

다중 부품선택이 존재하는 직렬구조 시스템의 최적 신뢰성설계를 위한 시뮬레이티드 어닐링 알고리즘

김호균[†] · 배창옥 · 백천현

동의대학교 공과대학 정보산업공학과

A Simulated Annealing Algorithm for the Optimal Reliability Design Problem of a Series System with Multiple Component Choices

Ho-Gyun Kim · Chang-Ok Bae · Chunhyun Paik

Department of Information & Industrial Engineering, Dong-Eui University, Busan, 614-714

This paper presents a simulated algorithm(SA) for the optimal reliability design problem of a series system with multiple component choices incorporated at each subsystem. The objective of the problem is to maximize the system reliability while satisfying some constraint on system budget. The problem is formulated as a nonlinear binary integer programming problem and characterized as an NP-hard problem. The SA algorithm is developed by introducing some solution-improvements methods. Numerical examples are tested and the results are compared. The results have demonstrated the efficiency and the effectiveness of the proposed SA algorithm.

Keywords: optimal reliability design, multiple component choices, simulated annealing

1. 서론

최적 신뢰성설계(ORD; optimal reliability design)란 주어진 자원 하에서 최대의 신뢰성이 확보되는 시스템 구조를 결정하는 것인데, 일반적으로 신뢰성 부품을 사용하는 방법과 부품의 리던던시(redundancy)를 도입하는 방법이 사용된다. ORD 문제 분야의 초기에는 서브시스템 내의 단일 부품에 대해 최적의 부품 리던던시 수를 결정하는 문제에 관한 연구들이 주로 수행되었다. 그러나 최근에는 부품기술의 발달로 동일 기능을 수행하면서도 신뢰도, 비용, 부피, 무게 등 특성이 다른 다양한 부품대안들이 존재하게 되었고, 이러한 추세에 맞추어, 주어진 자원 제약하에 다양한 부품대안들을 이용하여 시스템의 신뢰성을 최대화할 수 있는 구조를 결정하는 ORD 문제에 관한 연구들이 최근 들어 활발히 수행되고 있다. 이와 같이 다양한 부품대안

choices)라 하기로 한다. MCC를 고려한 ORD 문제는 서브시스템 내 대안들의 혼합(mixing) 사용 여부 그리고 다시 각 경우에 대해 부품 리던던시의 실시 여부에 따라 세분화될 수 있다. 대안들의 혼합이 허용되는 경우에는 한 가지 이상의 부품이 선택되는 상황을 의미하므로 리던던시를 실시하는 경우만이 이에 해당된다.

MCC를 고려한 ORD 문제를 위한 최적해 탐색을 위해 다양한 연구들이 수행되었다. Sung and Cho(2000)는 비용제약을 갖는 직렬구조 시스템의 최적해 탐색을 위하여 분지한계법(branch-and-bound method)을 제시하였다. 이 연구는 부품대안들의 혼합사용을 허용하지 않고 리던던시가 실시되지 않는 경우이다. 대안들의 혼합사용을 허용하지 않고 리던던시가 실시되는 시스템의 최적해 탐색에 관한 연구로는 동적 계획법(DP; dynamic programming)을 사용한 Fyffe *et al.*(1968), 대리제약(surrogate constraints) 기법을 이용한 Nakagawa and Miyazaki(1981)와 다중

[†]연락처 : 김호균 교수, 614-714 부산광역시 부산진구 엄광로 995 동의대학교 공과대학 정보산업공학과, Fax : 051-890-1619, E-mail : hgkim@deu.ac.kr

상태(multi-state)가 존재하는 직렬구조 시스템의 최적해 탐색을 위한 휴리스틱 방법을 제시한 Ramirez-Marquez and Coit(2004)가 있다. Hsieh(2002)는 대안들의 혼합사용을 허용하는 경우의 문제에 대해 선형근사해법(linear approximation method)을 제시하였다.

위에서 언급된 기존 연구들에서 제시된 해법들은 조합적 최적화 해법 또는 국부 탐색방법에 근거한 휴리스틱 범주에 포함된다. 따라서 이들은 규모가 큰 현실문제에 대해 적용하기 어렵거나, 단지 국부해(local optima)만을 탐색하는 등의 단점을 가지고 있다. 이러한 조합적 최적화 해법 및 국부 탐색방법의 한계를 극복하고자 최근에는 유전자 알고리즘(GA; genetic algorithm) 등의 메타 휴리스틱을 적용한 연구들이 활발히 수행되고 있다. Nahas and Nourelfath(2005)는 대안들의 혼합사용을 허용하지 않고 리던던시가 실시되지 않는 경우에 개미 시스템(ant system)을 적용하였고, Yokota *et al.*(1996)은 대안들의 혼합사용을 허용하지 않고 리던던시가 실시되는 경우에 대해 GA를 이용하였으며, Coit and Smith(1996a, b)는 대안들의 혼합사용을 허용하는 ORD 문제의 최적해 탐색에 GA를 이용하였다.

ORD 문제에 SA를 이용한 연구는 다른 메타 휴리스틱 방법에 비해 상대적으로 매우 적은 편이다. SA를 이용한 연구로는 Angus and Ames(1997)와 Ravi *et al.*(1997)이 있으며, 이는 MCC를 고려하지 않고 시스템 신뢰성을 최대화하기 위한 최적 리던던시 수를 결정하는 리던던시 할당문제에 관한 연구이다. MCC를 고려한 ROD 문제 분야의 기존 연구들을 대안들의 혼합사용 여부, 리던던시 실시 여부, 사용 알고리즘의 측면에서 <표 1>에 정리한다.

Kuo *et al.*(2001)에서는 SA가 복잡한 이산최적화 문제(complex discrete optimization problems)의 해법에 큰 장점을 가지고 있으나 ORD 문제 분야에서는 연구가 미흡하여 다양한 측면에서의 연구수행이 요구된다고 하였다. 이에 본 연구에서는 MCC를 고려한 ROD 문제의 최적해 탐색에 있어 이제까지 적용되지 않았던 SA를 적용하고자 한다. 제시한 SA 알고리즘에는 문제의 최적해를 효율적으로 탐색하기 위해 일정횟수의 가능해가 획득되어진 경우 내부 루프를 종료하도록 하는 NES(non-equilibrium SA)의 개념과 제약함수를 이용하여 효율적으로 인접해를 생성하는 잔여자원 개선(resource-based improvement) 개념을

응용하여 최적해 탐색성능을 향상시킨다. MCC를 고려한 ROD 문제들 중 구조가 상대적으로 간단한 경우인 대안들의 혼합사용을 허용하지 않고 리던던시가 실시되지 않는 직렬구조 시스템의 최적해 탐색에 이를 적용한다. 기존 연구 및 일반적인 SA 알고리즘의 결과와 개발된 SA 알고리즘을 이용한 경우의 성능 비교를 실시하여 제시한 SA 알고리즘의 성능을 평가한다. 이를 통해 본 연구는 MCC를 고려한 ROD 분야의 다양한 문제들의 최적해 탐색연구에 SA를 적용하기 위한 초석이 될 수 있을 것이다.

본 연구의 구성은 다음과 같다. 2장에서는 ORD 문제에 관한 일반적 사항과 기호를 설명하고, 3장에서는 개발된 SA 알고리즘의 개념과 관련 모수들에 관해 설명한다. 추출한 예제들에 대한 수치실험 및 실험결과를 4장에서 설명하며, 마지막으로 5장에서 결론과 추후연구에 관해 언급한다.

2. 최적 신뢰성설계 문제

MCC를 고려한 ORD 문제는 비용제약이 존재하는 직렬구조 시스템에서 전체 신뢰성을 최대화할 수 있는 구조를 결정하는 문제이다. 문제의 모형화에 사용될 몇 가지 기호들을 다음과 같이 정의한다.

- n : 서브시스템의 수
- m_i : 서브시스템 i 의 이용 가능한 부품대안의 수
- R_i^j : 서브시스템 i 에 부품대안 j 가 사용되었을 때의 서브시스템 신뢰도
- R_S : 시스템 신뢰도
- c_i^j : 서브시스템 i 에 사용된 부품대안 j 의 비용
- C : 시스템 비용의 상한값

본 문제에서의 의사결정변수인 $x_i^j (i=1,2,\dots, n \ \& \ j=1,2,\dots, m_i)$ 는 0과 1의 값을 가지는 이진정수로 서브시스템 i 에 대안 j 가 사용되었을 경우에는 1, 그렇지 않은 경우엔 0의 값이 된다. 본 연구에서 제시한 SA 알고리즘의 성능평가를 위해 사용될 직렬구

표 1. MCC를 고려한 기존 연구의 분류

기존 연구	대안들의 혼합사용 여부	리던던시 실시 여부	사용 알고리즘
Sung and Cho (2000) Nahas and Nourelfath(2005)	×	×	분지한계법 (B&B) 개미시스템 (AS)
Fyffe <i>et al.</i> (1968) Nakagawa and Miyazaki (1981) Yokota <i>et al.</i> (1996) Ramirez-Marquez and Coit (2004)	×	○	동적 계획법 (DP) 대리제약기법 (surrogate constraints method) 유전자 알고리즘 (GA) 휴리스틱 (heuristic)
Coit and Smith(1996a, b) Hsieh (2002)	○	○	유전자 알고리즘 (GA) 선형근사해법 (linear approximation method)

조 시스템 문제는 다음과 같은 최적화 문제로 나타낼 수 있다.

$$\text{Max } R_s = \prod_{i=1}^n \left(\sum_{j=1}^{m_i} x_i^j R_i^j \right)$$

$$\text{st } \sum_{i=1}^n \sum_{j=1}^{m_i} x_i^j c_i^j \leq C,$$

$$\sum_{j=1}^{m_i} x_i^j = 1 \quad \forall i = 1, 2, \dots, n,$$

$$x_i^j = \{0, 1\} \quad \forall i = 1, 2, \dots, n \ \& \ j = 1, 2, \dots, m_i.$$

위에 주어진 최적화 문제에서, 목적함수는 전체 시스템 신뢰성의 최대화이며, 첫 번째 제약식은 시스템의 비용제약을 나타내고, 두 번째는 각 서브시스템에서 선택해야 할 부품의 수가 한 가지임을 나타낸다.

3. SA 알고리즘 개발

SA는 전통적인 국부 탐색법(local search methods)의 대안으로 Kirkpatrick *et al.*(1983)과 Cerny(1985)에 의해 제시된 이후, 조합적 최적화 문제에 성공적으로 적용되어진 메타 휴리스틱 기법 중 하나이다. SA는 주어진 인접해 구조 (neighborhood structure) 하에서 현재해로부터 인접해로 전이하는 반복적 과정을 통하여 전체 최적해(global optimal solution)를 찾아 나가는 방법으로, 상대적으로 열등한 해로의 전이도 확률적으로 허용함으로써 특정 국부 최적해에서 빠져나올 수 있는 가능성을 열어두는 특징을 가지고 있다.

SA를 다양한 조합적 최적화 문제에 적용하기 위해서는 관련 모수인 해 표현방법 및 초기해의 설정, 인접해 생성방법, 초기 온도 및 내부 루프 반복수의 설정, 온도제어 모수의 감소를 위한 쿨링 스케줄(cooling schedule) 및 알고리즘 종료기준이 우선 정의되어야 한다. 본 연구에서는 MCC를 고려한 ORD 문제의 최적해 탐색을 위한 SA 알고리즘의 모수들을 다음과 같이 정의한다.

[해 표현방법 및 초기해의 설정]

문제의 해는 시스템의 서브시스템 수와 동일한 자릿수로 구성되며, 각 자리의 값은 서브시스템에서 사용할 부품대안을 의

미한다. 각 서브시스템에서 다양한 부품대안들의 식별번호는 신뢰도의 오름차순으로 표시된다. <그림 1>은 MCC를 고려한 직렬구조 시스템의 ORD 문제를 위한 해 표현의 예를 나타낸 것으로, 서브시스템 1의 3은 여러 가지 부품대안 가운데 세 번째 대안이 사용되었음을 의미한다.

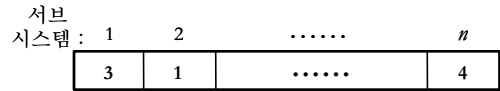


그림 1. 해의 표현 예.

초기해는 전체 자리에 대해 난수발생방법을 실시한 무작위해를 사용한다. 생성된 해를 평가하기 위한 SA의 성능평가함수인 에너지함수 E는 문제의 목적함수를 그대로 사용하며, 제약식을 위반하는 해일 경우 에너지함수 값이 0이 되도록 하였다.

[인접해 생성방법]

인접해 공간상에서 인접해를 효율적으로 생성하는 것은 SA 알고리즘의 성능향상에 있어 매우 중요하다. 본 연구에서는 2 단계로 구성된 인접해 생성방법(scheme)을 채택한다. 먼저, 단계 1에서 소위 pairwise-swapping 방법을 이용해 인접해 생성을 시도한다. 만약 다섯 번 연속으로 비가능해 (infeasible solution)를 생성하거나, 현재해(current solution)보다 개선된 해 탐색에 실패하는 경우 생성방법 단계 2(잔여자원 개선)를 실행하게 된다.

단계 1 (pairwise-swapping scheme) : 단계 1에서는 해의 전체 자리 가운데 임의로 두 개의 자리를 중복 없이 선택하여 그 자리의 값들을 서로 교환한 후 다시 임의의 한 자리를 선택하여 난수 발생 방법으로 값을 할당하는 것이다. 한 개의 자리만을 선택하여 변화를 주는 방법이 인접공간을 제약하지 않고 해를 생성할 수 있는 일반적인 방법이지만, 이 방법은 사전실험에서 해의 길이가 길어질수록 수렴속도가 매우 낮아짐을 확인할 수 있었다. <그림 2>는 단계 1에 의한 인접해 생성의 예를 나타낸 것이다.

단계 2 (resource-based improvement scheme) : 단계 2는 <그림 3>에 나타난 절차에 따라 인접해를 생성하는 방법으로, 예를 들어 세 개의 서브시스템으로 구성된 경우의 현재해가 {3, 2, 5}

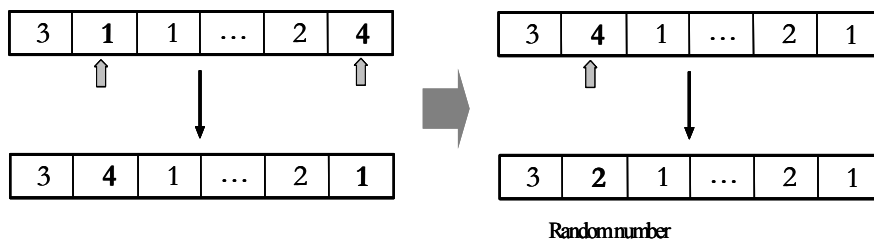


그림 2. 인접해 생성방법 단계 1의 생성 예.

인 경우에는 탐색절차에 의해 {4, 1, 5}, {4, 2, 4}, {2, 3, 5}, {2, 2, 6}, {3, 3, 4}, {3, 1, 6}의 인접해를 탐색한다. 생성된 해 가운데 자원제약함수를 기준으로 현재해보다 자원활용이 많은 것들이 선택되며, 이 가운데 목적함수가 최대인 해를 인접해로 선택하게 된다.

```

For  $i = 1$  to  $(n - 1)$ 
  부품  $i$ 에 사용된  $p$  대안을  $(p + 1)$  대안으로 변경한다.
  For  $k = (i + 1)$ 
    부품  $k$ 에 사용된  $q$  대안을  $(q - 1)$  대안으로 변경한다.
    현재 획득한 해를 저장한다.
  End
  부품  $i$ 에 사용된  $p$  대안을  $(p - 1)$  대안으로 변경한다.
  For  $k = (i + 1)$ 
    부품  $k$ 에 사용된  $q$  대안을  $(q + 1)$  대안으로 변경한다.
    현재 획득한 해를 저장한다.
  End
End
현재까지 획득한 모든 해를 평가한다.
가장 좋은 가능해를 인접해로 선택한다.

```

그림 3. 인접해 생성방법 단계 2의 절차.

[초기온도의 설정]

SA의 초기온도(T_0)는 초기에 이루어지는 대부분의 이동이 수락 가능하도록 충분히 높게 설정되어야 한다. 그러나 지나치게 높은 초기온도의 설정은 알고리즘의 탐색이 불필요하게 길어지게 하므로 SA를 이용한 기존 연구들에서는 예비실험을 통하여 적절한 수준의 온도를 설정하는 방법을 사용하고 있다. 본 연구에서는 예비실험을 실시하여 $T_0 = 50$ 의 값을 사용하였다.

[내부 루프 반복수의 설정]

특정 온도제어 모수값에서 반복되는 탐색과정의 수인 내부 루프 반복수는 일반적으로 인접해의 규모에 일정한 상수를 곱한 상수설정방법이 사용된다. 이 방법은 해의 수렴성을 유지하기 위한 평형상태(equilibrium state)에 도달하기 위하여 충분한 횟수만큼 내부 루프를 반복하기 위함인데, 이는 내부 루프를 불필요하게 길어지게 하는 결과를 초래할 수 있다(Yoon and Song, 1995). 이러한 SA의 단점을 보완하고 알고리즘의 수렴속도를 빠르게 하기 위하여 Cardoso *et al.*(1994)은 일정횟수의 가능해가 획득되어진 경우 내부 루프를 종료하도록 하는 NESAs (nonequilibrium SA)를 제안하였으며, Ravi *et al.* (1997)은 이를 리던던시 할당문제의 최적해 탐색에 적용하여 그 성능이 뛰어난 것을 입증하였다. 본 연구에서도 NESAs의 개념을 활용하여 내부 루프 내에서 가능해가 Q 번 획득되어진 경우 내부 루프를 종료하도록 설정하였으며, 예비실험을 통하여 $Q = 5n$ 의 값을 사용하였다.

[쿨링 스케줄 및 종료기준]

SA의 외부 루프에서 온도제어 모수를 탐색과정에 따라 감소시키는 것을 쿨링 스케줄이라고 하며, 이는 알고리즘의 수렴속도 및 해의 질에 영향을 미치게 된다. 본 연구에서는 온도 강하율 α ($0 \leq \alpha \leq 1$)에 의해 현재온도를 $T_C = \alpha T_{C-1}$ ($C = 1, 2, \dots$)로 조정하는 방법인 기하 스케줄(geometric schedule)을 사용하였다. 온도 강하율 α 는 일반적으로 0.5~0.99의 값을 사용하는데, 값이 클수록 수렴속도는 늦어지나 해의 질은 좋아지게 된다. 본 연구에서는 사전실험을 통해 $\alpha = 0.95$ 로 설정하였으며, 문제 크기가 큰 예제 4의 경우에는 $\alpha = 0.98$ 을 사용한다.

종료기준은 온도제어 모수가 미리 설정한 최종온도보다 작거나 같을 때까지 알고리즘이 수행되도록 설정한다. 한 내부 루프가 종료되면 쿨링 스케줄에 의해 온도를 조정하게 되며, 만약 조정된 현재온도가 최종온도보다 작으면 알고리즘을 종료하고, 그렇지 않으면 현재온도까지의 최선해를 현재해로 설정하고 다시 내부 루프를 반복하게 되는 것이다. 본 연구에서는 최종온도를 $T_F = 1$ 로 설정하였다.

4. 수치실험 및 결과

본 장에서는 제시된 SA 알고리즘의 성능을 객관적으로 평가하기 위해, 기존 연구(Nahas and Nourelfath(2005))에서 사용된 동일한 실험 데이터를 이용한 수치실험을 실시한다. 모든 예제에 대해 동일 조건하에서 10회 반복 실험하였으며, 알고리즘의 수렴성을 파악하기 위해 각 실험결과에서의 최대값, 최소값, 평균 및 표준편차를 표시하였다. 본 연구에서 제시한 SA 알고리즘은 프로그램 언어 C++로 코딩되었으며, 수치실험은 펜티엄 IV 2.0GHz IBM-PC에서 실행되었다.

수치실험에 이용된 예제는 Nahas and Nourelfath(2005)의 연구에서 사용된 직렬구조 시스템으로 예제 1~4에 대한 데이터가 부록 1~4에 나타나 있다. 시스템 내 전체 부품대안의 수가 61개, 80개, 100개, 166개인 경우의 네 가지 예제로 구성되며, 시스템 비용의 상한값 C 는 각각 1000, 900, 1000, 1400과 같다. 예제 1~3은 15개 서브시스템으로, 예제 4는 25개의 서브시스템으로 구성된다. 예제의 탐색영역은 각각 4.147×10^8 , 4.915×10^{10} , 1.572×10^{12} , 1.932×10^{20} 크기를 갖는다.

<표 2>는 네 가지 예제에 대해 본 연구에서 제시한 SA 알고리즘을 이용하여 최적해를 탐색한 결과를 나타낸 것으로 Nahas and Nourelfath(2005) 및 전형적인 SA 알고리즘의 결과와 비교하였다. 여기서 전형적인 SA 알고리즘이란 NESAs 개념과 잔여자원 개선개념을 적용하지 않는 SA 알고리즘이다. 즉, 내부 루프는 일정횟수를 반복하도록 상수(L)로 설정하여 $L = 100n$ 을 사용하였고, 인접해 생성단계에서는 앞에서 언급한 단계 1의 방법만 이용하는 것이다. 전형적인 SA 알고리즘과 본 연구에서 제시한 SA 알고리즘을 비교함으로써 NESAs 및 잔여자원

개선개념의 응용에 관한 성능을 평가하고자 한다.

예제 1과 2의 경우는 Nahas and Nourelfath (2005)의 개미시스템과 제시한 SA 알고리즘은 매 실행 시마다 최적해를 제공하는 것으로 나타났으나, 전형적인 SA 알고리즘의 경우는 약간의 편차(0.00201)를 보이는 것으로 나타났다. 예제 3과 4의 경우에는 제시한 SA 알고리즘 역시 개미시스템이나 전형적인 SA 알고리즘과 마찬가지로 약간의 편차를 가지는 것으로 나타났으며, 전형적인 SA 알고리즘의 경우에는 최적해를 탐색하지 못한 것으로 나타났다.

실험결과를 통해 세 가지 알고리즘 가운데 본 연구에서 제시한 SA 알고리즘의 성능이 가장 뛰어나고 전형적인 SA 알고리즘의 성능이 가장 못한 것을 알 수 있으며, 개미시스템과 제시한 SA 알고리즘의 성능차이는 그다지 크지 않은 것을 알 수 있다. 따라서 전형적인 SA 알고리즘에 NESAs 및 잔여자원 개선개념을 적용하는 것이 최적해 탐색성능을 향상시켰다고 할 수 있다. 각 예제에 대한 최적해를 <표 1>에 나타내었다.

예제 4의 경우에, 제시한 SA 알고리즘은 Nahas and Nourelfath (2005)의 최적해뿐만 아니라 동일한 시스템 신뢰성을 제공하는 대안 최적해(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)도 탐색하였다. 잔여비용은 각 예제에 대해 10, 5, 5, 0이었다. 제시한 SA 알고리즘의 각 예제에 대한 최적해 탐색에 소요된 평균 계산시간(CPU time)은 예제 1~3의 경우는 평균 1초 이내였으며, 상대적으로 문제의 크기가 큰 예제 4의 경우에는 평균 1.68초인 것으로 나타났다.

실험결과를 바탕으로 본 연구에서 제시한 SA 알고리즘은 MCC를 고려한 직렬구조 시스템의 ORD 문제에 대한 최적해 탐

색에 있어 그 성능이 뛰어난을 알 수 있었으며, 전형적인 SA 알고리즘에 NESAs 및 잔여자원 개선개념을 적용하는 것이 알고리즘의 최적해 탐색성능을 향상시키는 데 효과가 있었음을 알 수 있었다.

5. 결론

본 연구에서는 MCC를 고려한 직렬구조 시스템의 ORD 문제를 다루었으며, MCC를 고려한 ORD 문제의 최적해 탐색분야에서 지금까지 적용되지 않았던 SA 알고리즘을 제시하였다. 제시한 SA 알고리즘에는 일정횟수의 가능해가 획득되어진 경우 내부 루프를 종료하도록 하는 NESAs의 개념과 제약함수를 이용하여 효율적으로 인접해를 탐색하는 잔여자원 개선개념이 활용되었으며, 기존 연구에서 사용한 예제들에 대한 수치실험을 통해 알고리즘의 최적해 탐색성능을 평가하였다. 기존 연구에서 사용한 네 가지 예제로 구성된 문제들의 수치실험 결과 수초 이내에 최적해를 제공하여 알고리즘의 수렴성이 뛰어난을 알 수 있었다.

본 연구에서 고려한 문제는 서브시스템 내에 부품 리던던시가 실시되지 않는 경우이나, 리던던시가 있는 경우가 보다 현실적인 시스템이라 할 수 있으며 또한 비용제약 이외에도 무게, 부피 등의 다양한 자원제약이 존재할 수 있을 것이다. 문제의 특성에 따른 몇 가지 모수들의 조정만으로 다양한 문제에 적용 가능하다는 메타 휴리스틱의 장점을 이용하여, 제시한 SA 알고리즘을 서브시스템 내에 리던던시와 다양한 제약함수가 존재하는 경우에 대한 적용을 추후 연구에서 수행할 것이다.

표 2. 수치실험 결과

예제	Nahas and Nourelfath(2005)				전형적인 SA 알고리즘				본 연구의 SA 알고리즘				
	Min	Max	Average	S.D.	Min	Max	Average	S.D.	Min	Max	Average	S.D.	
1	0.85705	0.85705	0.85705	0	0.85055	0.85705	0.85589	0.00201	0.85705	0.85705	0.85705	0	
2	0.91504	0.91504	0.91504	0	0.90762	0.91504	0.91208	0.00341	0.91504	0.91504	0.91504	0	
3	0.96406	0.96513	0.96439	0.00050	0.87960	0.96513	0.94619	0.03517	0.96407	0.96513	0.96503	0.00033	
4	0.86465	0.86543	0.86491	0.00038	0.79320	0.84893	0.82474	0.02280	0.86465	0.86543	0.86536	0.00025	
최적해	예제 1 :		3-4-5-2-3-3-2-3-2-2-3-4-3-2										
	예제 2 :		3-3-3-4-2-3-3-2-4-1-2-3-4-3-1										
	예제 3 :		3-3-4-4-3-3-2-3-2-2-4-4-2										
	예제 4 :		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 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										

부 록

부록 1. 예제 1에 대한 데이터

부품		대안 1	대안 2	대안 3	대안 4	대안 5	대안 6	대안 7	대안 8
1	신뢰도 비용	0.9 20	0.99 40	0.999 60	0.9999 80	0.99999 100	0.999999 120	0.9999999 140	0.99999999 180
2	신뢰도 비용	0.85 30	0.9775 60	0.9966 90	0.9995 120	0.9999 150	-	-	-
3	신뢰도 비용	0.8 20	0.96 40	0.99 60	0.998 80	0.9997 100	-	-	-
4	신뢰도 비용	0.75 30	0.938 40	-	-	-	-	-	-
5	신뢰도 비용	0.85 20	0.99 40	0.999 65	-	-	-	-	-
6	신뢰도 비용	0.9 25	0.95 30	0.999 50	0.9999 70	-	-	-	-
7	신뢰도 비용	0.95 40	0.99 60	-	-	-	-	-	-
8	신뢰도 비용	0.85 10	0.995 30	0.999 60	0.9999 80	0.99999 120	-	-	-
9	신뢰도 비용	0.9 30	0.95 50	-	-	-	-	-	-
10	신뢰도 비용	0.99 15	0.999 40	0.9999 70	0.99999 100	0.999999 130	-	-	-
11	신뢰도 비용	0.95 20	0.999 40	0.9998 60	0.99999 80	0.999998 100	0.9999999 120	-	-
12	신뢰도 비용	0.8 40	0.9 60	0.99 85	-	-	-	-	-
13	신뢰도 비용	0.75 30	0.85 50	0.99 80	0.999 100	-	-	-	-
14	신뢰도 비용	0.8 10	0.95 30	0.99 40	-	-	-	-	-
15	신뢰도 비용	0.99 50	0.999 80	0.9999 110	0.99999 140	-	-	-	-

부록 2. 예제 2에 대한 데이터

부품		대안 1	대안 2	대안 3	대안 4	대안 5	대안 6	대안 7	대안 8
1	신뢰도 비용	0.9 20	0.99 40	0.999 60	0.9999 80	0.99999 100	0.999999 120	0.9999999 140	0.99999999 180
2	신뢰도 비용	0.85 30	0.9775 60	0.9966 90	0.9995 120	0.9999 150	-	-	-
3	신뢰도 비용	0.8 20	0.96 40	0.99 60	0.998 80	0.9997 100	-	-	-
4	신뢰도 비용	0.75 30	0.938 40	0.97 60	0.99 70	0.995 80	-	-	-
5	신뢰도 비용	0.85 20	0.99 40	0.999 65	0.9999 80	-	-	-	-
6	신뢰도 비용	0.9 25	0.95 30	0.999 50	0.9999 70	-	-	-	-
7	신뢰도 비용	0.95 40	0.99 60	0.999 80	0.9999 100	-	-	-	-
8	신뢰도 비용	0.85 10	0.995 30	0.999 60	0.9999 80	0.99999 120	-	-	-
9	신뢰도 비용	0.9 30	0.95 50	0.98 70	0.995 90	0.9999 120	-	-	-
10	신뢰도 비용	0.99 15	0.999 40	0.9999 70	0.99999 100	0.999999 130	-	-	-
11	신뢰도 비용	0.95 20	0.999 40	0.9998 60	0.99999 80	0.999998 100	0.9999999 120	-	-
12	신뢰도 비용	0.8 40	0.9 60	0.99 85	0.999 100	0.9999 120	0.99999 140	0.999997 155	0.9999995 170
13	신뢰도 비용	0.75 30	0.85 50	0.99 80	0.999 100	-	-	-	-
14	신뢰도 비용	0.8 10	0.95 30	0.99 40	0.996 60	0.9993 80	0.9999 95	0.99996 120	0.999998 140
15	신뢰도 비용	0.99 50	0.999 80	0.9999 110	0.99999 140	-	-	-	-

부록 3. 예제 3에 대한 데이터

부품		대안 1	대안 2	대안 3	대안 4	대안 5	대안 6	대안 7	대안 8
1	신뢰도 비용	0.9 20	0.99 40	0.999 60	0.9999 80	0.99999 100	0.999999 120	0.9999999 140	0.99999999 180
2	신뢰도 비용	0.85 30	0.9775 60	0.9966 90	0.9995 120	0.9999 150	-	-	-
3	신뢰도 비용	0.8 20	0.96 40	0.99 60	0.998 80	0.9997 100	0.9999 120	0.99999 140	0.999999 160
4	신뢰도 비용	0.75 30	0.938 40	0.98 50	0.999 60	0.9999 70	-	-	-
5	신뢰도 비용	0.85 20	0.99 40	0.999 65	0.9999 80	0.99998 100	0.999998 120	0.9999998 140	0.99999998 155
6	신뢰도 비용	0.9 25	0.95 30	0.999 50	0.9999 70	0.99999 90	-	-	-
7	신뢰도 비용	0.95 40	0.99 60	0.997 80	0.9997 100	0.99997 120	0.999997 140	0.9999997 160	0.99999997 180
8	신뢰도 비용	0.85 10	0.995 30	0.999 60	0.9999 80	0.99999 120	-	-	-
9	신뢰도 비용	0.9 30	0.95 50	0.995 70	0.9995 90	0.99995 110	0.999995 130	0.9999995 150	0.99999995 170
10	신뢰도 비용	0.99 15	0.999 40	0.9999 70	0.99999 100	0.999999 130	0.9999999 160	-	-
11	신뢰도 비용	0.95 20	0.999 40	0.9998 60	0.99999 80	0.999998 100	0.9999999 120	0.99999997 140	0.999999999 160
12	신뢰도 비용	0.8 40	0.9 60	0.99 85	0.999 110	0.9999 130	-	-	-
13	신뢰도 비용	0.75 30	0.85 50	0.99 80	0.999 100	0.9996 120	0.99996 140	0.999996 160	0.9999996 180
14	신뢰도 비용	0.8 10	0.95 30	0.99 40	0.999 60	0.9999 80	-	-	-
15	신뢰도 비용	0.99 50	0.999 80	0.9999 110	0.99999 140	0.999999 160	0.9999999 180	0.99999998 200	0.999999995 220

부록 4. 예제 4에 대한 데이터

부품		대안 1	대안 2	대안 3	대안 4	대안 5	대안 6	대안 7	대안 8
1	신뢰도	0.9	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	0.99999999
	비용	20	40	60	80	100	120	140	180
2	신뢰도	0.85	0.9775	0.9966	0.9995	0.9999	-	-	-
	비용	30	60	90	120	150			
3	신뢰도	0.8	0.96	0.99	0.998	0.9997	0.9999	0.99999	0.999999
	비용	20	40	60	80	100	120	140	160
4	신뢰도	0.75	0.938	0.98	0.999	0.9999	-	-	-
	비용	30	40	50	60	70			
5	신뢰도	0.85	0.99	0.999	0.9999	0.99998	0.999998	0.9999998	0.99999998
	비용	20	40	65	80	100	120	140	155
6	신뢰도	0.9	0.95	0.999	0.9999	0.99999	-	-	-
	비용	25	30	50	70	90			
7	신뢰도	0.95	0.99	0.997	0.9997	0.99997	0.999997	0.9999997	0.99999997
	비용	40	60	80	100	120	140	160	180
8	신뢰도	0.85	0.995	0.999	0.9999	0.99999	-	-	-
	비용	10	30	60	80	120			
9	신뢰도	0.9	0.95	0.995	0.9995	0.99995	0.999995	0.9999995	0.99999995
	비용	30	50	70	90	110	130	150	170
10	신뢰도	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	-	-
	비용	15	40	70	100	130	160		
11	신뢰도	0.95	0.999	0.9998	0.99999	0.999998	0.9999999	0.99999997	0.999999999
	비용	20	40	60	80	100	120	140	160
12	신뢰도	0.8	0.9	0.99	0.999	0.9999	-	-	-
	비용	40	60	85	110	130			
13	신뢰도	0.75	0.85	0.99	0.999	0.9996	0.99996	0.999996	0.9999996
	비용	30	50	80	100	120	140	160	180
14	신뢰도	0.8	0.95	0.99	0.999	0.9999	-	-	-
	비용	10	30	40	60	80			
15	신뢰도	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	0.99999998	0.999999995
	비용	50	80	110	140	160	180	200	220
16	신뢰도	0.9	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	0.99999999
	비용	20	40	60	80	100	120	140	180
17	신뢰도	0.85	0.9775	0.9966	0.9995	0.9999	-	-	-
	비용	30	60	90	120	150			
18	신뢰도	0.8	0.96	0.99	0.998	0.9997	0.9999	0.99999	0.999999
	비용	20	40	60	80	100	120	140	160
19	신뢰도	0.75	0.938	0.98	0.999	0.9999	-	-	-
	비용	30	40	50	60	70			
20	신뢰도	0.85	0.99	0.999	0.9999	0.99998	0.999998	0.9999998	0.99999998
	비용	20	40	65	80	100	120	140	155
21	신뢰도	0.9	0.95	0.999	0.9999	0.99999	-	-	-
	비용	25	30	50	70	90			
22	신뢰도	0.95	0.99	0.997	0.9997	0.99997	0.999997	0.9999997	0.99999997
	비용	40	60	80	100	120	140	160	180
23	신뢰도	0.85	0.995	0.999	0.9999	0.99999	-	-	-
	비용	10	30	60	80	120			
24	신뢰도	0.9	0.95	0.995	0.9995	0.99995	0.999995	0.9999995	0.99999995
	비용	30	50	70	90	110	130	150	170
25	신뢰도	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	-	-
	비용	15	40	70	100	130	160		

참고문헌

- Angus, J.E. and Ames, K. (1997), A simulated annealing algorithm for system cost minimization subject to reliability constraints, *Communications in Statistics: Simulation and Computation*, 26, 783-790.
- Cardoso, M.F., Salcedo, R.L. and de Azevedo, S.F. (1994), Nonequilibrium simulated annealing: a faster approach to combinatorial minimization, *Industrial Engineering Chemical Research*, 33, 1908-1918.
- Cerny, V. (1985), Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Applications*, 45, 41-51.
- Coit, D.E. and Smith A.E. (1996a), Reliability optimization of series-parallel systems using a genetic algorithm, *IEEE Transactions on Reliability*, 45(2).
- Coit, D.E. and Smith A.E. (1996b), Penalty guided genetic search for reliability design optimization, *Computers & Industrial Engineering*, 30(4), 895-904.
- Fyffe, D.E., Hines, W.W. and Lee, N.K. (1968), System reliability allocation and a computational algorithm, *IEEE Transactions on Reliability*, 17, 64-69.
- Hsieh, Y.C. (2002), A linear approximation for redundant reliability problems with multiple component choices, *Computers & Industrial Engineering*, 44, 91-103.
- Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983), Optimization by simulated annealing, *Science*, 220, 671-679.
- Kuo, W., Prasad, V.R., Tillman, F.A. and Hwang, C.L. (2001), *Optimal Reliability Design: Fundamentals and Applications*, Cambridge University Press, Cambridge.
- Nahas, N. and Noureldath, 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.
- Nakagawa, Y. and Miyazaki, S. (1981), Surrogate constraints algorithm for reliability optimization problems with two constraints, *IEEE Transactions on Reliability*, 30(2), 175-180.
- Ramirez-Marque, J.E. and Coit, D.W. (2004), A heuristic for solving the redundancy allocation problem for multiple-state series-parallel systems, *Reliability Engineering and System Safety*, 83, 341-349.
- Ravi, V., Muty, B. and Reddy, P. (1997), Nonequilibrium simulated-annealing algorithm applied reliability optimization of complex systems, *IEEE Transactions on Reliability*, 46(2), 233-239.
- Sung, C.S. and Cho, Y.K. (2000), Reliability optimization of a series system with multiple-choice and budget constraint, *European Journal of Operational Research*, 127, 159-171.
- Yokota, T., Gen, M. and Li, Y.X. (1996), Genetic algorithm for nonlinear mixed-integer programming and its applications, *Computers & Industrial Engineering*, 30(4), 905-917.
- Yoon, B.S. and Song, N.U. (1995), Effective variations of simulated annealing and their implementation for high level synthesis, *Journal of the Korean Institute of Industrial Engineers*, 21(1), 33-49.



김 호 균

서울대학교 자원공학과 학사
 서울대학교 산업공학과 석사
 서울대학교 산업공학과 박사
 현재: 동의대학교 정보산업공학과 교수
 관심분야: 신뢰성분석, 품질경영, 생산시스템
 성능분석 등



백 천 현

연세대학교 경영학과 학사
 한국과학기술원 경영과학 석사
 한국과학기술원 경영과학 박사
 현재: 동의대학교 정보산업공학과 부교수
 관심분야: 경영과학, 정보통신, 인터넷 활용기
 술 등



배 창 욱

동의대학교 산업공학과 학사
 동의대학교 산업공학과 석사
 현재: 동의대학교 정보산업공학과 박사과정
 관심분야: 최적 신뢰성 설계, 품질경영 등