

## 다구치 기법을 이용한 시뮬레이티드 어닐링 알고리즘의 최적화

### Optimizing Simulated Annealing Algorithms Using Taguchi Method

김호균 · 조형수 · 배창욱  
동의대학교 정보산업공학과

#### Abstract

The performance of simulated annealing (SA) algorithm such as solution optimality and computation time mainly depends on how to determine the SA-related parameters. Several schemes have been suggested to improve the performance of SA and several parameter design methods have been utilized to select parameter values of each scheme. In this paper, we propose a new SA algorithm design method that can determine schemes as well as parameter values simultaneously. The new SA algorithm design method is based on the Taguchi method which primarily selects the design parameters for a product or process to minimize the effect of noise parameters, so that the response is close to the desired target with minimum variation. To show the effectiveness of the proposed method, extensive computation experiments are conducted.

#### 1. 서론

시뮬레이티드 어닐링(SA; simulated annealing)은 고체 물리학에서 에너지 수준이 가장 낮은 상태인 결정을 얻기 위해서 이용하고 있는 어닐링(annealing) 과정을 조합적 최적화 문제에 모사한 것으로, Kirkpatrick *et al.*(1983)과 Cerny (1985)에 의해 제시된 이후 다양한 조합적 최적화 문제에 적용되어 성공적인 성과를 이룩한 메타휴리스틱(meta heuristic) 방법 중 하나이다. SA는 주어진 인접해 구조하에서 현재해로부터 인접해로 전이하는 반복적 과정을 통하여 전체 최적해를 찾아나가는 방법으로, 상대적으로 열중한 해로의 전이도 확률적으로 허용함으로써 특정 국부 최소해에

서 빠져 나올 수 있는 가능성을 열어둔다는 특징을 가지고 있다.

SA 알고리즘은 기존의 발견적 해법들에 비해 기본 개념이 간단하며, 알고리즘도 간단하다. 또한 융통성이 매우 크기 때문에 서로 다른 형태의 문제에 쉽게 적용 가능하며, 특정 최적화 문제에 활용하기가 매우 용이하다는 장점을 가지고 있다. SA를 실제 문제에 적용하기 위해서는 특정 문제에 관련된 모수인 문제별 모수(problem-specific parameter)와 특정 문제와는 어느 정도 무관한 일반 모수(generic parameter)가 먼저 결정되어야 한다. SA는 이론적으로는 전체 최적해에 수렴하는 효율적인 방법이지만, 긴 계산시간이 소요된다는 단점을 가지고 있다. 따라서 실제 적용상에서는 해의 최적성이나 수렴속도를 가속화하기 위해 문제별 모수 및 일반 모수의 결정에 대한 여러 가지 변형적인 방법들이 제시되었다(Ali *et al.*, 2002; Arts and Van Laarhoven, 1987; Huang *et al.*, 1986; Johnson *et al.*, 1989; Lundy and Mees, 1986; Youssef *et al.*, 1991). 그러나 일반모수들 간에는 상호작용이 존재함에도 불구하고 각 성능가속화 방법들이 동시에 고려되지 못하여 SA의 성능척도들을 동시에 향상시키지는 못하였다. 최근 요인실험법(factorial design) 및 심플렉스법(simplex method) 등을 이용하여 일반모수들의 상호작용이 고려된 최적의 모수값 결정에 관한 연구가 이루어졌다(Park and Kim, 1998). 그러나 특정한 성능가속화 방법이 주어진 후 모수값을 결정하게 되어 SA 성능 향상에는 어느 정도 한계가 존재하게 된다. Kim *et al.*(2002)은 SA의 두 가지 성능척도를 동시에 향상시킬 수 있는 방법으로 다구치(Taguchi)의 강건설계(robust) 기법에 기초한 SA 알고리즘의 최적 설계방법을 제시되었으나, 일반모수들의 교호효과(interaction effect)가 고려되지 않아 완전하지 못하였다. 이에 본 연구에서는 Kim *et al.*(2002)의 연구를 확장하여 일반

모수들의 교호효과와 설계시에 고려되어야 할 여러 가지 인자들을 고려한 SA 알고리즘의 최적 설계방법을 제시한다.

제시한 SA의 최적 설계를 위한 표본문제로는 혼합모델 조립라인에서의 투입순서 결정 문제를 사용한다. 혼합모델 투입순서 결정 문제는 조합적 최적화 문제 중 NP-hard 영역이어서 규모가 큰 문제는 계산시간 제약상 거의 비실행적인 경우가 대부분이다. 혼합모델 투입순서는 수요형태의 변화라든가 부품 품질의 발생으로 인하여 실시간으로 변경될 수 있어야 하기 때문에 계산시간은 투입순서 결정 문제를 해결하는 방법을 개발하는데 있어 매우 중요한 요소가 된다. 이러한 이유에서, 다양한 휴리스틱 방법들이 많은 연구자들에게 의해 개발되어져 왔다(Bard et al., 1994; Bolat et al., 1994; Dar-El, 1978). 최근에는 이러한 형태의 조합적 최적화 문제에 대해서, 시뮬레이티드 어닐링, 유전자 알고리즘(GA; genetic algorithm) 및 타부 서치(TS; Tabu search) 등과 같은 메타 휴리스틱 탐색기법들이 사용되어지고 있으며 성공적인 성과를 제공하였다(Celano et al., 1999; Kim et al., 1996; McMullen, 1998; McMullen and Frazier, 2000; Tamura et al., 1999). 혼합모델 투입순서 결정시 중요하게 고려되어야 할 목적으로는 부품사용률 변동의 최소화, 작업부하의 균형화, 정지위험의 최소화, 총 가외작업의 최소화, 작업준비비용의 최소화 등과 같은 다양한 목적들이 존재한다. McMullen and Frazier(2000)는 상충적 관계에 있는 목적들인 부품사용률 안정화 목적과 작업준비시간의 최소화란 두 가지 목적을 갖는 혼합모델 투입순서 결정 문제에 대하여 SA 알고리즘을 소개하였다. 본 연구에서 언급되는 수리적 모형은 McMullen and Frazier (2000)에 기초한다.

본 논문의 구성을 살펴보면, 2절에서는 SA 알고리즘 강건설계의 표본문제로 사용된 혼합모델 투입순서 결정 문제를 기술한다. 3절에서는 시뮬레이티드 어닐링의 다양한 성능 가속화 방법들을 소개하고 다구치의 강건 설계방법을 활용하여 성능 최적화된 알고리즘을 설계한다. 그리고 4절에서는 여러 가지 실험을 통하여 기존 연구(McMullen and Frazier, 2000)에 의한 알고리즘과 최적 설계된 알고리즘에 의한 해를 비교 분석하여, 제시된 성능최적화 알고리즘의 타당성을 검증한다.

## 2. 혼합모델 투입순서 결정 문제

혼합모델 조립라인은 보통 두 가지 이상의 서로 다른 모델들을 갖는 동일한 제품을 동시에 조

립하는데 사용된다. 생산될 제품들은 라인에서 연속적으로 한 모델에서 다른 모델로 전환되며 생산된다. 혼합모델 조립라인은 여러 가지 시장의 변동에 대해 유연하고 매우 빠르게 대처할 수 있게 된다. 혼합모델 조립라인을 효율적으로 운영하기 위해서는 회사 목적에 맞는 혼합모델들의 적절한 투입순서를 결정하는 것이 매우 중요하다.

혼합모델 조립라인의 투입순서 결정문제에서 고려되어야 할 목적들을 위한 수리적 모형에서 기본적으로 사용되는 기호는 다음과 같이 정의된다.

- $n$  : 최종 제품 모델의 수,
- $D$  : 모든 제품 모델들의 총 생산 수요량 혹은 투입순서를 결정할 총 수량,
- $d_i$  : 제품 모델  $i$ 에 대한 수요량,  $i=1, 2, \dots, n$ ,
- $n_i$  : 제품 모델  $i$ 에 대한 수요율 ( $=d_i/D$ ),  
 $i=1, 2, \dots, n$ ,
- $s_k$  : 생산단계  $k$ 에서의 작업준비시간,  
 $k=1, 2, \dots, D$ ,
- $x_{ik}$  : 생산단계  $k$ 까지 제품 모델  $i$ 의 총 생산량,  
 $i=1, 2, \dots, n, k=1, 2, \dots, D$ ,

여기서 생산단계(stage)란 하나의 제품이 생산되는 시간 단위를 의미하는 것으로,  $k$ 는 혼합모델 투입순서 중 특정 위치이고, 여러 제품들이 총  $k$  단위가 생산되었다면, 생산단계  $k$ 가 끝난 것이다. 생산단계  $k$ 는 최종 제품에 대한 총 수요량  $D$ 를 만족할 때까지 이어지게 된다.

혼합모델 투입순서 결정시에 고려되어야 할 목적에는 여러 가지가 존재하지만, 본 연구에서는 두 가지만을 고려한다. 첫 번째 목적인 부품사용률 변동의 최소화 목적은 각 제품 모델의 생산이 그 모델에 대한 수요와 동시에 진행되어 일정한 비율로 여러 제품들을 생산할 수 있도록 투입순서를 결정하는 것이다. 이것은 생산단계  $k$ 에서 생산한 각 모델별 실제 생산량에 대한 비율과 전체 수요율과의 변동값을 최소화함으로써 달성될 수 있으며, 총 생산 변동  $U$ 는 Miltenburg(1989)에 의해 제시된 척도로 다음 식에 의해 계산된다.

$$U = \sum_{k=1}^D \sum_{i=1}^n (x_{ik} - kr_i)^2 \quad (1)$$

혼합모델 투입순서 결정시에 고려되어야 할 두 번째 목적은 혼합모델들의 생산전환으로 인해 발생하는 작업준비시간을 최소화하는 것으로, 생산단계  $k$ 까지 발생한 총 작업준비시간의 합을 최

소화함으로써 달성될 수 있다. 총 작업준비시간의 합  $S$ 는 McMullen and Frazier (2000)에 의해 제시된 척도로 다음과 같이 계산될 수 있다.

$$S = \sum_{k=1}^D s_k \quad (2)$$

이때,  $s_1$ 은 초기 작업준비시간이며, 각 작업준비시간은 단위시간을 갖는다고 가정한다.

### 3. 시뮬레이티드 어닐링 성능향상을 위한 알고리즘 최적 설계

SA를 실제 문제에 적용하기 위해서는 문제별 모수인 해의 표현방법, 인접해의 구조, 비용함수 및 초기해의 획득 방법 등이 정의되어야 하며, 일반 모수인 초기 온도, 쿨링스케줄, 내부루프의 반복수 및 종료기준 등이 결정되어야 한다. 이때, 일반 모수들은 SA 알고리즘의 성능 즉, 해의 최적성 및 수렴속도(계산시간)에 크게 영향을 미치는 모수들로서, 각 모수값에 따라 알고리즘의 성능 척도들은 상충적 관계를 가진다. 따라서 상충적 관계를 갖는 성능 척도들을 동시에 만족시킬 수 있도록 일반 모수가 적절히 선택되어야 한다. 본 절에서는 상충적 관계를 갖는 성능 척도들을 동시에 만족시킬 수 있도록 SA의 일반 모수들을 최적 설계한다.

#### 3.1 최적 설계를 위한 실험 설계

SA 알고리즘의 최적 설계를 위해 다구치 방법을 적용하려면 먼저 제어인자, 잡음인자, 각 인자의 수준 및 수집할 특성치 등이 정의되어야 한다. 먼저 특성치에 영향을 미치는 인자들 중에서 제어 가능한 인자인 제어인자로는 초기온도, 내부루프 반복수, 쿨링스케줄 및 종료기준으로 설정한다. 각 제어인자에 대한 수준들로는 SA의 성능향상을 위해 고려될 수 있는 일반적인 성능 가속화 방법들 중 실험계획 구현의 편의성을 고려하여 주요 인자별로 세 가지 방법들을 고려한다.

#### 초기온도

Kirkpatrick *et al.*(1983)은 초기온도  $T_1$ 값을 모든 전이를 받아들일 수 있도록 충분히 높게 설정하여야 한다고 했다. 그러나 초기온도가 너무 높으면 계산시간이 길어지는 등 성능에 좋지 않은 영향을 줄 수 있다. 최적설계에 고려된 제어인자 중 초기온도를 구하는 방법으로, 첫 번째 방법은 초기온도  $T_1$ 값을 미리 결정된 상수값으로 설정하는 방법이다. 두 번째 방법은 Huang *et al.*(1986)에 의

해 제시된 것으로 예비실험을 통해 에너지 분포의 표준편차( $\sigma$ )를 구해서  $k\sigma$ 로 설정하는 방법을 고려한다. 이때  $k$ 값은 에너지 분포를 정규분포로 가정하고  $3\sigma$ 밖의 에너지 값을 확률  $p$ 로 받아들일 수 있도록  $k = -3/\ln p$ 로 결정된다. 세 번째 방법은 Arts and Van Laarhoven(1987)에 의해 제시된 방법으로 해의 이동을 받아들이는 비율을 사용하여 초기온도를 구하는 방법으로서, 초기실험을 수행하여 해를 받아들이는 비율이 어떤 정해진 값( $a_0$ )보다 크도록 초기온도를 다음과 같이 정하는 것이다.

$$T_1 = \frac{\Delta^+}{\ln(m_1(a_0 - 1)/m_2 + a_0)} \quad (3)$$

여기서,  $m_1$ 과  $m_2$ 는 예비실험에서 에너지함수값이 감소된 해의 수와 증가된 해의 수를 나타내며,  $\Delta^+$ 는 에너지함수값이 증가된 해들의 평균 에너지 증가값이고  $a_0$ 는 해의 이동을 받아들이고자 하는 기대값이다.

#### 내부루프 반복수 및 해의 이동과정

두 번째 제어인자인 내부루프의 반복수,  $L$ 은 해가 수렴할 수 있도록 충분히 크게 해주어야 한다. 그러나, 수렴성의 보장을 위해서  $L$ 을 필요 이상으로 크게 해주면 계산성능에 치명적인 영향을 미칠 수 있기 때문에 일반적으로  $L$ 은 인접의 규모( $N_h$ )에 일정한 상수( $v$ )를 곱하는  $L = v \times N_h$ 과 같은 형태를 취하고 있다. 이러한 방법을 상수 설정법이라 부르기로 한다. 상수로 설정하는 방법 외에 항상 동일한 길이를 갖는 것이 아니라 주어진 상황에 적절히 적용하는 방법들도 제안되었다(Kirkpatrick *et al.*, 1983). 두 번째 방법은  $L$ 을 상대적으로 작은 값을 1차적으로 설정하여 계산시간을 줄일 수 있도록 하고, 만약 내부루프의 1차 반복 후에도 에너지 함수값에 변화가 없으면 아직 안정상태에 도달하지 못한 것으로 간주하여 현재 해에 좀더 많은 변동을 줄 수 있는 방법으로 내부루프를 재차 반복하는 방법을 고려한다. 이를 재순환법이라 부르기로 한다. 세 번째 방법은 내부루프 수행 중 발생된 가장 낮은 에너지값( $f_1$ )과 가장 높은 에너지값( $f_h$ )의 차이에 대한 지수함수값을 기본적 내부루프 반복수에 더해가는 방법으로서, 해가 수렴해감에 따라 생성되는 에너지값의 차이가 점점 작아짐을 이용하여 내부루프 반복수를 신축적으로 설정하는 것으로 다음과 같이 설정된다.

$$L - L_B + \lfloor L_B F \rfloor \quad (4)$$

여기서,  $\lfloor x \rfloor$  는  $x$  를 넘지 않는 최대 정수이고,  $F = 1 - \exp(- (f_n - f_i)/f_n)$  이며,  $L_B$  는 기본적인 내부루프의 반복수를 나타내는 것으로 상수설정 방법과 마찬가지로  $v \times M_n$  로 계산된다. 이것은 Ali et al. (2002) 가 제시한 방법에 기초하여 변형된 것으로 지수 평활법이라 부르기라고 하고, 인접해 생성방법은 교환방법을 사용한다.

위 방법들에 의하면 해의 탐색공간은 항상 가능성(feasibility)을 유지하게 되며, 인접의 규모, 즉 발생 가능한 인접해의 개수는  $n(n-1)/2$  가 된다. 여기서,  $n$  은 문제크기를 나타내는 것으로서 투입 순서를 결정할 최종 제품 모델들에 대한 총 생산 수요량인  $D$  이다.

#### 쿨링스케줄

세 번째 제어인자인 쿨링스케줄에 대해서 기존의 연구들에서 사용되는 기본적인 쿨링스케줄은 다음과 같은 단순 기하스케줄을 사용하고 있다.

$$T_k = \alpha T_{k-1}, \quad k = 2, 3, \dots, \quad 0 < \alpha < 1 \quad (5)$$

이때  $\alpha$  는 온도 강하율이라 하며 대개 0.5와 0.99 사이에서 정해지지만, 일반적으로 0.9나 0.95를 많이 사용한다.

두 번째 쿨링스케줄은 Lundy and Mees(1986)에 의해 제시된 방법으로써 다음과 같은 식에 의해 온도를 강하시킨다.

$$T_k = T_{k-1} / (1 + \beta T_{k-1}), \quad \beta > 0 \quad (6)$$

이때,  $\beta = (T_1 - T_F) / ((L-1)T_1 T_F)$  로 계산되며,  $L$  은 내부루프의 반복수,  $T_F$  는 최종온도를 나타낸다. 이러한 쿨링스케줄 방법은 온도 강하속도를 첫 번째 방법인 단순 기하스케줄보다 더 느리게 만든 것으로 최적해 탐색의 가능성을 더 높게 된다. 이러한 방법을 LM 쿨링스케줄이라 부르기라고 한다.

세 번째 고려된 쿨링스케줄은 기본적으로는 단순 기하스케줄을 사용하지만 고려되는 온도 강하율에 변화를 주는 방법이다.  $A_c$  를 특정 온도에서의 내부루프 수행 도중 발생한 에너지 함수값(현재해)의 변화율이라 정의하고,  $P(A)$  를 열등한 해의 평균 수락확률이라 하면 온도 강하율  $\alpha$  는 다음과 같이 결정된다.

$$\alpha = \begin{cases} a_{\min}, & \text{만약 } A_c > \overline{P}(A) \\ a_{\max}, & \text{만약 } A_c < \overline{P}(A) \end{cases} \quad (7)$$

이 방법은 비교적 해의 변화가 많이 일어난 온도에서는 온도 강하율,  $\alpha_{\min}$  을 사용하여 온도를 많이 내리고, 그렇지 않고 해의 변화가 적게 일어난 온도에서는 온도 강하율,  $\alpha_{\max}$  를 사용하여 온도를 서서히 내리게 되는 적응적인 방법으로서 이를 적응적 기하스케줄이라 한다.

#### 종료 기준

종료기준은 알고리즘이 끝나는 조건으로서 이론적으로는 온도가 0에 수렴하는 곳에서 끝나야 한다. 하지만 알고리즘이 에너지값에 변화가 없이 낮은 온도에서 보내는 시간이 너무 많기 때문에, 해의 질과 계산시간을 적절히 만족하는 효과적인 종료기준의 결정은 SA 설계에서 중요한 인자 중 하나이다.

본 연구에서 고려된 네 번째 제어인자인 종료기준은 첫 번째 방법으로, 온도가 0에 가까운 상수값인 최종온도  $T_F$  보다 작아질 때 종료하는 방법으로서 최종온도 한계법이라 부르기라고 한다. 두 번째 방법은 Youssef et al.(1991)에 의해 제시된 방법으로서 외부루프가 일정 횟수( $R$ )를 반복한 후에도 에너지 함수값에 변화가 없을 때 알고리즘을 종료하는 방법으로 외부루프 한계법이라 부르기라고 하며, 세 번째 방법은 Johnson et al.(1989)에 의한 방법으로써 내부루프 중 현재해의 변화비율이 설정된 값  $F_{\min}$  보다 작을 때 알고리즘을 종료하는 방법으로 내부루프 한계법이라 부르기라고 한다.

앞서 언급된 제어인자와 그 수준은 다음 <표 1> 과 같이 요약된다.

표 1. 제어인자와수준

제어인자	수준		
	1	2	3
초기온도 (A)	상수 설정법	$kt$ 설정법	수락비율 설정법
내부루프 반복수(B)	상수 설정법	재순환법	지수 평활법
쿨링 스케줄(C)	단순 기하 스케줄	LM 쿨링 스케줄	적응적 기하스케줄
종료기준 (D)	최종온도 한계법	외부루프 한계법	내부루프 한계법

표 2. 신호인자와 수준

제어인자	수준	
	1	2
초기온도 (A)	A <sub>1</sub> ' T <sub>1</sub> =100	T <sub>1</sub> =50
	A <sub>2</sub> ' k=1.0014(p=0.05)	k=0.5164(p=0.003)
	A <sub>3</sub> '	
내부루프 반복수 (B)	B <sub>1</sub> ' L <sub>B</sub> =90	L <sub>B</sub> =50
	B <sub>2</sub> ' L <sub>B</sub> =70	L <sub>B</sub> =40
	B <sub>3</sub> ' L <sub>B</sub> =90	L <sub>B</sub> =50
플링 스케줄 (C)	C <sub>1</sub> ' α=0.98	α=0.90
	C <sub>2</sub> ' T <sub>F</sub> =10	T <sub>F</sub> =1
	C <sub>3</sub> ' α <sub>max</sub> =0.985, 0.95	α <sub>max</sub> =0.97, 0.90
종료기준 (D)	D <sub>1</sub> ' T <sub>F</sub> =10	T <sub>F</sub> =1
	D <sub>2</sub> ' R=5	R=1
	D <sub>3</sub> ' F <sub>min</sub> =0.05	F <sub>min</sub> =0.01

다음으로는 SA 알고리즘의 성능에 대해 의도한 출력을 위한 입력으로 작용하는 인자인 신호

인자를 고려한다. 신호인자로는 각 일반모수들을 결정하기 위한 성능가속화 방법들의 모수값으로써, 각 방법들에 대해 두 가지 수준씩을 고려한다. 신호인자와 수준 및 모수값은 다음 <표 2>에 요약되어 있다.

실험계획은 주효과와 교호작용이 있는 것으로 알려진 교호효과 B×C만을 고려한 제어인자를 배치하기 위해 내측 직교배열표는 L<sub>27</sub>(3<sup>4</sup>)을 사용하며(Johnson et al., 1989), 신호인자를 배치하기 위한 외측 직교배열표는 L<sub>8</sub>(2<sup>7</sup>)을 선정하여 <표 3>과 같이 제어인자와 신호인자를 할당하였다. 실험조건은 총 27×8=216가지로서, 예를 들어 실험조건 (77)은 제어인자(A<sub>2</sub>B<sub>3</sub>C<sub>1</sub>D<sub>2</sub>), 신호인자(A<sub>1</sub>B<sub>2</sub>C<sub>1</sub>D<sub>2</sub>)이다.

최종해의 최적화 정도를 나타내는 특성치로는 수렴된 최종 에너지함수값을 사용하고, 수렴속도를 나타내는 특성치로는 알고리즘 수행 중 검색한 임의해의 총 생성갯수를 특성치로 사용한다.

표 3. 제어인자와 신호인자의 직교배열표 할당

	A	B	C	D	1	1	2	2	1	1	2	2	A'
					1	2	1	2	2	1	2	1	B'
					1	2	1	2	2	1	2	1	C'
					1	2	1	2	2	1	2	1	D'
1	1	1	1	1	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	
2	1	1	2	2	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	
3	1	1	3	3	(17)	(18)	(19)	(20)	(21)	(22)	(23)	(24)	
4	2	2	1	1	(25)	(26)	(27)	(28)	(29)	(30)	(31)	(32)	
5	2	2	2	2	(33)	(34)	(35)	(36)	(37)	(38)	(39)	(40)	
6	2	2	3	3	(41)	(42)	(43)	(44)	(45)	(46)	(47)	(48)	
7	3	3	1	1	(49)	(50)	(51)	(52)	(53)	(54)	(55)	(56)	
8	3	3	2	2	(57)	(58)	(59)	(60)	(61)	(62)	(63)	(64)	
9	3	3	3	3	(65)	(66)	(67)	(68)	(69)	(70)	(71)	(72)	
10	2	3	1	2	(73)	(74)	(75)	(76)	(77)	(78)	(79)	(80)	
11	2	3	2	3	(81)	(82)	(83)	(84)	(85)	(86)	(87)	(88)	
12	2	3	3	1	(89)	(90)	(91)	(92)	(93)	(94)	(95)	(96)	
13	3	1	1	2	(97)	(98)	(99)	(100)	(101)	(102)	(103)	(104)	
14	3	1	2	3	(105)	(106)	(107)	(108)	(109)	(110)	(111)	(112)	
15	3	1	3	1	(113)	(114)	(115)	(116)	(117)	(118)	(119)	(120)	
16	1	2	1	2	(121)	(122)	(123)	(124)	(125)	(126)	(127)	(128)	
17	1	2	2	3	(129)	(130)	(131)	(132)	(133)	(134)	(135)	(136)	
18	1	2	3	1	(137)	(138)	(139)	(140)	(141)	(142)	(143)	(144)	
19	3	2	1	3	(145)	(146)	(147)	(148)	(149)	(150)	(151)	(152)	
20	3	2	2	1	(153)	(154)	(155)	(156)	(157)	(158)	(159)	(160)	
21	3	2	3	2	(161)	(162)	(163)	(164)	(165)	(166)	(167)	(168)	
22	1	3	1	3	(169)	(170)	(171)	(172)	(173)	(174)	(175)	(176)	
23	1	3	2	1	(177)	(178)	(179)	(180)	(181)	(182)	(183)	(184)	
24	1	3	3	2	(185)	(186)	(187)	(188)	(189)	(190)	(191)	(192)	
25	2	1	1	3	(193)	(194)	(195)	(196)	(197)	(198)	(199)	(200)	
26	2	1	2	1	(201)	(202)	(203)	(204)	(205)	(206)	(207)	(208)	
27	2	1	3	2	(209)	(210)	(211)	(212)	(213)	(214)	(215)	(216)	

### 3.2 실험 결과 분석 및 최적 알고리즘 결정

최적설계를 위한 실험은 문제의 크기에 따른 세 가지 문제군 중에서 제품 모델에 대한 수요량의 분포가 비교적 일반적인 형태를 갖는 문제(문제 II군 중 문제 6)를 선택하여 사용하였다(부록 참조). 실험은 인자들의 각 실험조건에 따라 10회 반복 실시하며, 성능 특성치인 에너지함수값 및 총 검색해의 수는 모두 작을수록 좋은 특성을 가지므로 각각 망소특성으로 분석하였다. 실험의 분석은 통계분석 S/W인 MINITAB ver13.30을 사용하였다.

실험결과는 <표 4>에 나타나 있듯이 최적화 정도를 나타내는 특성치인 에너지함수값에 대한 S/N비와 총 검색해의 수에 대한 S/N비로 나타내었으며, 이때 S/N비 값인 SN<sub>i</sub>는 다음 식에 의해 계산된다.

$$SN_i = -10 \log \left[ \frac{1}{m} \sum_{j=1}^m y_{ij}^2 \right] \quad (8)$$

여기서,  $y_{ij}$ 는 제어인자의  $i$  번째 실험조건과 신호인자의  $j$  번째 실험조건에서의 10회 반복에 대한 평균 성능 특성치이다. 성능 특성치  $y_{ij}$ 의 구체적인 값은 지면관계상 생략하였다.

최적화 정도 및 계산 성능이 동시에 고려된 최적의 조합으로 설정된 알고리즘을 RD-SA (robust designed SA)라고 한다.

### 4. 수치 실험

본 연구에서 제시되었던 최적 설계된 SA 알고리즘 RD-SA에 대한 타당성 평가를 위해 McMullen and Frazier(2000)에 의해 제시되었던 SA 알고리즘과의 비교실험을 실시한다. 편의상 McMullen and Frazier(2000)에 의해 제시되었던 SA 알고리즘을 MF-SA로 표시한다. 비교실험을 위해 사용된 문제는 McMullen and Frazier (2000)에 의해 제시되었던 문제이며, 문제는 세 가지 문제 군으로써, 비교적 작은 크기인 문제 I군(총 수요량=20)과 중간 크기인 문제 II군(총 수요량=100) 그리고 가장 큰 크기인 문제 III군(총 수요량=500)으로 구성된다.

SA 알고리즘의 실험인자에 대한 설정값은 <표 7>에 SA의 일반 모수인 초기 온도, 내부루프 반복수, 온도 강하율 및 종료조건에 대해서, RD-SA와 MF-SA별로 각각 나타내었다. 단 MF-SA에 대한 모수 설정값은 McMullen and Frazier (2000)에 의해 사용되었던 값을 그대로 사용한다.

SA 알고리즘은 프로그램 언어 C++를 이용하

여 코딩되었고, 펜티엄 III 866 MHz IBM-PC 호환 기종에서 실행되었다. 실험은 문제 I군, II군 및 III군의 모든 문제들에 대해서 각각 시행되었으며, 각 문제에 따라 RD-SA와 MF-SA로 각기 10회씩 반복 실험하였다. 수행된 결과는 각 문제군에 따라 최적화 정도를 나타내는 척도인 에너지함수값과 수렴속도를 나타내는 척도인 총 검색해의 수에 대해서 평균 및 표준편차로 나타내었다.

### 5. 결론

일반적으로 전형적인 SA 알고리즘은 질 높은 해를 생성하여 주지만, 비교적 긴 계산시간을 요구한다. 이렇듯 상충적 관계에 있는 해의 최적성 정도와 수렴 속도를 동시에 만족시키기 위해, 본 연구에서는 최소의 실험으로 최적의 수준조합을 찾을 수 있는 효과적 방법인 다구치의 강건 설계기법을 사용하여 최적의 SA 알고리즘 설계방법을 제시하였다. 유효성 검증용을 위하여 혼합모델 조립라인의 투입순서 결정문제에 최적 설계된 SA 알고리즘을 적용하였으며, 동일한 표본 문제군을 다루는 McMullen and Frazier(2000) 알고리즘과의 비교실험 결과 해의 최적성 및 수렴 속도의 측면에서 더 우수한 결과를 산출하여 준다는 것을 확인할 수 있었다.

### 참고문헌

- Aarts, E.H. L. and Van Laarhoven (1987), R. J. M., *Simulated Annealing: Theory and Applications*, Reidel, Dordrecht.
- Ali, M. M., Torn, A. and Viitanen, S. (2002), A direct search variant of the simulated annealing algorithm for optimization involving continuous variables, *Computers & Operations Research*, **29**, 87-102.
- Bard, J. F., Shtub, A. and Joshi, S. B. (1994), Sequencing mixed-model assembly lines to level parts usage and minimize line length, *International Journal of Production Research*, **32**, 2431-2454.
- Bolat, A., Savsar, M. and Al-Fawzan, M. A. (1994), Algorithms for real-time scheduling of jobs on mixed model assembly lines, *Computers and Operations Research*, **21**, 487-498.
- Celano, G., Fichera, S., Grasso, V., La Commare, U. and Perrone, G. (1999), An evolutionary approach to multi-objective scheduling of mixed model assembly lines, *Computers and Industrial*

- Engineering*, 37, 69-73.
- Cerny, V. (1985), Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Applications*, 45, 41-51.
- Dar-El, E. M. (1978), Mixed model assembly line sequencing problem, *OMEGA*, 6, 313-323.
- Huang, M. D., Romeo, F. and Sangiovanni-Vincentelli, A. L. (1986), An efficient general cooling schedule for simulated annealing, *Proceedings of IEEE International Conference on Computer-Aided Design*, Santa Clara, 381-384, November.
- Johnson, D. S., Aragon, C. R., McGeoch, L. A. and Schevon, C. (1989), Optimization by simulated annealing: An experimental evaluation; Part 1, Graph partitioning, *Operations Research*, 37, 868-892.
- Kim, H.G., Paik, C.H. and Cho, H.S. (2002), A robust design of simulated annealing approach: mixed-model sequencing problem, *IE Interfaces*, 15, 189-198.
- Kim, Y. K., Hyun, C. J. and Kim, Y. H. (1996), Sequencing in mixed-model assembly lines: a genetic algorithms approach, *Computers and Operations Research*, 23, 1131-1145.
- Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983), Optimization by simulated annealing, *Science*, 220, 671-679.
- Lundy, M. and Mees, A. (1986), Convergence of an annealing algorithm, *Mathematical Programming*, 34, 111-124.
- McMullen, P. R. (1998), JIT sequencing for mixed-model assembly lines with setups using Tabu search, *Production Planning and Control*, 9, 504-510.
- McMullen, P. R. and Frazier, G. V. (2000), A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line, *IIE Transactions*, 32, 679-686.
- Miltenburg, J. (1989), Level schedules for mixed-model assembly lines in just-in-time production systems, *Management Science*, 35, 192-207.
- Park, M. W. and Kim, Y. D. (1998), A systematic procedure for setting parameters in simulated annealing algorithms, *Computers and Operations Research*, 25, 207-217.
- Su, C. T. and Hsieh, K. L. (1998), Applying neural network approach to achieve robust design for dynamic quality characteristics, *International Journal of Quality & Reliability Management*, 15, 509-519.
- Tamura, T., Long, H. and Ohno, K. (1999), A sequencing problem to level part usage rates and work loads for a mixed-model assembly line with a bypass subline, *International Journal of Production Economics*, 61, 557-564.
- Youssef, G. S. and Vasant, B. R. (1991), Combinational optimization by stochastic evolution, *IEEE Transactions on Computer-Aided Design*, 10, 525- 535.

부록 (분석시 사용된 문제)

문제 I 군. 제품 모델에 대한 수요량(d)

문제 번호	제품 모델 i									
	1	2	3	4	5	6	7	8	9	10
1	11	1	1	1	1	1	1	1	1	1
2	10	2	1	1	1	1	1	1	1	1
3	9	3	1	1	1	1	1	1	1	1
4	8	4	1	1	1	1	1	1	1	1
5	7	5	1	1	1	1	1	1	1	1
6	6	5	2	1	1	1	1	1	1	1
7	5	5	3	1	1	1	1	1	1	1
8	4	4	4	2	1	1	1	1	1	1
9	2	2	2	2	2	2	2	2	2	2

**문제 II군. 제품 모델에 대한 수요량( $d_i$ )**

문제 번호	제품 모델 $i$														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	40	40	8	1	1	1	1	1	1	1	1	1	1	1	1
2	35	35	10	5	5	1	1	1	1	1	1	1	1	1	1
3	30	30	15	10	5	1	1	1	1	1	1	1	1	1	1
4	25	25	20	15	5	1	1	1	1	1	1	1	1	1	1
5	20	20	20	15	15	1	1	1	1	1	1	1	1	1	1
6	20	20	15	15	10	6	6	1	1	1	1	1	1	1	1
7	15	15	15	10	10	10	10	5	4	1	1	1	1	1	1
8	15	15	10	10	10	10	10	10	4	1	1	1	1	1	1
9	7	7	7	7	7	7	7	7	7	7	6	6	6	6	6

**문제 III군. 제품 모델에 대한 수요량( $d_i$ )**

문제 번호	제품 모델 $i$																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	105	105	105	105	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
2	35	35	35	35	35	35	35	35	35	35	15	15	15	15	15	15	15	15	15	15
3	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25