

대안 부품을 고려한 다계층 시스템의 최적 중복 설계

정일한^{*†}

^{*} 울산과학기술대학교 산업경영과

Optimization of Redundancy Allocation in Multi Level System considering Alternative Units

IL-HAN CHUNG^{*†}

^{*} Department of Industrial Management, Ulsan college

ABSTRACT

Purpose: System consists of a lot of units with coherent function. In design phase, various units could be considered with the same function. In this study, we consider the alternative units with the same function and redundancy allocation to maximize system reliability in multi level system.

Methods: The redundancy allocation problem with the alternative units in multi level system is formulated. Memetic algorithm(MA) is proposed to optimize the redundancy allocation problem. In addition, the performance of the proposed algorithm is explained by a numerical experiment.

Results: MA showed better results than genetic algorithm(GA) and the convergence of the solutions in MA was also faster than GA. In addition, we could know from experiment that system reliability is increased and the chosen unit for redundancy allocation is changed if cost limit is increased.

Conclusion: The chose unit for redundancy allocation is changed as resource constraints. It means we need to consider the alternative units in system design. In the future, we need to consider various problem related to redundancy allocation in multi level system and develop the better method to enhance search performance.

Key Words: Redundancy, Multi Level, Memetic Algorithm, Genetic Algorithm

● Received 5 January 2015, 1st revised 26 February 2015, accepted 17 March 2015

† Corresponding Author(ihchung@uc.ac.kr)

© 2015, The Korean Society for Quality Management

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-Commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

* 이 논문은 2014년 울산과학기술대학교 교내학술연구비 지원에 의해 수행됨

1. 서론

신뢰도는 철도, 항공기, 배 등 대형 시스템의 성능을 나타내는 중요한 척도로 사용되고 있다. 일단, 이러한 시스템은 아무리 빠른 정비가 가능하더라도 운영 중에 고장이 발생하면 큰 위험과 직결될 가능성이 있기 때문에 고장이 나지 않는 것이 중요하다. 시스템의 신뢰도를 향상시키기 위해서는 동일한 기능을 가지는 여러 부품들 중 신뢰도가 높은 부품을 선별하여 사용하는 방법과 동일한 기능을 갖는 부품을 중복으로 탑재하는 방법이 있다. 이러한 두 가지 방법은 비용적 측면과 기술적 측면을 함께 고려하여 설계에 반영해야 한다. 비용적 측면의 경우 성능이 좋은 부품은 다른 부품에 비하여 가격 또한 높으며, 부품을 중복으로 탑재할 경우 중복되는 수 만큼의 비용이 또한 필요로 한다. 기술적 측면에서는 부품을 선별 또는 중복할 경우 발생하는 시스템의 공간적, 중량적 그리고 기술적 난이도에 영향을 미치게 된다. 한 개의 부품을 사용할 경우보다 여러 개가 사용될 경우에 탑재되는 시스템에 더욱 많은 공간과 무게가 필요하게 되며, 이로 인해 기술적 측면에서 고려해야 하는 요소가 더욱 많아지게 된다. 본 연구에서는 부품의 선별과 중복 요소를 고려하여 시스템의 신뢰도 최적화 방법을 다룬다.

중복구조에 대한 중복할당 최적화 문제는 NP-hard 문제로 Fyffe et al.(1968) 이후로 다양한 문제와 최적화 방법에 대하여 연구가 되었다. 중복할당 문제에서 가장 기본적으로 다루어진 문제는 m 개의 서브시스템을 가지는 시스템의 병렬-직렬 문제로 각 i 번째 서브시스템에 중복 개수 n_i 를 결정하는 문제인데, 이 때 시스템의 개발 비용, 부피, 무게 등이 제약으로 사용된다. 이러한 기본 문제를 바탕으로 각 부품의 상태가 다중 상태를 가지거나, 각 서브시스템에 동일한 기능을 하는 대안 부품을 적용하는 것 등으로 확장이 되었다. 또한, 시스템의 구조 역시 직렬, 직렬-병렬, 다양한 구조가 함께 나타나는 복합 시스템을 대상으로 중복할당 문제가 확장되어 연구가 되었다. 뿐만 아니라, 이러한 다양한 문제를 대상으로 최적해를 찾는 많은 방법들이 연구되었으며, 이에 대한 최근 연구들은 Kuo and Prasad(2000)에서 찾아 볼 수 있다.

다계층 시스템의 중복할당 문제는 Yun and Kim(2004)이 처음 제안 하였으며, Yun and Kim(2004)은 다계층 시스템에서 직계선 상에 한 개의 부품 또는 모듈만 중복이 가능한 것으로 가정하여 중복할당 최적화 문제를 다루었다. 또한, Yun et al.(2007)은 다계층시스템에서 직계선 상에 여러 개의 부품 또는 모듈이 중복 가능한 것으로 Yun and Kim(2004)의 문제를 확장하여 연구결과를 제시하였다. Pourdarvish and Ramezani(2013)은 다계층 시스템에서 대기구조에 대한 중복을 고려하여 중복할당 문제를 연구하였다. 이러한 다계층 시스템의 중복할당 문제를 최적화 위하여 Yeh(2009)는 Yun et al.(2007)에서 다룬 문제를 대상으로 2단계 이산 입자군집 최적화 알고리즘을, Jang and Kim(2011)은 타부서치를 통해 최적화 방법을 다루었다. Pourdarvish and Ramezani(2013)와 Wang et al.(2010)은 미미틱(memetic) 알고리즘을 활용하여 다계층 시스템에 대한 중복 할당 문제에 대한 최적화를 다루었으며, Kumar et al.(2009)은 시스템의 계층적 구조를 유전알고리즘에 적용하여 탐색 성능을 향상시켰다. He et al.(2013) 역시 Yun et al.(2007)에서 다룬 문제에 대해 해의 인코딩 및 탐색방법을 개선하여 탐색 성능을 더욱 향상시켰다. 위에서 언급한 바와 같이 대부분의 연구는 Yun et al.(2007)에서 다룬 문제에 대해 탐색성능을 향상시키는 연구를 실시하였다. 본 연구에서는 Yun and Kim(2004)에서 적용한 직계선 상에 한 개의 중복만이 허용되는 다계층 시스템의 모듈 중복 모형에서 대안 부품을 고려한 경우에 대해서 다룬다. 다계층 시스템의 중복할당 문제에서 대안 부품을 고려할 경우 각 부품에 대한 결정 사항이 2개(중복대상, 중복 수)에서 적용 부품이 추가된 3개가 되며, 문제의 복잡도 및 해의 공간이 커져 기존의 알고리즘 보다는 탐색의 속도 및 질을 향상 시킬 필요가 있다. 이에, 본 연구에서는 최적화를 위하여 미미틱 알고리즘을 제안하고 수치예제를 통해 알고리즘에 대한 성능을 나타낸다.

2. 다계층 시스템 모형 설계

일반적으로 시스템은 Figure 1과 같이 나무 형태로 시스템을 구성하는 부품의 상하 관계를 나타낼 수 있다. 많은 중복할당 문제에서는 Figure 1의 왼쪽 그림과 같이 최하위 부품만을 중복 설계의 대상으로 고려하지만, 다계층 시스템의 모듈 중복의 경우 오른쪽 그림과 같이 다계층 시스템 내의 모든 부품이 중복이 가능한 것으로 고려한다.

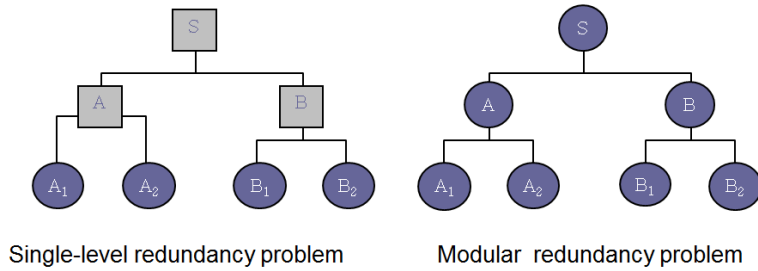


Figure 1. Structure of multi level system

단, 본 연구에서는 모든 품목이 중복 설계시 고려 대상이 되지만, 시스템의 직계에서는 단 한 개의 수준만이 중복 대상으로 선정되어지는 것을 가정한다. 즉, 직계인 $A_1 - A - S$ 에서 중복은 A_1, A, S 중 하나만이 중복 대상이 될 수 있다. 그 외 중복 설계 모형을 위한 가정은 아래와 같다.

<가정>

- (1) 각 부품들은 직렬로 연결되어 있다.
- (2) 중복 부품 및 모든 부품들이 통계적으로 독립이다.
- (3) 각 부품들의 신뢰도는 알려져 있고 확정적이다.
- (4) 각 부품과 그 부품의 대안 부품들 중에 단지 하나의 부품만이 중복 가능하다.
- (5) 직계에서 단 한 개의 수준만이 중복 대상으로 선정될 수 있다.

제안된 모형을 설계하기 위해 몇 가지 용어에 대한 정의가 필요하다. 우선 ‘품목’은 시스템, 서브시스템, 모듈, 부품을 나타내는 공통된 의미로 사용되며, 계층간의 상/하 관계에 따라 직계선 상에 대상 품목의 바로 위에 있는 품목을 부모, 모든 위에 있는 품목을 조상, 바로 아래에 있는 품목을 자식으로 나타낸다. 또한, 계층이 같으면서 같은 부모 품목을 가지는 품목을 형제, 부모 품목이 다른 경우를 사촌으로 표현한다.(Yun and Kim, 2004) 대안 부품을 고려하기 위해 위의 그림에서 A, B 등을 품목의 ‘군’으로 표현하며, 군에 사용 가능한 품목이 2종류 있는 경우 A^1, A^2, B^1, B^2 와 같이 표현한다.

<기호 및 정의>

R_s : 전체 시스템 신뢰도

i_f : i 군의 조상 품목 집합

m_i : 위치 i 에 대체 가능한 대안 부품들의 수

N : 품목 군의 수

y_i^j : i 군의 j 번째 중 품목의 사용 여부 (0, 1)

x_i^j : i 군의 j 번째 중 품목의 사용 개수

$g_{ri}^j(x_i^j)$: i 군의 j 번째 중 품목의 사용개수가 x_i^j 일 때 소모되는 자원 r 의 양($r = 1, 2, \dots, l$)

p_p^k : p 군의 k 번째 중의 품목이 지역 최적화 대상 품목으로 선택될 확률

\underline{R}_p^k : 중복할당 수를 고려하였을 경우 p 군의 k 번째 중 품목의 신뢰도

$C(x_p^k)$: 중복할당 수를 고려하였을 경우 p 군의 k 번째 중으로 인한 소요 비용

$$\text{Max } R_s = \prod_{i=1}^N \prod_{j=1}^{m_i} (1 - y_i^j (1 - R_i^j)^{x_i^j}) \quad (1)$$

subject to

$$\sum_{i=1}^N \sum_{j=1}^{m_i} [y_i^j \times g_{ri}^j(x_i^j)] \leq b_r, \quad r = 1, 2, \dots, l \quad (2)$$

$$\sum_{h=1}^{m_i} \left\{ y_i^h + \sum_{k \in \{i_j\}} \sum_{j=1}^{m_k} y_k^j \right\} = 1 \quad (3)$$

$$y_i^j = 0 \text{ 또는 } 1 \quad (4)$$

$x_i^j \geq 1$ 이고 정수이고, 식(3)에서 기호 i 는 최하위 품목을 나타냄

식(3)은 직계간에는 하나의 품목만을 사용할 있다는 것과 한 개의 군에서는 한 종의 품목만을 사용할 수 있다는 가정을, 식(2)는 자원에 대한 한계를 나타내고 있다. 본 문제는 각 부품에 대한 사용여부, 사용개수, 부품과 그 부품의 대안 부품들의 집합에 대한 사용여부를 포함한 세 가지 변수를 사용하여 시스템을 모델링 하였으며, 사용되는 자원에 대한 제약은 비용으로 비용함수의 발생식은 아래와 같다.

$$C(x_i^j) = c_i^j \times x_i^j + \lambda_j^{x_i^j} \quad (5)$$

3. 최적 중복할당을 위한 미미틱 알고리즘

미미틱 알고리즘은 혼합형 유전알고리즘의 일종으로 최적해를 구하는 과정에서 지역 최적화 알고리즘을 함께 사용하고 있다. 유전 알고리즘 과정에서 교차와 돌연변이 연산 과정 후에 이에 대한 결과를 바탕으로 주변 영역에 대해 해의 변화를 시도하게 된다. 본 연구에서는 Figure 2에 나타난 것과 같은 흐름으로 최적화를 실시한다.

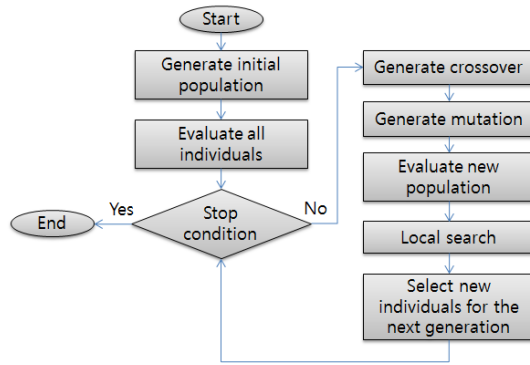


Figure 2. Flow chart of memetic algorithm

3.1 해의 표현

유전 알고리즘을 적용하기 위해서는 먼저 잠재해를 유전적 표현, 즉 염색체로 표현해야 한다. 이 유전적 표현은 유전 알고리즘의 다른 절차(적합도 평가와 유전 연산자 적용 등)에 많은 영향을 주기 때문에 문제의 특성에 맞게 적절한 방법으로 염색체를 표현해야 한다. 본 연구에서의 염색체의 표현은 다음과 같다. 즉, 아래와 같이 중복구조 대상으로 사용되는지를 나타내는 y_i^j 와 중복할당 수를 나타내는 x_i^j 의 배열로 나타낸다.

$$\nu_k = [(y_{k1}^1, x_{k1}^1), (y_{k1}^2, x_{k1}^2), (y_{k2}^1, x_{k2}^1), \dots, (y_{ki}^j, x_{ki}^j)] \quad (6)$$

이러한 표현 방법은 무엇보다도 염색체로 표현하는 부호화와 부호화된 염색체를 가능해로 나타내는 해석이 용이하므로 해를 평가하는데 소요되는 시간을 줄일 수 있다. 그리고 염색체가 균별, 종별로 구분되어, 각 부품을 하나의 유전 인자처럼 다루어 유전 연산자(교차, 돌연변이)를 수행할 수 있다. 따라서 각 부품이 가지고 있는 좋은 정보를 유지하고 효율적으로 자손에게 전파할 수 있다.

3.2 초기 모집단

유전 알고리즘에서는 정해진 수의 모집단을 운영하고, 세대를 거듭함으로써 각 염색체의 특성이 변화하게 된다. 초기 모집단은 유전 알고리즘에서 처음에 사용되는 염색체 집단으로 임의의 생성 또는 문제의 특성을 고려한 발견적 기법을 사용할 수 있다. 본 연구에서는 임의의 생성 방법으로 초기해를 생성할 경우 제약식을 만족시키는 해를 만드는 데, 상당한 시간이 소요됨으로 처음에는 최하위 품목 군에서 첫 번째 부품을 사용하는 것으로 정한다. 또한, 각 품목 군의 첫 번째 부품의 중복 개수는 적용이 가능한 부품 수의 상한을 고려해서 임의로 생성한다.

3.3 적합도 평가(fitness function)

미미틱 알고리즘에서 생성되는 모든 해는 적합한 방법에 따라 해의 질을 평가하게 된다. 생성되는 모든 해가 제약식 (2)~(4)를 모두 만족시키는 실행 가능해라면 목적함수를 해의 질을 위해 사용할 수도 있다. 본 연구에서는 제약식(3)은 항상 만족시키는 대안해를 생성하게 하고, 자원제약에 한해서는 실행 불가능해도 대안해로 생성이 가능한 것으로 고려한다. 단, 이러한 실행 불가능해에 대해서는 식(7)과 같이 벌금함수를 통해 해의 평가 결과를 나쁘게 할 수 있다.

$$eval(v_k) = \prod_{i=1}^N \prod_{j=1}^{m_i} (1 - y_{ki}^j (1 - R_i^j)^{x_{ki}^j}) - \sum_{r=1}^l Min \left\{ \left(\sum_{i=1}^N \sum_{j=0}^{m(i)} [y_{ki}^j \times g_{ri}^j(x_{ki}^j)] - b_r \right) / b_{r,0} \right\} \quad (7)$$

3.4 교차

교차는 서로 다른 두 염색체의 유전 인자들이 결합하여 자손을 생산하는 과정으로 본 연구에서는 Yun and Kim(2004)에서 활용한 것과 유사한 단순 일점교차 방법을 사용한다. 일점교차 과정에서 제약식(3)을 어길 경우에는 아래 절차에 따라 적합한 염색체로 만들게 된다.

단계 1 : 교환된 두 개의 유전자를 재교환하고 각 염색체에서 교환하고자 하는 유전자의 직계 중 y_i^j 가 1인 유전자를 찾는다.

단계 2 : y_i^j 에서 i 군의 첫 번째 품목인 j 가 1인 유전자를 찾는다.

단계 3 : 두 개의 유전자 중 높은 수준에 있는 품목을 선택해서 y_i^j 두 염색체에서 선택된 품목의 모든 군 및 중 품목과 하위 품목을 교환한다.

3.5 돌연변이

돌연변이 연산자는 유전알고리즘에서 해의 국부최적화를 예방하고, 다양한 해의 탐색을 가능하게 한다. 제약조건 (3)을 만족시키기 위하여 다음 절차를 따른다.(Yun and Kim(2004))

단계 1 : 유전자를 임의로 선택한다.(선택된 유전자를 Gene1로 나타낸다.)

단계 2 : 선택된 유전자의 y_i^j 값이 1이면 (y_i^j, x_i^j) 값을 $(1, random[1, max_i])$ 로 변경한다.

단계 3 : 선택된 유전자의 y_i^j 값이 0이고 같은 i 군에 있는 품목의 y_i^k 값이 1이면 (y_i^k, x_i^k) 를 $(0,0)$ 으로 하고 (y_i^j, x_i^j) 값을 $(1, random[1, max_i])$ 로 변경한다.

단계 4 : 선택된 유전자의 y_i^j 값이 0이면, 식(3)을 만족시키기 위해 다음 절차를 따른다.

4.1 Gene1의 직계 중에 조상 품목의 유전자의 y_i^j 값이 1인 경우 단계 4.1.1과 4.1.2를 따른다.(조상 품목의 유전자를 Gene2로 나타낸다.)

4.1.1 Gene2의 y_i^j 값을 0으로 설정한다.

4.1.2 Gene2의 자식 유전자에 대하여 각 품목군 i 중 임의의 종 j 를 선택하고 (y_i^j, x_i^j) 값을 $(1, random[1, max_i])$ 로 변경한다.

4.2 Gene1의 직계 중에 자손 품목의 유전자의 y_i^j 값이 1인 경우 단계 4.2.1과 4.2.2를 따른다.

4.2.1 Gene1에 대하여 (y_i^j, x_i^j) 값을 $(1, random[1, max_i])$ 로 변경한다.

4.2.2 Gene1의 자손 품목 유전자의 y_i^j 값을 0으로 변경한다.

3.6 국부 최적화

중복 구조에서 직계선상의 제약을 만족한 상태에서 국부 최적화를 진행한다. Wang et al.(2010)에서 적용한 방법

과 유사한 방법을 적용하되, 문제의 특성을 고려하여 변경하여 사용한다. 중복할당 대상으로 선정된 품목들 중 임의로 2개의 품목을 선택한다. 이때 각 두 품목의 선정은 아래의 식(8)의 확률로 임의로 선정하게 된다.

$$p_p^k = \frac{R_p^k / C(x_p^k)}{\sum_{i=1}^N \sum_{j=1}^{m_i} \{R_i^j / C(x_i^j)\}} \quad (8)$$

$$\underline{R}_p^k = 1 - y_i^j (1 - R_i^j)^{x_i^j} \quad (9)$$

1 단계 : $(y_i^j, x_i^j), (y_p^k, x_p^k)$ 에서 첫 번째 품목의 x_i^j 를 1 감소시킨 새로운 검색체를 생성한다. 동일한 방법으로 x_p^k 에 대해서 실시하고 새로운 검색체를 생성한다.

2 단계 : $(y_i^j, x_i^j), (y_p^k, x_p^k)$ 에서 첫 번째 품목의 x_i^j 를 1 증가시킨 새로운 검색체를 생성한다. 단, 자원의 한계를 초과할 경우 x_p^k 에서 1 감소시킨다.

3 단계 : $(y_i^j, x_i^j), (y_p^k, x_p^k)$ 에서 두 번째 품목의 x_p^k 를 1 증가시킨 새로운 검색체를 생성한다. 단, 자원의 한계를 초과할 경우 x_i^j 에서 1 감소시킨다.

4 단계 : $(y_i^j, x_i^j), (y_p^k, x_p^k)$ 로부터 각 두 개의 품목 대해 품목군은 변경하지 않고 종을 $1 \sim m_j, 1 \sim m_p$ 중에 임의로 한 개 선택한다. 선택된 종을 j', k' 이라고 하면 $y_i^{j'}$ 와 $y_p^{k'}$ 을 1로, $x_i^{j'} = x_i^j, x_p^{k'} = x_p^k$ 로 변경하고, $(y_i^{j'}, x_i^{j'}), (y_p^{k'}, x_p^{k'})$ 에 대해서 1~3단계를 실시한다.

3.6 탐색정지 조건

해의 탐색 정지를 위하여 일반적으로 반복회수(세대 수) n을 지정하는 방법과 탐색 중 해의 개선 없이 동일한 최적해가 반복적으로 n회 나타났을 경우 정지하는 방법을 적용한다. 본 연구에서는 세대 수를 지정하여 탐색을 실시하고 해의 탐색을 정지시키게 한다.

4. 실험 및 분석

제안된 알고리즘의 실험을 위해서 Yun and Kim(2004)과 Kumar et al.(2009)에서 적용한 데이터를 참고하여 사용한다. Figure 3은 계층이 3개인 시스템을 나타낸다. 시스템은 A, B, C 모듈로 구성이 되어 있으며, 모듈 A, B, C는 각각 3개, 2개, 2개의 하위 품목군으로 구성이 된다. Figure 4는 계층이 4개인 시스템으로 모듈과 부품이 15개로 구성이 되어 있다.

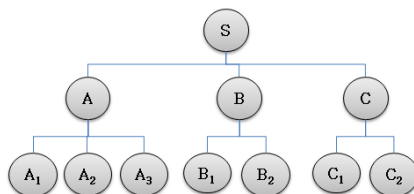


Figure 3. System structure in numerical example proposed at Yun and Kim(2004)

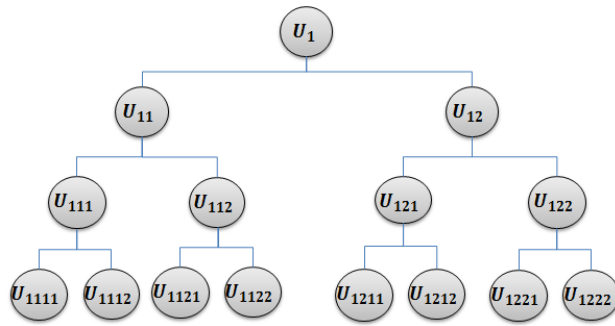


Figure 4. System structure in numerical example proposed at Kumar et al(2009)

각 품목군은 Table 1과 같이 1~3종의 대안 부품의 적용이 가능한 것으로 하였다. 시스템에서 각 품목별 비용은 $C(x) = cx + \lambda^x$ 를 따르는 것으로 하였다. 예를 들어, A^1 에 중복을 3개로 할당할 경우 A^1 로 인한 비용은 다음과 같이 계산되어진다.

$$C(3) = 26 \times 3 + 2^3 = 86$$

Table 1. Input data of three-level system

Unit	Parent unit	Reliability	Price	Additive cost
$S^1 \sim S^{12}$ (시스템)	-	0.424~0.327	72	2
A^1	$S^1 \sim S^3$	0.727	26	2
A^2	$S^4 \sim S^6$	0.689	20	2
A^3	$S^7 \sim S^9$	0.646	25	1
A^4	$S^{10} \sim S^{12}$	0.612	16	2
B^1	S^1, S^4, S^7	0.765	19	3
B^2	S^2, S^5, S^8	0.810	23	4
B^3	S^3, S^6, S^9	0.720	18	3
C	$S^1 \sim S^{12}$	0.720	21	2
A_1^1	A^1 or A^2	0.900	5	3
A_1^2	A^3 or A^4	0.800	4	2
A_2^1	A^1 or A^3	0.950	6	4
A_2^2	A^2 or A^4	0.900	4	4
A_3	$A^1 \sim A^4$	0.850	5	4
B_1	$B^1 \sim B^3$	0.900	6	4
B_2^1	B^1	0.850	7	4
B_2^2	B^2	0.900	9	5
B_2^3	B^3	0.800	6	3
C_1	C	0.900	8	3
C_2	C	0.800	7	4

첫 번째 실험은 Table 1의 데이터를 바탕으로 비용한계를 150에서 340까지 10씩 증가시키면서 미미틱 알고리즘과 유전알고리즘과의 비교를 실시한다. Table 2는 첫 번째 실험에서 각 비용한계마다 초기난수를 달리하여 30회를 실시한 결과 중 가장 좋은 해와 분산을 나타내고 있다. 비용한계 20가지 경우에서 미미틱 알고리즘의 가장 좋은 해는 all enumeration 방법의 최적해와 모두 동일한 결과를 나타내었으며, 유전알고리즘만 실시한 경우에는 4가지 경우에서 최적해보다 좋지 못한 해를 나타내었다. 또한, 미미틱 알고리즘의 분산이 유전알고리즘보다 적게 나타남을 알 수 있다.

Table 2. Comparison result between MA and GA

Cost limit	All enumeration		Memetic algorithm			GA		
	Total cost	Reliability	Total cost	Reliability	Variance	Total cost	Reliability	Variance
150	150	0.8342	150	0.8342	4.9504E-05	150	0.8342	0.00073
160	160	0.8620	160	0.8620	4.8661E-06	160	0.8620	0.00033
170	170	0.8811	170	0.8811	3.5825E-06	170	0.8708	0.000215
180	178	0.8923	178	0.8923	4.2624E-07	176	0.8871	0.000262
190	187	0.9086	187	0.9086	2.6255E-05	186	0.8961	0.00023
200	200	0.9202	200	0.9202	3.1341E-05	200	0.9202	0.000242
210	210	0.9305	210	0.9305	4.8166E-05	210	0.9305	0.000176
220	220	0.9346	220	0.9346	2.2038E-06	220	0.9333	0.000146
230	230	0.9409	230	0.9409	8.9787E-06	230	0.9409	0.000138
240	240	0.9515	240	0.9515	6.7138E-06	240	0.9515	0.000172
250	240	0.9515	240	0.9515	5.6517E-06	240	0.9515	0.000111
260	251	0.9563	251	0.9563	4.5921E-09	251	0.9563	8.57E-05
270	269	0.9669	269	0.9669	1.1651E-06	269	0.9669	0.000272
280	280	0.9717	280	0.9717	1.1670E-06	280	0.9717	0.000105
290	280	0.9717	280	0.9717	4.4541E-06	280	0.9717	8.17E-05
300	296	0.9723	296	0.9723	7.5375E-07	296	0.9723	7.35E-05
310	310	0.9755	310	0.9755	1.0898E-06	310	0.9755	6.61E-05
320	316	0.9781	316	0.9781	0	316	0.9781	5.8E-05
330	316	0.9781	316	0.9781	0	316	0.9781	5.54E-05
340	335	0.9790	335	0.9790	2.5558E-07	335	0.9790	4.94E-05

Table 3은 각 비용한계별 중복할당 결과를 나타낸다. 미미틱 알고리즘의 결과를 살펴보면, C군 품목을 비용한계 170에서는 중복할당으로 선택하지 않았다가 비용한계 180에서 부터는 첫 번째 종을 중복할당 대상 품목으로 선택하였다. 이는 C군의 모듈 중복을 3개로 하는 것이 C₁과 C₂의 중복을 3개로 하는 것 보다 비용을 낮추면서 신뢰도를 향상시키는데 유리하기 때문으로 해석이 가능하다. B₂의 경우 비용한계 180에서는 세 번째 종을 중복으로 선택하였으나, 비용한계 190에서는 첫 번째 종으로 220에서는 두 번째 종으로 변경이 되었다. 비용한계 190에서는 10만 원의 비용이 증가한 것을 신뢰도 향상으로 만들기 위해 더욱 신뢰도가 높은 첫 번째 종으로 선택한 것으로 판단되며, 220의 경우 다른 부품의 중복 수를 증가시키기 위해 다시 좀 더 비용이 낮은 두 번째 종으로 선택이 된 것으로 판단된다.

Table 3. Results of redundancy allocation

Cost limit	Algorithm	Unit	<i>S</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>A</i> ₁	<i>A</i> ₂	<i>A</i> ₃	<i>B</i> ₁	<i>B</i> ₂	<i>C</i> ₁	<i>C</i> ₂	
170	MA	Class	0	0	0	0	1	2	1	1	1	1	1	
		Number	0	0	0	0	2	2	2	2	2	2	2	
	GA	Class	0	2	1	0	0	0	0	0	0	0	1	1
		Number	0	3	2	0	0	0	0	0	0	0	2	2
180	MA	Class	0	0	0	1	2	2	1	1	3	0	0	
		Number	0	0	0	3	3	2	2	2	2	0	0	
	GA	Class	0	0	1	1	2	2	1	0	0	0	0	
		Number	0	0	2	3	3	2	2	0	0	0	0	
190	MA	Class	0	0	0	1	2	2	1	1	1	0	0	
		Number	0	0	0	3	3	2	2	2	2	0	0	
	GA	Class	0	2	1	1	0	0	0	0	0	0	0	
		Number	0	3	2	3	0	0	0	0	0	0	0	
220	MA	Class	0	0	0	1	2	2	1	1	2	0	0	
		Number	0	0	0	3	4	3	2	2	2	0	0	
	GA	Class	0	0	1	1	1	2	1	0	0	0	0	
		Number	0	0	3	3	2	3	2	0	0	0	0	

Table 4는 Appendix에 Table 5의 데이터를 바탕으로 4 수준 시스템에 대한 실험 결과를 나타낸다. 15개의 비용 조건에서 미미틱 알고리즘이 유전알고리즘 보다 3가지 경우에서 더욱 좋은 결과를 나타내었으며, 12개의 조건에서는 동일한 결과를 나타내었다.

Table 4. Results of redundancy allocation in four-level system

Cost limit	Memetic algorithm			GA		
	Total cost	Reliability	Variance	Total cost	Reliability	Variance
200	177	0.903933	0.00427	177	0.903933	0.00249
250	233	0.934064	0.00206	233	0.934064	0.00107
300	286	0.952339	0.00067	287	0.951507	0.00034
350	340	0.970123	0.00081	340	0.970123	0.00009
400	340	0.970123	0.00038	340	0.970123	0.00005
450	450	0.97128	0.00010	450	0.97128	0.00003
500	478	0.980064	0.00007	478	0.980064	0.00001
550	525	0.981796	0.00006	478	0.980064	0.00002
600	525	0.981796	0.00004	478	0.980064	0.00004
650	640	0.983907	0.00002	640	0.983907	0.00002
700	656	0.988604	0.00007	656	0.988604	0.00001
750	656	0.988604	0.00005	656	0.988604	0.00001
800	656	0.988604	0.00001	656	0.988604	0.00005
850	656	0.988604	0.00005	656	0.988604	0.00002
900	656	0.988604	0.00003	656	0.988604	0.00004

Figure 5는 Figure 3의 시스템을 대상으로 미미틱 알고리즘과 유전알고리즘의 세대별 적합도 결과를 나타낸다. 실험의 결과는 Table 2에서 실험한 각 비용한계당 30회 반복 실험 중 임의의 한 결과를 선택하여 반복회수 50회까지의 결과를 표시하였다. Figure 5로부터 미미틱 알고리즘이 유전알고리즘보다 적은 반복에서 최적해에 수렴해 가고 있음을 알 수 있다. 또한, 수렴하는데 소요되는 시간은 1회 해를 찾는데 있어 유전알고리즘은 0.2초, 미미틱 알고리즘은 1.7초의 시간이 소요되었다. 이는 미미틱 알고리즘의 국부최적화로 인해 계산시간이 추가적으로 소요되었기

때문이다. 위의 실험결과에서 볼 수 있듯이 미미틱 알고리즘의 경우 유전알고리즘보다 소요시간은 많이 소요되나, 더욱 질 좋은 결과를 산출하고 있음을 알 수 있다.

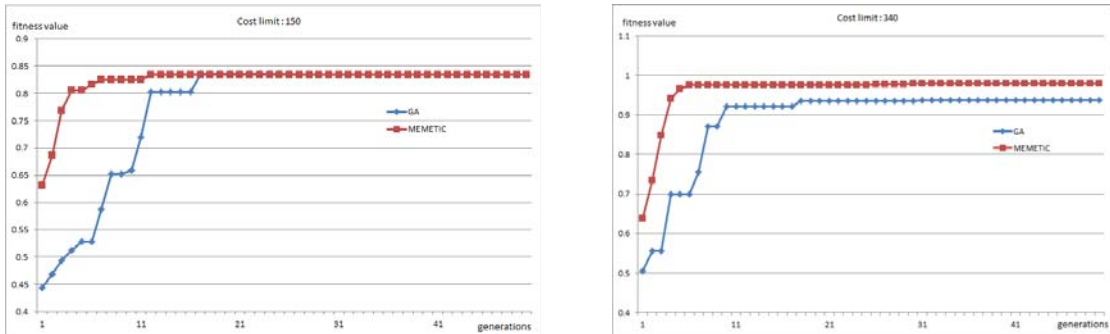


Figure 5. Comparison of evaluation values between MA and GA

5. 결론 및 추후 연구과제

본 연구에서는 대안 부품이 적용이 가능한 경우에 시스템의 최적 중복설계 방법에 대하여 다루었다. 다계층 시스템에서 대안 부품과 모듈 중복 문제를 함께 고려할 경우 최적화를 위한 해 공간은 대안부품으로 인해 크게 증가하게 된다. 이에 기존의 유전알고리즘에서 지역최적화 기법을 포함하는 미미틱 알고리즘을 제안하였다. 미미틱 알고리즘은 유전 알고리즘에 비해 더욱 좋은 해의 결과를 도출하였으며, 또한 빠른 수렴 속도를 나타내는 것을 실험을 통해 확인할 수 있었다. 모듈 중복과 관련하여서는 대안 부품을 비롯하여, 가용도 최적화, 다계층 시스템의 구성 방법 등 다양한 문제가 여전히 남아 있으며, 최적화도 다양한 방법으로 시도 및 연구가 필요할 것으로 생각된다.

REFERENCES

- Fyffe, D. E., Hines, W. W., and Lee, N. K. 1968. "System reliability allocation and a computational algorithm." Translated by Reliability. *International Journal of Industrial Engineering* 17(2):64-69.
- He, P., Wu, K., Xu, J., Wen, J., and Jiang, Z. 2013. "Multilevel redundancy allocation using two dimensional arrays encoding and hybrid genetic algorithm." *Computers and Industrial Engineering* 64:69-83.
- Jang, K. J., and Kim, J. H. 2011. "A tabu search for multiple multi-level redundancy allocation problem in series-parallel systems." *International Journal of Industrial Engineering* 18(3):120-129.
- Kumar, R., Izui, K., Yoshimura, M., and Nishiwaki, S. 2009. "Optimal multilevel redundancy allocation in series and series-parallel systems." *Computers and Industrial Engineering* 57:169-180.
- Pourdarvish, A., and Ramezani, Z. 2013. "Cold standby redundancy allocation in a multi-level series system by memetic algorithm." *International Journal of Reliability, Quality and Safety Engineering* 20(3):1-16.
- Prasad, V. R. 2000. "An annotated overview of system-reliability optimization." Translated by Reliability. *International Journal of Industrial Engineering* 49(2):176-187.
- Wang, Z., Tang, K., and Yao, X. 2010. "A memetic algorithm for multi-level redundancy allocation." Translated by Reliability. *International Journal of Industrial Engineering* 59(4):754-765.
- Yun, W. Y., and Kim, J. W. 2004. "Multi-level redundancy optimization in series systems." *Computers and Industrial Engineering* 46:337-346.

Yun, W. Y., Song, Y. M., and Kim, H. G. 2007. "Multiple multi-level redundancy allocation in series systems." Reliability Engineering and System Safety 92:308–313.

〈APPENDIX〉 실험예제 데이터

Table 5. Input data of four-level system

Unit	Reliability	Price	Additive cost	Unit	Reliability	Price	Additive cost
U_1^1	0.2198	102	2	U_{111}^1	0.7200	21	3
U_1^2	0.2068	102	2	U_{111}^2	0.6800	19	3
U_1^3	0.2041	102	2	U_{112}^1	0.7125	21	3
U_1^4	0.1921	102	2	U_{112}^2	0.6650	19	3
U_1^5	0.2051	102	2	U_{121}^1	0.6300	23	3
U_1^6	0.1931	102	2	U_{121}^2	0.5850	21	3
U_1^7	0.1905	102	2	U_{122}^1	0.6800	21	4
U_1^8	0.1793	102	2	U_{122}^2	0.6400	19	4
U_1^9	0.2076	102	2	U_{1111}^1	0.9000	7	4
U_1^{10}	0.1954	102	2	U_{1111}^2	0.8500	6	4
U_1^{11}	0.1927	102	2	U_{1112}	0.8000	6	4
U_1^{12}	0.1814	102	2	U_{1121}^1	0.7500	8	4
U_1^{13}	0.1937	102	2	U_{1121}^2	0.7000	7	4
U_1^{14}	0.1823	102	2	U_{1122}	0.9500	5	4
U_1^{15}	0.1799	102	2	U_{1211}^1	0.7000	9	4
U_1^{16}	0.1693	102	2	U_{1211}^2	0.6500	8	4
U_{11}^1	0.5130	48	2	U_{1212}	0.9000	6	4
U_{11}^2	0.4788	46	2	U_{1221}^1	0.8500	5	4
U_{11}^3	0.4845	44	2	U_{1221}^2	0.8000	4	4
U_{11}^4	0.4522	42	2	U_{1222}	0.8000	8	4
U_{12}^1	0.4284	50	2				
U_{12}^2	0.4032	48	2				
U_{13}^3	0.3978	46	2				
U_{14}^4	0.3744	44	2				