대한기계학회논문집 A 권, 제 29 권 제 10 호, pp. 1369~1376, 2005

〈 논문 〉

이산공간에서 순차적 알고리듬(SOA)을 이용한 전역최적화

조범상† · 이정욱* · 박경진**
(2005 년 4 월 27 일 접수, 2005 년 9 월 6 일 심사완료)

Global Optimization Using a Sequential Algorithm with Orthogonal Arrays in Discrete Space

Bum-Sang Cho, Jeong-Wook Yi and Gyung-Jin Park

Key Words : Discrete Design(이산설계), Sequential Algorithm(순차적 알고리듬), Genetic Algorithm(유전알고리듬), Orthogonal Arrays(직교배열표), Global Solution(전역해)

Abstract

In structural design, the design variables are frequently selected from certain discrete values. Various optimization algorithms have been developed for discrete design. It is well known that many function evaluations are needed in such optimization. Recently, sequential algorithm with orthogonal arrays (SOA), which is a search algorithm for a local minimum in a discrete space, has been developed. It considerably reduces the number of function evaluations. However, it only finds a local minimum and the final solution depends on the initial values of the design variables. A new algorithm is proposed to adopt a genetic algorithm (GA) in SOA. The GA can find a solution in a global sense. The solution from the GA is used as the initial design of SOA. A sequential usage of the GA and SOA is carried out in an iterative manner until the convergence criteria are satisfied. The performance of the algorithm is evaluated by various examples.

1. 서 론

최적설계 시 설계변수를 연속설계변수(continuous design variable)와 이산설계변수(discrete design variable)로 분류한다. 대부분의 최적설계 기법들은 연속설계변수를 다루는데 관심을 두고, 효율적인 알고리듬을 개발하는데 중점을 두고 있다. 그러나 실제 구조물의 최적설계에서는 설계변수를 어떤 특정한 값들 중에서 선택해야 하거나, 이미 규격화되어 있는 부품의 치수를 취급해야 하는 경우가 많다.[1~3] 따라서 이산설계변수를 처리하기 위한 연구가 많이 이루어지고 있다. 직교배열표를 이용한 순차적 알고리듬(sequential algorithm with orthogonal arrays, SOA),[4~7] 분단탐색법(branch and bound),[8,9] 시뮬레이티드 어닐링(simulated annealing, SA),[10~12] 유전알고리듬(genetic algorithm, GA)[13~16] 등이 그 예이다.

그 중 최근 발표된 바 있는 직교배열표를 이용한 순차적 알고리듬은 다른 알고리듬들과의 비교실험에서 우수한 결과를 제시함으로써 보다 효율적인 최적화 방법이다.[4~7] 하지만 순차적 알고리듬은 초기값으로부터 순차적으로 탐색공간을 이동해 가며 해를 찾아 가는 방법이므로, 초기 값에 크게 의존하고 탐색영역은 주어진 초기값으로부터 결정되는 국부 탐색영역이 된다.

유전알고리듬의 경우 전역해 공간에서 임의로 선정된 잠재 해 집단을 형성하고, 개체들의 적합도를 서로 비교하여 보다 우수한 해를 찾아 나가는 방법을 사용한다. 또한 유전 연산자를 통한 연산과정에서 목적함수(objective function) 값만을 필요로 하고, 연속성(continuity), 미분가능성(differentiability), 단봉성(unimodality) 등과 같은 다른 보조정보가 필요치 않으므로 이산공간에서의 최적화 알고리듬으로 우수한 성질을 갖고 있다.[17]

† 책임저자, 회원, 한양대학교 대학원 기계설계학과
* 대우 종합기계, 특수사업부
** 회원, 한양대학교 기계정보경영학부
E-mail : gjpark@hanyang.ac.kr
TEL : (031) 400-5246 FAX : (031) 407-0755

SOA와 GA는 그 특성에 있어 상호 보완할 수 있는 요소를 갖고 있다. 즉 SOA의 국부 탐색영역을 GA의 전역 탐색영역으로 확장시키고, GA의 함수계산량을 SOA를 이용하여 현저히 줄이는 것이다. 이러한 방법으로 국부 탐색능력은 그대로 유지하면서 전역 탐색능력을 활용하는 보다 효율적인 전역 탐색 알고리듬을 제안한다.

2. 이산설계공간에서의 최적화 방법

2.1 직교배열표를 이용한 순차적 알고리듬(SOA)

이산설계공간 **S** 에서 현재의 설계변수를 중심으로 하고, 주변의 설계변수를 3 수준으로 배치하여 탐색공간(search space)이라고 정의한다. 현재의 탐색공간에서 목적함수가 최소인 최소해를 구하고, 최소해를 중심으로 탐색공간을 재정의하여 이동하는 순차적 반복과정을 생각할 수 있다. 순차적 알고리듬은 탐색공간에서 최소해를 찾는 과정에서 전조합에 의한 함수계산 횟수를 효율적으로 줄이기 위하여 직교배열표를 이용한다. 설계변수 사이에 교호작용이 없다면 직교배열표를 이용하여 국부해를 찾을 수 있다. Fig. 1 에는 제한조건이 있는 문제에 대한 순차적 알고리듬의 흐름을 보이고 있다. 알고리듬 단계별 기본 구조는 다음과 같다.

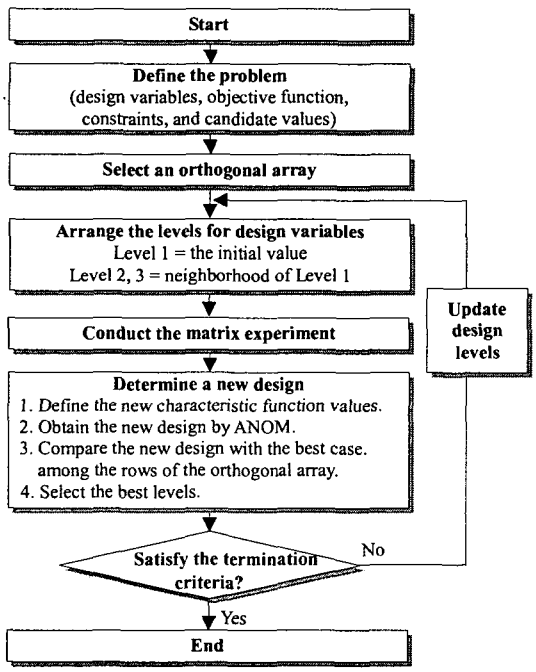

Fig. 1 Flow of sequential algorithm for constrained problems

1 단계 : 문제 정의

최적화를 위한 문제의 특성치(목적함수)와 인자(설계변수)를 설정한다. 이산설계를 위한 설계변수가 가질 수 있는 후보값들을 설정해야 한다. 행렬실험에서 수준수는 보통 2 수준과 3 수준을 사용한다. 제시된 알고리듬에서는 최대값이나 최소값 탐색에 용이한 3 수준을 사용하였다.[18]

2 단계 : 직교배열표 선택

인자 수와 수준수를 고려하여 행렬실험에서 사용할 직교배열표를 적절하게 선택한다. 직교배열표의 열의 수는 최대 적용 가능한 설계변수의 수를 나타내므로 이를 잘 고려하여야 한다.[19,20] 또한 선택한 직교배열표의 실험횟수에 따라 함수계산 횟수가 결정되므로 적절한 직교배열표를 선택하여야 한다. 직교배열표의 선택은 표준직교배열표[18]라고 부르는 기본적인 직교배열표를 참고로 필요한 설계변수를 포함할 수 있는 것을 선택한다.[21,22] 각각 3 수준을 갖는 설계변수 A, B, C, D 에 대하여 Table 1과 같이 $L_9(3^4)$ 직교배열표를 선택할 수 있다.

3단계 : 설계변수의 수준값 결정

행렬실험을 위해 직교배열표에 명시된 수준에 대응되는 각 설계변수의 이산값을 직교배열표에 배치해야 한다. 수준값들은 설계변수의 후보값들 중에서 선택하여야 한다. 초기 설계값은 3 개의 수준수 중 1 수준으로 설정하고 인접한 한 단계 낮거나 높은 후보값은 2 수준과 3 수준으로 설정한다. 이것은 다음 반복과정에서 초기 설계값보다 크거나 작은 범위로 탐색이 가능하도록 해 준다. 만약 최소값이나 최대값이 초기 설계값으로 주어진 경우 순차적으로 크거나 작은 값들을 2, 3 수준에 배치한다.[21] Fig. 2 는 설계변수의 수준값 결정에 대한 예를 보여준다.

한 번의 반복과정이 끝나는 7 단계까지 모두 거친 후 최적해가 선정되면 설계변수의 최적 조

Table 1 $L_9(3^4)$ orthogonal array

Expt. No.	Column number and factor assigned				Result (η)
	1	2	3	4	
	A(b_1)	B(b_1)	C(b_1)	D(b_1)	
1	1	1	1	1	η_1
2	1	2	2	2	η_2
3	1	3	3	3	η_3
4	2	1	2	3	η_4
5	2	2	3	1	η_5
6	2	3	1	2	η_6
7	3	1	3	2	η_7
8	3	2	1	3	η_8
9	3	3	2	1	η_9

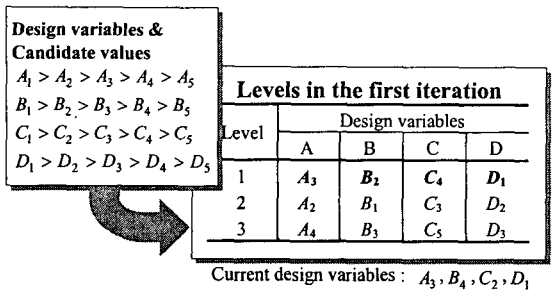

Fig. 2 An example of the selection of level values

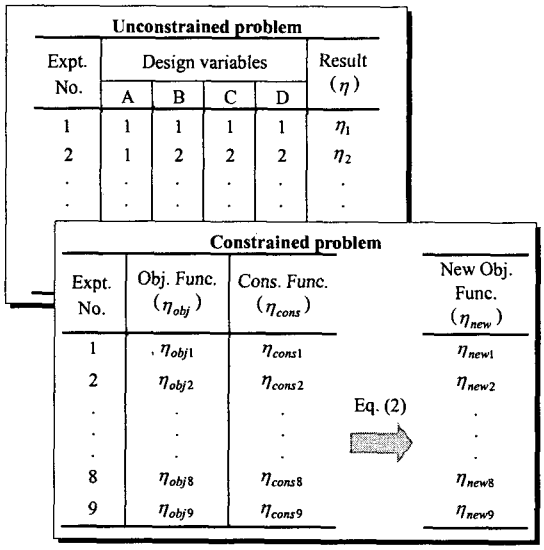

Fig. 3 An example of the characteristic function calculation

건을 1 수준으로 선정하고 인접한 한 단계 낮거나 높은 후보값들을 2 수준과 3 수준으로 선정하여 다음 반복과정을 실시한다.

4 단계 : 행렬실험 실시

Fig. 3 과 같이 직교배열표의 각 행마다 배치된 인자의 수준에 따른 특성치 η를 계산한다. 제한조건이 있는 문제에서는 목적함수에 대한 특성치 η_{obj}와 제한조건에 대한 특성치 η_{cons}(제한조건의 최대 위배량)을 구할 수 있다.

제한조건이 있는 문제의 경우 제한 조건을 고려하기 위하여 벌칙함수를 정의하여 제한조건의 위배정도에 따라 벌칙함수값이 부과된 새로운 특성치 η_{new}를 계산한다.

$$P = s \times \eta_{cons} = s \times \max[0, g_1, g_2, \ldots, g_l] \quad (1)$$

$$\eta_{new} = \eta_{obj} + P \quad (2)$$

벌칙함수는 식 (1)에 의하여 구할 수 있다. 여기서 P는 벌칙함수, g_i는 i 번째 제한조건의 위배량이고 s는 조절계수이다.

Table 2 One-way table for $L_9(3^4)$ orthogonal array

Design variables	Levels 1	2	3
A(b_1)	$m_{A1} = \frac{\eta_1+\eta_2+\eta_3}{3}$	$m_{A2} = \frac{\eta_4+\eta_5+\eta_6}{3}$	$m_{A3} = \frac{\eta_7+\eta_8+\eta_9}{3}$
B(b_2)	$m_{B1} = \frac{\eta_1+\eta_4+\eta_7}{3}$	$m_{B2} = \frac{\eta_2+\eta_5+\eta_8}{3}$	$m_{B3} = \frac{\eta_3+\eta_6+\eta_9}{3}$
C(b_3)	$m_{C1} = \frac{\eta_1+\eta_6+\eta_8}{3}$	$m_{C2} = \frac{\eta_2+\eta_4+\eta_9}{3}$	$m_{C3} = \frac{\eta_3+\eta_5+\eta_7}{3}$
D(b_4)	$m_{D1} = \frac{\eta_1+\eta_5+\eta_9}{3}$	$m_{D2} = \frac{\eta_2+\eta_6+\eta_7}{3}$	$m_{D3} = \frac{\eta_3+\eta_4+\eta_8}{3}$

5 단계 : 최적조건 선정

행렬실험의 결과를 바탕으로 평균분석(analysis of mean, ANOM[21])을 행한다. 평균분석으로는 일원표(one-way table)를 이용한다. A 인자의 A_1에 대하여 A_1의 주효과(main effect) M_{A1}을 구하면 다음과 같다.

$$M_{A1} = m_{A1} - m, \quad (3)$$

$$m_{A1} = \frac{1}{3}(\eta_1 + \eta_2 + \eta_3), \quad (4)$$

$$m = \frac{1}{9}\sum_{i=1}^{9}\eta_i \quad (5)$$

여기서 m은 특성치에 대한 전체평균, m_{A1}은 A_1의 영향이 포함된 특성치의 평균이다. 특성치에 대한 최적수준은 Table 2 와 같이 각 인자의 수준별 평균값을 계산하여 최소가 되는 인자들의 조합으로 구해진다. 만약 $m_{A1}, m_{B2}, m_{C3}, m_{D1}$가 인자에 대한 최소값이면, A_1, B_2, C_3, D_1이 인자의 최적수준이 된다.

6 단계 : 결과분석

직교배열표에 설계 변수를 배치할 때 이차 이상의 교호작용을 무시하였기 때문에 평균분석을 통한 분석에 오차가 발생할 수 있다. 따라서 평균분석을 통한 최적조합의 해와 비교하여 직교배열표상의 조합에 의한 해가 더 우수하다면 그 결과를 최적의 해로 간주한다.

7 단계 : 종료조건

종료조건은 (1)현재 설계변수가 변화 없이 이전 해와 동일한 경우 (2)제한조건을 위배하는 해가 연속적으로 구해지는 반복회수로 한다. 어떤 종료조건도 만족하지 않으면 3 단계로 돌아가 반복과정을 수행한다.

2.2 유전알고리듬(GA)

유전알고리듬은 자연현상을 모방하여 강건성과 문제비종속성의 이점을 가지는 혁신적인 탐색 알고리듬이다.[16]

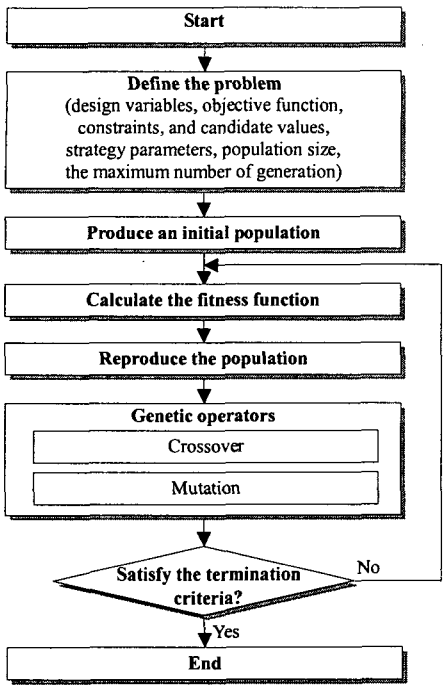

Fig. 4 Flow of the simple genetic algorithm

특히 파라미터의 코딩(유전자형)을 갖고 동작하므로 이산공간에서의 최적화에 적합한 알고리듬이다. 또한 확률적 천이 규칙을 사용하므로 순차적 알고리듬의 탐색공간 선정에 있어 우수한 영역을 선택할 확률적 근거를 마련해 준다. Fig. 4는 단순유전알고리듬(simple genetic algorithm)의 흐름을 보여주고 있다. 알고리듬 단계별 기본 구조는 다음과 같다.

1 단계 : 문제 정의

최적화를 위한 염색체(설계변수)와 유전자(후보값)에 대한 암호화(coding)를 실시한다. 함수 최적화 문제의 경우 일반적으로 실수코딩(real-number encoding)이 사용된다.[13] 또한 모집단의 크기, 최대 교배 횟수, 교차율(crossover ratio), 돌연변이율(mutation ratio)도 문제의 특성에 맞게 설정한다.

2 단계 : 초기 모집단 생성

설계변수의 후보값들로 이루어진 초기 모집단을 생성한다. 염색체를 이루는 각각의 유전자는 무작위로 추출된다. 반복과정 중 모집단 생성은 4단계의 재생산과정을 통해 이루어진다.

3 단계 : 목적함수 및 적합도 함수 계산

각 염색체에 대하여 목적함수를 계산한다. 모든 개체에 대하여 목적함수가 구해지면, 그 중 가장 우수한 값을 최적해로 선택한다.

목적함수를 이용하여 적합도 평가함수(fitness function)를 계산한다. 이때 평가함수는 설계자가 적절히 정식화한다.[13,14]

4 단계 : 재생산

초기 모집단의 개수 만큼 개체를 선택하여 다음 세대의 모집단을 구성한다. 이때 3단계에서 구한 적합도를 이용하여 룰렛휠방법(roulette wheel selection)[23]으로 선택한다.

5 단계 : 유전연산

선택된 개체들은 유전연산자인 교차와 돌연변이 연산를 통해 진화하게 된다.[13~16]

6 단계 : 종료조건

종료조건은 (1)설계자에 의해 주어진 진화횟수가 초과된 경우 (2)유전자형이 일정한 형태로 수렴한 경우 (3)해의 개선이 일정횟수이상 이루어지지 않은 경우를 사용한다. 종료조건을 만족하지 않으면 3단계로 돌아가 반복과정을 진행한다.

3. 유전알고리듬을 이용한 순차적 알고리듬(SOAGA)

앞에서 설명한 유전알고리듬과 순차적 알고리듬의 특성을 이용하여 직교배열표와 유전알고리듬을 이용한 순차적 알고리듬(sequential algorithm with orthogonal arrays and genetic algorithm, SOAGA)을 제안한다.

SOAGA는 Fig. 5의 단계를 반복적으로 실행하며 개선된 해를 찾아 간다. 이에 대하여 각 단계의 진행 과정을 설명하면 다음과 같다

1 단계 : 문제 정의

문제 정의 단계로 2.1의 1단계와 2.2의 1단계와 같은 방법으로 설계변수와 유전연산자 등을 설정한다.

2 단계 : 직교배열표 선택

2.1의 2단계와 같이 인자 수와 수준수를 고려하여 행렬실험에서 사용할 직교배열표를 적절하게 선택한다.

3단계 : 초기 모집단 생성

2.2의 2단계와 같은 방법으로 GA의 초기 모집단을 구성한다.

4단계 : 유전알고리듬으로부터 순차적알고리듬의 초기값 선정

2.2의 3~6단계와 같은 방법으로 해를 구하여 SOA 초기값으로 사용된다. 이때 구해진 해가 이전의 해와 같은 경우 5단계 진행을 유보한 채 최대 10회까지 진화시킨다. 이는 SOA가 초기값에 의존함으로 동일한 결과를 피하기 위함이다. 이때 종료조건 (1)의 만족 여부를 확인한다.

5단계 : 설계값 결정

2.1의 3~6단계와 같은 방법으로 해를 구한다. 4단계에서 얻어진 GA의 해는 SOA의 초기값으로 사

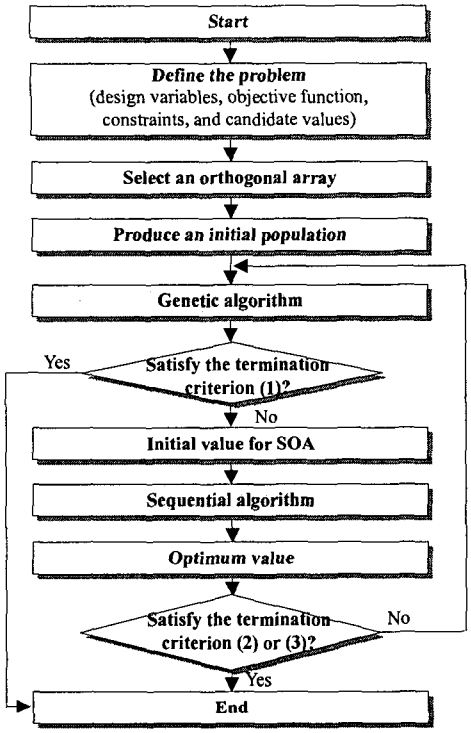

Fig. 5 Flow of sequential algorithm with orthogona arrays and genetic algorithm

용된다. 이때 얻어진 해는 SOAGA의 각 단계를 1회 실행한 결과에 대한 해가 된다.

6단계 : 종료조건

종료조건은 (1)GA가 새로운 해를 제공하지 못할 경우 (2)해의 개선이 일정 횟수에 걸쳐 이루어지지 않은 경우 (3)함수의 특성을 고려하여 지정한 반복횟수가 초과된 경우이다.

4. 예제 및 고찰

앞의 3장에서 제안한 알고리듬을 이용하여 이산공간에서의 최적화 알고리듬의 타당성을 검증하기 위해 수학예제, 10부재 트러스(ten-bar truss) 및 25부재 트러스(transmission tower) 예제를 예시한다. 또한 각각의 문제들은 순차적 알고리듬(Sequential Algorithm with Orthogonal Arrays, SOA), 다봉 유전알고리듬(Multi-Island Genetic Algorithm, MIGA), 랜덤방법(Random Method), 시뮬레이티드 어닐링(Adaptive Simulated Annealing, ASA) 등의 결과와 비교하였다.

유전알고리듬은 상용소프트웨어인 iSIGHT 7.0[24]에 포함된 MIGA 를 이용하였다. 모집단의 크기는 250(subpopulation=50, No. of island=5), 교차율 0.98, 돌연변이율 0.1 을 사용하였고, 종료조건은 함수계산 횟수의 비교를 위해 해의 개선이 이루어지지

Levels in the first iteration

Level	Design variables			
	b_1	b_2	b_3	b_4
1	-3	-1	-3	-1
2	-4	-2	-4	-2
3	-2	0	-2	0

Matrix experiment

Expt. No.	Design variables				Result(η)
	b_1	b_2	b_3	b_4	
1	-3	-1	-3	-1	19192
2	-3	-2	-4	-2	41661
3	-3	0	-2	0	9605
4	-4	-1	-4	0	52080.1
5	-4	-2	-2	-1	34934.1
6	-4	0	-3	-2	36691.4
7	-2	-1	-2	-2	6008.1
8	-2	-2	-3	0	11075.4
9	-2	0	-4	-1	27734.1

Fig. 6 The matrix experiment using $L_9(3^4)$ orthogonal array

않는 세대수가 5 회 이상일 때 종료하도록 하였다.

제안된 알고리듬은 유전알고리듬의 함수계산 횟수를 최대한 줄이기 위하여 모집단의 크기를 50으로 설정하였다. 모집단의 크기를 제외한 나머지 사항에 대해서는 유전알고리듬과 동일하게 적용하였다.

4.1 수학적 예제

$$\textbf{Find} \quad b_1, b_2, b_3, b_4 \tag{6}$$

$$\textbf{to minimize} \quad f(\mathbf{b}) = 100(b_2 - b_1^2)^2 + (1-b_1)^2 + 90(b_4 - b_3^2)^2 + (1-b_3)^2 + 10.1((b_2-1)^2 + (b_4-1)^2) + 19.8(b_2-1)(b_4-1)$$

$$\textbf{subject to} \quad -10 \le b_i \le 10 \qquad i = 1,2,3,4$$

먼저 수학 함수를 이용하여 유전알고리듬을 통한 국부해 영역선택 경향과 해의 개선여부를 타진해 보았다. 모든 알고리듬에 대하여 초기값은 $b_1 = -3$, $b_2 = -1$, $b_3 = -3$, $b_4 = -1$로 동일하게 주어졌다. 각 설계변수의 후보값은 $b_1, b_2, b_3, b_4 \in \{-10, -9, -8, -7, \ldots, 7, 8, 9, 10\}$로 설정하였다.

1단계에서 행렬실험과 GA를 위한 문제정의를 하고, 2단계에서는 설계변수가 4개 수준수가 3이므로 $L_9(3^4)$ 직교배열표를 선택하였다.

SOAGA는 GA로부터 SOA의 초기값이 주어지지만, 이 문제의 경우는 비교를 위한 SOA와 같은 초기값을 사용하였다. 이는 SOAGA가 기존의 SOA로부터 얻어진 해를 개선할 수 있는지 여부를 판단하기 위함이다. 따라서 3단계에서 초기 모집단 구성 후 첫 번째 반복과정에 대해서는 4단계과정은 생략하고 주어진 초기값을 가지고 5

Table 3 One-way table for mathematical problem

Design variables	Levels		
	1	2	3
A(b_1)	m_{A1} = 23486.0	m_{A2} = 41235.2	m_{A3} = 14939.2
B(b_2)	m_{B1} = 25760.1	m_{B2} = 29223.5	m_{B3} = 24676.8
C(b_3)	m_{C1} = 22319.6	m_{C2} = 40491.7	m_{C3} =16849.1
D(b_4)	m_{D1} = 27286.7	m_{D2} = 28120.2	m_{D3} = 24253.5

Table 4 Results of the mathematical problem

Method	No. of function evaluation	Optimum variables				Function value
		b_1	b_2	b_3	b_4	
SOA	30	-1	1	-1	1	8
Random	720	1	1	-1	1	8
SOAGA	670	1	1	1	1	0

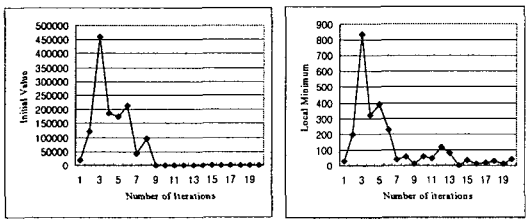

Fig. 7 History of the function value for a mathematical example

단계를 실행한다.

Fig. 6과 같이 설계변수를 직교배열표에 배치하고 행렬실험을 실시한다. 이 문제는 제한조건이 없으므로 새로운 특성치 계산은 하지 않는다.

실험결과로 설계변수들의 각 수준에 대한 일원표(one-way table)를 Table 3과 같이 구성하고 최적수준을 구한다. 일원표로부터 구해진 $A_3B_3C_3D_3$에 대하여 확인실험을 실시하고 직교배열표의 해와 비교하여 우수한 해를 최적해로 선택한다. 선택된 해는 다음 반복과정의 초기값으로 사용된다.

이 문제의 경우 같은 초기값을 사용하였으므로 SOAGA의 첫 번째 반복과정의 해는 기존 SOA의 해와 일치한다. 기존의 SOA는 여기서 알고리듬을 종료한다. 반면 SOAGA는 다른 초기값으로부터 총 20회의 탐색을 반복한다. 이때 첫 번째 반복과정에서 구해진 해를 GA의 초기 모집단에 포함시켜 다음 반복과정의 초기값 선택에 영향을 주었다.

두 번째 반복과정부터는 4단계로부터 SOA의 초기값을 구한다. 반복횟수는 총 20회이며, 각 반복횟수에 대한 초기값과 결과는 Fig. 7과 같다.

SOAGA는 탐색공간의 이동을 통하여 기존의 SOA로부터 얻어진 결과보다 개선된 해를 구할 수 있었다. 또한 랜덤방법의 경우와 달리 뚜렷한 수렴특성을 보이고 있다. 각 알고리듬에 대한 결과는 Table 4와 같다.

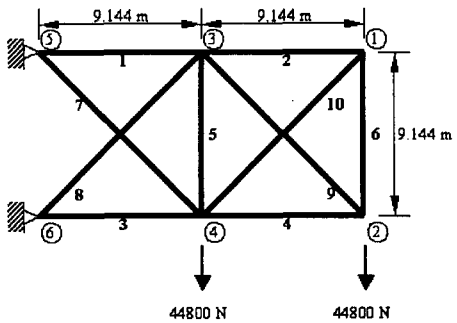

Fig. 8 Ten bar truss

Table 5 Design variables and their values of the initial level ($\times 10^{-5}$ m^2) for ten bar truss

Levels	Design variables									
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
1	729	729	729	465	465	465	465	465	465	465
2	645	645	813	400	400	400	400	400	400	400
3	813	813	645	542	542	542	542	542	542	542

Table 6 Assignment of design variables in $L_{27}(3^{13})$

Expt. No.	Column number and design variable assigned												
	1	2	3	4	5	6	7	8	9	10	11	12	13
	A_1	A_2	ignored	A_3	A_4	ignored	A_5	ignored	A_6	A_7	A_8	A_9	A_{10}
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	2	2
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
27	3	3	2	1	3	2	1	2	1	3	1	3	2

4.2 10부재 트러스

Fig. 8과 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($\rho$) 2770 kg/m^3, 포아송비($\nu$) 0.33을 사용하였다. 문제의 정식화는 식 (7)과 같다.

$$\begin{aligned} &\textbf{Find} && A_i \\ &\textbf{to minimize} && \text{mass (kg)} \\ &\textbf{subject to} && -172.4\text{MPa} \le \sigma_i \le 172.4\text{MPa} \\ &\textbf{candidate values} && A_i\ (\times 10^{-5}\ \text{m}^2) \in \{6, 97, 194, 323, \\ & && 400, 465, 542, 645, 729, 813\}\ (i = 1,2,\cdots,10) \end{aligned} \tag{7}$$

SOAGA를 제외한 모든 방법에서 Table 5와 같이 동일한 초기 설계값을 적용하였다. 설계변수의 수가 10개이고, 수준 수는 3인 점을 감안하여 $L_{27}(3^{13})$ 표준 직교배열표를 선택하여 Table 6과 같이 설계변수를 배치하였다.

SOAGA의 각 반복횟수에 대한 초기값과 결과는 Fig. 9와 같다. SOAGA는 총 10회의 탐색공간 이동을 통하여 가장 우수한 최적값을 구하였고, MIGA에 비하여 함수계산 횟수가 월등히 적었다. 또한

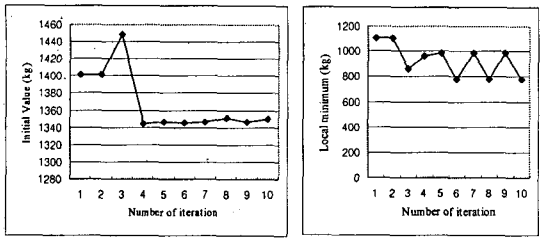

Fig. 9 History of the function value for ten bar truss

Table 7 Results of ten bar truss with SOA, ASA, MIGA and SOAGA

Method	Mass (kg)	No. of Function Evaluation	Optimum design variables ($\times 10^{-5}$ m^2)										Feasibility
			A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	
SOA	879.0	351	542	194	542	323	97	97	400	400	323	97	feasible
ASA	1087.5	327	465	323	729	97	323	400	465	400	97	465	feasible
MIGA	931.5	5369	542	97	97	542	465	600	97	465	465	600	feasible
SOAGA	781.7	2889	542	6	542	323	6	6	400	400	400	6	feasible

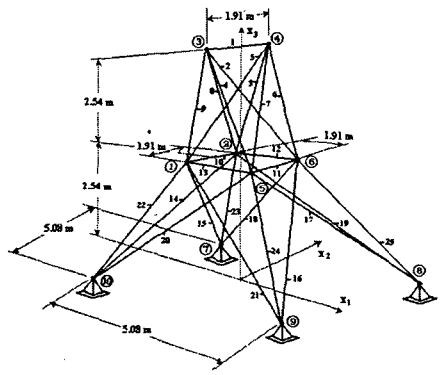

Fig. 10 Twenty-five member transmission tower

SOA와 ASA에 비하여 함수계산 횟수는 많았으나 가용영역에서의 초기값을 찾을 필요가 없고 전역해 영역에서의 탐색이라는 점에서 결과의 우수함을 확인할 수 있다. 각 알고리듬에 의한 결과는 Table 7와 같다.

4.3 25부재 트러스

Fig. 10과 같은 25부재 트러스 구조물에 대하여 Table 8의 2가지 하중조건에 대하여 응력에 대한 구속조건을 만족하면서 질량을 최소화 하는 부재의 단면적을 구한다. 모든 부재에 대하여 탄성계수(E) 68.9GPa, 밀도(ρ) 2770kg/m^3, 포아송비(ν) 0.33이다. 문제의 정식화는 식 (8)과 같다.

$$\begin{aligned} &\textbf{Find} && b_i \\ &\textbf{to minimize} && \text{mass (kg)} \\ &\textbf{subject to} && -275.8\text{MPa} \le \sigma_j \le 275.8\text{MPa} \\ & && (j = 1,2,\cdots,25) \end{aligned} \tag{8}$$

Table 8 Nodal load components for transmission tower

Load case	Node	x	y	z
1	1	4448.22	44482.22	-22241.11
	2	0	44482.22	-22241.11
	3	2224.11	0	0
	6	2224.11	0	0
2	5	0	88964.43	2224.11
	6	0	88964.43	-2224.11

Table 9 Design variables and their values of the initial level (m^2) for twenty five bar truss

Levels	Design variables						
	b_1	b_2	b_3	b_4	b_5	b_6	b_7
1	0.044	0.044	0.044	0.024	0.024	0.024	0.044
2	0.024	0.024	0.024	0.0094	0.0094	0.0094	0.024
3	0.064	0.064	0.064	0.044	0.044	0.044	0.064

Table 10 Assignment of design variables in $L_{18}(2^1\times 3^7)$

Expt. No.	Column number and design variable assigned							
	1	2	3	4	5	6	7	8
	ignored	b_1	b_2	b_3	b_4	b_5	b_6	b_7
1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
18	2	3	3	2	1	2	3	1

Table 11 Results of twenty five bar truss with SOA, ASA, MIGA and SOAGA

Method	Mass (kg)	No. of Function evaluation	Optimum design variables ($\times 10^{-4}$m^2)							Feasibility
			b_1	b_2	b_3	b_4	b_5	b_6	b_7	
SOA	4243	162	440	240	240	94	74	34	440	feasible
ASA	5450	262	640	440	94	54	240	74	440	feasible
MIGA	4570	2660	440	440	240	94	54	6	440	feasible
SOAGA	4058	1152	440	240	240	54	74	34	440	feasible

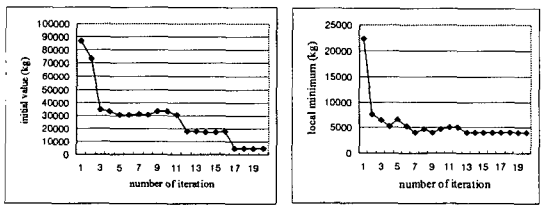

Fig. 11 History of the function value for twenty five bar truss

candidate values

$b_i(\times 10^{-4}\text{m}^2) \in \{0.74, 0.94, 2.4, 4.4, 6.4, 8.4, 14, 34, 54, 74, 94, 240, 440, 640\}$ $(i = 1,2,\cdots,7)$

설계변수는 각 부재의 단면적으로 대칭성을 고려하여 설계변수 연결을 사용하였다. $b_1=A_1$, $b_2=A_2=A_3=A_4=A_5$, $b_3=A_6=A_7=A_8=A_9$, $b_4=A_{10}=A_{11}=A_{12}=A_{13}$, $b_5=A_{14}=A_{15}=A_{16}=A_{17}$, $b_6=A_{18}=A_{19}=A_{20}=A_{21}$ 및 $b_7=A_{22}=A_{23}=A_{24}=A_{25}$ 이다. SOAGA를 제외한 모든 알고리듬에 대해서 Table 9와 같이 동일한 초기 설계변수를 설정하였다. 설계변수의 수가 7개이고, 수준 수는 3인 점을 감안하여 $L_{18}(2^1\times 3^7)$ 표

준 직교배열표를 선택하여 Table 10과 같이 설계변수를 배치하였다.

Fig. 11은 20회의 반복횟수에 대한 초기값과 해를 나타내고 있다. 25부재 트러스의 경우 10부재 트러스와 달리 함수값의 분포 영역이 넓어 모집단의 수렴특성이 뚜렷이 나타나는 것을 볼 수 있다.

SOAGA와 MIGA는 비슷한 결과값을 얻을 수 있었지만 함수 계산 횟수에서 많은 차이를 보이고 있다. SOA의 경우 우수한 값을 보였으나 초기값의 의존도가 높아 주어진 초기값에 따라 결과값이 다르게 나타날 수 있는 단점이 있다. 각 알고리듬에 대한 결과는 Table 11과 같다.

5. 결 론

이산 설계변수에 대하여 유전알고리듬을 이용한 순차적 알고리듬으로 최적해를 구하는 방법을 제안하였다. 또한 예제를 통하여 최적값 및 함수계산 횟수에서 그 우수함을 입증하였다. 이를 통하여 다음과 같은 결론을 얻을 수 있었다.

(1) 순차적 알고리듬은 유전알고리듬의 수행시 요구되는 비용과 시간을 최소화 하였다. 유전알고리듬은 순차적 알고리듬의 초기값 선택에 있어 적합도를 이용함으로써 전역최적해로 단정할 수 없지만 확률적으로 개선된 해를 찾게 하였다.

(2) 유전알고리듬의 전역 탐색 특성에 의해 전역해 영역으로의 확장과 함께 비가용 영역과 가용 영역의 이동이 가능하게 되었다.

(3) SOAGA에 사용된 유전알고리듬은 일반적으로 많이 사용되는 단순 유전알고리듬(simple genetic algorithm)으로써, 해의 정확성은 뛰어나나 계산의 속도가 느리고 재생산 방법으로 룰렛 휠 선택(roulette wheel selection)을 사용하므로 유전정보의 다양성을 유지하지 못하는 단점이 있었다.

(4) 적용사례는 해의 개선 가능성 타진을 그 목적으로 하였다. 예제 분석에 있어 보다 타당성 있고 공정한 비교를 위해서는 다양한 예제의 적용과 함께 유전 연산자 설정을 위한 각 파라미터들과 종료조건 등의 고려가 필요하다.

참고문헌

(1) 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.

(2) 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.

(3) 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.

(4) Yi, J.W., 2004, "A Sequential Algorithm Using Orthogonal Arrays in Discrete Space," Ph.D. Thesis, Hanyang University, Korea (in Korean).

(5) 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. 1651~1656 (in Korean).

(6) Yi, J.W. and Park, G.J., 2004, "Local Solution of a Sequential Algorithm Using Orthogonal Arrays in a Discrete Design Space," *Transactions of the KSME(A)*, Vol. 28, No. 9, pp. 1399~1407 (in Korean).

(7) Lee, K.H., Yi, J.W., Park, J.S. and Park, G.J., 2003, "An Optimization Algorithm Using Orthogonal Arrays in Discrete Design Space for Structures," *Finite Elements in Analysis and Design*, Vol. 40, pp. 121~135.

(8) Tseng, C.H., Wang, L.W. and Ling, S.F., 1995, "Enhancing Branch-and-Bound Method for Structural Optimization," *Journal of Structural Engineering, ASCE*, Vol. 121, pp. 831~837.

(9) Gutkowski, W., 1997, Discrete Structural Optimization, Springer-Verlag, New York.

(10) Pham, D.T. and Karaboga, D., 2000, Intelligent Optimization Techniques, Springer-Verlag, New York.

(11) Osman, I.H. and Kelly, J.P., 1996, Meta-Heuristics: Theory & Applications, Kluwer Academic Publishers, Boston.

(12) Kim, Y.K., Yoon, B.S. and Lee, S.B., 1999, Meta-Heuristic, Youngji (in Korean).

(13) Gen, M. and Cheng, R., 2000, Genetic Algorithms and Engineering Optimization, John Wiley & Sons, Inc., New York.

(14) Goldberg, D.E., 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Pub. Co.

(15) Gong, S.K., Kim, I.T., Park, D.H., Park, J.Y. and Shin, Y.H., 1997, Introduction of the Genetic Algorithm, Jinyoung (in Korean).

(16) Hwang, H.S., 2002, Evolution Design and Calculating Using the Computer, Naeha (in Korean).

(17) Janikow, C.Z., An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithm, Proceedings of the Fourth International Conference on GA.

(18) Peace, G.S., 1995, Taguchi Methods: A Hands-On Approach, Addison Wesley, Massachusetts.

(19) Park, S.H., 1990, Applications of the Design of Experiments, Youngji (in Korean).

(20) Park, S.H., 2002, Modern Design of Experiments, Minyoungsa (in Korean).

(21) Park, G.J., 2006, Analytic Methods for Design Practice, Spring-Verlag, New York.

(22) Phadke, M.S., 1989, Quality Engineering Using Robust Design, Prentice-Hall, New Jersey.

(23) Holland, J., 1992, Adaptation in Natural and Artificial systems, MIT Press, Massachusetts.

(24) iSIGHT 7.0, 2002, http://www.engineous.com, Engineous Software, Inc., NC.