한국정밀공학회지 제20권 제11호 (2003년 11월)
Journal of the Korean Society of Precision Engineering Vol. 20, No. 11. November 2003.

유전자 알고리듬을 이용한 대량의 설계변수를 가지는 문제의 최적화에 관한 연구

이원창#, 성활경*

A Study of A Design Optimization Problem with Many Design Variables Using Genetic Algorithm

Won Chang Lee#, Hwal Gyeong Seong*

ABSTRACT

GA(genetic algorithm) has a powerful searching ability and is comparatively easy to use and to apply as well . By that reason, GA is in the spotlight these days as an optimization skill for mechanical systems.[1] However, GA has a low efficiency caused by a huge amount of repetitive computation and an inefficiency that GA meanders near the optimum. It also can be shown a phenomenon such as genetic drifting which converges to a wrong solution.[8] These defects are the reasons why GA is not widdy applied to real world problems. However, the low efficiency problem and the meandering problem of GA can be overcomed by introducing parallel computation[7] and gray code[4], respectively. Standard GA(SGA)[9] works fine on small to medium scale problems. However, SGA done not work well for large-scale problems. Large-scale problems with more than 500-bit of gene's have never been tested and published in papers. In the result of using the SGA, the powerful searching ability of SGA doesn't have no effect on optimizing the problem that has 96 design valuables and 1536 bits of gene's length. So it converges to a solution which is not considered as a global optimum. Therefore, this study proposes ExpGA(experience GA) which is a new genetic algorithm made by applying a new probability parameter called by the experience value. Furthermore, this study finds the solution throughout the whole field searching, with applying ExpGA which is a optimization technique for the structure having genetic drifting by the standard GA and not making a optimization close to the best fitted value. In addition to them, this study also makes a research about the possibility of GA as a optimization technique of large-scale design variable problems.

Key Words : GA(Genetic algorithm,유전자 알고리듬), ExpGA(Experience GA,경험치를 도입한 GA), Gray code(비가중 코드의 일종), SGA(standard GA,표준 GA)

1. 서론

유전자 알고리듬은 자연생태계의 적자생존 원리를 모델로 개발된 최적설계용 알고리듬이다. 유

접수일: 2002년 12월 20일; 게재승인일: 2003년 8월 9일
교신저자: 창원대학교 기계공학과 대학원
Email: rainman@sccr.changwon.ac.kr Tel:(055) 263-4956
* 창원대학교 기계공학과

전자 알고리듬은 탐색 공간전역에 분포된 유전자의 환경 적합도(Fitness)를 평가하여 우수한 개체만을 선택하여 다음세대를 구성하는 개체의 복제에 참여시킴으로써 이전 세대보다 더욱 진화된 우수한 특성으로 이루어진 개체를 구성하는 과정을 되풀이하여 최적해를 찾는 방법이다. 현재 널리 사용되어지는 2진 코드를 사용하는 기존의 표준 유전자 알고리듬[1]에서는 설계변수의 수가 많아지면 개체의 유전정보가 길어지게 되어 일반적으로 사용하는 교배(Crossover)와 돌연변이와 같은 유전자 조작만으로는 유전정보의 충분한 변화를 일으킬 수 없고 따라서 탐색속도가 현저히 떨어지게 된다. 실험에 의하면 유전자의 길이가 500 비트(bit)이상이 되면 조기에 국소해에 수렴하여 더 이상 전역해의 탐색이 이루어지지 않는 경향을 보인다.[8] 이러한 유전자 알고리듬의 특성으로 많은 설계 변수를 가지는 최적화 문제에는 적용되어 지지 않고 있으며 이러한 문제를 해결하기 위한 방법으로 설계 변수를 몇 개의 그룹으로 그룹화 하여 실제 유전자를 구성한 설계변수의 수를 줄여 개체의 유전자 길이를 짧게 하여 사용하는 기법이 많이 사용되고 있다.[7,8] 하지만, 그룹화를 통하여 설계변수의 수를 줄였음에도 불구하고 설계변수의 수가 많을 경우 그로인해 유전자의 길이가 아주 길어져 버릴 경우에 대한 연구결과는 현재 어디에도 발표된 사례가 없으며 시도되고 있지도 않다. 본 연구에서는 기존의 표준 유전자 알고리듬[1,4]을 개선하고 경험치를 도입하는 새로운 기법을 적용한 ExpGA(경험치 GA)를 제안하며 많은 설계 변수를 가지는 문제에 ExpGA를 적용하여 몇 번의 실험으로 개체의 유전자 길이가 아주 길어 기존의 표준 유전자 알고리듬으로는 전역최적화가 이루어지지 않던 최적화 문제에 최적해를 구하고자 한다. 사용된 200멤버 2차원 트러스 구조물(Two hundred-member planar truss structure)[2]은 96개의 설계변수를 갖으며 한 개의 변수 당 16비트로 표현되어 한 개체의 총 유전정보의 길이가 1536비트이다.

2. 개선된 유전자 알고리듬

2.1 기존의 표준 유전자 알고리듬

유전자 알고리듬은 자연계의 적자 생존법칙을 응용한 탐색 알고리듬으로써 초기에 적당한 크기의 모집단을 생성시킨 후 모집단에 속한 각 개체에 대하여 주어진 환경에 대한 적합도(Fitness)를 평가함으로써 개체의 진화형태와 모집단의 진화방향이 결정된다. 이때 다음 세대를 구성하기 위한 조작으로 교차와 돌연변이 조작을 가하여 부모 세대의 우수한 형질을 이어 받음과 동시에 국소해를 벗어나 전역해를 찾는 시도를 행한다.[9] 기존의 표준 유전자 알고리듬 기본 조작인 교차와 돌연변이만으로는 유전자의 길이가 아주 길어져 버릴 경우 즉, 설계 변수의 수가 많아져 버릴 경우 충분한 탐색을 하지 못하고 국소해에 수렴해 버린다.[8]

2.2 그레이코드(Gray Code) 유전자 알고리듬

본 연구에서는 유전정보를 표시하기 위하여 그레이 코드를 사용하였다.[4] 그레이 코드는 일정한 자리 값이 없는 코드로 한 수에서 다음 수로 진행하는데 있어 인접 코드간 오직 하나의 비트만이 변화하도록 비트를 부여한 코드이다. 이러한 특성으로 인해 음성이나 영상의 경우에 발생되는 아날로그 신호를 컴퓨터가 처리하기 위한 디지털신호로 변환하거나 I/O(input/output)장치를 제어하는 코드로 사용되어져 왔다. 예를 들면 I/O 장치로부터 입력되는 신호의 크기가 7에서 8로 변할 때 BC(binary code)에서는 0111 -> 1000로 네 개의 비트가 한 번에 모두 변한다. 이런 경우 제일 좌측의 비트가 변화하는 시간이 다른 비트보다 느리다면 0000으로 인식될 가능성이 높다. 따라서, 이러한 경우 그레이 코드를 사용하면 유리하다. 즉, 그레이 코드에서는 같은 변화가 0100 -> 1100 으로 한 비트의 변화만으로 가능하기 때문이다. 이러한 연속적으로 변화하는 성질을 이용하면 유전자 알고리듬을 이용한 기계 구조물의 최적 설계 문제에서 인접한 두 설계치의 경계 부분에 인접한 그레이 코드를 할당하여 두 부분을 구분 짓는 경계선에서 탐지를 행하면 2진코드를 쓸 경우 보다 훨씬 모호성이 감소한다.

2.3 변수별 교차(Crossover per Variable)[10]

설계 변수가 많아 유전정보가 긴 경우 기존의 2점 혹은 3점 교차로는 유전정보의 충분한 변화를 일으키기 어렵다. 교차를 통해 부모로 선택된 두 개체의 유전자를 효과적이면서 다양하게 자식 세대에 물려주기 위하여 설계변수별 교차를 도입하였다. 변수별 교차는 설계변수의 차례대로 (i) 해당

변수의 교차여부를 결정하고, (ii) 교차가 결정되면 몇 점 교차 할 것인가를 결정하고, 그리고 (iii) 교차점의 위치를 결정한다. 모든 결정에는 난수발생기(Random Number Generator)가 사용된다.

2.4 변수별 돌연변이(Mutation per Variable)

기존의 표준 유전자 알고리듬에 사용되는 돌연변이 기법은 설계변수가 많아 유전자의 길이가 길어질 경우 단지 한 개의 Bit에 돌연변이를 일으키는 것만으로는 국소해에 수렴한 개체를 탈출시키기가 쉽지 않았다.[10] 본 연구에서는 그래이 코드가 가지는 이점을 활용하여 변수별 돌연변이를 도입하였다. 설계변수 별로 확률에 의해 (i) 돌연변이 여부를 결정한 후, (ii) 변수별로 선택된 임의의 비트를 바꾸어 줌으로써 보다 많은 변화를 줄 수 있도록 하였다.

2.5 경험치를 도입한 유전자 알고리듬 (Experience genetic algorithm : ExpGA)

표준 유전자 알고리듬은 일단 생성된 개체를 해석, 평가한 후 그 다음세대를 생성하는데 참여 또는 도태 시킬 것인가를 결정하고 우수한 개체는 보다 많은 교차 기회를 가지는 반면 덜 우수한 개체는 교차의 기회를 상대적으로 적게 부여 받음으로써 자연도태가 이루어진다.[5] 이 때 개체의 유전자 조작은 교차와 돌연변이에 의해서만 이루어지며 생성된 개체는 다음의 세대를 생성하기까지는 일체의 변화를 줄 수 없었다. 하지만, 실제 자연계에서는 한 개체가 태어나서 태어날 때의 모습 그대로 평가받고 도태되는 경우는 거의 없다고 해도 과언이 아니다. 비록, 처음 태어난 상태는 환경에 비교적 덜 적합하더라도 여러 가지 경험을 통하여 주어진 환경에 다른 어떤 개체보다 더 적응성이 뛰어난 개체로 스스로를 변화 시킬 수 있으며, 이러한 개체 개개의 경험은 우연히 혹은 스스로의 의도적 노력으로 자신의 몸에 축적한다. 그렇게 한 후 각 개체들은 소멸하기 전 자신의 유전자를 복제하는 노력을 기울이는 것이다. 물론 이때 평가의 기준은 태어났을 때의 모습이 아니라 경험을 축적한 최후의 형질로 평가를 받게 되는 것이다. 설계변수가 대단히 많아 이를 표현한 유전자 길이가 아주 길어 버릴 경우 기존의 유전자 알고리듬의 기본 조작인 교차와 돌연변이만으로는 전역 최적치에 도달할 만큼의 충분한 변화를 줄 수 없음으로 유전자 알고리듬의 최대 장점인 전역최적치를 찾지 못하고 국소최적치에 수렴하는 문제점을 가지고 있었다.[8] 본 연구에서는 자연계의 경험 축적을 통한 자기 발전 과정을 경험치와 경험확률이라는 개념을 도입하여 전산 프로그램으로 모델링하였다. 최초 생성 유전자에 확률적 변화를 주어 개체 자체의 성장 가능성을 부여하고 새로운 개체 생성에 참여 할 때의 평가 기준은 개체의 최종 성장이 끝난 상태에서 평가되고 이를 바탕으로 교차의 기회를 부여 받도록 하였다. 이러한 조작을 통한 장점은 개체의 유전자 전 영역을 대상으로 유전자 비트(Bit)를 확률적으로 변화시킴으로써 교차와 돌연변이만으로는 변화를 주기 힘든 영역까지도 다양하게 변화를 줄 수 있으며 이를 통해 새로운 우수한 유전정보를 가지는 개체를 찾을 수 있다. 이렇게 찾아진 개체는 다음 세대를 구성하기 위한 부모 개체로 교차에 참여 함으로써 유전자가 아주 길어질 경우에도 표준 유전자 알고리듬으로는 찾을 수 없었던 전역최적화가 이루어지도록 한 것이다. Fig. 1은 이러한 과정을 간단히 도식화한 것이다. F_{ori}는 최초 생성 개체의 적합도이며, $F_{\exp}$는 경험유전정보를 가진 개체의 적합도이다. $p_{\exp}$는 경험 축적 확률이며, r은 난수 발생기에 의해 발생된 0.0 ~ 1.0 사이의 값이다. 경험 확률에 따라 경험치를 축적할 수 있는 개체의 최대 경험축적횟수는 경험치를 도입하여 제한하고 있다. 경험치는 동일한 개체가 경험을 축적할 수 있는 최대 경험 횟수로 0보다 큰 정수를 사용한다. 여기서, 경험 축적 확률이 너무 크면 모든 개체가 경험치를 생성할 수 있으며, 경험축적횟수가 너무 많으면 한 세대의 해석 시간이 너무 길어짐으로 적절한 값을 선택하여야 한다. 본 논문에서는 몇 번의 예비 탐색으로 경험축적확률 값으로 0.1을 경험치로 10을 적용하여 최대 10번까지 한 개체의 유전자에 변화를 줄 수 있도록 하였다. (i)먼저 한 개체를 대상으로 0.0 ~ 1.0 사이의 난수를 발생시켜 발생된 난수가 경험축적확률보다 작다면 (ii)그 개체는 경험축적을 위해 설계변수 변수만큼의 난수를 발생 시켜 (iii)경험축적확률보다 작은 난수가 발생되는 변수에 해당하는 유전자의 임의 비트를 0이며 1로 1이면 0으로 변화 시킨다. (iv)이렇게 변화를 준 새로운 개체가 경험치가 축적된 개체로 개체의 적합도 평가과정을 거친 후 이전 개체와 우열을 비

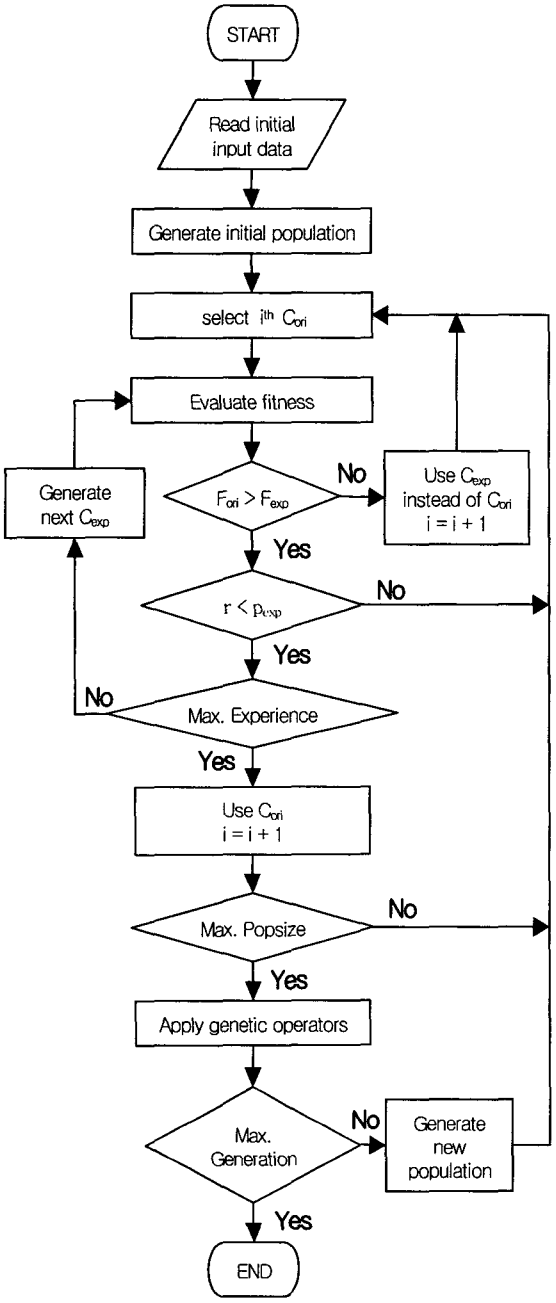

Fig. 1 Flowchart to illustrate the main algorithm used by the new method proposed in this paper

교한다. 만약 경험치가 축적된 개체가 이전 개체보다 우수하지 않다면 새로운 개체는 버려지고 다시 두 번째 경험축적을 위한 난수를 발생 시켜 앞에서 설명된 (i)~(iv)의 과정을 최대 경험치로 설정된 횟수만큼 반복 함으로써 개체의 형질을 개선 시킬 노력을 기울일 수 있다. (v)만약 경험치로 설정된 횟수만큼 진행되기 전에 이전 개체보다 우수한 형질을 가지는 개체가 발견되면 그 즉시 순수한 개체를 험치가 축적된 개체로 대체하고 그 다음의 새로운 순수한 개체를 평가하고 앞에 설명된 (i)~ (v)와 같은 조작을 되풀이 한다. Fig. 1에서 C_{ori}는 경험치가 없는 순수한 개체를 나타내며 $C_{\exp}$는 경험치가 축적된 개체이다. $F_{\exp}$가 F_{ori}보다 크면 해당 개체는 $C_{\exp}$가 C_{ori}를 대체하게 된다.

3. 트러스 구조물의 최적화

본 논문에서 제안한 ExpGA의 실효성을 검토하기 위하여 동일한 조건 하에서 (i) 설계민감도에서 구한 참고문헌의 최적값[2]과 (ii) 새로 제안한 유전자 알고리듬을 적용한 결과(ExpGA)와 (iii) 기존의 2점 교차와 돌연변이 조작을 이용한 표준 유전자 알고리듬을 적용하여 구한 값(Standard GA)을 비교,검토하였다.

3.1 트러스 구조물 최적화

200요소 2차원 트러스 구조물(Two-hundred member plane truss)[2]에 대한 예제는 Fig. 2와 같으며 설계문제는 다음과 같다. (i) 목적함수로는 구조물의 총질량이 사용되었고, (ii) 제한조건식은 구조물에 정하중이 작용 될 때의 변위량(displacement)과 응력(stress)[3]으로 하였다. 즉,

Minimize

$$f(\vec{x}) = \sum_{i=1}^{200} \rho A_i L_i \qquad (1)$$

Subject to

$$g_{j1}(\vec{x}) = |u_j| - \overline{u_j} \le 0$$

$$g_{j2}(\vec{x}) = |v_j| - \overline{v_j} \le 0 \qquad (2)$$

$$g_{j3}(\vec{x}) = |w_j| - \overline{w_j} \le 0$$

$$g_i(\vec{x}) = |\sigma_i| - \overline{\sigma_i} \le 0 \qquad (3)$$

(단, i=1,· · ·,200 , j = 1,· · ·,77)

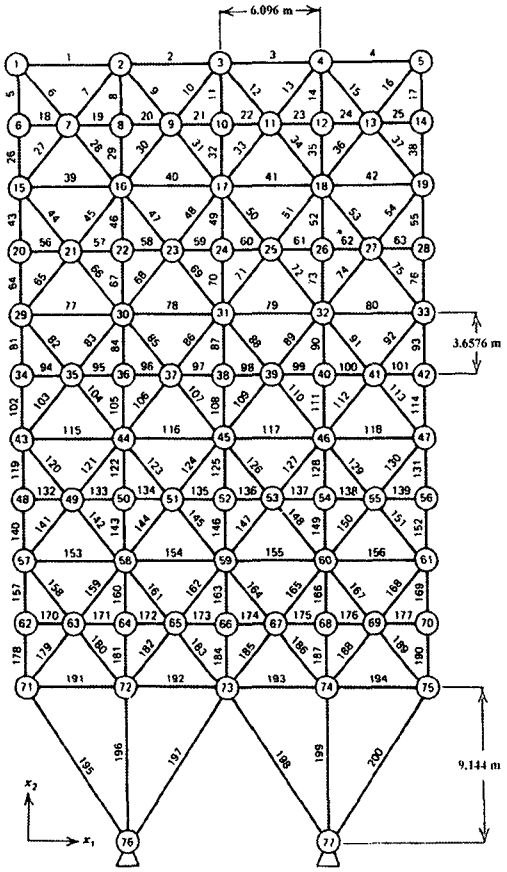

Fig. 2 Two-hundred member plane truss[2]

이다. 이때 u_j, v_j, w_j는 j번째 절점에서의 x-, y-, 및 z-방향 변위이며 $\overline{u_j}$, $\overline{v_j}$, $\overline{w_j}$는 각 방향 변위의 상한치이다. 그리고, σ_i는 i번째 부재에의 응력을 $\overline{\sigma_i}$는 i번째 부재의 응력상한치이다. 또한 ρ는 재질의 밀도이고, A_i 및 L_i는 i번째 부재의 단면적과 길이이다. 위의 최적화 문제는 유전자 알고리듬을 적용시키기 위하여 1차적으로 Unconstrained Optimization problem 으로 식(4)와 같이 변환된다.

$$F' = f + r\left[\sum_{j=1}^{77}\left(g_{j1}^2 + g_{j2}^2 + g_{j3}^2\right) + \sum_{i=1}^{200}\left(g_i^2\right)\right] \tag{4}$$

여기서, r은 벌칙 parameter이고, $\{g\}$ 함수는 아래와 같이 정의된다.

$$\{g\} = \begin{cases} 0, & g \le 0 \\ g, & g > 0 \end{cases} \tag{5}$$

F'은 최적화가 될 수록 작은 값을 취하게 되나, 유전자 알고리듬에서 판단의 기준이 되는 적응도(Fitness)는 최적화 될수록 크게 된다. 따라서 식(6)과 같이 적응도를 정의한다.

$$F = \frac{C_1}{C_2 + F'} \tag{6}$$

단, C_1및 C_2는 적응도의 크기를 조절하기 위해 사용되는 상수이며 본 연구에서 C_1으로는 10^7이 사용되었고 C_2로는 1이 사용되었다. 하중 및 물성(material property)으로는 Table 1이 사용되었다.

200개의 트러스 부재는 Table 2와 같이 96개의 그룹을 형성하고 각 그룹을 설계 변수로 하여 각 변수 당 16비트(bit)로 Encoding하여 총 1536비트(bit)의 유전자 정보를 갖는 설계문제로 표현하였다. 이 경우 각 그룹에서 선택될 설계변수는 몇 번의 예비 해석[6]을 통해 Table 1에서 언급한 $6.4516\text{x}10^{-5}$ m² 의 하한값과 Table 2와 같은 상한값 범위의 연속적인 값으로부터 선택 된다.

Table 1 Design data for 200-member truss[2]

Loading condition	Node points	Load and direction of load
step 1	1,6,15,20,29,34,43,48,57,62,71	4.45 kN acting in positive x_1- *dircetion*
step 2	1,2,3,4,5,6,8,10,12,14,15,16,17,18,19,20,22,24, ...,71,72,73,74,75	44.5 kN acting in negative x_2- *dircetion*
step 3	Loading Condition 1 and 2 acting together.	
Modulus of elasticity = 206.9 GPa Material density = 76.85 kN/m³ Lower limit on cross-sectional areas = 6.4516×10^{-5} m² Displacement limits at all nodes and in all directions = ± 0.0127 m Allowable stress limits = ± 206.85MPa		

Table 2 Strings assigned to design group and selectable sizing variables(cross-sectional area) for each topology[2]

Design group	Member Number	string (bit)	Cross sectional area Upper bound(m^2)
1	x_1,x_4	16	0.00387096
2	x_2, x_3	16	0.00387096
3	x_5, x_{17}	16	0.00774192
4	x_6, x_{16}	16	0.00387096
5	x_7, x_{15}	16	0.00387096
6	x_8, x_{14}	16	0.00387096
7	x_9, x_{13}	16	0.00387096
8	x_{10}, x_{12}	16	0.00387096
9	x_{11}	16	0.00387096
10	x_{18}, x_{25}, x_{56}, x_{63}, x_{94}, x_{101}, x_{132}, x_{139}, x_{170}, x_{177}	16	0.00387096
11	x_{19}, x_{20}, x_{23}, x_{24},	16	0.00387096
12	x_{21}, x_{22}	16	0.00387096
13	x_{26}, x_{38}	16	0.00774192
14	x_{27}, x_{37}	16	0.00387096
15	x_{28}, x_{36}	16	0.00387096
16	x_{29}, x_{35}	16	0.00387096
17	x_{30}, x_{34}	16	0.00387096
18	x_{31}, x_{33}	16	0.00387096
19	x_{32}	16	0.00387096
20	x_{39}, x_{42}	16	0.00387096
21	x_{40}, x_{41}	16	0.00387096
22	x_{43}, x_{55}	16	0.00774192
23	x_{44}, x_{54}	16	0.00387096
24	x_{45}, x_{53}	16	0.00387096
25	x_{46}, x_{52}	16	0.00387096
26	x_{47}, x_{51}	16	0.00387096
27	x_{48}, x_{50}	16	0.00387096
28	x_{49}	16	0.00387096
29	x_{57}, x_{58}, x_{61}, x_{62}	16	0.00387096
30	x_{59}, x_{60}	16	0.00387096
31	x_{64}, x_{76}	16	0.00774192
32	x_{65}, x_{75}	16	0.00387096
33	x_{66}, x_{74}	16	0.00387096
34	x_{67}, x_{73}	16	0.00387096
35	x_{68}, x_{72}	16	0.00387096
36	x_{69}, x_{71}	16	0.00387096
37	x_{70}	16	0.00774192
38	x_{77}, x_{80}	16	0.00387096
39	x_{78}, x_{79}	16	0.00387096
40	x_{81}, x_{93}	16	0.00774192
41	x_{82}, x_{92}	16	0.00387096
42	x_{83}, x_{91}	16	0.00387096
43	x_{84}, x_{90}	16	0.00774192
44	x_{85}, x_{89}	16	0.00387096
45	x_{86}, x_{88}	16	0.00387096
46	x_{87}	16	0.00774192
47	x_{95}, x_{96}, x_{99}, x_{100}	16	0.00387096
48	x_{97}, x_{98}	16	0.00387096
49	x_{102}, x_{114}	16	0.01096772
50	x_{103}, x_{113}	16	0.00387096
51	x_{104}, x_{112}	16	0.00387096
52	x_{105}, x_{111}	16	0.00774192
53	x_{106}, x_{110}	16	0.00387096
54	x_{107}, x_{109}	16	0.00387096
55	x_{108}	16	0.00774192
56	x_{115}, x_{118}	16	0.00387096
57	x_{116}, x_{117}	16	0.00387096
58	x_{119}, x_{131}	16	0.00387096
59	x_{120}, x_{130}	16	0.00387096
60	x_{121}, x_{129}	16	0.00387096
61	x_{122}, x_{128}	16	0.00774192
62	x_{123}, x_{127}	16	0.00387096
63	x_{124}, x_{126}	16	0.00387096
64	x_{125}	16	0.00774192
65	x_{133}, x_{134}, x_{137}, x_{138}	16	0.00387096
66	x_{135}, x_{136}	16	0.00387096
67	x_{140}, x_{152}	16	0.00774192
68	x_{141}, x_{151}	16	0.00387096
69	x_{142}, x_{150}	16	0.00387096
70	x_{143}, x_{149}	16	0.00774192
71	x_{144}, x_{148}	16	0.00387096
72	x_{145}, x_{147}	16	0.00387096
73	x_{146}	16	0.00774192
74	x_{153}, x_{156}	16	0.00387096

Design group	Member Number	string (bit)	Cross sectional area Upper bound(m^2)
75	x_{154}, x_{155}	16	0.00387096
76	x_{157}, x_{169}	16	0.00774192
77	x_{158}, x_{168}	16	0.00387096
78	x_{159}, x_{167}	16	0.00387096
79	x_{160}, x_{166}	16	0.01096772
80	x_{161}, x_{165}	16	0.00387096
81	x_{162}, x_{164}	16	0.00387096
82	x_{163}	16	0.00774192
83	x_{171}, x_{172}, x_{175}, x_{176}	16	0.00387096
84	x_{173}, x_{174}	16	0.00387096
85	x_{178}, x_{190}	16	0.00774192
86	x_{179}, x_{189}	16	0.00387096
87	x_{180}, x_{188}	16	0.00387096
88	x_{181}, x_{187}	16	0.01096772
89	x_{182}, x_{186}	16	0.00387096
90	x_{183}, x_{185}	16	0.00387096
91	x_{184}	16	0.00774192
92	x_{191}, x_{194}	16	0.00774192
93	x_{192}, x_{193}	16	0.00387096
94	x_{195}, x_{200}	16	0.01096772
95	x_{196}, x_{199}	16	0.0129032
96	x_{197}, x_{198}	16	0.00774192
Total		1536 bit	

특히 초기 모집단의 각 개체는 난수 발생기(random number generator)에 의해 생성되며 모집단의 크기는 150으로 하였다. 유전자 연산을 위해 교차확률(crossover probability)은 0.85로서 2점 교차를 사용하였으며 돌연변이 확률(Mutation probability)은 0.1로 하여 표준 유전자 알고리듬[5]으로 유전자를 생성되도록 하여 최적화를 시도하였다. 그리고, 앞의 기존 유전자 알고리듬을 적용한 문제와 같은 환경에 경험 축적확률은 0.1, 경험치는 10으로 하여 자기 개발 가능성을 부여하였다. 위의 확률 값들은 통상적으로 사용되어지는 범위 안에서 몇 번의 예비 테스트를 통하여 선택된 값들이다.

4. 결과 및 고찰

기존의 유전자 알고리듬을 이용한 구조물의 최적해는 314세대, 본 논문에서 제안한 ExpGA를 적용한 200요소 2차원 트러스 구조물의 최적해는 275세대만에 나타 났으며 적합도와 목적함수의 경향은 Fig. 3 및 Fig. 4와 같다. 그리고 벌칙 함수는 두 방법 모두 0으로 수렴되어 모든 구속 조건을 만족하는 상태였다.

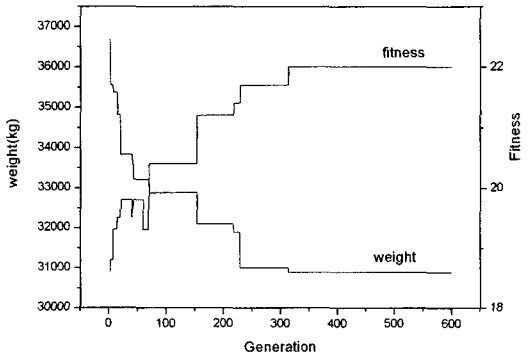

Fig. 3 Generaion-history of "weight and fitness" design for two-hundred member plane truss by previous GA

Fig. 3을 보면 기존의 GA를 이용한 결과에서는 적합도가 전형적인 엘리트전략을 사용한 양상을 보이며 높아지다 22.04부근에서 더 이상 변화를 보이지 않고 있으며 목적함수인 시스템의 질량도 30892.67 kg으로 수렴되었다.

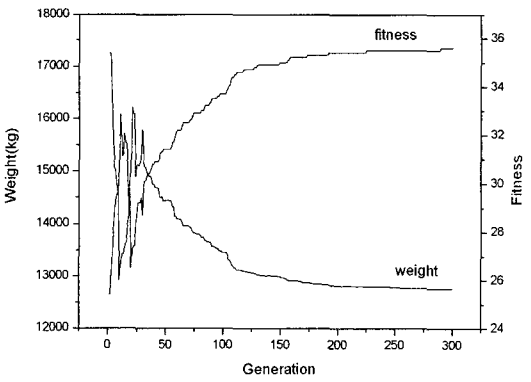

Fig. 4 Generaion-history of "weight and fitness" design for two-hundred member plane truss by ExpGA

Fig. 4에서는 ExpGA를 적용하여 구조물을 최적화한 결과를 나타낸 것으로 적합도가 꾸준히 높아지다가 35.54부근에 수렴하고 있으며 목적함수는 꾸준히 낮아지다가 12769.07kg에 수렴 되었다. 최적치에 도달했을때의 설계 변수는 Table 3과 같다.

Table 3 Comparison of topology results

Design group	Optimum topology (m^2)		
	Ref.[2]	SGA	ExpGA
1	0.000121161	0.000513083	0.000327741
2	0.000064516	0.000191967	0.000250193
3	0.003085929	0.006086805	0.002667801
4	0.000109871	0.000820166	0.000220645
5	0.000064516	8.97593E-05	8.15482E-05
6	0.001513674	0.002609724	0.001562319
7	0.000121032	0.008639311	7.90966E-05
8	0.000064516	0.005883701	0.000124774
9	0.001858641	0.002368172	0.002501608
10	0.000064516	0.000246121	7.58708E-05
11	0.000064516	0.00705698	6.74192E-05
12	0.000064516	0.011969317	8.81289E-05
13	0.004372056	0.012384246	0.004314572
14	0.000064516	0.010865192	0.000236516
15	0.000152322	0.004986844	0.00030658
16	0.002137609	0.005542629	0.002565285
17	0.000111742	0.004041991	0.000141677
18	0.000143677	0.012440862	0.000212774
19	0.002675672	0.000843829	0.002839285
20	0.000064516	0.002551148	0.000623418
21	0.000064516	0.003772424	0.000345032
22	0.005244635	0.008279236	0.004797152
23	0.000159742	0.000358766	0.000773095
24	0.000064516	0.009642545	0.000109226
25	0.002851994	0.000381687	0.003063091
26	0.000180774	0.006988414	0.000151548
27	0.000172451	0.002048062	0.000304451
28	0.003092187	0.003585139	0.003635864
29	0.000064516	0.002351716	0.000186967
30	6.4645E-05	0.000158943	0.000112258
31	0.006057343	0.003277175	0.005080377
32	0.000064516	0.003219774	0.000275096
33	0.000216903	0.002543312	0.001007288
34	0.00327309	0.001921311	0.003244832
35	0.000194064	0.008569765	0.000417612
36	0.000199742	0.003832371	0.000168129
37	0.00359638	0.00054762	0.004539152
38	0.000320451	0.00632057	0.000620515
39	0.000249354	0.006133481	0.000151935
40	0.006141665	0.001203512	0.00429683
41	0.000604257	0.003329089	0.001579932
42	0.000064516	0.009248578	9.8645E-05
43	0.004039798	0.001014267	0.004980119
44	0.000226322	0.000946092	0.000118322
45	0.000311935	0.005752248	0.000102903
46	0.003785734	0.011971081	0.004636184
47	0.000064516	0.000891238	0.000240709
48	0.000064516	0.005017014	0.000160451
49	0.006761277	0.00188977	0.00439554
50	7.14837E-05	0.004304308	0.000292516
51	0.000665354	0.004472592	0.001194127
52	0.004400185	0.001151793	0.005504957
53	0.000323354	0.001761256	0.000286193
54	0.000242193	0.006431257	0.000226129
55	0.004178572	0.012143086	0.00470812
56	0.001277868	0.000624023	0.001145998
57	0.000953805	0.000772128	0.000479547
58	0.005906182	0.011341048	0.004021863
59	0.002063157	0.001414307	0.0016749
60	0.000064516	0.000168934	0.001077933
61	0.005823924	0.007477002	0.006826567
62	0.000133806	0.004213408	0.000667612
63	0.000626902	0.006140925	0.001239546
64	0.004215346	0.005201753	0.004567023
65	0.000064516	0.006744707	0.000614257
66	7.8645E-05	0.003963628	0.000311161
67	0.006427342	0.01137729	0.006026246
68	8.6516E-05	0.006370722	0.000271806
69	0.002129028	0.001634505	0.00253135
70	0.006178762	0.004798579	0.005925537
71	0.00063316	0.002656153	0.001768513
72	0.000146387	0.000921016	0.000140451
73	0.004552313	0.002978614	0.005495344
74	0.001645158	0.009209985	0.001180062
75	0.00039187	0.001669964	0.000471096
76	0.004862958	0.00683835	0.002833801
77	0.002659091	0.004498843	0.001670126

Design group	Optimum topology (m^2)		
	Ref.[2]	SGA	ExpGA
78	0.000064516	0.008730408	0.000814644
79	0.008599338	0.01167154	0.010068948
80	0.001205869	0.002386587	0.00105503
81	0.000196451	0.00234623	0.000203419
82	0.004790055	0.004208315	0.00394954
83	0.000064516	0.00803122	0.001455675
84	0.000064516	0.00122467	0.002711349
85	0.005302118	0.008809553	0.002837478
86	0.000064516	0.004544881	0.002259931
87	0.002704253	0.003111634	0.00174819
88	0.008924498	0.006573093	0.010697721
89	0.000216387	0.003859602	7.94837E-05
90	0.001231094	0.000383059	0.00086116
91	0.005086441	0.007811022	0.003161865
92	0.003783799	0.00207353	0.001612771
93	0.002209544	0.002711987	0.00089174
94	0.006874825	0.005753031	0.004743926
95	0.011469009	0.007432728	0.009962045
96	0.004976764	0.004223791	0.004108185
Total weight (kg)	13149.2	30892.67	12772.45

Table 3에는 수렴결과를 비교하기 위하여 참고문헌상에 표기된 최적 결과값을 Ref[2]로 표기 했으며 표준 유전자 알고리듬을 이용한 결과를 SGA로 본 논문에서 제안한 새로운 방법을 도입한 유전자 알고리듬에 의한 결과는 ExpGA로 표기 하였다. 여기서, 참고 문헌의 결과값은 설계민감도 이론을 적용하여 구한 최적값 결과값을 의미한다.

5. 결 론

본 연구에서는 유전자 알고리듬으로는 시도된바 없는 매우 많은 설계변수를 가지는 구조물의 최적화 기법으로 유전자 알고리듬의 적용가능성과 실효성을 알아보기 위해 (i)유전자를 그레이 코드로 표현하고,(ii)변수별 교차와 같이 이미 단편적으로 시도되었던 방법과 (iii)변수별 돌연변이 및 (iv) 경험 확률 및 경험치라는 지금까지 시도된바 없는 새로운 방법을 도입하여 기존 유전자 알고리듬의 탐색 능력을 개선시킨 ExpGA를 96개의 설계변수를 갖는, 따라서 이를 표현하기 위해 1536-Bit의 유전정보를 갖는 최적화 문제에 적용시켜 다음과 같은 결론을 얻을 수 있었다.

1. 설계 변수 96개, 유전자 길이 1536비트의 복잡하고 큰 문제의 최적화 기법으로 ExpGA가 훌륭한 방법임을 알 수 있었으며, 동시에 매우 훌륭한 탐색 능력과 결과를 나타낸다는 것을 알 수 있었다. ExpGA를 이용하여 얻어진 최적화 결과는 기존의 유전자 알고리듬을 이용하여 얻어진 최적화 질량인 30892.67kg보다 58.7%, 참고문헌의 설계민감도에 의해 얻어진 최적화 질량인 13149.2kg보다 2.9% 개선된 12772.45kg의 질량을 가지는 설계치를 얻을 수 있었다. 무엇보다 기존의 유전자 알고리듬으로는 설계민감도 이론을 적용하여 구한 참고문헌상의 최적설계치에도 못 미치는 결과를 보여줌으로써 전역탐색이 이루어지지 않고 있음을 알 수 있었다.

2. 본 논문에서 제안한 ExpGA를 이용하여 최적화를 시도한 결과 2시간 55분만에 참고문헌상의 결과보다 더 감소된 12772.45kg의 질량을 가지는 전역최적해를 찾은 반면 기존의 유전자 알고리듬을 이용한 경우 2시간 37분만에 국소해인 30892.67kg에 수렴을 하였으나 이후 5시간동안 600세대까지 최적해 찾기를 시도하였으나 더 이상 개선이 이루어지지 않아 참고문헌상의 최적 설계치 도달에는 실패하였다. 이상에서 경험확률 개념을 도입한 개선된 유전자 알고리듬은 기존의 일반적인 알고리듬보다 더욱 우수한 전역 탐색 능력을 가진다는 것을 알 수 있었다.

3. 다만 전산효율의 문제점이 지적 될 수 있으나, 병렬화 등을 도입한다면 그다지 큰 제약은 아니라고 판단된다.

앞으로 아주 많은 설계변수를 가짐으로써 500-Bit이상의 유전정보를 가지는 실제적인 문제의 최적화에 적용시켜 보다 용이하게 여러 가지 문제의 최적화에 유전자 알고리듬이 적용될 수 있을 것이다.

참고문헌

1. Baek, W. T., Seong, H. G., "The Optimization of sizing and topology design for drilling machine by genetic algorithm," J. of the KSPE, Vol. 14, No. 12, pp. 25-27, 1997.
2. Haug, E. J., Arora, J. S., "Applied Optimal Design," John wiley & sons, pp. 249-254, 1979.
3. Cook, R. D., Malkus, D. S., and Plesha, M. E., Witt, R. J., "Concepts and Applications of Finite element analysis," John wiley & sons, pp.29-42, 2002.
4. Haupt, R. L., Haupt, S. E., "Practical Genetic Algorithm," John wiley & sons, pp. 88-91, 1998.
5. Goldberg, D. E., "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-wesley publishing company, pp.85-86, 1989.
6. Papadrakakis, M., "Solving Large-scale Problems in Mechanics," John wiley & sons, pp.418-424, 1993.
7. Lee, W. C., Ju, J. H. and Seong, H. G., "A study for improvement effect of paralleled genetic algorithm by using clustering computer system," J. of KSPE, Vol. 20, No. 4, pp.189-196, 2003.
8. Coello, C. A., Christiansen, A. D., "Multiobjective optimization of trusses using genetic algorithms," J. of Computers and Structures, Vol. 75, pp. 647-660, 2000.
9. Kwon, Y. D., Kwon, S. B., Jin, S. B., and Kim, J. Y., "Convergence enhanced genetic algorithm with successive zooming method for solving continuous optimization problems," J. of Computers and Structures, Vol. 81, pp. 1715-1725, 2003.
10. Nagendra, S., Jestin, D., Gurdal, Z., Haftka, R. T., and Watson, L. T., "Improved genetic algorithm for the design of stiffned composite pannels," J. of Computer and structures, Vol. 58, No. 3, pp. 543-555, 1996.