

## 부품선택이 존재하는 직렬시스템의 신뢰성 최적화 해법

김호균<sup>1\*</sup> · 배창옥<sup>2</sup> · 김재환<sup>3</sup> · 손주영<sup>4</sup>

<sup>1</sup>동의대학교 산업경영공학과 / <sup>2</sup>국방과학연구소 제3기술연구본부2부

<sup>3</sup>한국해양대학교 나노데이터시스템학부 / <sup>4</sup>한국해양대학교 컴퓨터 · 제어 · 전자통신공학부

### Solution Methods for Reliability Optimization of a Series System with Component Choices

Ho-Gyun Kim<sup>1</sup> · Chang-Ok Bae<sup>2</sup> · Jae-Hwan Kim<sup>3</sup> · Joo-Young Son<sup>4</sup>

<sup>1</sup>Department of Industrial & Management Engineering, Dong-Eui University, Busan 614-714, Korea

<sup>2</sup>The 3rd R&D Institute-2, Agency for Defense Development, Daejeon 305-714, Korea

<sup>3</sup>Division of Nano Data System, Korea Maritime University, Busan 606-791, Korea

<sup>4</sup>Division of Computer, Control, and Electronic Communications Engineering, Korea Maritime University, Busan 606-791, Korea

Reliability has been considered as an important design measure in various industrial systems. We discuss a reliability optimization problem with component choices (ROP-CC) subject to a budget constraint. This problem has been known as a NP-hard problem in the reliability design fields. Several researchers have been working to find the optimal solution through different heuristic methods. In this paper, we describe our development of simulated annealing (SA) and tabu search (TS) algorithms and a reoptimization procedure of the two algorithms for solving the problem. Experimental results for some examples are shown to evaluate the performance of these methods. We compare the results with the solutions of a previous study which used ant system (AS) and the global optimal solution of each example obtained through an optimization package, CPLEX 9.1. The computational results indicate that the developed algorithms outperform the previous results.

**Keywords:** Reliability Optimization Problem, Component Choices, Series System, Metaheuristics, CPLEX

#### 1. 서론

신뢰성은 지금까지 다양한 산업 분야에서 중요한 설계척도의 하나로 고려되어 왔으며, 가용한 자원을 활용하여 최적의 신뢰성을 달성할 수 있는 시스템 구조를 결정하는 문제를 신뢰성 최적화문제(ROP : reliability optimization problems)라 한다.

시스템을 설계할 때 사용되는 구성품들은 동일한 기능을 수행하지만 다양한 특성 면에서(신뢰성, 비용, 무게, 부피 등) 차이를 가지는 대안들을 가지며, 시장에서 이러한 대안 가운데 적절한 구성품을 선택하게 된다. 본 연구에서는 이러한 ROP를 부품선택 하에서의 ROP(ROP-CC : ROP with component choices)라 칭하기로 한다. ROP-CC는 각 서브시스템(직렬 구

조인 경우 구성품을 설계할 때 선택 가능한 몇 가지 대안들을 고려하여 시스템의 구조를 결정하는 문제이다 이러한 ROP-CC는 구성품의 중복설계(redundancy)를 고려하는가의 유무에 따라, 또한 중복설계를 고려할 경우 구성품 대안의 혼합사용을 허용하는가의 유무에 따라 세 가지로 구분 가능하다. 본 연구에서 다루고자 하는 문제는 이러한 세 가지 문제군 가운데 중복설계가 고려되지 않는 직렬시스템(series system)의 경우이다.

지금까지 많은 연구자들에 의해 ROP-CC 분야에 관한 연구가 수행되어 왔다. Sung and Cho(2000)는 중복설계를 고려하지 않은 경우에서 비용제약을 갖는 직렬시스템에 대한 최적해 탐색을 위해 분지한계법(branch-and-bound method)을 사용하였다. 중복설계를 고려하지만 구성품 대안의 혼합사용이 허용되지 않

본 연구는 동의대학교 교내연구과제의 지원을 통해 수행되었음(과제번호: 2007AA171).

\* 연락저자 : 김호균, 614-714 부산광역시 부산진구 염광로 995(가야동 산24번지), Fax : 051-890-2627, E-mail : hgkim@deu.ac.kr

2007년 07월 접수; 2007년 11월 수정본 접수; 2007년 12월 게재 확정.

는 직렬시스템(series-parallel system)의 최적 중복설계 구조를 결정하는 경우에 대해서 Fyffe *et al.*(1968)은 동적계획법(dynamic programming)을 활용한 휴리스틱 방법을 제시하였으며, Ramirez-Marquez and Coit(2004)는 목적함수가 시스템 총비용의 최소화인 다중상태(multi-state) 직렬시스템 문제에 대한 최적해 탐색을 고려하였다. 또한 구성품 대안의 혼합사용을 허용하는 경우로 Nakagawa and Miyazaki(1981)는 Fyffe *et al.*(1968)의 예제를 구성품 대안의 혼합사용을 허용하는 경우로 확장하고, 최적해 탐색을 위해 대리제약(surrogate constraints) 기법을 이용한 휴리스틱 방법을 제시하였다 Hsieh(2002) 역시 동일한 예제를 활용하여 문제의 최적해 탐색을 위한 효율적인 선형 근사해법(linear approximation)을 제시하였다.

국부 탐색법에 기초한 휴리스틱 방법은 국부 최적해에 도달한 경우 더 이상 해를 개선할 방법이 없으며 해결하고자 하는 문제마다 각기 그 특성에 맞추어 개발해야 한다는 단점을 가지고 있다. 따라서 최근 조합적 최적화문제(combinatorial optimization problems)의 해법에 관한 연구에서는 이러한 휴리스틱 방법의 단점을 보완한 GA(genetic algorithm), SA(simulated annealing), TS(tabu search), AS(ant system) 등의 메타휴리스틱 기법을 활용한 연구가 주로 수행되고 있다. 메타휴리스틱은 이해하기가 쉽고, 문제의 현실적 상황 및 다양한 제약조건을 반영하기가 용이하여 다양한 조합적 최적화문제의 해법에 대한 적용 가능성이 크며, 신뢰성 최적설계 분야에 대해서도 그러하다.

중복설계를 고려하지 않는 직렬시스템에서의 ROP-CC의 최적해 탐색에 관한 연구로서 Nahas and Nourelfath(2005)는 AS를, Kim *et al.*(2004)은 SA를 적용하였다. 중복설계를 고려한 ROP-CC에 대하여 Yokota *et al.*(1996)은 구성품 대안의 혼합사용이 허용되지 않는 문제의 최적해 탐색에 GA를 이용하였다. 구성품 대안의 혼합사용이 허용되는 경우에 대해서는 Coit and Smith(1996a, b)가 GA, Kulturel-Konak *et al.*(2003, 2004)이 TS, Liang and Smith(2004), Nahas *et al.*(2007), Zhao *et al.*(2007)이 ACO(ant colony optimization), Chen and You(2005)가 IA(immune algorithm)를 Fyffe *et al.*(1968)이 제시한 예제에 적용하고 그 결과를 이전 연구들의 결과와 비교하였다 Kuo *et al.*(2001)에 의하면 SA와 TS가 복잡한 이산 최적화문제(complex discrete optimization problems)의 해법에 있어 큰 장점을 가지고 있으나 신뢰성 최적설계 분야에서는 연구가 미흡하여 다양한 측면에서의 연구수행이 요구된다고 하였다

최근의 컴퓨터 성능 및 상용 최적화 소프트웨어의 성능향상에 따라 이전에 제시되지 못했던 다양한 최적화 문제들의 최적해가 탐색 가능해졌다. 이에 본 연구에서도 상용 최적화 소프트웨어의 하나인 CPLEX 9.1(ILOG, 2007)을 활용하여 기존 연구들에서 제공하지 못하였던 ROP-CC의 최적해를 최초로 제공해 보고자 한다. CPLEX는 ILOG사에서 개발한 최적화 소프트웨어로써 다양한 분야의 문제에 적용되어 뛰어난 성능을 나타내고 있다. Nahas and Nourelfath(2005)에서 사용된 직렬시

스템 예제에 대하여 CPLEX를 활용하여 최적해를 탐색해 보고, 문제의 해법을 위해 개발한 SA와 TS 알고리즘을 적용한다. 또한 기존 예제보다 사이즈가 큰 경우의 예제를 도출하여 TS로 탐색한 최적해를 SA의 초기해로 다시 설정하여 탐색하는 재최적화 절차(reoptimization procedure)도 평가해 보고자 한다.

제 2절에서는 본 연구에서 다루고자 하는 ROP-CC에 관한 수리적 모형에 대해 설명하며, 문제의 최적해 탐색을 위한 메타휴리스틱 알고리즘에 대해 제 3절에서 언급한다. 제 4절에서는 개발된 메타휴리스틱과 CPLEX를 활용한 수치실험의 결과에 대해 언급하며, 마지막으로 결론 및 추후연구를 제 5절에서 제시한다.

## 2. 수리적 모형

본 연구에서 다루고자 하는 ROP-CC는 중복설계를 고려하지 않는  $n$ 개의 구성품(서브시스템)으로 구성된 직렬시스템을 다루며, 각 구성품이 동일한 기능을 수행하지만 각기 다른 특성을 갖는 대안들을 가지는 경우에서의 시스템 구조를 결정하는 문제이다. 이 문제의 목적은 비용제약 하에서 최대의 시스템 신뢰도를 제공할 수 있는 구성품 대안의 최적 조합을 찾는 데 있다.

문제의 의사결정변수  $x_{ij}(i = 1, 2, \dots, n \ \& \ j = 1, 2, \dots, m_i)$ 는 다음과 같이 나타낼 수 있다.

$$x_{ij} = \begin{cases} 1, & \text{구성품 } i \text{에 대안 } j \text{를 사용한 경우} \\ 0, & \text{그렇지 않은 경우} \end{cases}$$

따라서 직렬시스템에서의 ROP-CC는 다음과 같은 최적화문제로 나타낼 수 있다.

$$\begin{aligned} \text{Maximize } R_s &= \prod_{i=1}^n \left( \sum_{j=1}^{m_i} x_{ij} r_{ij} \right) \\ \text{subject to } & \sum_{i=1}^n \sum_{j=1}^{m_i} x_{ij} c_{ij} \leq C \\ & \sum_{j=1}^{m_i} x_{ij} = 1, \quad \forall i = 1, 2, \dots, n \\ & x_{ij} = \{0, 1\}, \quad \forall i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, m_i \end{aligned}$$

여기서 비선형함수 형태인 문제의 목적함수는 대수(logarithm)를 취할 경우 선형함수로 표현할 수 있으나 중복설계를 고려하는 경우를 포함한 ROP-CC의 일반적인 형태에서와 같이 비선형함수로 표현하였으며, 기존 연구에서도 그러하다(Sung and Cho, 2000; Nahas and Nourelfath, 2005).  $r_{ij}$ 는  $i$ 번째 구성품에 사용된  $j$ 번째 대안의 신뢰성,  $c_{ij}$ 는  $i$ 번째 구성품에 사용된  $j$ 번째 대안의 비용을 의미한다. 또한 첫 번째 제약함수는 비용

제약(여기서 가용한 시스템 전체비용  $C$ 는 상수)을 나타내며, 두 번째 제약함수는 중복설계를 고려하지 않는 ROP-CC 문제임을, 세 번째 제약함수는 문제의 의사결정변수를 나타낸다. 이 문제는 비선형 목적함수를 갖는 이진정수계획법문제(binary integer programming problem)로 NP-hard 문제라 알려져 있어 지금까지 휴리스틱 방법의 개발에 관한 연구가 주로 수행되어 왔다.

### 3. 메타휴리스틱 알고리즘

메타휴리스틱은 국부탐색법(local search)에 기초한 휴리스틱 방법의 단점을 보완하고자 개발되었으며 쉬운 개념과 뛰어난 탐색능력 때문에 다양한 조합적 최적화문제의 해법에 있어 널리 사용되고 있다. 본 연구에서는 직렬시스템에서의 ROP-CC의 최적해 탐색을 위해 SA와 TS 알고리즘을 적용하며, 사이즈가 매우 큰 문제의 해결을 위해 SA와 TS 두 가지를 혼합한 재최적화 절차를 제시하고자 한다. ROP 분야에서는 최적해 탐색을 위해 GA를 활용하여 뛰어난 결과를 얻은 연구도 많이 있으나, Kulturel-Konak *et al.* (2003)은 GA가 모집단 단위의 탐색에 따라 많은 계산을 수행해야 한다는 단점을 가지기도 한다고 하였다. 이에 본 연구는 이후 중복설계를 고려한 보다 복잡한 ROP-CC로의 확장을 위해 복잡한 이산 최적화문제에 있어 뛰어난 성능을 나타내는 SA와 TS를 활용한다. SA는 열등한 해로의 전이도 확률적으로 허용(stochastic move)함으로 인해 국부최적해에서 빠져 나올 수 있다는 장점을 가지며, TS는 확정적 탐색(deterministic search)에 기반을 두어 변동성이 적어 최적해로의 수렴이 뛰어나다는 장점을 가진다. Jain and Meeran(1999)은 이러한 SA와 TS의 장점을 혼합한 하이브리드 알고리즘(hybrid algorithm)의 경우 다른 알고리즘들에 비해 생성능이 매우 뛰어나다고 하였으며, Zhang *et al.*(2008)은 TS와 SA의 혼합 알고리즘(TSSA)을 JSP(job shop scheduling problem)에 적용하여 그 결과가 뛰어난 것을 보였다.

#### 3.1 SA 알고리즘

SA는 전통적인 국부탐색법의 한 대안으로 Kirkpatrick *et al.*(1983)과 Cerny(1985)에 의해 제시된 이후, 다양한 조합적 최적화문제에 성공적으로 적용되어진 메타휴리스틱 기법 중 하나이다. SA는 주어진 인접해 구조 하에서 현재해로부터 인접해로 전이하는 반복적 과정을 통하여 전체 최적해를 찾아 나가는 방법으로 전통적인 국부탐색법의 단점을 극복하기 위해 상대적으로 열등한 해로의 전이도 확률적으로 허용함으로써 특정 국부최적해에서 빠져 나올 수 있는 가능성을 열어둔다는 특징을 가지고 있다. SA는 해의 탐색과정에서 해의 이동이 인접해 중에서의 무작위적 선택에 의해 이루어지고 해의 수락기준이 메트로폴리스 기준을 따르며 충분한 탐색시간이

주어진다면 전체 최적해로의 수렴성을 확률적으로 보장한다. 다양한 조합적 최적화문제에 쉽게 적용 가능한 SA 알고리즘을 개발하기 위해서는 우선 해 표현방법 및 성능평가함수를 정의하여야 하고, 초기해, 초기온도, 쿨링 스케줄 및 종료기준이 초기화되어야 한다.

ROP-SCC를 위한 SA 알고리즘의 해는 시스템 구성품 수, 즉  $n$ 개의 자리수로 표현되며 이는 각 구성품에서 사용할 부품 대안을 의미한다. <Figure 1>은 직렬시스템에서의 ROP-CC를 위한 해 표현의 예를 나타낸 것으로 첫 번째 자리의 3은 세 번째 대안을 첫 번째 구성품으로 사용하였음을 의미한다. SA의 성능평가함수인 에너지 함수  $E$ 는 문제의 목적함수를 그대로 사용하며, 제약식을 위반하는 해일 경우 에너지 함수값이 0이 되도록 한다. 초기 가능해는 전체  $n$  자리에 대해 난수발생방법을 실시한 무작위 해를 사용하며, 생성된 초기해는 현재해  $X_C$ 와 최선해  $X_B$ 로 초기 설정되도록 한다. 제어인자인 온도의 초기값과 최종값은 각각  $T_0$ 와  $T_F$ 로 나타내며, 일반적으로  $T_0$ 는 모든 가능한 이동을 받아들일 수 있도록 충분히 높게,  $T_F$ 는 가능한 0에 가까운 값으로 설정하기로 한다. 본 연구에서는  $T_0 = 100$ ,  $T_F = 1$ 의 값을 사용하였다.

현재해  $X_C$ 에서 인접해  $X_N$ 을 효율적으로 생성하는 것은 SA 알고리즘의 성능향상에 있어 매우 중요하다. 본 연구에서는 다음과 같이 2단계로 구성된 인접해 생성방법(scheme)을 제시한다.

(1) 단계 1 : 단계 1에서는 먼저 해의 전체  $n$  자리 가운데 임의로 2자리를 중복 없이 선택한 후 그 자리의 값들을 서로 교환하고, 다시 1자리를 임의로 선택하여 값을 다시 할당하는 방법이다. 생성된 해가 제약식을 만족하는 가능해인가를 판정하게 되며, 비 가능해일 경우 다시 해를 생성하게 된다. 만약 단계 1에서 다섯 번 연속으로 비 가능해를 생성할 경우에는 단계 2를 실행하게 된다.

(2) 단계 2 : 단계 2에서는 Nahas and Nourelfath(2005)와 같이 국부탐색법과 유사한 절차에 따라 이용 가능한 잔여자원을 고려하여 인접 가능해를 생성한다. 예를 들어 3개의 구성품으로 구성된 해의 표현이 {3, 2, 5}인 경우 탐색 절차에 의해 {4, 1, 5}, {4, 2, 4}, {2, 3, 5}, {2, 2, 6}, {3, 3, 4}, {3, 1, 6}의 인접해를 생성하여 자원제약 함수를 기준으로 해를 평가하게 되며, 이 가운데 목적함수를 최대화 하는 가능해를 선택하게 되는 것이다. Glover and Kochenberger(2001)와 Nahas and Nourelfath(2005)에서는 메타휴리스틱 기법과 이러한 인접해 탐색법을 함께 사용한 경우가 그렇지 않은 경우보다 일반적으로 성능 면에서 뛰어나다고 하였다.

생성된 인접해  $X_N$ 의 에너지 함수값  $E_N$ 이 현재해  $X_C$ 의 에너지

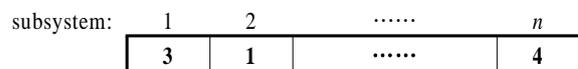


Figure 1. A sample path of the solution representation

지 함수값  $E_C$ 보다 크다면( $E_N > E_C$ ), 수락확률  $P(A) = 1$ 로 인접해  $X_N$ 을 수락하고 이를 현재해  $X_C$ 로 대체한다. 또한 이를 현재까지의 최선해  $X_B$ 의 에너지 함수값  $E_B$ 와 비교하여 그 값이 더 크다면( $E_N > E_B$ ), 인접해  $X_N$ 을 새로운 최선해  $X_B$ 로 대체한다. 만약 생성된 인접해  $X_N$ 의 에너지 함수값  $E_N$ 이 현재해  $X_C$ 의 에너지 함수값  $E_C$ 보다 작은 경우에는( $E_N < E_C$ ) 이를 무조건 기각시키는 것이 아니라 Metropolis *et al.*(1953)에 의해 제시된 Metropolis 기준에 따라 다음과 같이 계산된 수락확률  $P(A)$ 에 의해 확률적으로 인접해  $X_N$ 을 현재해  $X_C$ 로 받아들이게 된다. 여기서  $\Delta E = E_C - E_N$ 이며,  $k_B$ 는 볼츠만 상수(Boltzmann constant)를 나타낸다.

$$P(A) = \begin{cases} 1, & E_N > E_C \text{ or } E_N > E_B \\ \exp(-\Delta E/k_B T), & E_N < E_C \end{cases}$$

전형적인 SA 알고리즘은 크게 현재 온도를 적절히 조정하는 역할을 하는 외부루프와 설정된 온도에서 일정회수 만큼의 인접해 생성단계를 반복하는 내부루프로 구성된다. 내부루프는 일반적으로 해의 수렴성을 유지하기 위한 평형상태(equilibrium state)에 도달하기 위하여 충분한 횟수( $L$ )만큼 내부루프를 반복하게 된다. 그러나 이는 내부루프를 불필요하게 길어지도록 하는 결과를 초래할 수 있다.

이러한 SA의 단점을 보완하고 알고리즘의 수렴속도를 향상시키기 위해 Cardoso *et al.*(1994)은 일정횟수의 가능해가 획득되어진 경우 내부루프를 종료하도록 하는 개념인 NESA(non-equilibrium SA)를 제안하였으며, Ravi *et al.*(1997)은 이를 부품선택을 고려하지 않는 ROP 문제의 최적해 탐색에 적용하여 그 성능이 뛰어난 입증하였다. 본 연구에서도 NESA의 개념을 활용하여 내부루프 내에서 가능해가  $Q$ 번 획득되어진 경우 내부루프를 종료하도록 설정하였다. 본 연구에서는 예비실험을 통하여 내부루프 반복수  $L = 300$ ,  $Q = 10n$ 의 값을 사용한다.

내부루프가 종료되면 온도 강하율  $a$  ( $0 \leq a \leq 1$ )에 의해 현재 온도를  $T_p = aT_{p-1}$  ( $p = 1, 2, \dots$ )로 조정한다. 만약 조정된 현재 온도  $T_p$ 가 최종 온도  $T_f$ 보다 작으면( $T_p < T_f$ ) 알고리즘을 종료하고, 그렇지 않으면 현재 온도까지의 최선해  $X_B$ 를 현재해  $X_C$ 로 설정하고 다시 내부루프를 반복한다. 본 연구에서는 예비실험을 통해 예제 1~3에 대해서는  $a = 0.95$ 의 값을 사용하며, 예제 4와 5에는  $a = 0.98$ 의 값을 사용한다.

### 3.2 TS 알고리즘 및 재최적화 절차

TS는 Glover(1986)에 의해 처음으로 제안된 이후 TSP(traveling salesman problem), VRP(vehicle routing problem) 등의 다양한 조합적 최적화문제에 대해 많은 연구가 진행되었다. Kulturel-Konak *et al.*(2003, 2004)은 중복설계를 고려한 직병렬시스템(series-parallel system)에서 구성품의 혼합사용을 허용하는 ROP-CC의 최적해 탐색을 위한 TS 알고리즘을 개발하였으며,

본 연구에서 다루고자 하는 문제의 최적해 탐색을 위한 TS 알고리즘 또한 Kulturel-Konak *et al.*(2003, 2004)이 제시한 TS 알고리즘을 토대로 구성하였다. 다만, 본 연구의 경우에는 중복설계를 고려하지 않는 경우이므로 인접해를 탐색하는 전략을 수정하였다. Kulturel-Konak *et al.*(2003, 2004)의 TS 알고리즘을 설명하기 위한 기호는 다음과 같다.

BEST MOVE : 탐색 전략에 의한 인접해로의 이동 중 가장 좋은 것

BEST SO FAR : 한 Iteration 내에서의 가능해 또는 비가능해 중에서 가장 좋은 해

BEST FEASIBLE SO FAR : Iteration 중의 가능해 중에서 가장 좋은 해

TL : 선정된 BEST MOVE들의 집합(타부 리스트)

MAXITER : 중단 Iteration의 개수

Kulturel-Konak *et al.*(2003, 2004)의 TS 알고리즘을 간략히 기술하면 다음과 같다.

단계 0 : 초기 가능해 구성

단계 1 : 탐색전략에 의해 BEST MOVE를 선정. 선정된 BEST MOVE가 BEST SO FAR보다 좋지도 않으면서 TL에 있으면, 단계 1을 반복. 그렇지 않으면, BEST MOVE로 채택

단계 2 : 선정된 BEST MOVE를 TL에 입력

단계 3 : BEST FEASIBLE SO FAR를 개선시키지 못한 Iteration의 개수가 MAXITER 보다 크면, 종료. 그렇지 않으면, 단계 1을 반복

Kulturel-Konak *et al.*(2003, 2004)은 인접해의 탐색 전략으로 두 가지의 경우를 고려하였는데 그 중 하나는 BEST MOVE의 중복설계의 개수 1개를 늘여보고 줄여보는 것이고, 두 번째의 전략은 BEST MOVE의 대안 구성품을 그 외의 다른 모든 구성품으로 바꾸어 보는 것이다. 또한, TL의 크기도 매 20 Iteration마다  $U[n, 3n]$  사이의 값을 랜덤하게 선택하여 다양성(diversification)을 줌으로써 국부 최적해(local optimum)에 빠지는 위험을 줄이는 전략을 사용하였다. MAXITER은 실험결과 1000개가 적당하다고 제시하였고, 비 가능해의 탐색을 위해 벌과함수(penalty objective function)를 도입하였다.

본 연구에서의 ROP-CC의 해법을 위한 TS 알고리즘은 Kulturel-Konak *et al.*(2003, 2004)이 제시한 TS를 그대로 적용하였다. 다만 앞에서 언급한 바와 같이 본 연구는 중복설계를 고려하지 않는 경우이므로 Kulturel-Konak *et al.*(2003, 2004)의 두 가지 탐색전략 중 두 번째 전략만을 적용하였다.

또한 본 연구에서는 TS로 탐색한 최적해를 SA의 초기해로 다시 설정하여 탐색하는 재최적화 절차(reoptimization procedure)를 고안해 보았으며 제 4절의 수치실험을 통해 그 성능을 비교하였다.

## 4. 수치실험 및 결과

본 절에서는 제시한 알고리즘의 성능 평가를 위해 몇 가지 예제를 활용한 수치실험을 실시하며, 예제들의 최적해 탐색을

Table 1. Component data for the example 4

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

위해 상용 최적화 소프트웨어의 하나인 ILOG사의 CPLEX 9.1을 활용한다. CPLEX의 정수계획법 모듈에서는 비선형 목적함수를 다룰 수 없기 때문에 문제의 목적함수를 다음 식과 같이 대수 변환한 문제로 바꾸며, 탐색된 결과를 다시 지수 환원하여 원문제의 목적함수 값을 구한다. 직렬시스템에서의 ROP-CC의 경우 의사결정변수인  $x_{ij}$ 가 이진변수(binary variables)이기 때문에 다음과 같이 대수 변환 시키면 선형 목적함수가 된다.

$$\text{Maximize } \ln R_s = \ln \left( \prod_{i=1}^n \left( \sum_{j=1}^{m_i} x_{ij} r_{ij} \right) \right) = \sum_{i=1}^n \sum_{j=1}^{m_i} x_{ij} \ln r_{ij}$$

본 연구에서 사용된 예제는 Nahas and Nourelfath(2005)의 연구에서 제시된 4가지 예제와 이 가운데 가장 복잡한 경우인 예제 4의 데이터를 다시 직렬구조 형태로 연결하여 큰 사이즈로 형성한 예제들로 구성된다. 우선 기존 4가지 예제들은 각각 61, 80, 100, 166개의 구성품 대안이 시스템 내에 존재하며 예제 1~3은 15개, 예제 4는 25개의 구성품이 직렬로 연결된 시스템이다. 가용한 시스템 비용은 각각 1000, 900, 1000, 1400과

같다. 예제 1~3에 대한 구성품 대안들의 데이터는 Nahas and Nourelfath(2005)를 참고하기 바라며, 예제 4의 데이터를 <Table 1>에 나타내었다.

<Table 2>는 예제 1~4 각각에 대하여 개발된 SA 및 TS 알고리즘을 활용하여 최적해 탐색실험을 10회 실시한 결과를 최소값, 최대값, 평균 및 표준편차로 나타내고, CPLEX를 통해 탐색한 문제의 최적해를 나타낸 것이다. 제시한 SA 및 TS 알고리즘은 C++로 코딩하였으며, 모든 수치실험은 Pentium IV 3.0GHz PC에서 수행되었다. 예제 1~4 모든 경우에 대해서 개발된 SA 및 TS 알고리즘이 10번 시행에서 편차 없이 모두 최적해를 제공하는 것으로 나타났으며, 잔여비용은 각 예제에서 10, 5, 5, 0으로 나타났다. 최적해 탐색시간은 예제 1~3의 경우 SA 알고리즘은 2초 이내 TS 알고리즘은 1초 이내, 예제 4의 경우 SA 알고리즘은 4초 이내, TS 알고리즘은 2초 이내를 소요한 것으로 나타나 TS 알고리즘이 탐색속도가 빠른 것으로 나타났다.

수치실험 결과를 통해 제시한 SA 및 TS 알고리즘이 Nahas and Nourelfath(2005)에서 사용하였던 AS 알고리즘보다 그 성능이 뛰어난을 알 수 있었으며, 알고리즘의 수렴성으로 보아

Table 2. The experimental results for the examples 1~4

Example	Nahas and Nourelfath(2005)				SA algorithm				TS algorithm			
	Min	Max	Average	S.D.	Min	Max	Average	S.D.	Min	Max	Average	S.D.
1	0.85705	0.85705	0.85705	0	0.857054	0.857054	0.857054	0	0.857054	0.857054	0.857054	0
2	0.91504	0.91504	0.91504	0	0.915042	0.915042	0.915042	0	0.915042	0.915042	0.915042	0
3	0.96406	0.96513	0.96439	0.00050	0.965134	0.965134	0.965134	0	0.965134	0.965134	0.965134	0
4	0.86465	0.86543	0.86491	0.00038	0.865439	0.865439	0.865439	0	0.865439	0.865439	0.865439	0
CPLEX	Example 1 : 0.857054 (3-4-5-2-3-3-2-3-2-2-2-3-4-3-2)											
	Example 2 : 0.915042 (3-3-3-4-2-3-3-2-4-1-2-3-4-3-1)											
	Example 3 : 0.965134 (3-3-4-4-3-3-2-2-3-2-2-4-4-4-2)											
	Example 4 : 0.865439 (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) 0.865439 (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)											

Table 3. The experimental results for the example 5

C	CPLEX	SA algorithm			TS algorithm		
		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	<b>0</b>
7250	<b>0.900167</b>	0.899438	0.001050	-0.7302	0.899984	0.000305	-0.1833
7300	<b>0.904599</b>	0.903866	0.001027	-0.7683	0.904414	0.000390	-0.1939
7350	<b>0.908866</b>	0.908405	0.001202	-0.5058	<b>0.908866</b>	0.000480	<b>0</b>
7400	<b>0.913154</b>	0.912601	0.000499	-0.6368	0.913064	0.000337	-0.1036
7450	<b>0.917184</b>	0.916815	0.000510	-0.4456	0.917093	0.000494	-0.1099
7500	<b>0.921141</b>	0.920770	0.000743	-0.4705	<b>0.921141</b>	0.000365	<b>0</b>
7550	<b>0.925023</b>	<b>0.925023</b>	0.000590	<b>0</b>	<b>0.925023</b>	0.000502	<b>0</b>
7600	<b>0.929013</b>	0.928269	0.000696	-1.0481	<b>0.929013</b>	0.000445	<b>0</b>
7650	<b>0.931526</b>	<b>0.931526</b>	0.000388	<b>0</b>	<b>0.931526</b>	<b>0</b>	<b>0</b>

※ %MPI = (Max - CPLEX) / (1 - CPLEX) × 100%

Table 4. The experimental results for the example 6

C	CPLEX	TS algorithm			TS Reoptimization		
		Max	S.D.	%MPI	Max	S.D.	%MPI
14400	<b>0.802546</b>	0.802218	0.000425	-0.1661	0.802218	0.000000	-0.1661
14450	<b>0.806496</b>	0.806167	0.000396	-0.1700	0.806167	0.000000	-0.1700
14500	<b>0.810301</b>	0.809890	0.000519	-0.2167	0.809970	0.000000	-0.1745
14550	<b>0.814290</b>	0.813792	0.000352	-0.2682	0.813792	0.000000	-0.2682
14600	<b>0.818299</b>	0.817388	0.000391	-0.5014	0.817798	0.000053	-0.2757
14650	<b>0.822160</b>	0.821656	0.000891	-0.2834	0.821656	0.000000	-0.2834
14700	<b>0.826207</b>	0.824787	0.000709	-0.8171	0.825364	0.000142	-0.4851
14750	<b>0.830105</b>	0.829263	0.000815	-0.4956	0.829427	0.000026	-0.3991
14800	<b>0.833851</b>	0.833428	0.000891	-0.2546	0.833510	0.000026	-0.2052
14850	<b>0.837614</b>	0.837448	0.000824	-0.1022	<b>0.837614</b>	<b>0</b>	<b>0</b>
14900	<b>0.841310</b>	0.840805	0.000786	-0.3182	<b>0.841310</b>	0.000107	<b>0</b>
14950	<b>0.844856</b>	0.844009	0.000506	-0.5459	<b>0.844856</b>	0.000215	<b>0</b>
15000	<b>0.848500</b>	0.848332	0.000610	-0.1109	<b>0.848500</b>	0.000027	<b>0</b>
15050	<b>0.852076</b>	0.851991	0.000722	-0.0575	<b>0.852076</b>	<b>0</b>	<b>0</b>
15100	<b>0.855751</b>	0.855582	0.000679	-0.1172	<b>0.855751</b>	<b>0</b>	<b>0</b>

보다 큰 사이즈의 문제에 대해서도 충분히 최적해 탐색이 가능할 것으로 판단되었다. 이에 본 연구에서는 예제 중 사이즈가 가장 큰 예제 4의 시스템 구조를 직렬로 다시 연결한 형태로 구성품의 수가 100개, 200개인 경우의 문제(이를 예제 5, 6이라 함)를 생성하였다. CPLEX를 활용한 예비실험을 통해 원 문제의 비용제약을 선형으로 증가시킬 경우 시스템 전체 신뢰도가 현저히 낮아져 현실적이지 못하여 사용 가능한 비용제약을 각각 7200, 14400으로부터 시작하여 50씩 증가시킨 10개와 15개의 경우에 대한 수치실험을 실시하였다.

<Table 3>는 100개의 구성품이 직렬로 연결된 예제5에 대한 CPLEX 최적해 및 SA와 TS 알고리즘의 최적해 탐색결과를 나타낸 것이다. 이 경우 TS 알고리즘의 결과가 10가지 경우에서 6번 최적해를 제공하였으며, %MPI 수치와 같이 최적해를 제공하지 못한 경우에도 SA 알고리즘의 결과보다는 더 나은 TS 알고리즘의 탐색능력이 SA 알고리즘보다 뛰어난 것을 알 수 있다. 최적해 탐색시간 면에서 또한 SA 알고리즘이 평균 12.07초를 소요한 반면 TS 알고리즘은 평균 4.41초를 소요하여 TS 알고리즘이 보다 단시간 내 더 나은 해를 제공함을 알 수 있었다.

예제 6은 200개의 구성품이 직렬로 연결된 시스템이다. 예제 5의 수치실험을 통해 TS 알고리즘이 SA 알고리즘보다 나은 해를 제공하긴 하나 최적해를 제공하지 못하는 것을 알 수 있었다. 이에 본 연구에서는 TS 알고리즘에서 탐색한 최선해를 SA의 초기해로 하여 최적해를 재탐색하는 방법을 사용하는 재최적화 절차를 예제6에 적용해 보았으며, 수치실험 결과를 <Table 4>에 나타내었다. 각 경우에서 비용제약이 커질수록 탐색 가능한 영역 내에서 재최적화 절차가 보다 나은 해를 찾을 가능성이 높아짐을 알 수 있으며, 일정 수준 이상의 비용

제약 하에서는 재최적화 절차가 최적해를 제공해주는 것을 알 수 있다. CPLEX의 경우 문제 사이즈가 매우 큰 예제6의 경우에서도 모두 최적해를 제공함을 알 수 있다. 본 연구는 보다 복잡한 구조를 갖는 중복설계를 고려한 경우의 ROP-CC로의 확장을 위한 기초 연구이다. 중복설계를 고려할 경우 구성품 대안을 활용한 서비스시스템의 구성에 있어 경우의 수가 증가함에 따른 변수의 증가로 CPLEX의 해 탐색에 한계가 발생하게 된다. 이러한 경우 본 연구에서 제안한 메타휴리스틱 알고리즘 및 재최적화 절차가 유용하게 적용될 수 있을 것이다.

## 5. 결론 및 추후연구

본 연구에서는 부품선택이 존재하는 직렬시스템에서의 신뢰성 최적설계 문제를 다루었으며, CPLEX 9.1, SA, TS 및 TS의 재최적화 알고리즘을 활용하여 문제의 최적해를 탐색해 보았다. 제시한 각 알고리즘을 기존 연구에서 다루었던 예제에 적용해 보았으며, 보다 큰 사이즈의 예제를 도출하고 해법들을 적용하여 그 성능을 평가해 보았다. 전역 최적해 제공을 위해 사용된 CPLEX의 경우 문제 사이즈가 매우 큰 경우에도 최적해를 제공함을 알 수 있었으며, TS의 재최적화 절차가 개별 알고리즘에서보다 나은 결과를 나타냄을 알 수 있었다.

본 논문에서는 상대적으로 간단한 구조를 갖는 직렬시스템을 대상으로 하였으며, 시스템 비용제약만이 문제의 제약함수로 사용되었다. 중복설계 및 구성품 대안의 혼합사용을 고려한 더욱 복잡한 구조를 갖는 직렬시스템에서의 ROP-CC에 대한 적용과 다양한 자원제약을 고려할 수 있을 것이다. 또한 효율적인 하이브리드 메타휴리스틱 알고리즘의 개발도 추후

연구로 남겨두고자 한다

## 참고문헌

- 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.
- Chen, T. C. and You, P. S. (2005), Immune algorithms-based approach for redundant reliability problems with multiple component choices, *Computers in Industry*, **56**, 195-205.
- Coit, D. E. and Smith, A. E. (1996a), Reliability optimization of series-parallel systems using a genetic algorithm, *IEEE Transactions on Reliability*, **45**(2), 254-260.
- 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.
- Glover, F. (1986), Future paths for integer programming and links to artificial intelligence, *Computers and Operations Research*, **13**, 533-549.
- Glover, F. and Kochenberger, G. A. (2001), *Handbook of Metaheuristics*, Kluwer Academic Pub.
- Hsieh, Y. C. (2002), A linear approximation for redundant reliability problems with multiple component choices, *Computers & Industrial Engineering*, **44**, 91-103.
- ILOG, Inc. (2007), *ILOG CPLEX*, <http://www.ilog.com/products/cplex/>.
- Jain, A. S. and Meeran, S. (1999), Deterministic job shop scheduling: past, present and future, *European Journal of Operational Research*, **113**, 390-434.
- Kim, H. G., Bae, C. O. and Paik, C. H. (2004), A simulated annealing algorithm for the optimal reliability design problem of a series system with multiple component choices, *IE Interfaces*, **17**(Special Edition), 69-78.
- Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983), Optimization by simulated annealing, *Science*, **220**, 671-679.
- Kulturel-Konak, S., Smith, A. E. and Coit, D. W. (2003), Efficiently solving the redundancy allocation problem using tabu search, *IEE Transactions*, **35**, 515-526.
- Kulturel-Konak, S., Norman, B. A., Coit, D. W. and Smith, A. E. (2004), Exploiting tabu search memory in constrained problem, *INFORMS Journal on Computing*, **16**(3), 241-254.
- Kuo, W., Prasad, V. R., Tillman, F. A. and Hwang, C. L. (2001), *Optimal Reliability Design : Fundamentals and Applications*, Cambridge University Press, Cambridge.
- Liang, Y. C. and Smith, A. E. (2004), Ant colony optimization algorithm for the redundancy allocation problem (RAP), *IEEE Transactions on Reliability*, **53**(3), 417-423.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. (1953), Equation of state calculations by fast computing machines, *Journal of Chemical Physics*, **21**, 1087-1092.
- 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.
- Nahas, N., Noureldath, M. and Ait-Kadi, D. (2007), Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series-parallel systems, *Reliability Engineering and System Safety*, **92**, 211-222.
- 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-Marquez, 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.
- Zhang, C. Y., Li, P. G., Rao, Y. Q. and Guan, Z. L. (2008), A very fast TS/SA algorithm for the job shop scheduling problem, *Computers & Operations Research*, **35**, 282-294.
- Zhao, J. H., Liu, Z. and Dao, M. T. (2007), Reliability optimization using multiobjective ant colony system approaches, *Reliability Engineering and System Safety*, **92**, 109-120.