

유전자 알고리즘을 활용한 혼합 다수준 리던던시 할당문제의 신뢰성 최적화

Reliability Optimization for Multiple Multi-level Redundancy Allocation Problems using Genetic Algorithm

김 호 균¹ · 배 창 옥¹ · 윤 원 영²

동의대학교 산업경영공학과¹

부산대학교 산업공학과²

Abstract

지금까지 대부분의 리던던시 할당문제(RAP: redundancy allocation problems) 관련 연구들에서는 최상위 수준에서의 시스템 리던던시보다는 최하위 수준인 부품의 리던던시를 고려하였다. 이는 최하위 수준에서의 리던던시가 최상위 수준의 리던던시보다 효과적이라고 알려진 일반적 원리 때문이었다. 최근 한 연구에서는 동일하지 않은 예비부품을 사용하여 리던던시를 실시하는 경우 직렬구조의 시스템에서도 일반적 원리와 다른 결과가 나타날 수 있음을 보이고, 시스템을 구성하는 모든 수준에서 리던던시가 가능한 다수준 리던던시 할당문제(MRAP: multi-level RAP)를 제시하였다. 그러나 MRAP는 모든 수준에서의 리던던시를 고려하지만 단지 한 수준을 선택하여 리던던시를 할 수 있다는 가정사항을 포함하고 있다.

본 연구에서는 MRAP의 이러한 가정사항을 완화하여 시스템을 구성하는 모든 수준에서 리던던시를 위한 수준을 복수로 선택 가능한 혼합 다수준 리던던시 할당문제(MMRAP: multiple MRAP)를 제시하고 모형화하며, 문제의 해법을 위한 유전자 알고리즘(GA: genetic algorithm)을 제시한다. 제시한 GA를 활용한 몇 가지 수치실험을 통해 MMRAP 모형이 기존의 RAP 경우보다 효과적임을 입증한다.

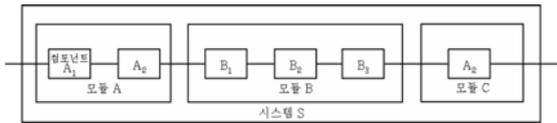
1. 서론

신뢰성(reliability)은 다양한 산업시스템에서 중요한 설계변수로 고려되고 있으며, 시장에서의 생존을 위한 기업 경쟁력 강화에 있어 제품 신뢰성의 향상은 필수라 할 수 있다. 일반적으로 제품 신뢰성의 향상이 제조상에서나 판매 후에 수행된다면 수반되는 비용이 매우 높아지므로 제품 설계단계에서 이를 수행하는 것이 바람직할 것이다. 최적 신뢰성 설계(optimal reliability design)는 시스템 설계단계에서 주어진 자원 제약 하에서 최대의 신뢰성이 확보되는 시스템 구조를 결정하는 것인데, 일반적으로 부품 자체의 신뢰성을 향상시키는

방법과 부품의 리던던시(redundancy)를 도입하는 방법이 사용된다. 여기서 전자의 방법을 신뢰성 할당(reliability allocation)이라 하며, 후자를 리던던시 할당(redundancy allocation)이라 한다. 최근에는 리던던시 할당에 관련된 연구가 신뢰성 할당에 관한 연구보다 활발히 수행되고 있는데, 이는 오늘날의 발전된 부품기술 하에서의 더 이상의 부품 신뢰성 향상이 힘들기 때문이라 할 수 있다.

다양한 구조에 대하여 리던던시 할당문제(RAP: redundancy allocation problem)가 모형화되어 왔으며, 이를 해결하기 위한 다양한 해법들이 제시되었다. 최근 연구인 Ha and Kuo(2006)는 RAP의 최적해 탐색을 위한 정수계획법(Integer Programming)을 제시하였으며, Agarwal and Gupta(2005), Ramirez-Marquez *et al.*(2004)은 휴리스틱 방법을, Hsieh(2002)는 선형근사해법(linear approximation method)을 활용하였다. RAP에 관한 최근 연구에서는 주로 메타 휴리스틱 방법을 활용하고 있으며, 시스템 내에 리던던시 가능한 아이템들의 몇 가지 대안이 존재하는 경우의 연구가 다수 이루어지고 있다. Coit *et al.*(1996a, b)과 Yokota *et al.*(1996)은 유전자 알고리즘(GA: genetic algorithm)을 활용하였으며, Liang and Smith(2004), Nahas and Nourelfath(2005)는 개미 알고리즘(ACO: ant colony optimization)을 적용하였다. 자세한 기존 연구에 대한 설명은 Kuo and Prasad(2000)와 Kuo *et al.*(2001)을 참고하길 바란다. 또한 Levitin(2002), Levitin(2003), Wu & Chan(2003)에서는 메타 휴리스틱 방법을 활용하여 다상태(multi-state)를 갖는 경우의 시스템에 관한 연구를 실시하였다. 신뢰성 할당문제에 관한 최근 연구로는 타부서치(TS: Tabu search)를 활용한 Kotz *et al.*(2003), 휴리스틱 방법을 제시한 Coit *et al.*(2004), 동적계획법(DP: dynamic programming)을 활용한 Yalaoui *et al.*(2005) 등이 있다.

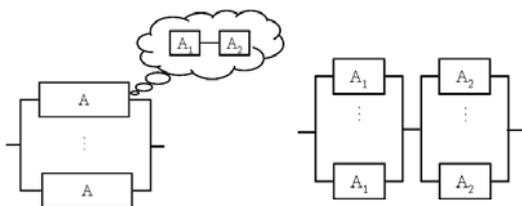
현재까지의 RAP에 관한 연구들에서는 시스템의 최하위 수준인 컴포넌트 수준의 리던던시(컴포넌트 리던던시)가 최상위수준인 시스템 수준의 리던던시(시스템 리던던시)보다 효과적이라고 알려진 일반적 원리 때문에 대부분의 연구들이 컴포넌트 리던던시에 국한되어



[그림 1] 한 개의 시스템, 3개의 모듈, 6개의 컴포넌트

왔다. 그러나 Boland and EL-Newehi(1995)는 동일하지 않은 예비부품을 이용하여 리던던시를 실시하는 경우에 직렬시스템에서도 일반적 원리와 다른 결과가 나타날 수 있음을 조사하였다. 실제 현실 시스템에서는 중간 수준인 모듈 수준에서의 리던던시(모듈 리던던시) 등 다양한 수준에서의 리던던시가 고려될 수 있을 것이며, 자원 제약 하에서 최대의 신뢰성이 달성될 수 있도록 시스템의 구조가 결정될 것이다. 이와 같이 시스템의 다양한 수준에서의 리던던시를 고려하는 RAP를 MRAP(multi-level RAP)로 나타내기로 한다.

일반적으로 MRAP에서의 시스템은 몇 개의 컴포넌트들로 구성된 모듈들이 직렬(series) 또는 병렬(parallel)로 연결되어 있고, 컴포넌트부터 시스템까지의 모든 수준에서 리던던시가 이루어 질 수 있다. 그림 1에서는 시스템 S가 A, B, C 세 개의 모듈이 직렬로 연결되어 있으며, 각 모듈은 각각의 컴포넌트들로 구성되어 있다. 모듈 A의 비용은 컴포넌트 A₁과 A₂의 비용 및 각 컴포넌트의 조립비용으로 산출될 수 있다. 그림 2와 같이 모듈 A의 리던던시와 컴포넌트 A₁, A₂의 리던던시를 고려해 보자. 두 가지 리던던시는 그 구성은 비슷하나 병직렬(parallel-series) 구조가 되는 컴포넌트 A₁과 A₂의 리던던시(component redundancy)가 컴포넌트 A₁과 A₂의 직병렬(series-parallel) 구조로 해석되는 모듈 A의 리던던시(modular redundancy)보다 신뢰성이 뛰어나게 된다. 그러나 컴포넌트 리던던시의 경우는 더 많은 시간과 기술을 요구하게 되므로 모듈 리던던시가 컴포넌트 리던던시보다 경제적이일 수도 있다.



[그림 2] 모듈 A vs. 컴포넌트 A₁, A₂의 리던던시

Yun and Kim(2004)은 이러한 MRAP의 개념을 제시하였으며, GA를 이용한 수치실험을 통해 모듈 리던던시가 컴포넌트 리던던시보다 효과적임을 입증하였다. 그러나 이 연구는 시스템 수준에서 컴포넌트 수준까지의 모든 수준에서 단지 한 수준만이 리던던시 대상으로 선택 가능하다고 가정하였다.

본 연구에서는 Yun and Kim(2004)에서 제시한 가정사항을 완화하여 시스템 내 모든 수준이 리던던시 대상으로 중복 선택 가능한 MRAP를 고려하고자 하며, 이를 MMRAP(multiple MRAP)라 칭하기로 한다. 본 연구에서는 모듈들이 직렬로 연결된 주어진 시스템 구조에서 비용제약을 만족시키면서 시스템 신뢰성을 최대화하는 MMRAP를 모형화하고 이를 해결하기 위한 GA를 제시하고자 한다. 본 연구의 구성은 다음과 같다. 2절에서는 MMRAP에 관한 기호 및 일반적 개념에 대해 언급하며, 3절에서는 MMRAP의 최적해 탐색을 위한 GA의 개념 및 관련 모수들에 관해 설명한다. 개발된 GA를 활용하여 Yun and Kim(2004)에서 사용된 예제 및 확장된 예제에 대한 수치실험 결과를 4절에서 설명하며, 마지막으로 5장에서 결론 및 추후연구에 관해 언급할 것이다.

2. 혼합 다수준 리던던시 할당문제의 모형화

본 절에서는 시스템 내 모든 수준을 리던던시 대상으로 고려하는 MMRAP에 관한 개념을 설명하고 이를 모형화할 것이다. MMRAP의 신뢰성 분석을 위해 시스템의 계층적 수준(level)을 사용하는데, 일반적으로 시스템, 모듈, 컴포넌트 등으로 구분될 수 있다. 우선 MMRAP의 모형화를 위해 사용될 몇 가지 기호들을 다음과 같이 정의한다.

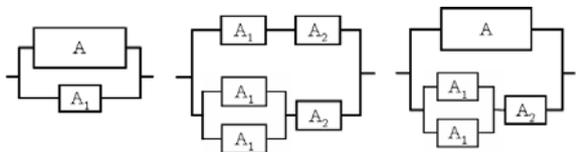
x_i :	기본 아이템 i 의 리던던시 수 기본 아이템 i 의 사용여부를 나타내는
y_i :	표시자(indicator), $\begin{cases} 0 & ; x_i=0 \\ 1 & ; x_i>0 \end{cases}$
$i_p, \{i_j\}$:	선택된 기본 아이템 i 의 부모 아이템과 자식 아이템 집합
$\{i_j\}, \{i_j\} $:	선택된 기본 아이템 i 를 포함한 형제 아이템 집합 및 수
N :	기본 아이템의 총 수
r_i :	기본 아이템 i 의 신뢰도
$R_{\{i\}}$:	하이퍼 아이템 $\{i\}$ 의 신뢰도
$g_r(x)$:	기본 아이템 i 가 추가 사용될 때 자원 r 의 사용되는 양, $r = 1, \dots, n_r$ 시스템 구조의 가능성 여부를 나타내는
f_i :	표시자(indicator), $\begin{cases} 0 & ; x_i=0 \\ 1 & ; \{i_j\} > 0 \end{cases}$
b_r :	자원 r 의 가용한 양

여기서 기본 아이템(basic item)이란 주어진 시스템에서 리던던시 대상으로 고려되는 모든 수준에서의 아이템들을 의미하며, 선택된 기본 아이템과 해당 기본 아이템의 차순위 하위수준의 기본 아이템들로 구성된 아이템을 하이퍼 아이템(hyper item)이라 정의한다. 자식 아이템(child item)은 해당 기본 아이템을 구성하는 하위수준의 기본 아이템을 말하며, 기본 아이템의 차순위 상위수준의 아이템을 부모 아이

템(parent item)이라 한다. 또한 동일한 부모 아이tem을 가지는 기본 아이tem을 형제 아이tem (sibling item)이라고 한다. 따라서 하나의 기본 아이tem은 자신의 모든 자식 아이tem들의 직렬 연결로 구성되며, 자신의 모든 형제 아이tem들과 직렬로 연결되어 부모 아이tem을 이루게 된다. 기본 아이tem의 고장은 서로 독립적으로 발생한다.

본 연구에서 고려하고자 하는 MMRAP는 시스템 내 모든 수준에서의 리던던시를 고려하지만 리던던시의 결합에 관한 규칙이 없어 기본 아이tem의 수의 증가에 따라 문제의 탐색 영역(search)이 기하급수적으로 넓어지게 된다. 따라서 불가능한 리던던시 결합 형태의 시스템 구조를 배제하고, 문제의 탐색 영역을 적절히 제한하기 위해 다음과 같은 가정 사항을 설정한다.

- (1) 모든 수준에서의 리던던시가 가능하지만 그 결합이 최상위 수준(시스템)의 기능을 만족해야 한다. 또한 임의의 기본 아이tem이 사용되려면 해당 기본 아이tem의 형제 아이tem들이 모두 사용되거나, 형제 아이tem의 모든 자식 아이tem들의 사용을 통해 형제 아이tem과 동일한 기능이 만족되어야 한다.
- (2) 리던던시를 위해 동일한 기본 아이tem이 사용된다면 그 아이tem들은 반드시 병렬(parallel)로 연결된다.



(i)불가능한 구조 (ii)불가능한 구조 (iii)가능한 구조
[그림 3] 다양한 리던던시 구조

그림 3과 같이 A₁, A₂의 자식 아이tem을 갖는 기본 아이tem A의 몇 가지 리던던시 예가 있다고 하자. 첫 번째 그림은 가정 (1)을 만족하지 못하고, 두 번째 그림은 가정 (2)에 위반된 구조로서 불가능한 구조(impossible structure)이다. 마지막 그림은 가정 (1) 및 (2)를 만족하는 가능한 구조(possible structure)이다.

기본 아이tem이 시스템에서 가능한 구조로 결합되려면 해당 기본 아이tem을 포함한 형제 아이tem들이 모두 사용되거나 또는 모두 사용되지 않아야 한다. 임의의 기본 아이tem *i*가 사용되었다면 그 기본 아이tem *i*의 모든 형제 아이tem들(*i*_j)에 대한 사용여부의 합($\sum_{j=1}^{|(i)|} y_j$)은 선택된 기본 아이tem *i*를 포함한 형제 아이tem의 수(|*i*_j|)와 동일해야 한다. 반대로, 기본 아이

tem *i*가 사용되지 않았다면 $\sum_{j=1}^{|(i)|} y_j$ 는 0이어야 한다. 만약 기본 아이tem을 포함한 형제 아이tem들이 모두 사용된다면 해당 기본 아이tem의 부모 아이tem이 사용된 것과 동일하다고 할 수 있다.

$$\sum_{j=1}^{|(i)|} y_j = f_i$$

MMRAP의 가능한 구조가 결정되면, 이제 시스템의 신뢰도가 계산되어야 한다. 신뢰도는 자식 아이tem들에 대한 하이퍼 아이tem의 신뢰도를 이용하여 계산될 수 있으며 다음 식으로 표현된다. 신뢰도는 순환적(recursive) 방법으로 최하위의 하이퍼 아이tem들에서 시작하여 최상위의 하이퍼 아이tem의 신뢰도까지 계산되며 시스템 신뢰도는 R_(S)가 된다.

$$R_{(i)} = 1 - (1-r)^{x_i} \left(1 - \prod_{j=1}^{|(i)|} R_{(j)} \right)$$

본 문제에서의 의사결정변수 *x_i*는 비음의 정수로 기본 아이tem *i*의 리던던시 수를 의미한다. 이와 같은 기호들을 이용하여 MMRAP을 모형화 한 최적화 문제는 다음과 같이 나타낼 수 있다.

$$\max R_{(S)} = 1 - (1-r)^{x_S} \left(1 - \prod_{j=1}^{|(S)|} R_{(j)} \right)$$

$$\text{subject to } \sum_{i=1}^N g_{r_i}(x_i) \leq b_r, \quad \forall r \quad (1)$$

$$\sum_{j=1}^{|(i)|} y_j = f_i, \quad \forall i \quad (2)$$

$$x_i \in \text{nonnegative integer}, \quad \forall i \quad (3)$$

$$y_i = \begin{cases} 0 & ; x_i = 0 \\ 1 & ; x_i > 0 \end{cases}, \quad \forall i \quad (4)$$

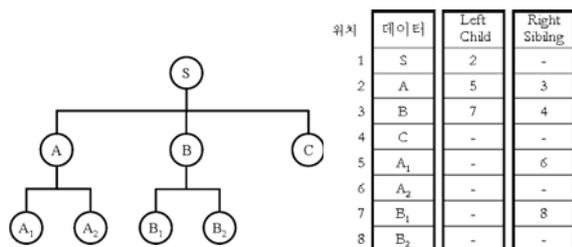
문제의 목적함수는 전체 시스템 신뢰도의 최대화이고, (1)은 비용, 무게, 부피 등과 같은 가용한 자원에 대한 제약함수를 나타낸 것이다. 그러나 본 연구에서는 비선형적 비용에 관한 제약함수, C_r(x_i) = c_rx_i + λ_r^{x_i}로 가정하였다. (2)는 시스템의 가능한 구조에 대한 제약을 나타내며, 문제의 의사결정변수에 관한 제약은 (3), 의사결정여부에 종속적인 표시자에 대한 제약은 (4)이다.

3. 유전자 알고리즘 개발

유전자 알고리즘(GA)은 생산 스케줄링(scheduling) 및 투입순서(sequencing) 결정문제, 운송계획(transportation scheduling) 문제, 운송경로(vehicle routing) 문제, 설비배치(facility layout) 문제, 신뢰성 설계문제 등을 포함하는 다양한 조합적 최적화 문제(combinatorial optimization problems)의 해법에 있어 널리 활용되고 있는 메타휴리스틱 기법 중 하나로 Holland(1975)에 의해 제시되었다. Chern(1992)은 직렬구조 시스템에서의 RAP가 조합적 최적화 문제 가운데 NP-hard 문제임을 입증하였으며, 이보다 더욱 복잡한 MRAP 및 MMRAP는 탐색영역이 매우 넓어 기본 아이템의 수가 많아지면 최적해를 구하는 것이 불가능한 문제가 된다. 본 절에서는 이러한 MMRAP의 최적해 탐색을 위한 GA를 제시하고자 하며, GA를 활용하기 위해 정의되어야 할 몇 가지 모수들에 대해 설명한다.

정의되어야 할 모수들 중 첫 단계는 문제의 해를 표현하는 방법이다. 우선 시스템의 계층적 구조(hierarchical structure)를 나타내는 방법에 대해 알아보도록 하자. 계층적인 구조를 가지는 시스템을 표현하기 위한 몇 가지 방법 중 본 연구에서는 순차적 저장방법을 사용하기로 한다. 순차적 저장방법은 다시 시스템의 계층적 관계를 해의 자리에 반영한 방법과 그렇지 않은 방법으로 구분할 수 있다. 전자의 방법은 시스템의 계층적 구조가 반영되어 있어 사용이 편리하다는 이점이 있지만, 일반적인 계층적 구조를 가지는 시스템에서는 모든 기본 아이템의 형제 아이템의 수가 동일하지 않아 불필요한 저장 공간이 발생하게 된다는 단점을 가지게 된다. 후자의 방법은 저장 공간의 효율적 사용이 가능하지만 계층적 구조를 명확하게 반영하기 어렵다는 단점이 있다.

본 연구에서는 해의 자리가 반영되지 않는 순차적 저장방법과 MMRAP의 계층적 구조를 나타내기 위해 두 개의 링크필드(left child 및 right sibling)를 함께 사용한다. 첫 번째 링크필드인 Left child는 선택된 기본 아이템의 자식 아이템 중 가장 왼쪽에 위치하는 기본 아이템에 대한 링크정보를 저장하는 곳이며, 두 번째 링크필드인 Right sibling은 선택된 기본 아이템의 오른쪽의 형제 아이템에 대한



[그림 4] 링크 필드를 이용한 시스템의 계층적 저장방법

	x_1	x_2	x_3	...	x_{N-2}	x_{N-1}	x_N
# of redundancy	1	5	3	...	5	2	1

[그림 5] 해 표현의 예

링크정보를 저장한다. 그림 4는 두 개의 링크 필드를 이용하여 계층적 구조를 가지는 시스템의 구조를 표현한 예를 나타낸 것이다.

사용된 문제의 해(v_j)는 아이템의 총 수와 동일한 자리수로 구성되며, 각 자리의 값은 기본 아이템의 리턴던시 수를 의미한다. 그림 5는 MMRAP의 신뢰성 최적 설계 문제를 위한 해 표현의 예를 나타낸 것이다.

GA를 수행하기 위해 필요한 모집단의 수 (N_p), 교차율(P_c), 돌연변이율(P_m) 및 세대수(GEN)와 같은 모수들은 알고리즘의 성능 향상을 위해 적절히 결정되어야 하는데, 일반적으로 사전실험을 통해 결정하게 된다. 사전실험은 두 단계로 구성되는데 단계 1에서는 Yun & Kim(2004)에서 사용한 $N_p=100$ 과 GEN = 100을 사용하여 P_c 및 P_m 의 값을 결정하며, 단계 2에서는 결정된 P_c 및 P_m 의 값을 이용하여 새로운 N_p 와 GEN을 결정한다. 표 1은 사전실험을 통해 결정된 GA 모수들의 값을 나타낸 것이다.

<표 1> 결정된 GA 모수

N_p	120
P_c	0.40
P_m	0.25
GEN	120

결정된 모수를 이용하여 MMRAP의 최적해 탐색을 위한 GA의 절차는 다음과 같다.

단계 1 : 초기모집단 생성

MMRAP의 해 v_j ($j = 1, \dots, N_p$)는 N개의 자리로 표현되며, 각 자리에 0부터 x_{max} 의 값 중 하나를 난수(random number)발생 방법을 이용하여 할당하게 된다. 제약함수를 이용하여 생성된 초기해의 가능성(feasibility) 여부를 검사하며, 모집단 수 N_p 만큼의 가능해로 초기모집단을 구성한다. 다음은 x_{max} 의 값을 계산하는 식을 나타낸 것이다.

$$x_{max} = \max\{x[\min c_i], x + [\min \lambda_i]x \leq C, \quad i=1, \dots, N\}$$

단계 2 : 적응도 평가

문제의 목적함수를 이용하여 생성된 모집단의 각 개체 v_j 에 대한 적응도 $eval(v_j)$ 를 평가하며, 각 개체의 적응도에 대한 누적 확률(G_j)을 산출한다.

<표 2> 예제 1 : 11개의 기본 아이টে을 가지는 시스템

basic item	parent item	r_i	G_i	λ_i
S	-	0.40029	72	2
A	S	0.72675	26	2
B	S	0.76500	19	3
C	S	0.72000	21	2
A ₁	A	0.90000	5	3
A ₂	A	0.95000	6	4
A ₃	A	0.85000	5	4
B ₁	B	0.90000	6	4
B ₂	B	0.85000	7	4
C ₁	C	0.90000	8	3
C ₂	C	0.80000	7	4

단계 3 : 선별 (selection)

[0, 1] 사이의 난수를 발생하여 그 난수가 누적 확률구간 $[q_{j-1}, q_j]$ 이면 j 번째 개체를 선택하는 확률바퀴(roulette wheel) 방법을 사용하여 선별한다. 이 과정에서 N_p 만큼 선별하여 새로운 모집단을 구성한다.

단계 4 : 교차 (crossover)

선별에 의해 생성된 새로운 모집단에 교차를 적용하는 단계이다. 모집단에 있는 각 개체 v_j 에 대해 [0, 1] 사이의 난수를 할당하고, 그 값과 교차율 P_c 를 비교하여 교차율보다 작은 난수가 발생된 개체들을 선정한다. 대상 개체의 수가 홀수개이면 선택되지 않은 개체 중 임의의 한 개체를 추가로 선정한 다음 두 개씩 쌍을 형성하며, 교차를 수행할 개체 쌍에 임의의 절단점(pos)을 다시 난수발생방법으로 선정한 후 교차를 실시한다.

단계 5 : 돌연변이 (mutation)

모집단 내의 각 개체 v_j 에 대해 [0, 1] 사이의 난수를 발생하여 돌연변이율 P_m 보다 작은 난수가 발생된 개체에 대해서 돌연변이를 수행한다.

단계 6 : 적응도 평가

교차 및 돌연변이에 의한 유전연산에 의해 새로이 구성된 모집단의 개체들에 대한 적응도를 평가한다. 생성된 개체들 가운데 비가능해에 대해서는 벌과(penalty)를 부과하여 시스템 신뢰도를 0으로 할당한다. 모집단의 각 개체에 대해 적응도를 평가한 후 각 개체의 적응도에 대해 누적 확률을 구한다.

단계 7 : 종료조건

종료조건인 세대수(GEN=120)를 만족하면 끝낸다. 그렇지 않으면 단계 3으로 이동하여 종료조건이 만족될 때까지 알고리즘을 반복 수행한다.

<표 3> 예제 2 : 16개의 기본 아이টে을 가지는 시스템

basic item	parent item	r_i	G_i	λ_i
S	-	0.3338	130	2
A	S	0.7650	15	2
B	S	0.8572	32	3
C	S	0.7650	14	2
D	S	0.8200	25	2
E	S	0.8114	24	3
A ₁	A	0.9000	6	3
A ₂	A	0.8500	5	3
B ₁	B	0.9600	8	5
B ₂	B	0.9500	7	4
B ₃	B	0.9400	6	3
C ₁	C	0.9000	5	3
C ₂	C	0.8500	4	3
E ₁	E	0.9200	7	4
E ₂	E	0.9000	5	3
E ₃	E	0.9800	3	3

<표 4> 예제 1에 대한 수치실험결과

C	Yun & Kim (2004)						실험결과																				
	reliability	cost	solution						reliability	cost	solution																
150	0.8056	149	0	2	2	2	0	0	0	0	0	0	0	0	0	0.8056	143	0	1	1	1	1	1	1	1	1	1
160	0.8255	157	0	0	2	0	2	1	2	0	0	2	2			0.8316	160	0	1	1	1	1	1	2	1	1	1
170	0.8511	169	0	2	0	0	0	0	0	2	2	2	2			0.8576	170	0	1	1	0	1	1	2	1	1	2
180	0.8668	175	0	0	2	0	2	2	2	0	0	2	2			0.8773	180	0	0	1	1	2	2	2	1	1	1
190	0.8878	186	0	0	0	0	2	2	2	2	2	2	2			0.8920	188	0	0	2	2	2	2	2	0	0	1
200	0.9010	199	0	3	0	0	0	0	0	2	2	2	2			0.9136	199	0	0	0	2	2	2	2	2	1	1
210	0.9136	202	0	0	0	3	2	2	2	2	2	0	0			0.9319	209	0	0	2	2	2	2	2	1	1	1
220	0.9272	215	0	3	0	3	0	0	0	2	2	0	0			0.9457	219	0	2	2	2	1	1	1	1	1	1
230	0.9272	215	0	3	0	3	0	0	0	2	2	0	0			0.9535	229	0	2	2	1	1	1	1	1	1	2
240	0.9319	228	0	0	3	3	2	2	2	0	0	0	0			0.9587	240	0	2	2	1	2	1	1	1	1	2
250	0.9457	241	0	3	3	3	0	0	0	0	0	0	0			0.9641	247	0	1	2	1	2	2	2	1	1	2
260	0.9469	257	0	0	3	4	2	2	2	0	0	0	0			0.9694	259	0	1	1	1	2	2	2	2	2	2
270	0.9609	270	0	3	3	4	0	0	0	0	0	0	0			0.9739	270	0	1	2	2	2	2	2	1	1	2
280	0.9609	270	0	3	3	4	0	0	0	0	0	0	0			0.9773	280	0	2	2	2	2	1	2	1	1	2
290	0.9609	270	0	3	3	4	0	0	0	0	0	0	0			0.9808	286	0	3	1	3	1	1	1	2	2	1
300	0.9608	270	0	3	3	4	0	0	0	0	0	0	0			0.9835	299	0	2	2	2	2	1	2	2	2	2
310	0.9754	304	0	4	3	4	0	0	0	0	0	0	0			0.9861	309	0	2	2	3	2	1	2	2	2	1
320	0.9754	304	0	4	3	4	0	0	0	0	0	0	0			0.9888	319	0	3	2	2	1	1	1	2	2	2
330	0.9754	304	0	4	3	4	0	0	0	0	0	0	0			0.9903	330	0	3	2	2	2	1	1	2	2	2
340	0.9754	304	0	4	3	4	0	0	0	0	0	0	0			0.9918	335	0	2	2	2	2	2	2	2	2	2

4. 수치실험

본 절에서는 MMRAP의 몇 가지 예제를 이용하여 개발된 GA의 성능평가를 위한 수치실험을 실시한다. 제시된 GA는 프로그램 언어 C++로 코딩되었으며, 펜티엄 IV 2.0 GHz IBM-PC에서 모든 예제에 대해 동일한 조건 하에서 10회 반복 실험을 수행한다.

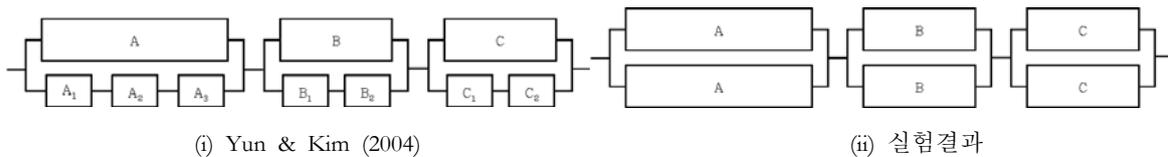
본 연구는 Yun & Kim(2004)의 연구에서 사용된 예제(예제 1) 및 확장된 예제(예제 2)로 실험되었다. 예제 1은 기본 아이템의 수가 11개이고, 시스템 비용의 상한값 C가 150~340으로 총 20가지의 하위문제로 구성된다. 예제 2는 16개의 기본 아이টে으로 구성되고, C의 범위가 260~350로 10가지의 하위문제로 구성된다. 표 2~3은 수치실험에 사용될 문제의 예제 1~2에 대한 데이터를 나타낸 것이다.

예제 1의 수치실험 결과를 통해 본 연구가 Yun & Kim(2004)보다 우수한 신뢰성을 제공하는 것으로 나타났다. 표 4는 예제 1의 20가지 문제에 대한 신뢰도, 비용 및 해를 나타내었다. Yun & Kim(2004)에서는 시스템 비용의 상한값 C가 150, 250 및 270보다 큰 경우의 해만 시스템을 구성하는 수준이 모듈수준의 리던던시의

형태로 나타났다. 예제 1에 대한 본 연구결과와 대부분이 모듈 리던던시와 컴포넌트 리던던시의 혼합 형태로 나타났고, 대부분이 Yun & Kim(2004)보다 목적함수인 전체 신뢰도에 대해 뛰어난 해를 제공하였다. <그림 6>은 예제 1의 C=150인 경우에 Yun & Kim(2004) 및 실험결과에 대한 해를 표현한 것이다.

<그림 7>은 예제 2에 대한 Yun & Kim(2004) 및 실험결과의 신뢰도를 비교한 것으로, 대부분의 경우, 본 연구의 실험결과가 Yun & Kim(2004)보다 더 좋은 신뢰도를 탐색하였다. 신뢰도가 가장 차이가 많이 나는 경우는 C=260으로, MMRAP의 신뢰도는 0.8859이고 Yun& Kim(2004)은 0.8714로 약 1.66%의 차이를 보이고 있다. 그림에서 C=340인 경우에는 신뢰도의 차이 가장 적은 경우로 0.9598과 0.9592로 약 0.1%의 차이를 보여주고 있다. 예제 2에 대해 MMRAP 와 Yun & Kim(2004)의 신뢰도에서 MMRAP의 신뢰도가 평균적으로 0.65%의 좋은 결과를 나타내었다.

실험결과를 바탕으로 본 연구에서 제시한 MMRAP가 Yun & Kim(2004)의 MRAP보다 최대의 신뢰성을 갖는 시스템 구조를 제시하는 것을 알 수 있었다.



[그림 6] 예제 1의 C=150일 때 시스템 구조 표현

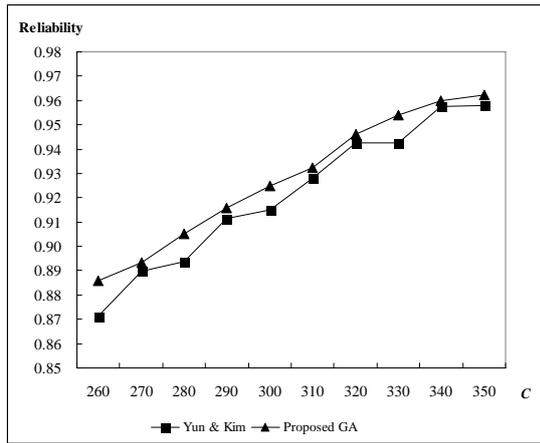


그림 7. 예제 2에 대한 실험결과

5. 결론 및 추후연구

이전까지의 리던던시 할당문제에서는 최하 위수준인 컴포넌트 리던던시에 관한 연구가 대부분이었다. 본 연구에서는 시스템 내 모든 수준에서의 리던던시가 가능하지만 한 수준만이 선택 가능한 MRAP의 가정사항을 완화하여 시스템 내 모든 수준이 리던던시 대상으로 선택 가능한 MMRAP를 제시하고 모형화하였다. 모형화 한 MMRAP와 MRAP의 수치실험을 위해 GA를 개발하였으며, 수치실험 결과를 통해 대부분의 문제에서 MMRAP가 MRAP의 경우보다 동일한 자원제약 하에서 높은 신뢰도를 가지는 시스템 구조를 가진다는 것을 보여주었다.

MMRAP를 위한 가정사항의 완화, 보다 다양하고 복잡한 예제 도출 및 알고리즘 적용과 타 메타휴리스틱 기법과의 성능비교는 추후 연구로 남겨두기로 한다.

참고 문헌

[1] Agarwal, M. and Gupta, R. (2005), Penalty function approach in heuristic algorithms for constrained, *IEEE Transactions on Reliability*, **54**(3), 549-58.

[2] Boland, P. J. and EL-Newehi, E. (1995), Component redundancy vs. system redundancy in the hazard rate ordering, *IEEE Transactions on Reliability*, **44**(4), 614-619.

[3] Chern, M. S. (1992), On the computational complexity of reliability redundancy allocation in a series system, *Operations Research Letters*, **11**, 309-315.

[4] Coit, D. W. and Smith, A. E. (1996a), Reliability optimization of series-parallel systems using a genetic algorithm, *IEEE Transactions on Reliability*, **45**(2), 254-260.

[5] Coit, D. W. and Smith, A. E. (1996b), Penalty guided genetic search for reliability design optimization, *Computers & Industrial Engineering*, **30**(4), 895-904.

[6] Coit, D. W. and Jin, T. and Wattanapongsakorn N. (2004), System optimization with component

reliability estimation uncertainty; A multi-criteria approach, *IEEE Transactions on Reliability*, **53**(3), 369-380.

[7] Ha, C. and Kuo, W. (2006), Reliability redundancy allocation: an improved realization for nonconvex nonlinear programming problems, *European Journal of Operations Research*, **171**, 24-38.

[8] Heish, Y. H. (2002), A linear approximation for redundant reliability problems with multiple component choices, *Computers & Industrial Engineering*, **44**, 91-103.

[9] Holland, J. (1975), *Adaption in natural and artificial systems*, University of Michigan press, Ann Arbor.

[10] Kotz, S., Lai, C. D. and Xie, M. (2003), On the effect of redundancy for systems with dependent components, *IIE Transactions*, **35**, 1103-1110.

[11] Kuo, W. and Prasad, V. R. (2000), An annotated overview of system reliability optimization, *IEEE Transactions on Reliability*, **49**, 176-87.

[12] Kuo, W., Prasad, V. R., Tilman, F.A. and Hwang, C. L. (2001), *Optimal reliability design: fundamentals and applications*, Cambridge Univeristy Press.

[13] Levitin, G. (2002), Optimal series-parallel topology of multi-state system with two failure modes, *Reliability Engineering and System Safety*, **77**, 93-107.

[14] Levitin, G. (2003), Optimal multilevel protection in series-parallel systems, *Reliability Engineering and System Safety*, **81**, 93-102.

[15] Liang, Y. H. and Smith, A. E. (2004), An ant colony optimization algorithm for the redundancy allocation problem (RAP), *IEEE Transactions on Reliability*, **53**(3), 417-423.

[16] Nahas, N. and Nourelfath, M. (2005), Ant system for reliability optimization of a series system with multiple-choice and budget constraints, *Reliability Engineering and System Safety*, **87**, 1-12.

[17] Ramirez-Marquez, J. E., Coit, D. W. and Konak, A. (2004), Redundancy allocation for series-parallel systems using a max-min approach, *IIE Transactions*, **36**, 891-898.

[18] Wu, S. and Chan, L. Y. (2003), Performance utility analysis of multi-state systems, *IEEE Transactions on Reliability*, **52**(1), 14-21.

[19] Yalaoui, A., Chu, C. and Chatelet, E. (2005), Reliability allocation problem in a series-parallel system, *Reliability Engineering and System Safety*, **90**, 55-61.

[20] Yokota, T., Gen, M. and Li, Y. X. (1996), Genetic algorithm for non-linear mixed integer programming problems and its applications, *Computers & Industrial Engineering*, **30**(4), 905-917.

[21] Yun, W. Y. and Kim, J. W. (2004), Multi-level redundancy optimization in series system, *Computers & Industrial Engineering*, **46**, 337-346.