

## 직렬시스템의 신뢰도 최적 설계를 위한 Hybrid 병렬 유전자 알고리즘 해법

김기태 · 전건욱<sup>†</sup>

국방대학교 운영분석학과

## A Hybrid Parallel Genetic Algorithm for Reliability Optimal Design of a Series System

Ki Tae Kim · Geonwook Jeon<sup>†</sup>

Department of Operations Research, Korea National Defense University

Reliability has been considered as a one of the major design measures in various industrial and military systems. The main objective is to suggest a mathematical programming model and a hybrid parallel genetic algorithm(HPGA) for the problem that determines the optimal component reliability to maximize the system reliability under cost constraint in this study. Reliability optimization problem has been known as a NP-hard problem and normally formulated as a mixed binary integer programming model. Component structure, reliability, and cost were computed by using HPGA and compared with the results of existing meta-heuristic such as Ant Colony Optimization(ACO), Simulated Annealing(SA), Tabu Search(TS) and Reoptimization Procedure. The global optimal solutions of each problem are obtained by using CPLEX 11.1. The results of suggested algorithm give the same or better solutions than existing algorithms, because the suggested algorithm could paratactically evolved by operating several sub-populations and improving solution through swap and 2-opt processes.

**Keywords** : Reliability Optimization Problem, Hybrid Parallel Genetic Algorithm, CPLEX

### 1. 서 론

오늘날 첨단 과학기술을 바탕으로 일상생활에서 사용되는 생활용품에서부터 각종 전자제품, 자동차, 선박, 항공기 등의 첨단 시스템에 이르기까지 기능과 구조가 점차 복잡화되면서 고장 발생의 우려가 높아지고 있다. 반면 소비자는 높은 품질과 신뢰도를 갖는 제품이나 시스템을 요구하고 있으며, 생산자는 시장에서의 생존을 위해 품질향상을 통한 제품의 경쟁력 강화와 신뢰도 향상을 필수적으로 고려하게 되었다. 신뢰도는 일정기간 동안 주어진 조건하에서 사용자의 요구조건에 부합하는 기

능을 원활하게 수행할 수 있는 확률을 말하며, 신뢰도가 높은 제품이나 시스템이 품질 또한 우수하다고 할 수 있다.

시스템의 신뢰도 향상이 제조상에서나 판매 이후에 수행되면 막대한 비용이 소요되므로 시스템의 설계단계에서부터 수행하는 것이 바람직하다. 신뢰도는 다양한 산업분야의 각종 시스템 설계에 있어 중요한 척도로 고려되어 왔으며, 가용한 자원을 활용하여 시스템의 최적 신뢰도를 달성할 수 있도록 개별 신뢰도를 갖는 구성품들을 선정하는 문제를 신뢰도 최적화 문제(Reliability Optimization Problem, ROP)라 한다.

본 연구는 신뢰도와 비용으로 이루어진 다양한 구성품들이 존재하며, 각 서브시스템 내에 중복(Redundancy) 설계를 고려하지 않는 단일 구성품을 사용하는 직렬시스템의 신뢰도 최적 설계 문제를 다룬다. 구성품의 신뢰도는 0과 1사이의 실수 형태, 구성품의 수량은 0과 1로 이루어진 2진 정수 형태로 표현되는 혼합 2진 정수 계획 모형(Mixed Binary Integer Programming Model)이 되며, Garey and Johnson[1]은 배낭문제(Knapsack Problem)와 동일하고, NP-hard인 조합 최적화 문제(Combinatorial Optimization Problem)라 하였다.

신뢰도 최적화 문제의 해를 구하기 위하여 다양한 연구들이 수행되었다. Sasaki et al.[7]은 목표 신뢰도를 만족하면서 비용 최소화를 위하여 구성품을 선택하는 신뢰도 최적화 문제를 해결하였으며, Sung and Chof[8]는 중복 설계를 고려하지 않고, 비용의 제약을 갖는 직렬시스템에 대한 최적해를 탐색하기 위하여 가능한 해집합에서 최적해를 구성하는데 도움이 되지 않는 불필요한 변수들을 제거하는 두 가지 특성을 적용한 새로운 분지한계법(Branch-and-Bound Method)을 제안하였다.

최근 조합 최적화 문제의 최적해 탐색에 관한 연구에서는 계산에 방대한 시간이 소요되는 최적화 기법(Exact Method)과 국부 최적해(Local Optima)에 빠지기 쉬운 휴리스틱 방법(Heuristic Method)의 단점을 보완하고, 해결하고자 하는 문제가 갖는 특성에 크게 구속되지 않으며, 다양한 문제에 적용이 가능한 유전자 알고리즘(Genetic Algorithm, GA), 개미 군집 최적화(Ant Colony Optimization, ACO), 모의 담금질(Simulated Annealing, SA), 금단 검색법(Tabu Search, TS) 등의 메타 휴리스틱(Meta-Heuristic) 기법을 이용한 연구가 주로 수행되고 있다. Kuo et al.[5]은 최근 신뢰도 최적화 문제를 다루기 위하여 메타 휴리스틱 기법이 주요하게 적용된다고 하였고, Gen et al.[2]은 신뢰도 최적화 문제를 해결하기 위하여 최초로 GA를 적용하였으며, Nahas and Nourelfath[6]는 ACO로 중복 설계를 고려하지 않는 단일 구성품을 사용하는 직렬시스템의 신뢰도 최적 설계 문제에 적용하였다. Kim et al.[4]은 SA를 이용하여 최적해를 탐색하였고, Kim et al.[3]은 Nahas and Nourelfath[6]가 제시한 직렬시스템 예제에 대하여 SA와 TS를 적용하였으며, 크기를 확장한 예제를 도출한 후 TS로 구한 최적해를 SA의 초기해로 다시 설정하여 탐색하는 재최적화 절차(Reoptimization Procedure)를 제시하였다.

본 연구에서는 Nahas and Nourelfath[6]가 제시한 다양한 신뢰도와 비용으로 이루어진 구성품들이 존재하며, 각 서브시스템 내에 중복 설계를 고려하지 않고, 단일 구성품을 사용하는 직렬시스템의 신뢰도 최적화 예제와 Kim et al.[3]이 확장한 문제에 상용 최적화 소프트웨어

인 CPLEX 11.1을 활용한 최적해를 산출과 Hybrid 병렬 유전자 알고리즘(Hybrid Parallel Genetic Algorithm, HPGA)을 이용한 해법을 제안하였다. HPGA를 이용하여 구성품의 구성과 신뢰도를 산출한 후 산출 결과를 CPLEX로 구한 최적해와 메타 휴리스틱 기법을 적용한 이전의 연구들과 비교하여 제안한 알고리즘의 성능을 평가하였다.

## 2. 수리모형의 구축

본 연구에서 다루고자 하는 신뢰도 최적화 문제는 동일한 기능을 갖지만 신뢰도와 비용이 서로 다른 다양한 구성품들이 비용의 제약을 만족하면서 각 서브시스템 내에 중복 설계를 고려하지 않고, 단일 구성품을 사용하는 직렬시스템의 신뢰도를 최대화하기 위한 최적의 구성품을 결정하는 문제이며, 본 모형에서 사용되는 용어 정의와 결정변수는 다음과 같다.

### • 용어정의

$i$  : 서브시스템( $i = 1, 2, \dots, s$ )

$j$  : 구성품( $j = 1, 2, \dots, m$ )

$R_S$  : 시스템의 신뢰도

$C_S$  : 시스템의 가용한 비용

$r_{ij}$  : 서브시스템  $i$ 에 사용된 구성품  $j$ 의 신뢰도

$c_{ij}$  : 서브시스템  $i$ 에 사용된 구성품  $j$ 의 비용

### • 결정변수

$x_{ij} = \begin{cases} 1, & \text{서브시스템 } i \text{에 구성품 } j \text{를 사용하면} \\ 0, & \text{그렇지 않으면} \end{cases}$

직렬시스템의 신뢰도를 최대화하고, 비용의 제약을 만족하는 신뢰도 최적화 문제의 수리모형은 다음과 같다.

$$\text{Maximize } R_S = \prod_{i=1}^s \left( \sum_{j=1}^m r_{ij} \cdot x_{ij} \right) \quad (1)$$

Subject to

$$\sum_{i=1}^s \sum_{j=1}^m c_{ij} \cdot x_{ij} \leq C_S \quad (2)$$

$$\sum_{j=1}^m x_{ij} = 1, \quad i = 1, 2, \dots, s \quad (3)$$

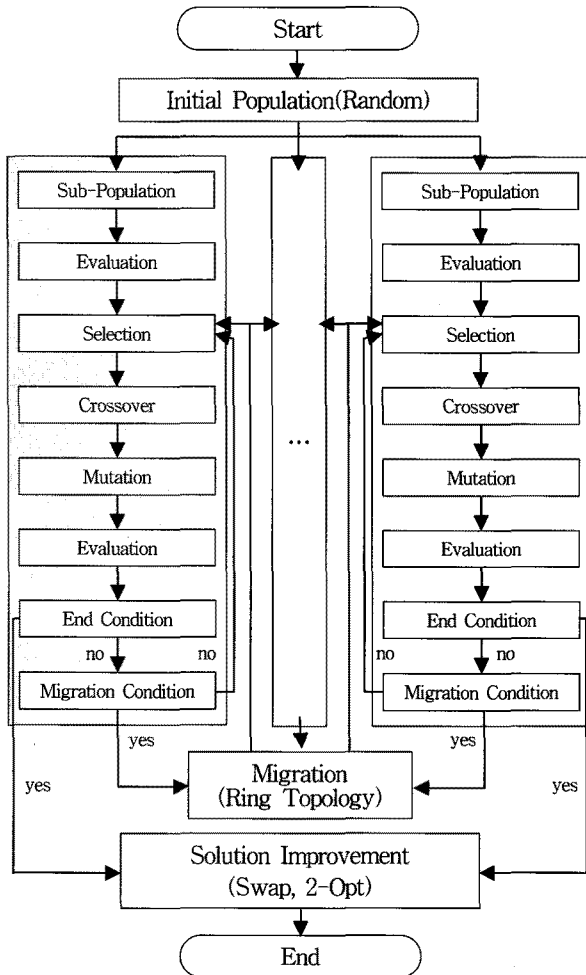
$$x_{ij} = \{0, 1\}, \quad i = 1, 2, \dots, s, \quad j = 1, 2, \dots, m \quad (4)$$

목적함수 식 (1)은 직렬시스템의 전체 신뢰도를 최대화 하는 것이고, 제약함수 식 (2)는 시스템의 자원제약을 의미하는 것으로 전체 시스템을 구성하고 있는 구성품들의 비용 총합이 가용한 비용을 초과할 수 없음의

미하며, 제약함수 식 (3)은 중복 설계를 고려하지 않는 단일 구성품을 사용하는 문제임을 나타내며, 제약함수 식 (4)는 구성품의 사용여부 의사를 결정하는 결정변수를 나타낸다.

### 3. Hybrid 병렬 유전자 알고리즘 모형

병렬 유전자 알고리즘(Parallel Genetic Algorithm, PGA)은 일반적인 GA의 단일 모집단을 여러 개의 부분 모집단으로 나누어 격리시키고, 격리된 각 부분 모집단이 서로 독립적이면서 병렬적으로 진화해 가는 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 최적해를 구하기 위하여 HPGA 모형을 구축하였으며, 세부 절차는 다음 <그림 1>과 같다.



<그림 1> Hybrid 병렬 유전자 알고리즘

### 3.1 유전자 표현

본 연구의 유전자 표현은 다음 <표 1>과 같이 정수 문자열(Integer-String)로 표현된 일차원 배열의 형태로 나타내었다. 첫 번째 행은 각 서브시스템의 번호를 나타내지만 불필요한 부분으로 실제 구성은 하지 않으며, 두 번째 행은 각 서브시스템에서 가용할 수 있는 구성품 번호로 표현하였다.

<표 1> 유전자 표현

Subsystem No.	1	2	3	4	5	6
Component No.	3	2	4	1	3	5

예를 들어 <표 1>에서 서브시스템 1은 구성품 3번, 서브시스템 2는 구성품 2번, ..., 서브시스템 6은 구성품 5번을 사용하는 것을 의미한다.

### 3.2 모집단

병렬 유전자 알고리즘(PGA)의 모집단(Population)은 초기 모집단과 부분 모집단으로 구성된다. 표준 유전자 알고리즘(GA)은 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 랜덤하게 분배하였다.

### 3.3 적합도 평가

적합도 평가(Evaluation)는 목적함수를 만족시키는 정도를 의미한다. 임의생성기법에 의해 생성된 해는 실행 불가능 해를 포함할 수 있으므로 전체 시스템의 가용한 비용을 초과하는 경우 벌금함수(Penalty Function)를 적용하였으며, 본 연구의 벌금함수는 다음 식 (5)와 같다.

$$P_C = \begin{cases} 1, & \text{total cost} \leq C_s \\ \frac{1}{\text{total cost}}, & \text{otherwise} \end{cases} \quad (5)$$

본 연구의 적합도 평가 함수는 비용의 제약을 만족하면서 시스템의 신뢰도를 최대화하고, 실행 불가능 해가

선택되지 않도록 하기 위하여 목적함수와 벌금함수의 곱으로 하였으며, 적합도 평가함수는 다음 식 (6)과 같다.

$$fitness = R_S \cdot P_C \quad (6)$$

### 3.4 선별

선별(Selection)은 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 보전전략을 적용하였다.

### 3.5 교차

교차(Crossover) 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, 세부 절차는 다음 <그림 2>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

<그림 2> 균등 교차

### 3.6 돌연변이

돌연변이(Mutation) 연산은 의도적으로 부모 개체에 없

는 형질을 갖게 하여 개체를 재생산하는 과정으로, 국부 최적해에 빠지거나 열등한 개체집단으로 수렴하는 것을 방지하는 유전 연산자이다. 본 연구에서는 균등 돌연변이(Uniform Mutation)를 적용하였으며, 세부 절차는 다음 <그림 3>과 같다.

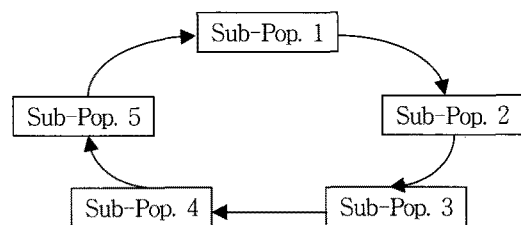
- 단계 1 : 모든 개체의 비트마다 0~1사이의 난수를 발생시킨 후 돌연변이율과 비교하여 돌연변이 대상 비트를 선정한다.
- 단계 2 : 선정된 비트에 각 서브시스템에서 가용할 수 있는 구성품 번호 중에서 난수를 발생시켜 새로운 값으로 대체한다.

$O_1$	3	5	2	3	3	5
RN	0.55	0.91	0.43	0.02	0.18	0.09
$O_1$	3	5	2	3	3	5

<그림 3> 균등 돌연변이

### 3.7 이주

이주(Migration) 연산은 이웃하는 부분 모집단 간에 서로 유용한 정보를 주고받기 위하여 일정 주기 또는 임의의 주기로 적응도가 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, 이주주기, 이주율 등의 이주 파라미터가 있다. 본 연구에서의 이웃구조는 다음 <그림 4>와 같이 원형 이웃구조(Ring Topology)를 사용하였고, 이주시킬 개체 및 교체될 개체의 선별은 식 (6)의 적합도 평가 함수를 적용하였으며, 이주 파라미터는 다음 <표 2>와 같다.



<그림 4> 원형 이웃구조

<표 2> 이주 파라미터

Migration Parameter	Sub-Pop. Size	Migration Interval	Migration Rate
Value	20	50	0.2

### 3.8 유전 파라미터

표준 유전자 알고리즘(GA)이 확률과 파라미터에 의하여 다양한 결과를 발생시키기 때문에 알고리즘의 성능을 좌우하는 몇 가지 요인들을 고려해야 한다. 이러한 요인들로는 모집단 크기(Pop-size), 교차 확률(Crossover Rate,  $P_c$ ), 돌연변이 확률(Mutation Rate,  $P_m$ ), 종료 세대수(Number of Generation) 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 다음 <표 3>과 같다.

<표 3> 유전 파라미터

Genetic Parameter	Pop. Size	Crossover Rate( $P_c$ )	Mutation Rate( $P_m$ )	Number of Generation
Value	100	0.8	0.02	1,000

### 3.9 해 개선

PGA에서 생성된 해를 개선하고 최종해를 산출하기 위하여 Swap 및 2-Opt 기법을 적용하였다. Swap 기법은 PGA에서 생성된 5개의 해 중에서 2개의 해를 선택하여 하나의 비트를 상호 교환하는 방법이다. Swap 기법을 적용한 후 적합도가 가장 우수한 해를 PGA의 해로 선정하였으며, 선정된 하나의 해에서 2개의 비트를 선택하여 상호 교환하는 2-Opt 기법을 적용한 후 적합도가 가장 우수한 해를 HPGA의 최종해로 산출하였다.

## 4. 수치 실험 및 결과 분석

### 4.1 실험 예제

본 연구에서 제안한 알고리즘의 성능을 평가하기 위하여

여 몇 가지 예제를 활용하여 실험을 실시하였다. 실험 예제는 Nahas and Nourelfath[6]의 연구에서 제시된 4가지 예제와 이 중에서 가장 복잡한 경우인 예제 4의 데이터를 Kim et al.[3]이 직렬구조 형태로 다시 연결하여 확장한 예제들로 구성된다. Nahas and Nourelfath[6]의 4가지 예제들은 각각 60, 80, 100, 166개의 구성품으로 이루어져 있고, 예제 1~3은 15개, 예제 4는 25개의 서브시스템이 직렬로 연결된 시스템이며, 전체 시스템의 가능한 비용은 각각 1000, 900, 1000, 1400이다.

CPLEX에서는 비선형 목적함수를 다룰 수 없기 때문에 본 연구의 비선형함수 형태인 목적함수 식 (1)에 대수(Logarithm)를 취하여 다음 식 (7)과 같이 선형함수로 변환하였으며, 탐색된 결과를 다시 지수 환원하여 원문제의 목적함수 값을 구하였다.

$$\begin{aligned}
 \text{Maximize } \ln R_s &= \ln \left\{ \prod_{i=1}^s \left( \sum_{j=1}^m r_{ij} \cdot x_{ij} \right) \right\} \\
 &= \sum_{i=1}^s \sum_{j=1}^m x_{ij} \cdot \ln r_{ij} \quad (7)
 \end{aligned}$$

### 4.2 실험 결과 분석

실험 예제 1~4에 대하여 CPLEX와 HPGA를 적용하여 실험하였으며, 실험 결과 산출된 구성품의 구조, 신뢰도 및 비용은 다음 <표 4>와 같다.

HPGA를 이용하여 실험한 결과 모든 예제에서 최적해를 제공하였으며, 각 예제에서의 잔여비용은 10, 0, 5, 0~5로 나타났다. 예제 4에서는 3개의 최적해를 산출하였으며, 기존 연구에서 발견하지 못한 새로운 최적해(3-3-5-2-3-2-2-3-1-2-3-4-3-1-3-3-3-4-2-3-2-2-3-1)를 도출하였다.

본 연구에서 제안한 HPGA를 활용하여 최적해 탐색 실험을 10회 실시하여 산출한 신뢰도 결과와 Nahas and Nourelfath[6]의 ACO, Kim et al.[4]의 SA, Kim et al.[3]의 TS 등 메타 휴리스틱 기법을 이용한 기존 연구의 신뢰도 결과를 비교하였으며, 다음 <표 5>와 같다.

<표 4> CPLEX와 Hybrid 병렬 유전자 알고리즘에 의한 예제 1~4 실험 결과

Example No.	Number of Subsystem	Number of Variable	$C_s$	CPLEX	HPGA		
				Reliability	Components Structure	Reliability	Cost
1	15	60	1000	<b>0.857054</b>	3-4-5-2-3-3-2-3-2-2-2-3-4-3-2	<b>0.857054</b>	990
2	15	80	900	<b>0.915042</b>	3-3-3-4-2-3-3-2-4-1-2-3-4-3-1	<b>0.915042</b>	900
3	15	100	1000	<b>0.965134</b>	3-3-4-4-3-3-2-2-3-2-2-4-4-2	<b>0.965134</b>	995
4	25	166	1400	<b>0.865439</b>	3-3-3-5-2-3-2-2-3-1-2-3-4-4-1-2-3-3-4-2-3-2-2-3-1	<b>0.865439</b>	1395
					3-3-3-5-2-3-2-2-3-1-2-3-4-3-1-3-3-3-4-2-3-2-2-3-1	<b>0.865439</b>	1395
					2-3-3-4-2-3-2-2-3-1-2-3-3-4-1-3-3-3-5-3-3-2-2-3-1	<b>0.865439</b>	1400

<표 5> 메타 휴리스틱 기법과의 실험 결과 비교(10회 실험)

Ex. No.	ACO[6] (Nahas and Nourelfath)			SA[4] (Kim et al.)			TS[3] (Kim et al.)			HPGA (This Study)		
	Max	Ave.	S.D.	Max	Ave.	S.D.	Max	Ave.	S.D.	Max	Ave.	S.D.
1	<b>0.85705</b>	0.85705	0	<b>0.85705</b>	0.85705	0	<b>0.857054</b>	0.857054	0	<b>0.857054</b>	0.857054	0
2	<b>0.91504</b>	0.91504	0	<b>0.91504</b>	0.91504	0	<b>0.915042</b>	0.915042	0	<b>0.915042</b>	0.915042	0
3	0.96512	0.96439	0.00050	0.96513	0.96503	0.00033	<b>0.965134</b>	0.965134	0	<b>0.965134</b>	0.965134	0
4	0.86543	0.86491	0.00038	0.86543	0.86536	0.00025	<b>0.865439</b>	0.865439	0	<b>0.865439</b>	0.865439	0

본 연구에서 제안한 알고리즘이 모든 예제의 10회 실험에서 편차 없이 모두 최적해를 제공하는 것으로 나타났으며, ACO와 SA보다는 우수한 결과를 도출하였고, TS와는 동일한 결과를 얻을 수 있었다. 알고리즘별 해의 탐색 시간은 ACO가 0.3초 이내, SA와 TS가 1초 이내, HPGA가 0.17초를 각각 소요하여 본 연구에서 제안한 알고리즘의 탐색속도가 빠른 것으로 나타났다.

Nahas and Nourelfath[6]의 연구에서 제시된 4가지 예제 중에서 가장 복잡한 경우인 예제 4의 데이터를 Kim et al.[3]이 직렬구조 형태로 다시 연결하여 서비스시스템의 수를 100, 200개로 확장한 예제 5, 6에 대한 확대 실험을 실시하였다. CPLEX를 활용한 예비실험으로 원문제의 비용 제약을 선형으로 증가시킬 경우 시스템의 전체 신뢰도가 낮아져 비현실적이라는 확인을 통하여 예제 5와 6의 비용 제약을 7200, 14400에서부터 시작하였으며, 비용을 각각 50씩 증가시킨 10, 15가지의 경우에 대하여 수치실험을 실시하였다. 또한 알고리즘의 우수성을 확인하기 위하여 신뢰도의 개선 정도를 나타내는 MPI(Maximum Possible Improvement)를 CPLEX 최적해를 기준으로 식 (8)을 이용하여 산출하였다.

$$\%MPI = \frac{(Max - CPLEX)}{(1 - CPLEX)} \times 100 \quad (8)$$

100개의 서브시스템이 직렬로 연결된 예제 5에 대한 HPGA의 실험을 10회 실시하여 산출한 신뢰도 결과는 다음 <표 6>과 같으며, 비용 제약을 고려한 10가지의 경우 중에서 SA와 TS를 이용한 기존 연구의 신뢰도 결과에 비해 더 많은 8번 최적해를 제공 하였고, 최적해를 제공하지 못한 경우에도 %MPI가 동일하거나 우수한 결과를 도출하였다. 알고리즘별 해의 탐색시간은 SA가 평균 12.07초, TS가 평균 4.41초, HPGA가 4.26초를 각각 소요하여 본 연구에서 제안한 알고리즘의 탐색속도가 빠른 것으로 나타났다.

서브시스템이 200개인 예제 6에 대한 실험을 10회 실시하여 산출한 신뢰도 결과는 다음 <표 7>과 같으며, 15가지의 경우 중에서 TS와 SA를 혼합한 재최적화 연구의 신뢰도 결과보다 더 많은 9번 최적해를 제공하였고, 최적해를 제공하지 못한 경우에도 %MPI가 동일하거나 우수하였다.

이는 HPGA가 여러 개의 부분 모집단을 운용하여 서

<표 6> 예제 5 실험 결과(10회 실험)

C <sub>s</sub>	CPLEX	SA[3] (Kim et al.)			TS[3] (Kim et al.)			HPGA (This Study)		
		Max	S.D.	%MPI	Max	S.D.	%MPI	Max	S.D.	%MPI
7200	<b>0.895758</b>	0.895575	0.001342	-0.1756	<b>0.895758</b>	0.000312	0	<b>0.895758</b>	0.001017	0
7250	<b>0.900167</b>	0.899438	0.001050	-0.7302	0.899984	0.000305	-0.1833	0.899984	0.000236	-0.1833
7300	<b>0.904599</b>	0.903866	0.001027	-0.7683	0.904414	0.000390	-0.1939	<b>0.904599</b>	0.000529	0
7350	<b>0.908866</b>	0.908405	0.001202	-0.5058	<b>0.908866</b>	0.000480	0	<b>0.908866</b>	0.000424	0
7400	<b>0.913154</b>	0.912601	0.000499	-0.6368	0.913064	0.000337	-0.1036	0.913114	0.000107	-0.0461
7450	<b>0.917184</b>	0.916815	0.000510	-0.4456	0.917093	0.000494	-0.1099	<b>0.917184</b>	0.000229	0
7500	<b>0.921141</b>	0.920770	0.000743	-0.4705	<b>0.921141</b>	0.000365	0	<b>0.921141</b>	0.000156	0
7550	<b>0.925023</b>	0.925023	0.000590	0	<b>0.925023</b>	0.000502	0	<b>0.925023</b>	0.000172	0
7600	<b>0.929013</b>	0.928269	0.000696	-1.0481	<b>0.929013</b>	0.000445	0	<b>0.929013</b>	0	0
7650	<b>0.931526</b>	0.931526	0.000388	0	<b>0.931526</b>	0	0	<b>0.931526</b>	0	0

<표 7> 예제 6 실험 결과(10회 실험)

$C_s$	CPLEX	TS[3] (Kim et al.)			TS+SA Reoptimization[3] (Kim et al.)			HPGA (This Study)		
		Max	S.D.	%MPI	Max	S.D.	%MPI	Max	S.D.	%MPI
14400	<b>0.802546</b>	0.802218	0.000425	-0.1661	0.802218	0	-0.1661	0.802364	0.000110	-0.0922
14450	<b>0.806496</b>	0.806167	0.000396	-0.1700	0.806167	0	-0.1700	0.806251	0.000076	-0.1266
14500	<b>0.810301</b>	0.809890	0.000519	-0.2167	0.809970	0	-0.1745	<b>0.810301</b>	0.000094	0
14550	<b>0.814290</b>	0.813792	0.000352	-0.2682	0.813792	0	-0.2682	0.813792	0.000182	-0.2682
14600	<b>0.818299</b>	0.817388	0.000391	-0.5014	0.817798	0.000053	-0.2757	0.817984	0.000003	-0.1734
14650	<b>0.822160</b>	0.821656	0.000891	-0.2834	0.821656	0	-0.2834	<b>0.822160</b>	0	0
14700	<b>0.826207</b>	0.824787	0.000709	-0.8171	0.825364	0.000142	-0.4851	0.825774	0.000325	-0.2491
14750	<b>0.830105</b>	0.829263	0.000815	-0.4956	0.829427	0.000026	-0.3991	<b>0.830105</b>	0.000407	0
14800	<b>0.833851</b>	0.833428	0.000891	-0.2546	0.833510	0.000026	-0.2052	0.833604	0.000450	-0.1487
14850	<b>0.837614</b>	0.837448	0.000824	-0.1022	<b>0.837614</b>	0	0	<b>0.837614</b>	0	0
14900	<b>0.841310</b>	0.840805	0.000786	-0.3182	<b>0.841310</b>	0.000107	0	<b>0.841310</b>	0	0
14950	<b>0.844856</b>	0.844009	0.000506	-0.5459	<b>0.844856</b>	0.000215	0	<b>0.844856</b>	0	0
15000	<b>0.848500</b>	0.848332	0.000610	-0.1109	<b>0.848500</b>	0.000027	0	<b>0.848500</b>	0	0
15050	<b>0.852076</b>	0.851991	0.000722	-0.0575	<b>0.852076</b>	0	0	<b>0.852076</b>	0	0
15100	<b>0.855751</b>	0.855582	0.000679	-0.1172	<b>0.855751</b>	0	0	<b>0.855751</b>	0	0

로 독립적이고 병렬적으로 진화함으로써 알고리즘의 효율성을 높일 수 있었으며, PGA 이후 지역 탐색 알고리즘인 Swap, 2-Opt 휴리스틱 과정을 통하여 해를 개선함으로써 우수한 해를 찾을 수 있었다.

수치실험을 통해 본 연구에서 제안한 HPGA의 성능이 우수함을 알 수 있었으며, 서브시스템의 수를 1000개로 확장한 예제 7을 생성하여 추가 확대 실험을 실시하였다. 비용 제약은 90000에서부터 시작하였으며, 비용을 1000씩 증가시킨 10가지의 경우에 대하여 10회의 수치실험을 실시하였으며, 실험을 통하여 산출한 신뢰도 결과는 다음 <표 8>과 같다.

<표 8> 예제 7 실험 결과(10회 실험)

$C_s$	CPLEX	HPGA		
		Max	S.D.	%MPI
90000	0.831082	0.830757	0.000681	-0.1924
91000	0.847706	0.846918	0.000594	-0.5174
92000	0.860003	0.859647	0.000317	-0.2543
93000	0.871659	0.871516	0.000183	-0.2228
94000	0.883369	0.883275	0.000262	-0.0806
95000	0.895226	0.895185	0.000208	-0.0391
96000	0.904832	0.904832	0.000079	0
97000	0.913836	0.913791	0.000055	-0.0522
98000	0.920716	0.920716	0.000016	0
99000	0.924869	0.924869	0	0

예제 7에 대한 HPGA의 실험을 통하여 신뢰도를 산출한 결과 3번 최적해를 제공 하였으며, %MPI를 통해 비용 제약이 커질수록 최적해에 근접한 해를 도출하는 것을 알 수 있었다.

### 5. 결론

본 연구에서는 동일한 기능을 갖지만 신뢰도와 비용이 서로 다른 다양한 구성품들이 비용의 제약을 만족하면서 각 서브시스템 내에 중복 설계를 고려하지 않고, 단일 구성품을 사용하는 직렬시스템의 신뢰도를 최대화하기 위한 최적의 구성품을 결정하는 신뢰도 최적화 문제에 대하여 CPLEX를 활용하여 최적해를 산출하였으며, HPGA를 이용한 해법을 제시하였다.

HPGA를 이용하여 구성품의 구성과 신뢰도를 산출한 후 산출 결과를 CPLEX로 구한 최적해와 메타 휴리스틱 기법을 적용한 이전의 연구들과 비교하여 제안한 알고리즘의 성능을 평가하였다.

본 연구에서 제안한 알고리즘이 모든 소형 문제에서 CPLEX와 동일한 최적해를 제공하였고, 메타 휴리스틱 기법을 이용한 기존 연구와 비교한 결과 본 연구가 동일하거나 우수한 해를 얻은 것으로 확인되었으며, 대형 문제에서 TS와 SA를 이용한 재최적화 절차보다 알고리즘 성능이 우수함을 알 수 있었다. 이는 HPGA가 여러 개의 부분 모집단을 운용하여 서로 독립적이면서 병렬

적으로 진화함으로써 알고리즘의 효율성을 높일 수 있었으며, PGA를 수행한 이후 지역 탐색 알고리즘인 Swap, 2-Opt 휴리스틱 과정을 통하여 해를 개선함으로써 우수한 해를 찾을 수 있었다.

본 연구는 신뢰도와 가용한 자원간의 관계를 고려하여 대형화, 복잡화, 고도화되는 시스템의 설계에 적용이 가능하고, 중량, 부피, 기술적인 제약 등이 추가되는 경우에 있어서도 유연하게 대응할 수 있으며, 시스템의 설계와 관련된 정책수립과 의사결정 자료로 활용될 수 있을 것이다.

### 참고문헌

- [1] Garey, M. and Johnson, D.; *Computers and Intractability : A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco, 1976.
- [2] Gen, M., Ida, K., and Taguchi, T.; "Reliability Optimization Problems : A Novel Genetic Algorithm Approach," Technical Report, ISE93-5, Ashikaga Institute of Technology, Ashikaga, Japan, 1993.
- [3] Kim, H. G., Bae, C. O., Kim, J. H., and Son, J. Y.; "Solution Methods for Reliability Optimization Problem of a Series System with Component Choices," *Journal of the Korean Institute of Industrial Engineers*, 34 (1) : 49-56, 2008.
- [4] Kim, H. G., Bae, C. O., and Paik, C. H.; "A Simulated Annealing Algorithm for the Optimal Reliability Design Problem of a Series System with Component Choices," *IE Interfaces*, 17(Special Edition) : 69-78, 2004.
- [5] Kuo, W., Prasad, V. R., Tillman, F. A., and Hwang C. L.; *Optimal Reliability Design : Fundamentals and Applications*, Cambridge University Press, UK, 2001.
- [6] Nahas, N. and Nourelfath, M.; "Ant System for Reliability Optimization of a Series System with Multiple Choice and Budget Constraints," *Reliability Engineering and System Safety*, 87(1) : 1-12, 2005.
- [7] Sasaki, G., Okada, T., and Shingsa, S.; "A New Technique to Optimize System Reliability," *IEEE Transaction on Reliability*, 32(2) : 175-182, 1983.
- [8] Sung, C. S. and Cho, Y. K.; "Reliability Optimization of a Series System with Multiple Choice and Budget Constraint," *European Journal of Operational Research*, 127(1) : 159-171, 2000.