었다. 이는 하이브리드 병렬 유전자 알고리즘이 여러 개의 부분 모집단을 운용하여 서로 독립적이고 병렬적으로 진화함으로써 알고리즘의 효율성을 높일 수 있었으며, 병렬 유전자 알고리즘 이후 지역 탐색 알고리즘인 Swap, 2-Opt, Interchange 휴리스틱 과정을 통하여 해를 개선함으로써 우수한 해를 찾을 수 있었다.

기존의 메타휴리스틱 기법으로 연구한 결과를 최대한 개선할 수 있는 신뢰도 중에서 본 연구에서 개선한 정도를 나타내는 Maximum Possible Improvement(MPI)는 식 (18)~식 (21)를 이용하여 확인하였으며, 개선 결과는 <Figure 7>과 같다.

$$L1(GA) = \frac{R_{HPGA} - R_{GA}}{1 - R_{GA}} \times 100 \quad (18)$$

$$L2(TS) = \frac{R_{HPGA} - R_{TS}}{1 - R_{TS}} \times 100 \quad (19)$$

$$L3(ACO) = \frac{R_{HPGA} - R_{ACO}}{1 - R_{ACO}} \times 100 \quad (20)$$

$$L4(IA) = \frac{R_{HPGA} - R_{IA}}{1 - R_{IA}} \times 100 \quad (21)$$

- $L1(GA)$: GA 연구 결과의 MPI(%)
- $L2(TS)$: TS 연구 결과의 MPI(%)
- $L3(ACO)$: ACO 연구 결과의 MPI(%)
- $L4(IA)$: IA 연구 결과의 MPI(%)
- R_{HPGA} : HPGA를 이용하여 구한 신뢰도
- R_{GA} : GA를 이용하여 구한 신뢰도
- R_{TS} : TS를 이용하여 구한 신뢰도
- R_{ACO} : ACO를 이용하여 구한 신뢰도
- R_{IA} : IA를 이용하여 구한 신뢰도

본 연구에서 제안한 하이브리드 병렬 유전자 알고리즘은 TS를 제외한 이전의 메타휴리스틱 연구 결과를 중량이 큰 1~20번 문제에서 신뢰도를 개선하였으며, 21~33번 문제에서는 거의 동일한 결과를 얻었다. 또한 TS 연구 결과와는 1~28번 문제에서 거의 동일하였으며, 중량이 작은 29~32번 문제에서는 신뢰도를 개선하였다.

실험을 통해 하이브리드 병렬 유전자 알고리즘이 기존의 메타

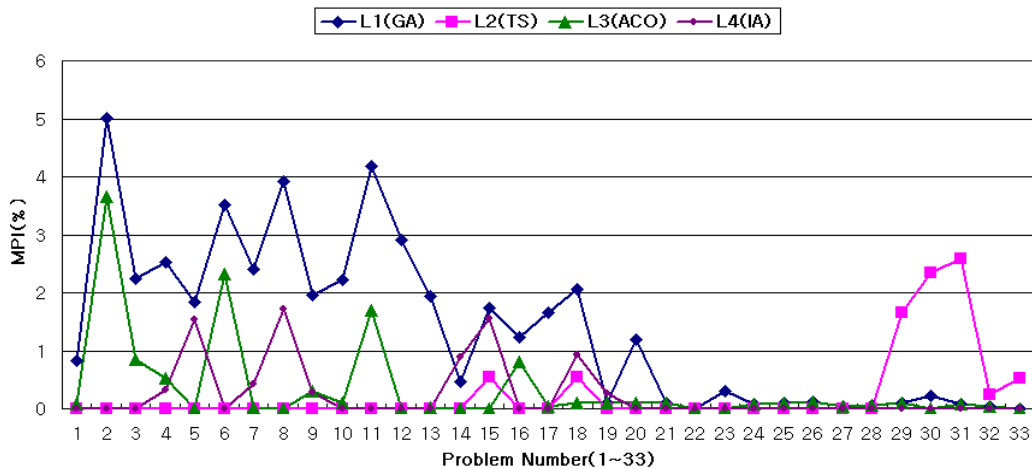


Figure 7. MPI Comparison of HPGA and Meta-Heuristics

Table 7. Numerical Results of the Large-Scale Examples

No.	Number of Subsystems	C	W	CPLEX	HPGA(10 runs)		
					Max	S.D.	Number of Optimal
1	14	130	191	0.9868110	0.9868110	0.000021	8/10
2	28	260	382	0.9740720	0.9740720	0.000147	6/10
3	42	390	573	0.9612374	0.9612374	0.000564	4/10
4	56	520	764	0.9488162	0.9488162	0.001095	1/10
5	70	650	955	0.9370413	0.9370413	0.001732	1/10

휴리스틱 기법보다 성능이 우수함을 알 수 있었으며, 대형 문제에서의 알고리즘 성능을 측정하기 위하여 실험 예제 1(C=130, W=191)을 직렬로 다시 연결한 형태로 서브시스템의 수가 28(C=260, W=382), 42(C=390, W=573), 56(C=520, W=764), 70(C=650, W=955)개인 경우의 예제를 생성하였다. 하이브리드 병렬 유전자 알고리즘을 이용하여 10회 실험한 후 CPLEX 최적해와 비교하였으며, 비교한 결과는 <Table 7>과 같다.

본 연구에서 제안한 하이브리드 병렬 유전자 알고리즘이 대형 문제에서도 CPLEX와 동일한 최적해를 제공하였으며, 14, 28, 42개의 서브시스템으로 구성된 실험 예제에서는 10회 실험 중에 4~8번, 56, 70개의 서브시스템으로 구성된 실험 예제에서는 10회 실험 중에 1번의 최적해를 각각 제공하였다.

5. 결 론

본 연구에서는 구성품의 혼합사용이 허용되는 신뢰도-중복 할당 문제에 CPLEX를 이용하여 최적해를 산출하고, 하이브리드 병렬 유전자 알고리즘을 이용한 해법을 제시하였다. 하이브리드 병렬 유전자 알고리즘을 이용하여 구성품의 구조와 신뢰도, 비용, 중량을 산출한 후 산출 결과를 CPLEX 최적해 및 메타휴리스틱 기법을 적용한 기존의 연구들과 비교하였으며, MPI를 이용하여 이전의 메타휴리스틱 기법을 적용한 연구 결과에 대한 본 연구의 신뢰도 개선 정도를 확인하였다.

본 연구에서 제안한 알고리즘이 모든 실험 예제 및 대형 문제에서 CPLEX와 동일한 최적해를 제공하였으며, 메타휴리스틱 기법을 이용한 연구와 비교한 결과 본 연구가 모든 문제에서 동일하거나 우수한 해를 얻은 것으로 확인됨으로써 알고리즘의 성능이 우수함을 알 수 있었다. 이는 하이브리드 병렬 유전자 알고리즘이 여러 개의 부분 모집단을 운용하여 서로 독립적이면서 병렬적으로 진화함으로써 알고리즘의 효율성을 높일 수 있었으며, 병렬 유전자 알고리즘을 수행한 이후 지역

탐색 알고리즘인 Swap, 2-Opt, Interchange 과정을 통하여 해를 개선함으로써 우수한 해를 찾을 수 있었다.

본 연구는 신뢰도와 가용한 자원간의 관계를 고려하여 대형화, 복잡화되는 시스템의 설계에 적용이 가능하고, 부피, 기술적인 제약 등이 추가되는 경우에 있어서도 유연하게 적용이 가능하며, 비교적 빠른 시간에 시스템의 설계와 관련된 정책 수립과 의사결정 자료로 활용될 수 있을 것이다.

참고문헌

- Chen, T. C. and You, P. S. (2005), Immune Algorithms Based Approach for Redundant Reliability Problems with Multiple Component Choices, *Computers in Industry*, **56**(2), 195-205.
- Chern, M. S. and Jan, R. H. (1986), Reliability Optimization Problems with Multiple Constraints, *IEEE Transactions on Reliability*, **35**(4), 431-436.
- Coit, D. W. and Smith, A. E. (1996), Reliability Optimization of Series-Parallel Systems using a Genetic Algorithm, *IEEE Transactions on Reliability*, **45**(2), 254-260.
- Fyffe, D. E., Hines, W. W., and Lee, N. K. (1968), System Reliability Allocation and a Computational Algorithm, *Operations Research*, **17**(2), 64-69.
- Hsieh, Y. C. (2002), A Linear Approximation for Redundant Reliability Problems with Multiple Component Choices. *Computers and Industrial Engineering*, **44**(1), 91-103.
- Kulturel-Konak, S., Smith, A. E., and Coit, D. W. (2003), Efficiently Solving the Redundancy Allocation Problem using Tabu Search, *IIE Transactions*, **35**(6), 515-526.
- Liang, Y. C. and Smith, A. E. (2004), An Ant Colony Optimization Algorithm for the Redundancy Allocation Problem, *IEEE Transactions on Reliability*, **53**(3), 417-423.
- Nakagawa, Y. and Miyazaki, S. (1981), Surrogate Constraints Algorithm for Reliability Optimization Problems with Two Constraints, *IEEE Transactions on Reliability*, **R-30**(2), 175-180.
- Yokota, T., Gen, M., and Li, X. (1996), Genetic Algorithm for Non-linear Mixed Integer Programming Problems and Its Applications, *Computers and Industrial Engineering*, **30**(4), 905-917.



김기태

해군사관학교 기계공학과 학사
국방대학교 운영분석학과 석사
현재 : 국방대학교 운영분석학과 박사과정
관심분야 : 조합최적화, 경로문제, 메타휴리스틱, 신뢰도



전건욱

공군사관학교 전자공학과 학사
고려대학교 산업공학과 석사
University of Louisville 산업공학 박사
Rutgers University 교환교수
현재 : 국방대학교 운영분석학과 부교수
관심분야 : 최적화기법 응용, 휴리스틱, 신뢰도 최적화, 셀형 제조시스템