

# 이산화 변수를 사용한 트러스 구조물의 최적 설계

## Optimum Design of Truss Structures Using Discrete Variables

박 성화 \*  
Park, Seong Hwa

이 종권 \*\*  
Lee, Jong Kweon

이 병해 \*\*\*  
Lee, Byung Hai

---

### ABSTRACT

This study presents the applicable possibility of numerical optimization and Genetic Algorithm in the design of truss structures using discrete variables and real constraints. The introduction of Genetic Algorithm in the design of truss structures enables us to do easier formulation and handle discrete variables. To investigate these applicable possibility, the design of 15 - bar truss structures has been studied using GT/STRUDL and Genetic Algorithm and the results of Genetic Algorithm are compared with GT/STRUDL's

---

### 1. 서론

구조 최적설계에 대한 연구는 지난 수십년간 이론적으로 괄목할 만한 성과를 이루어 왔으며, 이론의 실제적인 적용에 대해서도 활발한 연구가 진행되고 있다. 최근 구조물이 대형화되면서 나타나는 대경간 지붕 구조물은 건축 구조물 중에서도 비교적 규칙적인 형상을 지니고 있다. 따라서 해석 프로그램의 입력 데이터를 작성할 때 가장 많은 시간을 소비하는 구조물의 모델링도 대수학 함수를 사용함으로써 몇개의 변수만으로도 가능하게 되었다.[1] 이와같이 트러스 구조물은 해석이나 설계 과정을 일괄적으로 처리하기가 용이하기 때문에 최적 설계에서 상대적으로 많은 연구가 있어왔다.[2]

본 연구에서는 트러스 구조물을 실제 설계 조건(AISC CODE)을 적용하여 최적 설계를 수행하였다. 그리고 최적 설계의 적용에 있어, 기존 최적화 방법의 문제점이 무엇이며 최근 많은 연구가 진행중인 Genetic Algorithm(유전적 알고리즘)의 건축 트러스 구조물에 대한 적용 가능성을 검토하였다. 대경간 지붕 구조물의 최적설계 전단계로서 15-bar 트러스 구조물을 설계 대상 구조물로 정하였고, 이산화된 설계 변수를 사용하여 최적설계를 수행하였다.

### 2. 기존의 최적화 연구와 Genetic Algorithm 의 도입 배경

일반적인 최적화 기법으로는 수치 최적화(numerical optimization)에 대한 연구가 주로 많이 진행되어 왔다. 수치 최적화를 위해 많이 사용되고 있는 프로그램중, IDESIGN[3]과 NASTRAN[4]을 예로 들면 트러스 구조물에 적용할때 다음과 같은 문제점이 제기된다.

첫째, IDESIGN 은 정식화 과정이 복잡하고 많은 시간이 소요된다. 즉 목적함수와 제약조건이

- 
- \* 한양대 건축공학과 석사과정
  - \*\* 한양대 건축공학과 박사과정
  - \*\*\* 한양대 건축공학과 교수

모두 설계 변수로 표현가능해야 하고, 각 설계 변수에 대한 목적 함수와 제약 조건의 편미분을 미리 구해야한다는 문제점이 있다. 현재 실무에서는 대경간 지붕 구조물에 대해 경제적이라는 이유로해서 그룹핑을 하지 않고 설계하고 있다. 그렇기 때문에 많은 부재로 구성된 대경간 지붕 구조물에 적용할 경우 이런 문제점은 더욱 커진다고 할 수 있다.

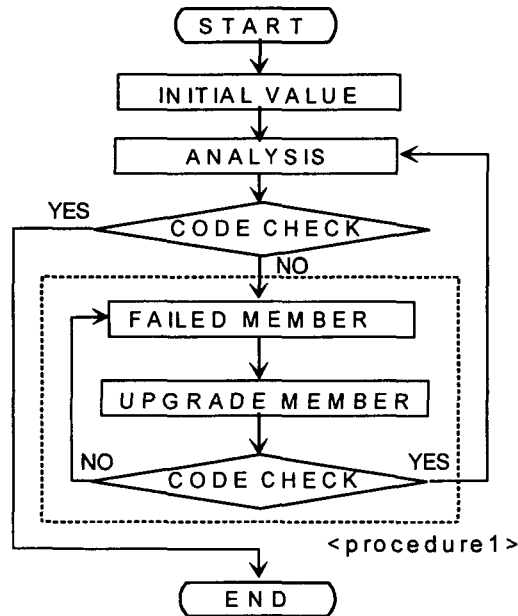
둘째, NASTRAN 은 해석 모듈이 포함되어 있고, IDESIGN 처럼 부프로그램을 작성하지 않아도 되기 때문에 입력 데이터만 형식에 맞게 작성하면 된다. 하지만 제약조건으로 응력을 고려할때 상수값으로만 줄 수 있어 실제 제약 조건식을 표현할 수가 없다.

수치 최적화 기법으로는 이상과 같은 문제점이 있기 때문에 대경간 지붕 구조물에 적용하기가 비효율적이라 판단된다. 따라서 본 연구에서는 목적 함수나 제약 조건의 편미분을 구할 필요가 없고 정식화 과정이 비교적 간단하며 이산화 변수를 직접 고려할 수 있는 Genetic Algorithm 을 도입하고자 한다.

확률론적 최적화 방법인 Genetic Algorithm 의 경우, 트러스 구조물에 대한 기존 연구는 제약조건을 응력 하나로 보고 허용응력도 상수로 가정하여(예를들어,  $\sigma = \pm 25,000 \text{psi}$ ) 설계했기 때문에 실제 구조물의 설계와는 차이가 있다고 볼 수 있다. 따라서 본 연구에서는 실제 트러스 구조물을 설계할때 사용하는 AISC CODE 를 적용하고 이를 일반적인 설계 과정을 통해 얻은 값과 비교하였다. 여기서 한국 기준을 사용하지 않은 이유는 본 논문에서 비교 검토한 설계가 AISC CODE 를 사용하는 GT/STRUDL 에 의한 것이기 때문이다.

### 3. Genetic Algorithm 의 적용 방법

건축이나 토목 구조물을 설계할때는 해석 과정이 반복적으로 사용이 되기 때문에 최적설계에서 이에 대한 고려가 필요하다. 그러므로 최적 설계를 하기 위해서는 새로운 해석 모듈을 만들어 첨가하거나 기존의 프로그램을 이용해야 한다. Genetic Algorithm 도 이러한 문제점을 가지고 있기 때문에, 이를 위해 본 연구에서는 상용 해석 패키지인 GT/STRUDL 을 사용하였다.



[그림 1] 일반적인 설계 흐름도

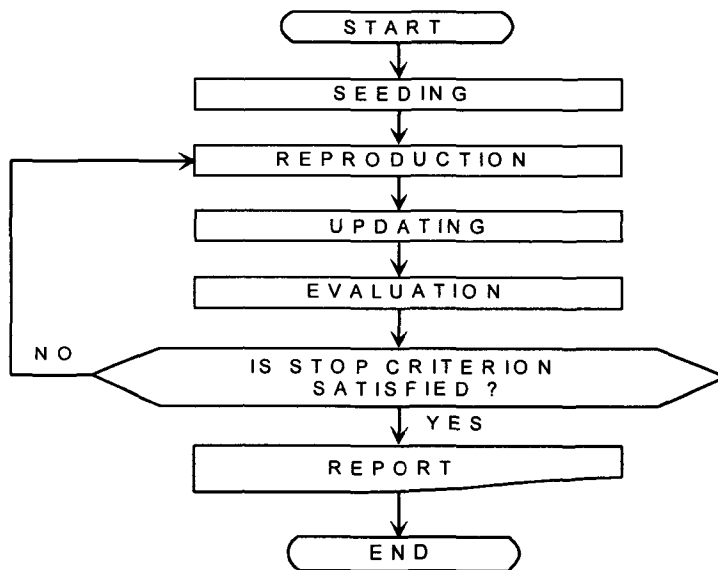
많은 프로그램중 GT/STRUDL을 사용한 이유는 실무에서 트러스 구조물을 해석,설계하는데 많이 이용되고 있고, 또 본 연구에서 실제 제약 조건을 가지고 비교 검토하려는 목적에 부합되기 때문이다. GT/STRUDL의 출력 결과에서 필요한 데이터(예를들어, 단면적, 부재 길이, 단면 2차 모멘트,critical 하중 조합등)를 추출하여 Genetic Algorithm의 목적함수나 제약조건을 정의하는 부분에서 사용할 수 있도록 추가하였다.

기존 트러스 구조물의 최적설계를 살펴보면 10-bar 트러스 경우 iteration이 1000 - 1500 번 정도 수행된 것으로 나타났다.[5] 이는 설계 변수가 발생(generation)될 때마다 해석을 해야 하기 때문이며 구조물이 커질 경우 매번 해석을 수행한다는 것은 비효율적이다. 그러므로 [그림 1]에서와 같이 일반적으로 실무에서 트러스 구조물을 설계할때 사용하는 방법을 채택하였다.

매번 해석을 하지 않고 <PROCEDUREI>에서와 같이 앞에서 수행한 해석 결과의 부재력을 가지고 CODE CHECK 만을 반복하여 만족한 설계 변수를 얻는다. 만족한 설계 변수를 가지고 다시 해석을 하게 되는데 이렇게 되면 부재가 바뀌고 나서의 하중 조건(selfweight)이 변경되지만 이 점은 무시하는 것으로 한다. 이렇게 5-6 번 정도를 반복하게 되면 제약 조건을 만족시키는 설계 변수 값을 찾을 수 있게 된다.

#### 4. Genetic Algorithm

Genetic Algorithm은 최근에 와서야 많은 연구가 진행중인 방법이며, 아직 그 특성과 장단점이 널리 인식되지 못해 부분적으로 활용되고 있는 방법이기도 하다. 사실 이들 방법은 공히 자연계의 진화 적응성을 최적화 방법에 적용하려는 노력에서 고안된 것으로서, 기존의 최적화 방법과는 달리 확률론적 최적화 방법을 사용한다는 특징을 가지고 있다.



[그림 2] Genetic Algorithm의 흐름도

Genetic Algorithm의 흐름도는 [그림 2]와 같고 각 과정을 살펴보면 다음과 같다.

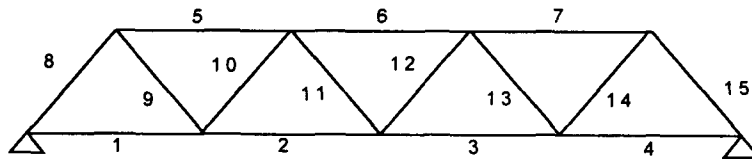
- (a) START
  - 입력 데이터(input file)가 주어지고, 일부 변수들이 초기화된다.
- (b) SEEDING

- (i) 초기 설계 집단을 무작위로(randomly) 발생시킨다.
  - (ii) decoding - 2 진수로 된 문자열(string)을 10 진수로 변환한다.
  - (iii) fitness value 를 계산한다. - 발생된 설계 집단의 적합성을 조사한다.
  - (iv)  $X_{max}$ ,  $X_{min}$ , sumfitness 등의 값을 계산한다.
- (c) REPRODUCTION
- (i) selection - 적합성이 높은 설계 변수를 선택한다.
  - (ii) crossover - 선택된 문자열의 인수들을 교체하여 보다 적합한 문자열을 만든다.
  - (iii) mutation - crossover 에 의해 문자열이 한쪽으로 치우치는 것을 보완한다.
- (d) UPDATING
- 이전의 설계집단을 새로운 설계집단으로 재정의한다.
- (e) EVALUATION
- (b) SEEDING 의 (ii),(iii),(iv) 과정을 재실행한다.
- (f) IS STOP CRITERION SATISFIED?
- 제약 조건을 만족하는지 검토한다.
- (g) REPORT
- 결과를 프린트한다.
- (h) END

## 5. 적용 사례

본 연구에서 적용한 구조물은 15-bar 트러스 구조물이며 그림 3 과 같다. 모든 부재는 트러스 요소로 모델링하였고 지지점은 단순지지로 정의하였다. 부재들에 대한 재료적 성질은 다음과 같다.

- 탄성계수(Young's Modulus):  $2.1 \times 10^6 \text{ kg/cm}^2$
- 포아송비(Poisson's Ratio) : 0.33      • 밀도 (Weight Density):  $7.93 \times 10^{-3} \text{ kg/cm}^3$



[그림 3] 대상 구조물

### 5.1 하중조건

$$\text{LOAD COM1} = 1.0 \times \text{D.L} + 1.0 \times \text{L.L}$$

$$\text{LOAD COM2} = 0.75 \times \text{D.L} + 0.75 \times \text{W.L}$$

고정하중(Dead Load) 과 적재하중(Live Load)의 각 100%를 조합한 경우와, 고정하중과 풍하중(Wind Load)의 각 75%를 조합한 경우, 2 가지에 대해 고려하였다.

### 5.2 정식화 과정

최적화 과정에서 각 트러스 부재의 단면적을 설계 변수로, 전체 중량을 목적함수로 정의하였다. AISC CODE를 제약 조건으로 적용하고 부재 단면적의 상한값과 하한값은 각각 0.2cm<sup>2</sup>와 60cm<sup>2</sup>로 설정하였다.

**Minimize** ;  $F(X_i) = \Sigma \rho A L$   
 ( $\rho$ : 밀도     $A$ : 부재 단면적     $L$ : 부재 길이)

**Subject To** ;  $\lambda = KL/r$  ( $r = \sqrt{I/A}$ ,  $K=1$ )

$G_i(A) = 200/\lambda - 1.0 > 0$

when  $f_a < 0$  ( $f_a = \frac{P}{A}$ )

when  $\lambda < C_c$

$$F_a = \frac{1 - \frac{\lambda^2}{2C_c^2} F_y}{FS}$$

$$FS = \frac{5}{3} + \frac{3\lambda}{8C_c} - \frac{\lambda^3}{8C_c^3}$$

$G_j(A) = f_a / F_a + 1.0 > 0$  ( $F_a$ : 허용 압축응력)

when  $\lambda > C_c$

$$F_a = \frac{12\pi^2 E}{3(KL/r)^2}$$

$G_k(A) = f_a / F_a + 1.0 > 0$

when  $f_a > 0$

$F_a = 0.6F_y$  ( $F_y$ : 항복 응력)

$G_l(A) = 1.0 - f_a / F_t > 0$  ( $F_t$ : 허용 인장응력)

### 5.3 설계 결과 및 비교 검토

GT/STRUDL 과 Genetic Algorithm, 그리고 수정된 Genetic Algorithm(표 1에서 \*GENETIC에 해당)에 의해 각각의 설계 과정을 수행한 후 그 결과를 비교하였다. 설계 과정에서 사용한 이산화 변수는 GTTABLE에 있는 것을 참조하였다.(예를들어, 단면적 0.354 → P1/8X10S) Genetic Algorithm에서는 32 가지 부재 종류와 64 가지 부재 종류에 대해 각각 이산화 변수로 mapping 하여 최적설계를 수행하였다.

GT/STRUDL에 의한 설계는 초기값으로 제일 작은 부재를 사용하여 해석을 한 다음 두번의 member select를 수행하여 [표 1]과 같은 결과를 얻었다. 여기서의 목적 함수값은 국부 최적점(local minimum)이라 할 수 있다. 이는 15개의 부재중 어느 하나라도 한 단계 아래 부재 크기를 사용하면 제약 조건을 만족시키지 못하기 때문이다.

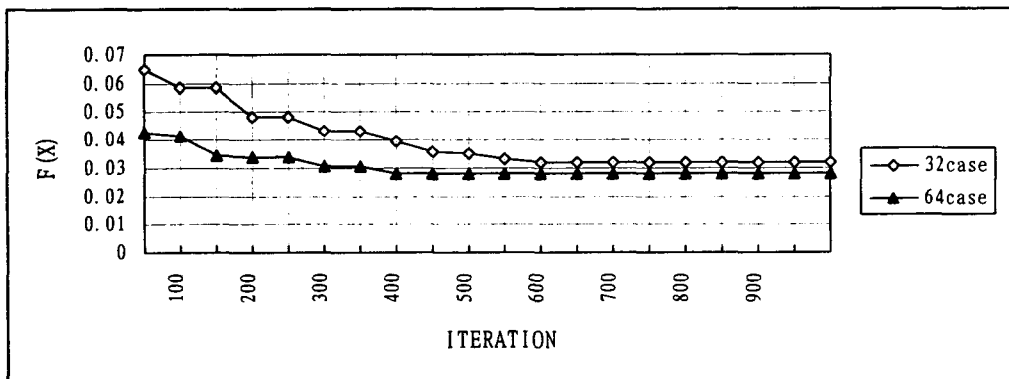
[표 1]에서 보면, Genetic Algorithm에 의한 최적설계 결과가 GT/STRUDL에 의한 값보다 크게 나온 것을 알 수 있다. GT/STRUDL이나 다른 일반적인 설계 과정에서는 부재 크기를 작은 것부터 차례로 code check 해가면서 증가시킨다. 반면 Genetic Algorithm에서는 무작위로 부재를 발생시키기 때문에 일단 큰 부재가 선택되어 제약조건을 만족하면 부재를 줄여야 한다. 이를 위해서 GT/STRUDL에서처럼 허용 응력비(allowable stress ratio)의 상,하한치(upper bound, lower bound)를 정해주어야 하지만 하한치를 정해주어 실행해 보면 결과가 더 이상 진척되지 않음을 알 수 있었다. 이는 무작위로(randomly) 발생된 부재 중에서 제약조건을 만족하는 부재가 상,하한치의 허용응력비를 만족하도록, 주어진 테이블에서 구하기가 쉽지 않기 때문이다.

[그림 4]는 32 가지 부재 종류와 64 가지 부재 종류의 결과치에 대해서 iteration 이 진행되면서 목적 함수값이 수렴해 가는 정도를 보여주고 있다. 이들을 비교해 보면 64case 의 값이 국부 최적점에 좀 더 근접한 값을 알 수 있다. 결국 부재의 종류가 다양할수록, 즉 연속 변수와 같아질수록 더 좋은 결과값을 얻을 수 있음을 알 수 있다.

여기서 설계 변수를 연속 변수로 처리하고 최적 설계를 하고 난 결과값을 이산화된 부재로 바꾸는 방법을 고려해 볼 수도 있다. 하지만 제약 조건중에서 부재의 단면 2 차 모멘트값이 필요한 경우(세장비, 압축 응력 검토시), 무작위로 발생된 단면적에 대해서는 단면 2 차 모멘트값을 정해줄 수가 없다. 그렇기 때문에 결국 트러스 구조물에서 실제 설계 조건을 가지고 Genetic Algorithm 을 사용하려면 이산화 변수를 사용해야 한다. [표 1]의 Genetic Algorithm(32case,64case)은 상한치만 주고 하한치는 주지 않은 결과이다.

		GT/STRU DL	*GENETIC	GENETIC(64case)	GENETIC(32case)
부 재 단 면 적  (cm <sup>2</sup> )	X <sub>1</sub>	0.354	0.354	0.354	0.354
	X <sub>2</sub>	0.354	0.354	0.354	0.354
	X <sub>3</sub>	0.354	0.354	0.354	0.354
	X <sub>4</sub>	0.354	0.354	0.354	0.354
	X <sub>5</sub>	1.298	1.298	1.298	1.647
	X <sub>6</sub>	1.298	1.298	1.647	1.298
	X <sub>7</sub>	1.298	1.298	2.101	1.647
	X <sub>8</sub>	2.101	2.101	2.664	2.101
	X <sub>9</sub>	1.647	1.647	1.647	4.697
	X <sub>10</sub>	1.647	1.647	1.647	1.647
	X <sub>11</sub>	1.647	1.647	1.647	2.417
	X <sub>12</sub>	1.647	1.647	1.647	2.417
	X <sub>13</sub>	1.647	1.647	2.101	2.101
	X <sub>14</sub>	1.647	1.647	2.417	1.647
	X <sub>15</sub>	1.647	1.647	1.647	2.101
Obj. Value(ton)		0.024166	0.024166	0.027838	0.034664
ITERATION		3	793	378	793
해석 횟수		3	1	1	1

[표 1] 설계 결과 비교



[그림 4] 32case 와 64case 와의 목적 함수값 비교

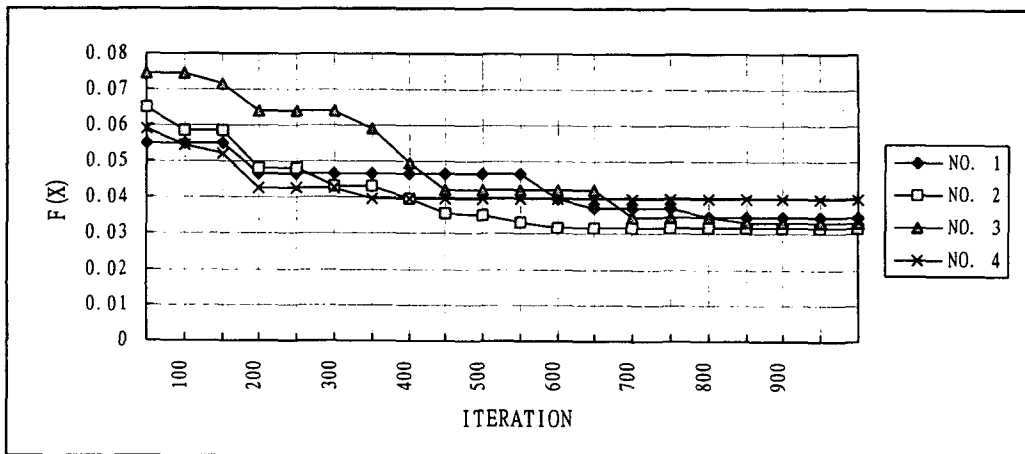
수정된 Genetic Algorithm(표에서 \*GENETIC)에서는 기존 프로그램의 하한치를 주기 어렵다는 문

제점 때문에, Genetic Algorithm 을 사용해서 결과를 얻은 이후에 하한치를 주어 결과값을 찾아낸 것이다. 이렇게 해서 얻은 결과가 국부 최적점과 일치함을 알 수 있다.

[표 2]와 [그림 5]는 p<sub>cross</sub>, p<sub>mutation</sub> 의 값에 따른 목적 함수의 값을 비교한 것이다. 무작위로 발생된 설계 변수들은 crossover(교배)와 mutation(돌연변이)의 과정을 거치게 되는데 이때 이들 과정을 얼마의 확률을 가지고 수행할 것인지를 정해야 한다. 그 확률을 p<sub>cross</sub>, p<sub>mutation</sub> 으로 표시한다. 연속 변수를 가지고 수행할 때는 초기 p<sub>cross</sub>, p<sub>mutation</sub> 이나 마지막 p<sub>cross</sub>, p<sub>mutation</sub> 값이 대부분 0.9, 0.01, 0.9, 0.01 일때가 가장 효과적으로 수렴 되지만 이산화 변수를 사용했을 때는 [표 2]에서 보는 것처럼 뚜렷한 특징을 발견하기가 힘들다. [표 2]는 32case 에서 수행한 결과이며 64case 에서는 0.9, 0.01, 0.9, 0.01 일때 효과적으로 수렴하였다.

NO.	I_CROSS	I_MUTATION	F_CROSS	F_MUTATION	OBJ. VALUE
1	0.9	0.01	0.9	0.01	0.034660
2	0.9	0.01	0.7	0.05	0.031699
3	0.8	0.01	0.8	0.01	0.032936
4	0.7	0.01	0.7	0.01	0.039587
5	0.92	0.001	0.92	0.001	0.050661
6	0.8	0.05	0.8	0.05	0.049837
7	0.7	0.1	0.7	0.1	0.05840
8	0.9	0.01	0.92	0.001	0.043559
9	0.9	0.01	0.7	0.07	0.03826
10	0.9	0.01	0.8	0.05	0.03321
11	0.7	0.05	0.9	0.01	0.033903

[표 2] p<sub>cross</sub>, p<sub>mutation</sub> 에 따른 목적 함수값 비교



[그림 5] p<sub>cross</sub>, p<sub>mutation</sub> 값에 따른 수렴 정도 비교

## 6. 결론

구조물이 대형화 되어 가는 추세에 그 필요성이 커지고 있는 대경간 지붕 구조물에 대해 확률론적 탐색 기법인 Genetic Algorithm 의 가능성과 문제점을 검토한 결과 다음과 같은 결론을 얻었다

- (1) 기존 수치 최적화 방법은 구조물이 대형화되고, 특히 대경간 지붕 구조물에서처럼 그룹핑을

하지 않는 경우에 정식화 과정이 복잡하고 많은 시간이 소요되기 때문에 비효율적이다. 반면 Genetic Algorithm 을 사용한 경우 정식화 과정을 비교적 간단하게 수행할 수 있었다.

- (2) 기존의 Genetic Algorithm 에서처럼 generation 될 때마다 해석을 하지 않고 해석 횟수를 줄임으로써 대경간 지붕 구조물의 최적 설계시 소요 시간을 줄일 수 있을 것으로 기대된다.
- (3) Genetic Algorithm 을 사용하여 실제 설계 조건을 갖는 트러스 구조물을 최적화할 때에는 연속변수로 처리할 수가 없고 이산화된 변수를 사용해야 함을 알 수 있었다. 그렇기 때문에 최적값이 지나는 한계점을 보완하기 위해, 최근에 많은 연구가 진행중인 shape optimization 을 Genetic Algorithm 에 도입하면 좀 더 효율적인 최적 설계값을 얻을 수 있을 것으로 기대된다.

### 참 고 문 헌

- [1] 최 은규, "최적화 기법을 이용한 3 차원 트러스 구조물의 설계 자동화", 한양대학교 대학원, 1993, p25 - 29
- [2] M.Z. Cohn, Fellow and A.S.Dinovitzer, "Application of Structural Optimization", ASCE, Vol.120, No.2, February, 1994
- [3] Arora,J.S., Tseng,C.H., "IDESIGN User's Manual Version 3.5", University of Iowa, 1987
- [4] Michael Reymond and Mark Miller,Editors, "Quick Reference Guide - Version 68", The Macneal-schwendler corporation(MSC)
- [5] 양 영순,김 기화, "전체 최적화를 위한 확률론적 탐색기법", 전산구조공학 제 5 권 제 2 호, 1992.6 , p 93-104
- [6] Arora,J.S., "Introduction to Optimum Design", McGRAW-HILL INTERNATIONAL EDITIONS, 1989
- [7] Shyue-Jian Wu and Pei-Tse Chow, "Genetic Algorithms for Nonlinear Mixed Discrete-Integer Optimization Problems Via Meta-Genetic Parameter Optimization", Eng.Opt., 1995. Vol.24,pp137-159
- [8] S.Rajeev and C.S. Krishnamoorthy, "Discrete Optimization of Structures Using Genetic Algorithms" ASCE, Vol.118, No.5, May, 1992.
- [9] R. A. Canfield, R.V. Grandhi, and V.B.Venkayya, "Optimum Design of Structures with Multiple Constraints", AIAA, Vol. 26, No. 1, 1986.
- [10] David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning",Addison Wesley Publishing Company, Inc , 1989
- [11] Zbigniew Michalewicz, "Genetic Algorithms + Data Structures = Evolution Problems", Springer-Verlag Berlin Heidelberg, 1992