이산공간에서 순차적 알고리듬(SOA)을 이용한 전역최적화

조범상† · 이정욱* · 박경진**

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

Bum-Sang Cho†, Jeong-Wook Yi*, Gyung-Jin Park**

Key Words : Discrete Design (이산설계), Sequential algorithm(순차적 알고리듬), Genetic algorithm(유전 알고리듬), Orthogonal arrays(직교배열표), Global Solution(전역해)

Abstract

In the optimized design of an actual structure, the design variable should be selected among any certain values or corresponds to a discrete design variable that needs to handle the size of a pre-formatted part. Various algorithms have been developed for discrete design. As recently reported, the sequential algorithm with orthogonal arrays(SOA), which is a local minimum search algorithm in discrete space, has excellent local minimum search ability. It reduces the number of function evaluation using orthogonal arrays. However it only finds a local minimum and the final solution depends on the initial value. In this research, the genetic algorithm, which defines an initial population with the potential solution in a global space, is adopted in SOA. The new algorithm, sequential algorithm with orthogonal arrays and genetic algorithm(SOAGA), can find a global solution with the properties of genetic algorithm and the solution is found rapidly with the characteristics of SOA.

1. 서 론

최적설계의 설계변수를 연속설계변수(continuous design variable)와 이산설계변수(discrete design variable)로 분류해보면, 대부분의 최적설계 기법들은 연속설계변수를 다루는데 관심을 두고, 효율적인 알고리듬을 개발하는데 중점을 두고 있다. 그러나 실제 구조물의 최적설계에서는 설계변수가 어떤 특정한 값들 중에서 선택해야 하거나, 이미 규격화되어 있는 부품의 치수를 취급해야 하는 경우가 많다[1,2,3]. 물론 이산설계변수를 처리하기 위한 연구도 많이 이루어지고 있으며, 직교배열표를 이용한 순차적 알고리듬(SOA)[4], 분단탐색법(branch and bound)[5,6], 시뮬레이티드 어닐링(simulated annealing, SA)[7-9], 유전 알고리듬(genetic algorithm, GA)[7-12] 등이 그 예이다.

그 중 최근 발표된 바 있는 직교배열표를 이용한 순차적 알고리듬(sequential algorithm with orthogonal arrays, SOA)은 다른 알고리듬들과의 비교 실험에서 우수한 결과를 제시함으로써 보다 효율적인 최적화 방법이다. 하지만 순차적 알고리듬은 사용자에 의해 주어진 초기값으로부터 탐색공간을 순차적으로 이동해 가며 해를 찾아 가는 방법이므로 초기 값에 크게 의존하고, 탐색영역은 주어진 초기값으로부터 결정되는 국부 영역이 된다. 따라서 사용자에 의해 결정되는 탐색영역을 전역해 영역으로 확장하기 위해서는 전역해 공간 내의 임의의 값을 순차적 알고리듬의 초기 값으로 하는 방법을 생각할 수 있다.

유전 알고리듬의 경우, 전역해 공간에서 임의로 선정된 잠재 해 집단을 형성하고 개체들의 적합도를 서로 비교하여 보다 우수한 해를 찾아 나

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

가는 방법을 사용한다. 또한 유전 연산자를 통한 연산과정에서 이득함수(payoff function) 또는 목적함수(objective function) 값만을 필요로 하고, 연속성(continuity), 미분가능성(differentiability), 단봉성(unimodality) 등과 같은 다른 보조정보를 요구하지 않으므로 이산공간에서의 최적화 알고리듬으로서 우수한 성질을 갖고 있다[13]. 이산설계공간에서 순차적 알고리듬의 뛰어난 국부탐색 능력과 유전알고리듬의 전역 탐색 능력을 상호 보완한 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 3수준으로 배치하여 탐색공간(search space)이라고 정의한다. 현재의 탐색공간에서 목적함수가 최소인 최소해를 구하고, 최소해를 중심으로 탐색공간을 재정의하여 이동하는 순차적 반복과정을 생각할 수 있다.[4]

또한 함수값 비교 시 직교배열표를 이용하면 전조합을 실시한 것과 같은 효과를 얻을 수 있다[14,15]. 그러므로 설계변수 사이에 교호작용이 없다면 직교배열표를 이용하여 국부해를 찾을 수 있다. 그러나 설계변수 사이에 교호작용이 존재한다면 근사적으로 국부해를 찾을 수 있다. 그러므로 순차적 반복과정을 통하여 근사적으로 국부해를 구한다.

2.2 유전 알고리듬(GA)

유전 알고리듬은 자연현상을 모방하여 강인성과 문제비종속성의 이점을 가지는 혁신적인 탐색 알고리듬이다[16].

순차적 알고리듬의 탐색영역을 선택하기 위한 알고리듬으로 유전 알고리듬을 이용하는 이유를 간단히 정리하면, 다음과 같은 4가지 면에서 기존의 알고리즘과 다른 특성을 갖기 때문이다[9].

- 파라미터 그 자체(표현형)가 아니라 파라미터의 코딩(유전자형)을 갖고 동작한다.
- 해공간에서 한 점이 아닌 집단에서 동작한다.
- 미분값이나 보조정보를 사용하지 않고, 목적함수 값에 해당하는 적합도 값만을 사용한다.
- 결정론적 규칙이 아닌 확률적 천이 규칙을 사용한다.

위와 같은 특성은 이산 전역해 공간에서 우수한 성능을 발휘하며, 순차적 알고리듬의 탐색공간 선정에 있어 우수한 영역을 선택할 확률적 근거를 마련해 준다.

3. 유전 알고리듬을 이용한 순차적 알고리듬(SOAGA)

앞에서 설명한 유전알고리듬과 순차적 알고리듬의 특성을 이용하여 직교배열표와 유전 알고리듬을 이용한 순차적 알고리듬(sequential algorithm with orthogonal arrays and genetic algorithm, SOAGA)을 제안한다. SOAGA는 다양한 국부해 공간을 탐색 할 수 있는 장점 외에도 비가용영역과 가용영역을 가변적으로 이동 할 수 있고, 같은 국부해 영역 안에서도 적합도 평가에 의해 우수한 개체를 선택하여 순차적 알고리듬의 초기값으로 활용하게 되므로 순차적 알고리듬 상의 반복횟수를 줄이는 효과를 얻을 수 있다.

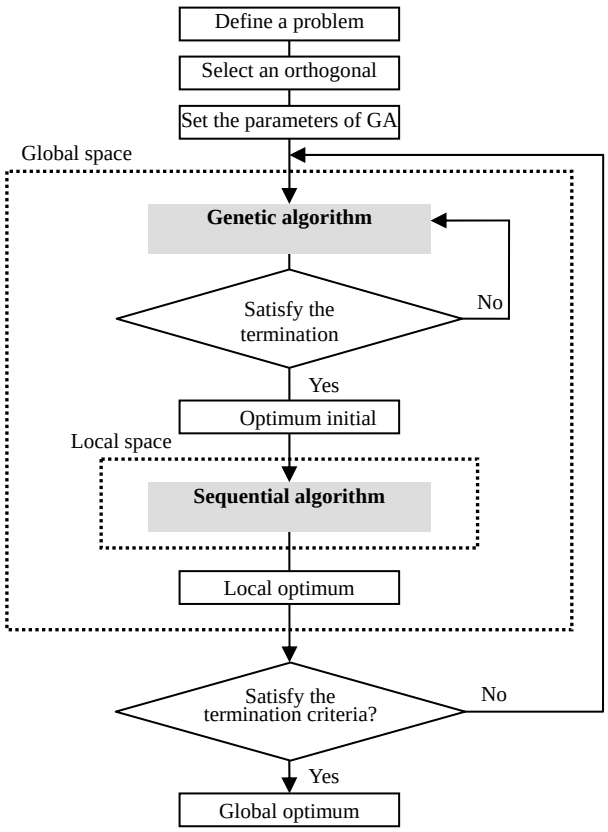

Fig. 1 Flow of sequential algorithm with orthogonal arrays and genetic algorithm

SOAGA는 Fig. 1의 단계를 반복적으로 실행하며 개선된 해를 찾아 간다. 이에 대하여 각 단계의 진행 과정을 설명하면 다음과 같다

1 단계 : 문제 정의

최적화를 위한 문제의 특성치(목적함수)와 인자(설계변수)를 설정한다. 순차적 알고리듬의 행렬실험 특성치와 유전 알고리듬의 적합도 평가 함

수는 목적함수에 해당하고, 순차적 알고리듬의 인자와 유전 알고리듬의 염색체를 이루는 각각의 인자는 설계변수에 해당한다. 행렬실험의 수준값과 염색체는 설계변수가 갖는 이산화 된 후보값들 중에서 선택한다. 제시된 알고리듬에서는 최대값이나 최소값을 탐색하는데 용이한 3수준을 사용하였다[17].

2 단계 : 직교배열표 선택

인자 수와 수준수를 고려하여 행렬실험에서 사용할 직교배열표를 적절하게 선택한다. 또한 선택된 직교배열표의 실험횟수에 따라 함수계산 횟수가 결정되므로 적절한 직교배열표를 선택하여야 한다. 직교배열표의 선택은 표준 직교배열표[17]라고 부르는 18개의 기본적인 직교배열표를 참고로 연구에 필요한 설계변수를 포함할 수 있는 것을 선택한다.

3 단계 : 파라미터 설정

유전 알고리듬의 유전 연산자 및 진화 연산자의 파라미터를 설정한다. 진화에 참여할 개체 집단 크기, 교배 확률, 돌연변이 확률, 적합도 평가 함수 등이 이 이에 해당한다. 제안된 알고리듬에서는 실수코딩이 사용되었으며 교배 확률 0.6, 돌연변이 확률 0.01로 설정하였다.

4단계 : 유전 알고리듬을 이용한 순차적 알고리듬의 초기값 선정

적합도 평가를 통하여 잠재적인 우수 해를 포함하고 있는 국부해 공간을 선택하는 단계이다. 이때 제한 조건이 있는 문제의 경우 제한 조건을 고려하기 위하여 순차적 알고리듬의 경우와 같이 제한조건의 위배 정도에 따라 벌칙함수값이 부과된 새로운 반응치를 계산한다[4].

위배량이 반영된 새로운 반응치로부터 개체 집단의 적합도 평가를 하고 재생산 과정을 통해 적합도가 높은 개체들로 형성된 다음 세대의 집단을 형성하게 된다.

5단계 : 순차적 알고리듬을 이용한 국부해 탐색

유전 알고리듬으로부터 얻어진 높은 적합도를 갖은 염색체는 순차적 알고리듬의 초기값으로 쓰여지게 된다. 탐색 공간 내의 최적값을 중심으로 하는 3수준의 탐색 공간을 재정의하고 최적값을 갱신하며 탐색 공간을 이동해나가는 반복과정을 실시한다. 이때 탐색 공간 내의 해의 비교는 앞서 설정한 직교배열표에 의하여 이루어지며 전조합 실시의 효과를 얻게 된다.

6단계 : 종료 조건

새로운 국부영역을 선정하고 국부해를 찾는 전체 과정은 다음과 같은 종료조건을 통해 반복 여부를 판단한다. 첫째, 전역해 공간에서 추출된 모집단으로부터 새로운 국부해 영역을 선택할 수 없을 경우, 즉 유전 알고리듬의 개체 집단이 일정한 값으로 수렴한 경우, 둘째, 해의 개선이 일정 회수에 걸쳐 이루어지지 않은 경우, 셋째, 함수의 특성을 고려하여 사용자가 지정한 반복회수가 초과된 경우이다.

4. 예제 및 고찰

4.1 수학적 예제

Find b_1, b_2, b_3, b_4

to minimize
$$\begin{aligned} f(\mathbf{b}) = {} & 100\left(b_2 - b_1^{\,2}\right)^2 + \left(1 - b_1\right)^{2\,2} \\ & + 90\left(b_4 - b_3^{\,2}\right) + \left(1 - b_3\right)^2 \\ & + 10.1\left(\left(b_2 - 1\right)^2 + \left(b_4 - 1\right)^2\right) \\ & + 19.8\left(b_2 - 1\right)\left(b_4 - 1\right) \end{aligned}$$

subject to $-10 \le b_i \le 10 \qquad i = 1,2,3,4 \qquad (1)$

SOAGA에서는 유전 알고리듬에 의한 각 세대의 최적값이 SOA의 초기값으로 주어졌으며, 반복횟수에 따른 종료조건을 사용하였다. ASA의 종료조건은 연속적으로 해의 개선이 이루어지지 않는 경우의 횟수가 2회, 5회의 경우를 실시하여 횟수와 최적해가 더 좋은 것을 채택하였다. MIGA의 종료조건은 세대 수, 개체군의 크기(size of sub-population), 섬의 수(number of islands)에 의하여 결정된다.

위의 예제는 SOAGA를 제외한 알고리듬에 대하여 초기치가 b_1 = -3, b_2 = -1, b_3 = -3, b_4 = -1 이고 함수값은 19192 이다. 각 설계변수의 후보값은 $b_1, b_2, b_3, b_4 \in$ {-10, -9, -8, -7, …, 7, 8, 9, 10}로 설정하였다. 위 예제를 각 알고리듬으로 풀어본 결과는 Table 1과 같다.

함수의 비선형성이 높아 대부분의 알고리듬이 최적해에 도달하지는 못하였으나, SOAGA는 20회의 국부해 영역 선택을 통해 연속공간에서의 최

Table 1 Results for mathematical example

method	no. of function evaluation	optimum variables				Function value
		b_1	b_2	b_3	b_4	
SOA	28	-1	1	-1	1	8
Random	613	1	1	-1	1	8
SOAGA	1290	1	1	1	1	0

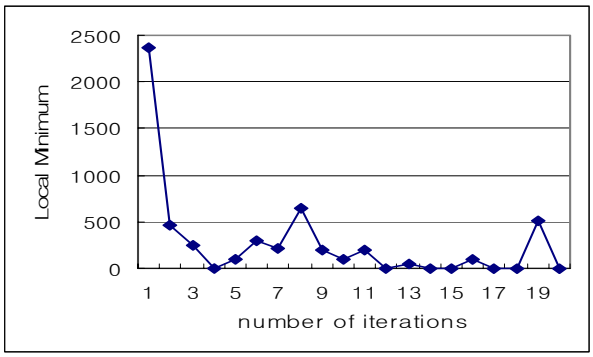

Fig. 2 History of the function value using GA for mathematical example

적해와 같은 값을 얻어냈다. 특히, 20회에 걸쳐 임의로 국부해 영역을 선택한 랜덤(Random) 탐색의 경우와 비교해 볼 때, Fig. 2에서 볼 수 있는 바와 같이 함수 값이 작은 영역, 즉 적합도가 높은 영역으로 이동하며 일관된 방향성과 수렴특성을 갖고 있음을 알 수 있다. 이러한 특성은 국부해 영역을 선택함에 있어 우수한 해를 찾을 수 있는 확률이 높은 영역을 선택한다는 것을 의미한다.

4.2 10 부재 트러스

Fig. 3과 같이 2개의 하중조건이 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) 2770 kg/m^3, 포아송비(ν) 0.33을 사용하였다.

Find A_i
to minimize mass (kg)
subject to $-172.4\text{MPa} \le \sigma_i \le 172.4\text{MPa}$
candidate values

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

설계변수의 수가 10개이고, 수준 수는 3인 점을 감안하여 $L_{27}(3^{13})$ 표준 직교배열표를 선택하여 설계변수를 배치하였다.

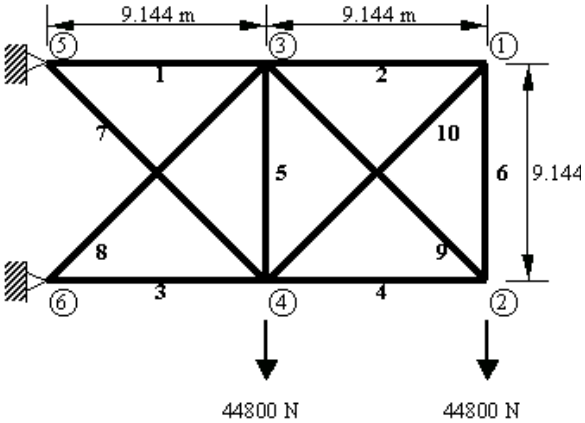

Fig. 3 Ten bar truss

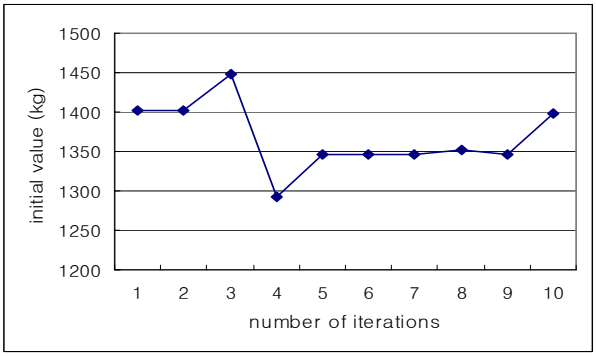

Fig. 4 History of the initial value for ten bar truss

각 알고리듬에 의한 결과는 Table 2과 같으며 SOAGA를 제외한 모든 방법에서는 SOA와 같은 초기 설계값과 설계변수 후보값을 적용하였다. 최적값은 10회의 반복횟수로부터 얻어진 SOAGA의 값이 가장 우수하였고, MIGA에 비하여 함수계산 횟수가 월등히 적었다. 또한 SOA와 ASA에 비하여 함수계산 횟수는 많았으나 가용영역에서의 초기값을 찾을 필요가 없고 전역해 영역에서의 탐색이라는 점에서 결과의 우수함을 확인할 수 있다.

Fig. 4, 5는 10회의 전체 반복횟수에 대한 초기값(initial value)과 각의 초기값으로부터 얻어진 국부해(local minimum)를 나타내고 있으며, 20회의 반복횟수에 대한 결과 역시 10회 부근에서 비슷한 값으로 수렴하였다.

Table 2 Results of ten bar truss with SOA, ASA, MIGA and SOAGA

method	Mass (kg)	no. of function. eval.uations	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	365	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	1110	542	6	542	323	6	6	400	400	400	6	feasible

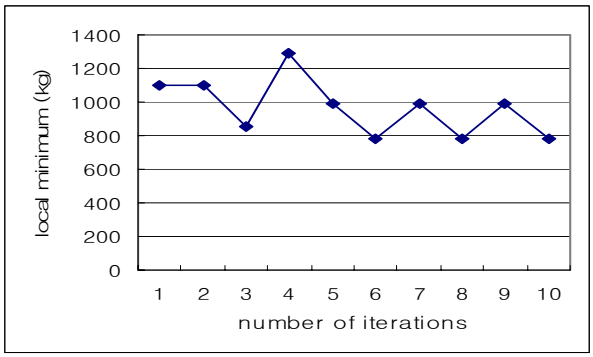

Fig. 5 History of the local minimum for each iteration numbers for ten bar truss

4.3 25 부재 트러스

Fig. 6과 같은 25부재 트러스 구조물에 대하여 Table 3의 2가지 하중조건에 대하여 응력에 대한 구속조건을 만족하면서 질량을 최소화 하는 부재의 단면적을 구한다. 모든 부재의 탄성계수(E) 68.9GPa, 밀도(ρ) 2770kg/m^3, 포아송비(ν) 0.33이다. 문제의 정식화는 식(3)과 같다.

Find b_i

to minimize mass (kg)

subject to $-275.8\text{MPa} \le \sigma_j \le 275.8\text{MPa}$

$$(j = 1,2,\cdots,25) \qquad (3)$$

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. 설계변수의 수가 7개이고, 수준 수는 3인 점을 감안하여 $L_{18}(2^1\times3^7)$ 표준 직교배열표를 선택하여 설계변수를 배치하였고 SOAGA를 제외한 알고리듬들에 대해서는 같은 초기 설계변수를 설정하였다.

Table 4에서 보듯이 최적값은 SOAGA와 MIGA

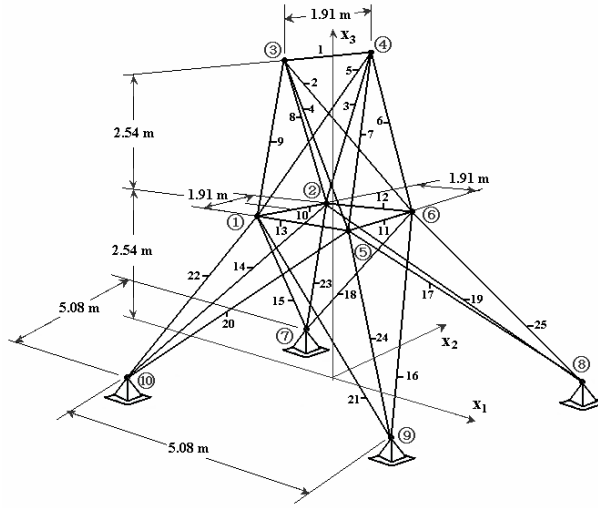

Fig. 6 Twenty-five member transmission tower

의 경우 모두 비슷한 함수 값을 보였지만 함수 계산 횟수에서 많은 차이를 보이고 있으며, SOA의 경우 초기값의 의존도가 높아 사용자에 의해 주어진 초기값에 따라 결과값이 다르게 나타날 수 있는 단점이 있다. 그러나 SOAGA의 경우 초기 모집단으로부터 진화를 통해 국부해 공간을 선정하고 직교배열표의 반복 사용을 통한 국부해 탐색을 행함으로써 유전 알고리듬에 비하여 현저히 적은 함수 계산 횟수와 잠재적인 우수해 포함 영역으로의 이동을 통한 탐색을 통해 주어진 초기치로부터 얻어진 SOA의 해보다 개선된 우수한 해를 찾아낼 수 있었다.

Fig. 7은 20회의 반복횟수에 대한 해를 나타내고 있다. 25부재 트러스의 경우 10부재 트러스와 달리 함수값의 분포 영역이 대단히 넓고, 그에 따라 수렴특성을 뚜렷이 나타내는 경향을 보이고

Table 3 Nodal load components for twenty five bar truss

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

method	Mass (kg)	no. of function evaluation	optimum design variables (m^2)							Feasibility
			b_1	b_2	b_3	b_4	b_5	b_6	b_7	
SOA	4243	172	0.044	0.024	0.024	0.0094	0.0074	0.0034	0.044	feasible
ASA	5450	262	0.064	0.044	0.0094	0.0054	0.024	0.0074	0.044	feasible
MIGA	4570	2660	0.044	0.044	0.024	0.0094	0.0054	0.0006	0.044	feasible
SOAGA	4058	1290	0.044	0.024	0.024	0.0054	0.0074	0.0034	0.044	feasible

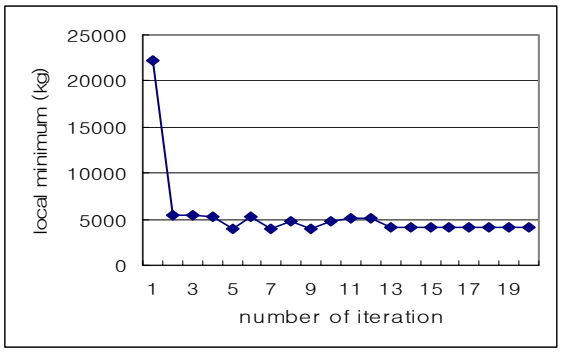

Fig. 7 History of the local minimum for each iteration numbers for twenty five bar truss

있다. 따라서 유전 알고리듬의 염색체의 수렴 정도, 순차적 알고리듬을 통해 얻어진 최종 결과의 수렴 정도를 종료조건으로 사용할 수 있을 것으로 판단된다.

5. 결론

이산공간에서 보다 효율적으로 최적 해를 구하기 위하여 국부해 탐색능력이 우수한 순차적 알고리듬과 전역해 탐색능력이 우수한 유전 알고리듬의 장·단점을 상호 보완한 직교배열표와 유전 알고리듬을 이용한 순차적 알고리듬을 제안하였다. 순차적 알고리듬은 유전 알고리듬의 수행 시 요구되는 비용과 시간을 최소화 하였으며, 유전 알고리듬은 순차적 알고리듬의 국부탐색을 위한 국부 해 영역 선택에 있어 높은 적합도를 갖는 잠재 해 영역을 선택 함으로써 확률적으로 개선된 해를 찾는 방향특성을 갖게 하였다. 또한 수학적, 구조적인 예제 등, 전역해 탐색 사례연구를 통하여 제안한 알고리듬의 우수함을 증명하였다.

또한 유전 알고리듬의 전역 탐색 특성에 의해 전역해 영역으로의 확장과 함께 비가용 영역과 가용 영역의 이동이 가능하게 되었으며, 동일 국부해 영역에 존재하는 개체들 중 적합도가 높은 개체를 선택함으로써 순차적 알고리듬을 통한 국부해 탐색 시 알고리듬의 반복 횟수를 줄일 수 있는 부가적인 효과를 얻을 수 있었다.

SOAGA에 사용된 유전 알고리듬은 일반적으로 많이 사용되는 단순 유전 알고리듬(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.

적용사례의 선정 및 파라미터 설정은 순차적 알고리듬 개발 시 사용되었던 예제를 활용하였으며, 개선된 해를 획득할 수 있는 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. Arora, and 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) 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.
(6) Gutkowski, W., 1997, Discrete Structural Optimization, Springer-Verlag, New York.
(7) Pham, D.T.and Karaboga, D., 2000, Intelligent Optimization Techniques, Springer-Verlag, New York.
(8) Osman, I.H. and Kelly, J.P., 1996, Meta-Heuristics: Theory & Applications, Kluwer Academic Publishers, Boston.
(9) Kim, Y.K., Yoon, B.S., Lee, S.B., 1999, Meta-Heuristic, Youngji. (in Korean)
(10) Gen, M. and Cheng, R., 2000, Genetic Algorithms and Engineering Optimization, John Wiley & Sons, Inc., New York.
(11) Jin, G.K., 2002, Genetic Algorithm and Applications, Kyowoo. (in Korean)
(12) Gong, S.K., Kim, I.T., Park, D.H., Park, J.Y. and Shin, Y.H., 1997, Introduction of Genetic Algorithm, Jinyoung. (in Korean)
(13) Hwang, H.S., 2002, Evolution Design and Calculating using Computer, Naeha. (in Korean)
(14) Park, S.H., 1990, Applications of the Design of Experiments, Youngji. (in Korean)
(15) Park, S. H., 2002, Modern Design of Experiments, Minyoungsa. (in Korean)
(16) Cezary Z. Janikow, An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithm, Proceedings of the fourth international conference on GA.
(17) G.S. Peace, 1995, Taguchi Methods: A Hands-On Approach, Addison Wesley, Massachusetts.