이산설계공간에서 직교배열표를 이용한 순차적 알고리듬의 국부해

이정욱† · 박경진*

Local Solution of Sequential Algorithm Using Orthogonal Arrays in Discrete Design Space

Jeong-Wook Yi, Gyung-Jin Park

Key Words : Discrete Design Space(이산설계공간), Orthogonal Arrays(직교배열표), Design of Experiments(실험계획법), Local Solution(국부해), Structural Optimization(구조최적설계)

Abstract

The structural optimization has been carried out in the continuous design space or in the discrete design space. Generally, available designs are discrete in design practice. But methods for discrete variables are extremely expensive in computational cost. In order to overcome this weakness, an iterative optimization algorithm was proposed for design in the discrete space, which is called as a sequential algorithm using orthogonal arrays (SOA). We focus to verify the fact that the local solution can be obtained throughout the optimization with this algorithm. The local solution is defined in discrete design space. Then the search space, which is the set of candidate values of each design variables formed by the neighborhood of current design point, is defined. It is verified that a local solution can be founded by moving sequentially the search space. The SOA algorithm has been applied to problems such as truss type structures. Then it is confirmed that a local solution can be obtained using the SOA algorithm

1. 서 론

최적설계는 설계의 주요 성능 인자들과 요구사항을 목적함수와 구속조건으로 정식화하여, 구속조건을 만족하면서 목적함수를 최소화 하는 자동화된 설계기법이다[1-3]. 특히 구조설계 분야에서는 유한요소법의 발달과 컴퓨터의 성능향상으로 최적설계의 적용이 활발히 진행되고 있다[4,5].

최적설계는 설계변수의 종류에 따라 연속설계변수와 이산설계변수로 분류할 수 있다. 연속설계변수를 갖는 문제의 최적화에 대해서는 함수의 구배(gradient)를 이용한 많은 연구가 이루어져 효율적인 알고리즘이 개발되어 있으나, 이산설계변수를 갖는 문제는 연속설계변수를 갖는 문제의 최적화 방법을 그대로 적용할 수 없다[6-8]. 그러나 실제 구조물의 최적설계에서는 설계변수가 어떤 특정한 값들 중에서 선택해야 하거나, 이미 규격화되어 있는 부품의 치수를 취급해야 하는 경우가 많다. 그러므로 이산설계변수를 갖는 문제를 처리하기 위한 방법이 필요하다.

이산설계 최적화를 위한 여러가지 알고리즘이 제안된 바 있다. Land와 Doig[9]는 선형계획 영역에서 고전적인 분단탐색법(branch and bound) 방법을 개발하면서 이산최적화의 분기점을 마련하였고, Salajegheh 등[10]은 분단탐색법을 정수계획법(integer programming)을 위해 전개하였다. Kirkpatrick 등[11]은 조합적 최적화문제의 한 해법으로 시뮬레이티드 어닐링(simulated annealing)을

† 한양대학교 최적설계신기술연구센터
E-mail : yijwook@ihanyang.ac.kr
* 한양대학교 기계정보경영학부
E-mail : gjpark@hanyang.ac.kr
TEL : (031)400-5246 FAX : (031)407-0755

제안하였다. 이는 Gidas[12], Mitra[13]에 의하여 전역 최소점에로의 수렴성이 이론적으로 증명되었고, 기본개념의 단순성과 범용성이 두드러져 특별한 대안이 없을 때 편리하게 사용할 수 있다. 그러나 실제 문제에 적용 시, 초기치 설정, 변동량, 수락기준 등 많은 문제점들을 가지고 있다. 유전알고리듬(genetic algorithm)은 유전학과 자연진화를 흉내낸 적응탐색법으로 Holland[14]가 처음 소개하였다. Michalewicz[15]는 특수 연산자, 부적합한 해를 복구해주는 복구 알고리듬에 대한 연구를 수행하였고, Goldberg[16]는 해의 정밀도 개선과 구속조건을 쉽게 다룰 수 있도록 실수코딩 염색체를 사용하는 방법을 제안하였다. 또한 Kido 등[17]은 타부탐색법(taboo search), 시뮬레이티드 어닐링의 다른 탐색 알고리듬과의 결합을 시도하였다. 이들 알고리듬은 최근까지 많은 연구가 이루어져 효율성이 향상되었으나, 아직 함수계산 횟수가 많다는 단점이 있다.

실험계획법을 이용한 최적화 방법에 관한 연구도 이루어지고 있다. 이권희 등[18]은 연속설계공간에서 구조물의 최적화를 수행한 후, 후처리 단계에서 다구찌법(Taguchi method)에 의하여 이산설계공간에서 최적점을 구하는 방법을 제안하였다. Rao[19]는 목적함수를 다구찌법의 특성함수로 정의하고 설계변수의 영역을 줄여가면서 반복적으로 최적의 해를 구하는 알고리듬을 제안하였다.

최근 이정욱 등[20]은 직교배열표에 의한 행렬실험을 반복적으로 수행하여, 함수계산 횟수를 효율적으로 줄이면서 최적의 해를 찾는 알고리듬을 제안하였다. 그러나 제안한 알고리즘을 통하여 구한 해가 국부해를 찾을 수 있는가에 대한 의문을 안고 있다. 그러므로 이산설계공간에서의 국부해를 정의하고, 순차적 알고리듬이 국부해를 찾을 수 있음을 증명하였다. 또한 직교배열표를 이용한 순차적 알고리듬이 근사적으로 국부해를 구할 수 있도록 수정된 알고리듬을 제안하였다. 이를 통하여 이산설계공간에서 직교배열표를 이용한 순차적인 접근방법이 국부해를 찾을 수 있음에 타당성을 부여하고자 한다.

2. 이산설계공간에서의 국부해

2.1. 전역 최적화

최적설계는 부과된 조건을 만족하면서 설계자가 원하는 특성치를 극대화시키는 설계변수를 결정하는 방법이다. 설계자가 원하는 특성치를 목적함수

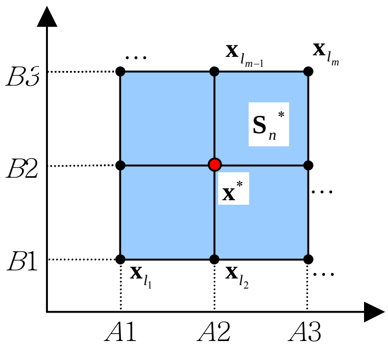

Fig. 1 Global and local solutions in the discrete design space

$f(\mathbf{x})$, 부과된 조건을 등제한조건 $h(\mathbf{x})$와, 부등제한조건 $g(\mathbf{x})$로 정의할 때 이산설계공간에서의 최적설계를 위한 정식화는 식 (1)과 같이 나타낼 수 있다.

$$
\begin{array}{lll}
\textbf{Find} & \mathbf{x} & \\
\textbf{to minimize} & f(\mathbf{x}) & \\
\textbf{subject to} & h_i(\mathbf{x})=0, & i=1,\cdots,n \\
 & g_j(\mathbf{x})\le 0, & j=1,\cdots,m \\
 & \mathbf{x}\in\mathbf{S} &
\end{array}
\tag{1}
$$

여기서 $\mathbf{x}$ 는 이산설계변수이고, $\mathbf{S}$ 는 설계변수가 존재할 수 있는 공간으로 이산설계공간(discrete design space)을 의미한다.

만약 $\mathbf{S}$ 에서 정의된 어떤 함수 $f(\mathbf{x})$가 $\forall\mathbf{x}^*\in\mathbf{S}$ 에 대해 $f(\mathbf{x}^*)\le f(\mathbf{x})$를 만족한다면, 한 점 $\mathbf{x}^*\in\mathbf{S}$ 는 전역해(global solution)이다. 정의대로 전역해는 이산설계공간에서 주어진 설계변수에 대하여 모든 조합에 대한 함수 값의 계산을 필요로 한다. 설계변수가 증가함에 따라서 함수 값의 계산은 기하급수적으로 증가하기 때문에, 전조합 계산을 통하여 최적의 해를 구하는 방법은 유용하지 않다.

2.2. 국부 최적화

이산설계공간 $\mathbf{x}_l^*\in\mathbf{S}$ 에서 정의된 설계변수 $\mathbf{x}_l^*$ 주변의 이산설계공간 $\mathbf{S}_l^*$ 이 있다. 여기서 $\mathbf{S}_l^*\in[\,\mathbf{x}_{l_1}\ \ \mathbf{x}_{l_2}\ \ \mathbf{x}_{l_3}\ \ \ldots\ \ \mathbf{x}_{l_{m-1}}\ \ \mathbf{x}_{l_m}\ \ \mathbf{x}_l^*\,]$, $m=3^n-1$이고 n은 설계변수의 개수이다. Fig. 1은 설계변수 A와 B의 2개를 갖는 이산설계공간 $\mathbf{S}_l^*$를 표현하고 있다. 만일 $\forall j$ 에 대하여 $f(\mathbf{x}_l^*)\le f(\mathbf{x}_{l_j})$ 이라면, $\mathbf{x}_l^*$는 국부해(local solution)이다.

Fig.2를 통하여 앞 절에서 정의한 전역해와 국부해를 살펴보면, 점 B는 전역해임 알 수 있다.

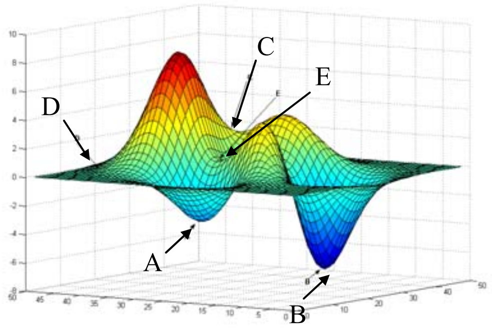

Fig. 2 Global and local solutions in the discrete design space

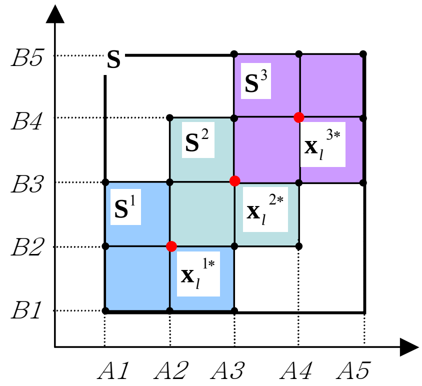

Fig. 3 Search space with two design variables in the discrete space

그러나 전역해를 구하는 것은 비용이 너무 많이 들기 때문에 국부해를 찾는 것이 하나의 방법일 수 있다. 국부 최적화는 설계변수의 주어진 영역에서 국부지역을 선택하여 목적함수를 최소화하는 해를 찾고, 다시 해의 지역을 중심으로 최소화하는 해를 찾는 반복적인 과정을 통하여 구하는 방법이다. 이 과정을 통하여 구한 최적해는 초기치의 위치에 따라서 점 B일 수 있지만, 점 A를 구할 수도 있다. 또한 경우에 따라서는 점 C, D, E 등을 최적해로 구할 수 있다. 그러나 초기의 특성치에 비하여 성능을 향상시킬 수 있는 해를 구한다면 공학적인 문제에서 유용하게 쓰일 수 있다.

3. 직교배열표를 이용한 순차적 알고리듬에 의한 국부해

3.1. 순차적 알고리듬에 의한 국부해

이산설계공간 $\mathbf{S}$에서 현재의 설계변수를 중심으로 하고, 주변의 설계변수를 배치한 탐색공간 (search space) $\mathbf{S}^j$를 정의한다. 설계변수가 2개인 경우, 이산설계변수의 현재 값을 2수준에 배치하고 2수준보다 크거나 작은 주변의 설계변수를 각각 1수준과 3수준에 배치하여 Fig.3과 같이 탐색공간을 나타낼 수 있다.

여기서 A와 B는 설계변수이고 Ai와 Bi (i = 1,2,3,4,5)의 전조합으로 이루어지는 공간은 이산설계공간 $\mathbf{S}$이다. 또한 $\mathbf{S}^j$와 $\mathbf{x}_l^{\,j}$는 j 번째 탐색공간과 탐색공간에서의 설계변수 벡터이다. j 번째 탐색공간에서 $\mathbf{x}_l^{\,j}$를 중심으로 주변의 설계값은 전조합에 대한 후보값 중에서 선택할 수 있다.

현재의 탐색공간에서 목적함수가 최소인 최소해를 구하고, 최소해를 중심으로 탐색공간을 재정의하여 이동하는 순차적 반복과정을 생각할 수 있다. 2.2절에서 정의한 국부해의 정의에 따르면, j 번째 탐색공간에서 전조합에 대한 함수값을 비교하여 가장 작은 최소해인 $\mathbf{x}_l^{\,j*}$를 찾을 수 있다. 이때 $\mathbf{x}_l^{\,j*}$는 국부해이다. 만일 j 번째 탐색공간에서 $\mathbf{x}_l^{\,j*}$가 국부해가 아니라면, $\mathbf{x}_l^{\,j+1*} \neq \mathbf{x}_l^{\,j*}$이므로 탐색공간을 이동하고, 이동한 공간에서 $f\left(x_l^{\,j+1*}\right) \leq f\left(x_l^{\,j*}\right)$인 최소해를 찾을 수 있다. 그러나 $\mathbf{x}_l^{\,j*}$가 국부해라면, 다음 반복과정에서도 같은 탐색공간으로 이동하므로 반복과정을 종료할 수 있다. 그러므로 탐색공간을 순차적으로 이동함으로써 국부해를 구할 수 있다.

3.2. 직교배열표를 이용한 순차적 알고리듬

앞 절에서는 탐색공간을 순차적으로 이동함으로써 국부해를 구할 수 있음을 알 수 있었다. 이 절에서는 이산설계공간에서 직교배열표를 이용하여 효율적으로 최적화 과정을 수행하는 알고리듬을 간략히 소개하겠다. 이는 직교배열표를 이용한 순차적 알고리듬(sequential algorithm using orthogonal arrays; SOA)으로 이전에 제안된 바 있다 [20]. 제안된 알고리듬은 함수계산 횟수를 효율적으로 줄이기 위하여 직교배열표를 이용한다. 이는 전조합에 대한 함수계산을 한 것과 같은 효과를 누릴 수 있었다. 또한 국부해를 찾기 위하여 제안된 알고리듬을 수정하였다. Fig.4에는 제한조건이 있는 문제에 대한 순차적 알고리듬의 흐름을 보이고 있다.

- 1 단계: 문제설정

문제의 특성치(목적함수)와 인자(설계변수)를 설정한다. 각 인자의 후보값은 이산설계공간에서 설계변수의 경계(boundary)를 벗어나지 않는 가능한 영역 내에 있는 이산 값들을 설정한다.

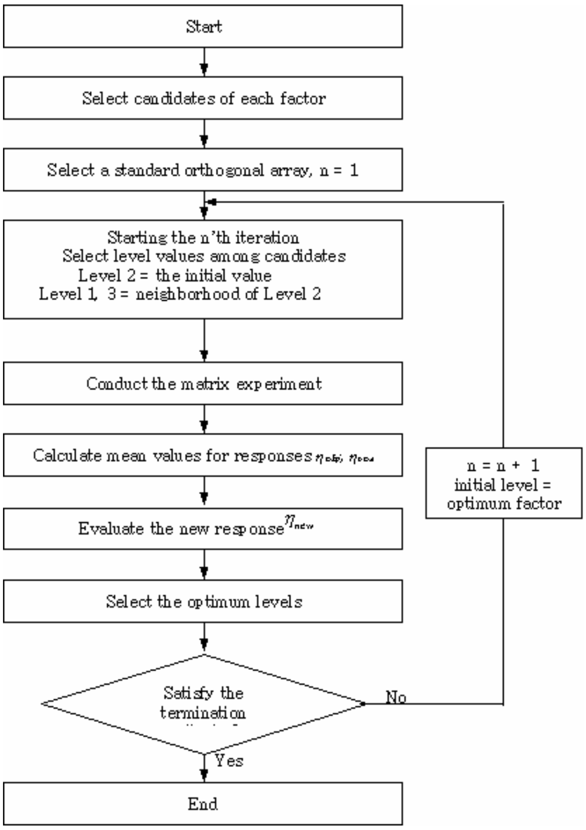

Fig. 4 Flow of discrete design for constrained problems

● 2단계: 직교배열표 선택 및 초기치 설정

행렬실험에서 사용할 3수준 계의 직교배열표를 선택한다. 주의할 점은 국부해를 찾을 수 있도록 표준직교배열표의 2수준을 1수준으로 변환한 수정된 직교배열표를 이용한다. 이는 최적수준을 검색할 때, 모든 설계변수에 대한 2수준 값을 포함시키기 위함이다. 다음으로 인자의 수준값들을 배치한다. 수준값들은 인자의 후보값들 중에서 선택하여야 하며, 초기치는 2수준에 배치하고 1 과 3 수준에는 초기치의 전·후에 위치한 후보값을 배치한다. 한번의 반복과정이 끝나고 탐색공간에서의 최적해를 선정하면, 각 인자에 대하여 최적조건의 후보값을 2수준에 배치하고 인접한 후보값들을 1 과 3 수준에 배치한다.

● 3단계: 행렬실험을 실시

직교배열표의 각 행마다 배치된 인자의 수준에 따른 반응치를 구한다. 제한조건이 있는 문제에서는 목적함수에 대한 반응치 η_{obj} 와 각 제한조건에 대한 반응치 η_{cons} 를 구할 수 있다.

● 4단계: 현 탐색공간에서의 최적조건 선정

행렬실험을 통해서 나온 결과를 바탕으로 전조합에 대한 목적함수와 제한조건의 추정치 $\hat{\eta}_{obj}$, $\hat{\eta}_{cons}$ 를 구한다. 제한조건이 있는 문제를 해결하기 위해서 벌칙계수를 포함하는 벌칙함수(penalty function)를 도입하였다. 벌칙함수는 식 (2)에 의하여 구한다.

$$\hat{P} = s \times \sum_{i=1}^{n} \max\left[0, \hat{v}_i\right] \tag{2}$$

여기서 $\hat{P}$ 는 추정치의 벌칙함수, $\hat{v}_i$ 는 i 번째 제한조건의 최대 위배량(maximum violation)의 추정치이고 s 는 조절계수(scale factor)이다.

전조합에 대한 새로운 반응치는 식 (3)과 같이 벌칙함수를 포함하며 각 인자의 수준에 대한 추정치이다.

$$\hat{\eta}_{new} = \hat{\eta}_{obj} + \hat{P} \tag{3}$$

이들을 올림차순으로 정렬하고, 확인실험을 통하여 실제로 제한조건을 만족하는 최소의 추정치에 대한 최적수준을 구한다. 또한 최적수준에서 확인실험 후의 새로운 반응치와 직교배열표 상의 행렬실험에 따른 반응치를 비교하여 가장 최적인 수준조합을 선택한다.

● 5단계: 종료조건

종료조건은 (1)현재 설계변수의 탐색공간과 최적수준을 중심으로 한 설계변수의 탐색공간이 같은 경우 (2)제한조건을 위배하는 해가 연속적으로 구해지는 반복횟수로 한다. 종료조건을 만족시키지 않는다면 2단계로 돌아가서 같은 과정을 반복적으로 수행한다. 종료조건에서 (1)의 경우는 앞에서 정의한 국부해를 찾은 경우이다. (2)의 경우는 모든 행렬실험이 제한조건을 위배하는 경우로 비가용 영역(infeasible region)으로 빠져든 경우로 강제적으로 반복과정을 중지시키고 지금까지 구한 해 중에서 제한조건을 만족하는 최적해를 선택한다.

다음 장에서는 알고리듬을 적용하여 예제를 풀고 국부해를 찾을 수 있음을 검증한다.

4. 예제 적용 및 고찰

4.1. 3 부재 트러스

Fig. 5와 같이 2개의 하중조건이 3부재 트러스 구조물에 작용할 때, 단면적 A_1, A_2, A_3를 결정하는 문제이다. 물성치는 탄성계수(E) 68.9GPa, 밀도(ρ) 2770kg/m^{3}을 사용하였다. 문제의 정식화는 식 (4)

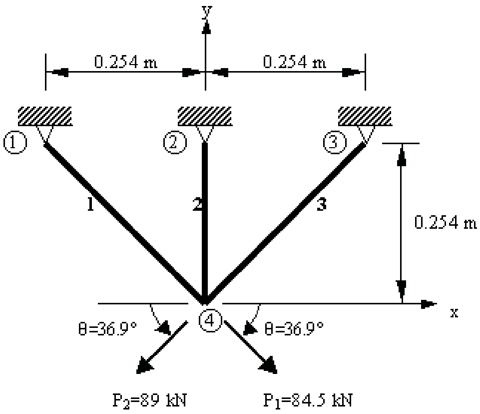

Fig. 5 Three bar truss

Table 1 Results of example 1 with SOA

Design variables	Initial values ($\times 10^{-5}$m^2)	Optimum levels ($\times 10^{-5}$m^2)
A_1	57	46
A_2	57	44
A_3	57	49
Mass (kg)	1.54	1.25
No. of Iteration	·	6
No. of fn. evaluations	·	61

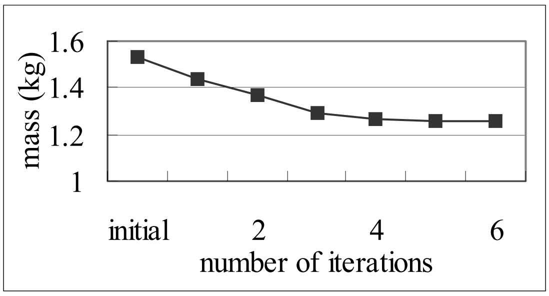

Fig. 6 History of objective function for example 1

와 같다.

$$
\begin{aligned}
&\textbf{Find} \quad A_i \\
&\textbf{to minimize} \quad \text{mass} \qquad (4) \\
&\textbf{subject to} \quad -103.4\text{MPa} \le \sigma_i \le 137.9\text{MPa} \quad (i=1,2,3)
\end{aligned}
$$

candidate values

A_1, A_2, A_3 ($\times 10^{-5}$ m^2) $\in$ {10, 13, 15, 18, 21, 23, 26, 28, 31, 34, 36, 39, 41, 44, 46, 49, 52, 54, 57, 59}

여기서 σ_i 는 각 부재에 작용하는 응력을 의미한다. 설계변수가 가질 수 있는 후보값들은 임의로 20개의 이산값으로 선정하였다. 설계변수의 수가 3개이고, 수준 수는 3인 점을 감안하여 $L_9(3^4)$ 표준 직교배열표를 선택하여 행렬실험을 실시하고 반복과정을 실시하였다. 순차적 알고리듬을 적용하여 최적화를 수행한 결과는 Table 1과 같고, 6회의 반복과정을 중 61회의 함수계산 후에 종료하였다. 종료 시 종료조건은 현재 설계변수의 탐색공간과 최적수준을 중심으로 한 설계변수의 탐색공간이 같은 경우였다. 즉 국부해를 구하였음을 알 수 있었다. Fig. 6에서는 반복과정에 따른 목적함수의 변화를 보이고 있다. 순차적으로 과정을 통하여 초기의 질량이 서서히 감소해 나아감을 알 수 있다.

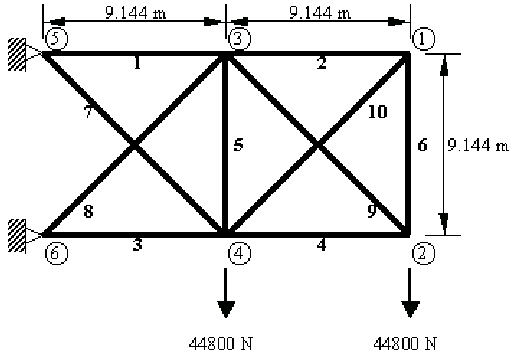

Fig. 7 Ten bar truss

Table 2 Results of example 2 with SOA

Design variables	Initial values	Optimum levels ($\times 10^{-5}$m^2)
A_1	729	542
A_2	729	194
A_3	729	542
A_4	465	323
A_5	465	97
A_6	465	97
A_7	465	400
A_8	465	400
A_9	465	323
A_{10}	465	97
Mass (kg)	1551.24	879.03
No. of Iteration	·	6
No. of fn. evaluations	·	169

4.2. 10 부재 트러스

Fig. 7와 같이 2개의 하중이 10부재 트러스 구조물에 작용할 때 각 부재의 단면적 A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10}을 결정하는 문제이다. 물성치는 탄성계수(E) 68.9GPa, 밀도(ρ) 2770kg/m^{3}을 사용하였다. 문제의 정식화는 식 (5)와 같다.

$$
\begin{aligned}
&\textbf{Find} \quad A_i \\
&\textbf{to minimize} \quad \text{mass} \\
&\textbf{subject to} \quad -172.4\text{MPa} \le \sigma_i \le 172.4\text{MPa} \qquad (5)
\end{aligned}
$$

candidate values

A_i ($\times 10^{-5}$ m^2) $\in$ {6, 97, 194, 323, 400, 465, 542, 645, 729, 813} $(i = 1,2,\cdots,10)$

여기서 σ_i 는 각 부재에 작용하는 응력을 의미한

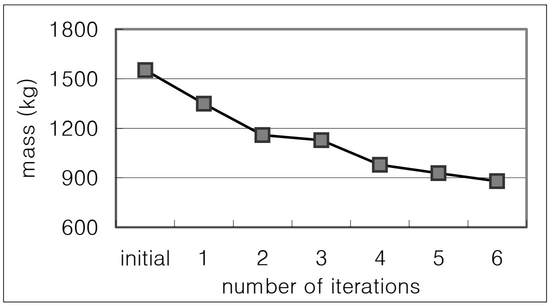

Fig. 8 History of objective function for example 2

다. 설계변수가 가질 수 있는 후보값들은 임의로 10개의 이산값으로 선정하였다. 설계변수의 수가 10개이고, 수준 수는 3인 점을 감안하여 $L_{27}(3^{13})$ 표준 직교배열표를 선택하여 행렬실험을 실시하고 반복과정을 실시하였다. 순차적 알고리듬을 적용하여 최적화를 수행한 결과는 Table 2와 같고, 6회의 반복과정을 중 169회의 함수계산 후에 종료하였다. 종료 시 종료조건은 탐색공간이 같은 경우로 국부해를 구하였음을 알 수 있다. Fig. 8에서는 반복과정에 따른 목적함수의 변화를 보이고 있다.

5. 결론

이산설계공간에서 직교배열표를 이용한 순차적 알고리듬(SOA)에 대한 국부해의 존재를 증명하였다. 이산설계공간에서 전역해와 국부해를 정의하고 국부해를 찾기 위하여 순차적 알고리듬이 수정하였고, 알고리듬이 해를 탐색하는 과정을 설명하였다. 또한 예제를 통하여 국부해를 찾을 수 있음을 검증하였다. 이를 통하여 이산공간에서의 최적화 알고리듬으로 순차적 알고리듬을 이용하여 함수계산 횟수를 줄이면서 국부해를 찾을 수 있어, 실제 현장에서 설계 시 유용하게 이용할 수 있다.

참고문헌

(1) Arora, J.S., 1989, *Introduction to Optimum Design*, McGraw-Hill, New York.

(2) Haug, E.J. and Arora, J.S., 1979, *Applied Optimal Design*, John Wiley & Sons, Inc., New York.

(3) Vanderplaats, G.N., 1984, *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill, New York.

(4) Haftka, R.T., Gurdal, Z. and Kamat, M., 1990, *Elements of Structural Optimization*, Kluwer Academic Publishers, Dordrecht.

(5) Kirsch, U., 1981, *Optimum Structural Design*, McGraw-Hill, New York.

(6) Arora, J.S. and Huang, M.W., 1994, “Methods for Optimization of Nonlinear Problems with Discrete Variables: a review,” *Structural Optimization*, Vol.8, pp.69-85.

(7) Huang, M.W. and Arora, J.S., 1997, “Optimal Design with Discrete Variables: Some Numerical Experiments,” *International Journal for Numerical Methods Engineering*, Vol.40, pp.165-188.

(8) Thanedar, P.B. and Vanderplaats, G.N., 1994, “A Survey of Discrete Variable Optimization for Structural Design,” *Journal of Structural Engineering, ASCE*, Vol.121, pp.301-306.

(9) Land, A.H. and Doig, A., 1960, “An Automatic Method of Solving Discrete Programming Problems,” *Econometrica*, Vol.28, No.4, pp.297-520.

(10) Salajegheh, E. and Vanderplaats, G.N., 1993, “Optimum Design of Structures with Discrete Sizing and Shape Variables,” *Structural Optimization*, Vol.6, No.2, pp.79-85.

(11) Kirkpatrick, S., Gelatt, C.D. and M.P. Vecchi, 1983, “Optimization by Simulated Annealing,” *Science*, Vol.220, pp.671-680.

(12) Gidas, B., 1985, “Non-stationary Markov Chains and Convergence of the Annealing Algorithms,” *Journal of Statistical Physics*, Vol.39, pp.73-131.

(13) Mitra, D., Romeo, F. and Sangiovanni-Vincentelli, A.L., 1986, “Convergence and Finite-time Behavior of Simulated Annealing,” *Advances in applied Probability*, Vol.18, pp.747-771.

(14) Holland, J.H., 1975, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Michigan.

(15) Michalewicz, Z. and Janikow, C.Z., 1991, “Handling Constraints in Genetic Algorithms,” *Proceeding 4th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, CA., pp.151-157.

(16) Goldberg, D.E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York.

(17) Kido, T., Kitano, H. and Nakanishi, M., 1993, “A Hybrid Search for Genetic Algorithms: Combining Genetic Algorithms, Tabu Search, and Simulated Annealing,” *Proceeding 5th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, CA.

(18) Lee, K.H., Eom, I.S., Park, G.J. and Lee, W.I., 1996, “Robust Design for Unconstrained Optimization Problems Using the Taguchi Method,” *AIAA*, Vol.34, No.5, pp.1059-1063.

(19) Ku, K.J., Rao, S.S. and Chen, L., 1998, “Taguchi-Aided Search Method for Design Optimization of Engineering Systems,” *Engineering Optimization*, Vol.30, pp.1-23.

(20) Yi, J.W., Park, J.S., Lee, K.H. and Park, G.J., 2001, “Development of an Optimization Algorithm Using Orthogonal Arrays in Discrete Design Space,” *Transactions of the KSME A*, Vol.25, No.10, pp.1621-1626.