

# 실수형 Genetic Algorithm에 의한 최적 설계

## A Real Code Genetic Algorithm for Optimum Design

양 영 순\*

Yang, Young-Soon

김 기 화\*\*

Kim, Gi-Hwa

### 요 약

Genetic Algorithms(GA)는 생명체의 자연진화 법칙에 기초한 최적화 방법으로 그 범용성이 높고 평가되어지고 있다. 기존의 GA는 대부분 설계변수로 2진수형 코드를 사용하는데, 이는 실수형 설계변수로 구성된 최적화 문제를 해결하기 위해 컴퓨터 주 기억용량을 많이 사용하여야 하며, 계산 시간 면에서도 비효율적이고 또한 국부탐색 능력도 떨어지는 단점이 있다. 따라서 본 연구에서는 GA에 의한 최적화과정에서 실수형 설계변수를 직접 사용할 수 있도록 교배와 돌연변이 과정을 새로이 정식화 하였다. 그리고 여러 형태의 단일 및 다목적함수 최적화 문제에 대해 실수형 GA와 2진수형 GA의 결과를 비교 검토하였다. 비교 검토 결과, 실수형 GA의 성능이 2진수형 GA보다 우수함을 알 수 있었고, 일반 최적화 방법으로 실수형 GA를 사용하여도 무방하리라 본다.

### Abstract

Genetic Algorithms(GA), which are based on the theory of natural evolution, have been evaluated highly for their robust performances. Traditional GA has mostly used binary code for representing design variable. The binary code GA has many difficulties to solve optimization problems with continuous design variables because of its large computer core memory size, inefficiency of its computing time, and its bad performance on local search. In this paper, a real code GA is proposed for dealing with the above problems. So, new crossover and mutation processes of GA are developed to use continuous design variables directly. The results of real code GA are compared with those of binary code GA for several single and multiple objective optimization problems. As a result of comparisons, it is found that the performance of the real code GA is better than that of the binary code GA, and concluded that the real code GA developed here can be used for the general optimization problem.

\* 서울대학교 조선해양공학과 부교수  
\*\* 서울대학교 조선해양공학과 연구원

이 논문에 대한 토론을 1995년 12월 31일까지 본 학회에 보내주시면 1996년 6월호에 그 결과를 게재하겠습니다.

## 1. 서 론

Genetic Algorithms(이하 GA)는 생명체의 진화법칙인 적자생존의 원리에 기초한 탐색 및 최적화 기법이다. GA는 1975년 Holland[1]에 의해 기존의 연구를 종합, 체계화 시켰으며, 공학분야에는 Goldberg[2][3]에 의해 개스 송수관에 대한 최적설계가 시도된 이래 많은 연구가 행해지고 있으며, 그 범용성이 높기 평가되어 지고 있다. 특히 기존의 최적화 방법이 국부탐색을 행함으로 해서 전체최적점(global optimum)을 찾기 위해 다수의 초기점에 대해 반복 계산을 해왔으나, GA는 하나의 설계점이 아닌 설계집단을 사용하여 확률론적인 탐색을 행함으로 해서 전체최적점에 도달할 신뢰도가 매우 크다.[4] 또한 공학설계 문제에 종종 나타나는 정수 및 이산적 설계변수의 처리에 있어서도 2진수(binary) 코드를 사용하여, 그 처리 과정이 매우 단순하고 효율적이다.[5] 그리고 다목적함수 최적화 문제에서도 Pareto 최적해 집합을 구하기 위해 기존의 방법이 다목적함수 문제를 임의의 설계상수를 도입하여 단일 목적함수로 구성된 문제로 치환하여 다수의 설계상수에 대해 반복 계산하였으나, GA는 탐색을 행하는 설계집단이 최적해가 되어 한번의 최적화 계산으로 최적해 집합을 구할 수 있는 MOGA(Multicriteria Optimization by Genetic Algorithm)가 개발되어 그 효율성이 매우 크다.[6]

하지만 GA에서 일반적으로 사용하는 2진수 코드가 실수형 변수로 구성된 최적화 문제에 대해 여러가지 문제점을 나타낸다. 실수형으로 표현된 설계변수를 2진수 코드로 표현하기 위해서는 2진수 문자열의 길이를 요구되는 정밀도를 표현할 정도로 길게 선정해야 하는데, 문자열의 길이가 길수록 문자열을 표시하는 배열의 크기도 매우 커져 컴퓨터 주 기억용량을 과도하게 사용하여야 하며, 교배(crossover)과정 및 2진수형 변수를 실제값으로 변환하는 데에도 많은 시간이 요구되는 단점이 있다. 이러한 2진수형 GA의 단점은 설계변수의 수가 증가할수록 더욱 심각하게 나타난다. 그리고 2진수형 GA의 또 하나의 문제점은 최적해 부근까지는 매우 빠른 속도로 접근하나, 그 근처

에서의 국부탐색 능력은 상당히 떨어진다는 것이다.

따라서 본 연구에서는 GA를 사용하여 실수형 변수로 이루어진 최적화 문제를 해결하는 한 방법을 제시하였다. 이를 위해 새로운 교배 및 돌연변이(mutation) 과정을 개발하였으며, 국부탐색 능력을 향상시키기 위해 설계변수의 증분량을 조절하여 주었다. 실수형 GA의 성능을 검토하기 위해 다양한 종류의 단일 및 다목적함수 문제에 대해 최적화를 수행하며, 그 결과를 2진수 코드에 의한 결과와 비교, 검토하였다.

## 2. Genetic Algorithms

### 2.1 Genetic Algorithm의 특징

Genetic Algorithm(GA)의 다아윈이 주장한 자연진화의 법칙인 적자생존(survival of fittest)과 자연도태(natural selection)의 원리를 토대로 하여 정립된 최적화 알고리즘이다. 자연진화의 법칙에 의하면 일정한 환경에 무리를 지어 살고 있는 생명체들은 그 환경에 적합한 형질을 가진 개체가 생존할 확률을 크게 가지며 교배(crossover)와 변종(mutation)의 과정을 통해 보다 좋은 방향으로 진화해 나가고 부적합한 형질의 개체들은 진화의 과정에서 점차 도태되어 간다. 이와 같은 진화의 과정을 되풀이 하면 결국 주어진 환경에 가장 적합한 형질의 개체들로 형성될 것이다. GA는 이러한 생명체의 자연진화 법칙을 일반 최적화 문제에 적용한 방법으로, 설계영역에 다수의 설계점을 분포시켜 목적함수 값과 제한조건의 위반정도에 따라 각 설계점에 적합성(fitness)을 부여한다. 적합성이 클수록 다음 단계인 교배와 변종의 과정에 참여할 확률은 크게 하여(reproduction) 적합성이 좋은 설계점에 비슷한 형질을 가진 설계점들이 다음 단계의 보다 많이 형성되어 계산이 진행될수록 전체 설계점들은 좋은 방향으로 탐색이 진행된다. 현재까지 대부분의 GA는 기존의 최적화 방법과는 다른 다음과 같은 특징이 있다.[3]

#### (1) 설계변수로 코드를 사용

GA는 설계변수로 실제값을 직접 사용하는 대

신 코드형태(2진수)의 문자열(string)을 사용한다.(예 : 100101) 이러한 문자열 형태의 설계변수의 사용은 염색체에 유전인자가 나열해 있는 것과 같은 형태를 지니고 있어 교배와 돌연변이의 변환 과정을 수행하기가 매우 단순하고 편리하다. 또한 이러한 문자열은 이산적인 성질을 지니고 있어 정수 또는 이산적 설계변수를 포함하는 혼합형 최적화 문제에 효과적으로 사용할 수 있다. 하지만 이러한 이산적인 형태의 코드는 실수형 설계변수를 처리하는 데는 비효율적이다.

(2) 다수의 설계점들이 집단 탐색

대부분의 최적화 방법이 한점에서 한점으로서의 이동에 의한 국부적인 탐색과정을 가지는 반면에, GA는 여러개의 설계점들이 집단(population)을 이루어 동시에 탐색을 행하여 보다 넓은 설계영역에 대한 정보를 활용함으로써, 전체 최적점에 수렴할 확률이 기존의 방법에 비해 매우 크다.

(3) 직접 탐색방법

GA는 목적함수와 제한조건의 값만을 사용하고 미분값이나 그 외의 다른 정보를 필요로 하지 않는 직접 탐색방법을 이용하므로 실제 구조물의 설계와 같은 복잡하고 다양한 최적화 문제에 적합하다.

2.2 Genetic Algorithm의 변환과정

GA는 자체의 기본 성질에서 타 방법과는 다른 특성을 가지고 있다. 이러한 특성을 적절히 활용하기 위해 다음과 같은 변환과정을 거친다.

• 도태(Reproduction, Selection)

각 설계점의 적합성의 크기에 따라 교배와 변종의 과정에서 선택될 확률을 부여한다. 즉 적합성이 큰 설계점은 많이 선택되도록 하고 나쁜 설계점은 적게 선택되도록 한다. 이를 위해 여러가지 방법이 있지만 본 논문에서는 가장 단순한 방법으로 각 설계점이 선택될 확률은  $f_i / \sum f_i$ 로 하였다. 여기서  $f_i$ 는  $i$  설계점의 적합성의 크기를 뜻한다. 이러한 과정을 통해 설계점들의 집단은 적합성의 크기에 따라 새로이 구성되고, 집단 전체의 적합성의 평균은 상승되는 효과가 있다.

• 교배(Crossover)

도태의 과정을 통해 좋은 설계점들은 강조되고

나쁜 설계점들은 도태되었지만 더 좋은 방향으로 진일보 하지는 못하였다. 따라서 설계점들의 향상을 위해 생명체의 유전인자 교환과 같은 교배과정을 거친다.(0,1) 사이의 임의의 수가 확률  $P_c$  보다 작을 경우 임의의 2개의 설계점을 도태의 과정을 통해 재구성된 집단 속에서 선택한다. 이때 설계점들은 문자열 형태로 구성되어 있다. 문자열의 길이 보다 작은 임의의 수  $k$ 를 선택하여, 짝지은 2개의 문자열의  $k$  아래쪽 부분을 서로 교환한다. 예를 들면 2진수로 구성된 문자열의 초기 상태를  $P_1, P_2$ 라 하고, 교배가 일어난 후의 상태를 각각  $C_1, C_2$ 라 하고  $k=4$ 일 때 다음과 같이 변환된다.

$$\begin{matrix} P_1=0000|000 & C_1=0000|111 \\ P_2=1111|111 & C_2=1111|000 \end{matrix} \rightarrow$$

이와 같이 교배과정은 적합성이 높은 설계점과 비슷한 형질의 설계점이 많이 생성되어 계산이 되풀이 될수록 가장 좋은 설계점 주위에 다수의 설계점이 집중하게 된다.

• 돌연변이(Mutation)

교배에 의해 설계점들이 너무 한쪽 방향으로만 치우치는 것을 보완하기 위해 아주 작은 확률( $P_m$ )로 각 문자열의 진수 단위로 돌연변이를 시켜준다. 즉 2진수 문자열일 경우 0은 1로 1은 0으로 바꾸어 준다. 만일 00101의 문자열에 돌연변이가 일어날 위치가 2라면 01101로 된다.

이와 같이 GA의 전체 전개 과정은 매우 단순하다. 하지만 다수의 설계집단과 도태, 교배 그리고 돌연변이의 과정이 합쳐진 결과는 매우 강력하다. 따라서 GA의 가장 큰 장점은 설계모델에 제약성(변수의 연속성, 미분값의 존재, unimodality 등)이 매우 적어 다양한 종류의 문제에 대해 범용성(robustness)이 매우 넓다는 것이다.

3. 실수형 변수를 사용한 GA의 최적화 과정

GA는 설계변수로 2진수 코드를 사용함으로써, 이산적 문제에는 적용이 매우 단순하고 그 결과도 기존의 타 방법(예 : branch-bound 방법)에

비해 우수하여, 그 효율성이 높기 평가되고 있다. [5] 하지만 설계변수가 실수로 이루어져 있을 때, 2진수 코드를 사용하기 위해서는 설계변수를 표현하는 2진수 문자열의 길이를 상당히 길게 선정해야 한다. 예를 들어  $F(x)=x^2, 0 \leq x \leq 5$ 의 문제의 경우 설계변수가 등간격으로  $x=(0, 1, 2, 3, 4, 5)$ 의 값을 가질 때, 이에 대응되는 목적함수 값은  $F=(0, 1, 4, 9, 16, 25)$ 로  $x$ 값이 커질수록  $F$ 값은 그 간격이 점점 증가한다. 일반적으로 대부분의 최적화 문제에서 목적함수 설계공간의 경사가 최적점 부근에서는 타부분 보다 급해지므로, 2진수 문자열의 길이를 충분히 길게 하지 않으면 최적점이 고려 대상에서 제외될 가능성도 있고, 실제 중요한 최적점 부근에서의 탐색에 어려움이 예상된다. 하지만 2진수 코드를 사용할 경우 문자열의 길이를 길게 하더라도 여전히 국부 탐색에는 한계가 존재한다. 예를 들어 2진수로 표현된 설계변수 10000이 최적값이라 할 때, 그와 가장 가까운 01111과 10001의 문자열이 생성되었다 하더라도 이 두 문자열은 교배과정을 통해서는 최적값을 형성할 수 없으며, 오직 두번째 문자열의 마지막 인수가 돌연변이에 의해 바뀌어 저야 하는데 문자열의 길이가 길수록 그 과정이 발생할 확률은 극히 희박해진다. 또한 문자열의 길이가 길수록 교배과정 및 2진수형 변수를 실제값으로 변환하는 데에도 많은 시간이 요구되며, 문자열을 표시하는 배열의 크기도 매우 커져 컴퓨터 주 기억용량을 과도하게 사용하는 단점이 있다. 따라서 본 연구에서는 GA를 사용하여 실수형 변수로 이루어진 최적화 문제를 해결하는 한 방법을 제시하였다.

### 3.1 최적화 과정

#### (1) 설계집단의 초기화

요구되는 수 만큼의 설계점들을 설계공간내에 발생시킨다. 이 때 각 설계변수의 값은 최소값과 최대값 사이에서 무작위로 발생시킨다.

#### (2) 도태 과정

Roulette wheel selection[3]을 사용하였다. 즉 각 설계점이 선택될 확률은 적합성(fitness)의 크기에 비례한다.

#### (3) 교배 및 돌연변이 과정

가장 특징적인 부분으로 다음 절에서 설명하겠다.

#### (4) Elitist procedure[3]

이전 세대의 가장 좋은 설계점의 적합성 값이 현 세대에서의 가장 좋은 설계점의 적합성 값보다 클 때는, 현 세대의 가장 나쁜 설계점과 교체하여 주어, 현재까지 구한 가장 좋은 정보를 간직한다.

### 3.2 교배 과정

먼저(0,1) 사이에서 임의의 수를 발생시켜 그 값이 교배 확률(Pc) 보다 작으면 1과 설계변수 갯수 사이의 임의의 수  $k$ 를 선택한다. 교배가 수행되기 전의 설계점이 각각 A와 B라면, 수행된 후는 다음과 같이 C와 D가 된다.

$$\begin{matrix} A=a_1 a_2 \cdots a_k a_{k+1} \cdots a_n & C=a_1 a_2 \cdots x_k b_{k+1} \cdots b_n \\ B=b_1 b_2 \cdots b_k b_{k+1} \cdots b_n & D=b_1 b_2 \cdots y_k a_{k+1} \cdots a_n \end{matrix} \rightarrow$$

여기서  $k$ 번째 설계변수  $a_k$ 와  $b_k$ 가 각각  $x_k$ 와  $y_k$ 로 변환되는 과정은 다음과 같다.

$$x_k = a_k + \delta \cdot \text{step} \tag{1}$$

$$y_k = b_k - \delta \cdot \text{step}$$

여기서  $\delta$ 와  $\text{step}$ 의 값은 다음과 같다.

$$\delta = \text{rnd}_1 \times b_k - \text{rnd}_2 \times a_k \tag{2}$$

여기서  $\text{rnd}_1$ 과  $\text{rnd}_2$ 는 모두 0과 1 사이의 임의의 수이다. (2)식에서  $\delta$ 는 교배과정에 의한 변수의 증분량을 뜻하며, 각 변수에 전달하여 주는 값의 상대적인 차이에 의해 증분량의 크기와 방향이 결정되어, 2진수형 문자열에서 사용한 교배과정과 비슷한 의미를 가진다.

그리고  $\text{step}$ 은  $\delta$ 에 의한 설계변수의 증분량의 크기를 조절하는 값으로, 탐색이 진행될수록 우수한 설계점 주위의 국부 탐색을 강화하기 위해, 세대가 진행될수록 그 크기를 감소시키며, 그 감소량( $d\text{step}$ )은 다음과 같다.

$$\begin{aligned} \text{mg} &= \text{maximum number of generations} \\ \text{step}_1 &= \text{step size of first generation} = 1.0 \end{aligned}$$

이라 할 때, 감소량은

$$dstep = (step_{mg})^{1 \cdot mg} \quad (3)$$

이 되고,  $i$ 번째 세대에서의 값은 다음과 같다.

$$step_i = step_{i-1} \times dstep = step_1 \times dstep^{i-1}, \quad i=2, 3, \dots, mg \quad (4)$$

그리고  $x_k$ 와  $y_k$ 의 값이 각각의 최소치 혹은 최대치를 초과하게 되면, 각각의 최소치 또는 최대치 값을 사용한다.

### 3.3 돌연변이 과정

돌연변이 과정은 교배과정에 의해 설계점들이 너무 한쪽 방향으로 치우치는 것을 보완하기 위해 아주 작은 확률( $P_m$ )로 문자열을 형성하는 각 설계변수 단위로 돌연변이율을 시켜준다. 즉 다음 문자열에서  $k$ 번째 설계변수에서 돌연변이가 일어난다고 하면 그 과정은 다음과 같다.

$$\begin{array}{l} \text{before mutation} \quad \rightarrow \quad \text{after mutation} \\ a_1 a_2 \dots a_k a_{k+1} \dots a_n \quad \rightarrow \quad a_1 a_2 \dots x_k a_{k+1} \dots a_n \end{array}$$

여기서

$$x_k = \begin{cases} a_k + \delta & \text{if } rnd > 0.5 \\ a_k - \delta & \text{if } rnd < 0.5 \end{cases} \quad (5)$$

$$\delta = rnd \times (0.5a_k)$$

$rnd$  = random number between 0 and 1

그리고 교배과정에서의 마찬가지로  $x_k$ 가 최소치와 최대치를 초과하지 못하도록 하였다.

## 4. 적용례 및 고찰

본 연구에서 새로이 정의된 실수형 GA의 성능을 평가하기 위해 단일 및 다목적함수 최적화 문제에 대해 기존의 2진수형 설계변수를 사용한 GA와 정확성과 계산시간의 관점에서 그 효율성을 비교, 검토하였다. 본 계산에 사용된 컴퓨터는 IBM

PC 486/66Mhz이다.

### 4.1 단일목적함수 최적화

#### 4.1.1 Shekel 함수

Shekel 함수는 설계변수가 4개이고 국소최소점이 다수 존재하는 문제로 다음과 같다.[7]

$$\begin{aligned} \text{Minimize } F(x) &= -\sum_{i=1}^m \left[ \sum_{j=1}^n (x_j - a_{ij})^2 + c_i \right]^{-1} \quad (6) \\ 0 \leq x_j &\leq 10.0, \quad j=1, 2, \dots, n \end{aligned}$$

여기서  $n=4$ ,  $m=10$ 이며 상수  $a_{ij}$ ,  $c_i$ 의 값은 Table 1에 수록하였다. 본 함수는 10개의 국소최소점이 존재하고 그 값은 다음과 같다.

$$\begin{aligned} x_i^* &\approx (a_{i1}, a_{i2}, a_{i3}, a_{i4}) \\ F_i^* &\approx -1/c_i, \quad i=1, 2, \dots, 10 \end{aligned}$$

Table 1. Input data for the Shekel function

| i  | $a_{ij}$ |     |     |     | $c_i$ |
|----|----------|-----|-----|-----|-------|
| 1  | 4.0      | 4.0 | 4.0 | 4.0 | 0.1   |
| 2  | 1.0      | 1.0 | 1.0 | 1.0 | 0.2   |
| 3  | 8.0      | 8.0 | 8.0 | 8.0 | 0.2   |
| 4  | 6.0      | 6.0 | 6.0 | 6.0 | 0.4   |
| 5  | 3.0      | 7.0 | 3.0 | 7.0 | 0.4   |
| 6  | 2.0      | 9.0 | 2.0 | 9.0 | 0.6   |
| 7  | 5.0      | 5.0 | 3.0 | 3.0 | 0.3   |
| 8  | 8.0      | 1.0 | 8.0 | 1.0 | 0.7   |
| 9  | 6.0      | 2.0 | 6.0 | 2.0 | 0.5   |
| 10 | 7.0      | 3.6 | 7.0 | 3.6 | 0.5   |

2진수형 GA와 실수형 GA에 의한 최적화를 위한 설계 상수는 Table 2와 같으며, 그 결과는 Table 3와 Fig. 1에 나타내었다. 각 방법의 초기 설계집단은 모두 전체 설계공간에 골고루 분포하였으며, 설계집단중 가장 좋은 설계점의 목적함수 값은 2진수형이  $-0.7316$ 이고 실수형은  $-0.8551$ 이므로 비슷한 환경에서 출발하였다.

본 예제의 전체최소점은  $x^*=(4, 4, 4, 4)$ 이고, 이 때  $F^*=-10.5364$ 가 되는데 Fig. 1에서 보여준 A점(4.057, 3.734, 4.047, 3.842)과 B점(4.448, 3.973, 3.956, 3.950)은 각각 2진수형과 실수형 GA에 의해 전체최소점 영역에 도달한 시점에서의 위

치를 보여주고 있다. 여기서 전체최소점 영역이란 전체최소점 부근의 점을 의미하는 것으로서, A점을 얻기 바로 직전의 설계점을 보면  $x=(4.995, 4.985, 2.806, 3.011)$ 이고  $F=-3.429$ 로서 Table 1의 7번째 국소최소점에 빠져 있음을 알 수 있고, B점을 얻기 바로 직전의 설계점은  $x=(4.962, 5.011, 2.972, 3.002)$ 이고  $F=-3.806$ 로서 마찬가지로 Table 1의 7번째 국소최소점에 빠져 있음을 확인할 수 있다.

Table 2. Design parameters of binary and real code GA for the Shekel function

|                         | Binary code GA | Real Code GA |
|-------------------------|----------------|--------------|
| population size         | 50             | 50           |
| chromosome length       | 10             | 1            |
| total chromosome length | 40             | 4            |
| crossover probability   | 0.9            | 0.9          |
| mutation probability    | 0.01           | 0.01         |
| max. generation No.     | 1000           | 1000         |
| step <sub>mg</sub>      | -              | 0.001        |

Table 3. Optimization results for the Shekel function

|                     | Binary code GA | Real code GA |
|---------------------|----------------|--------------|
| $x_1$               | 3.9980         | 4.0006       |
| $x_2$               | 4.0665         | 4.0005       |
| $x_3$               | 3.9980         | 3.9996       |
| $x_4$               | 4.0665         | 3.9995       |
| $F(x)$              | -9.7246        | -10.5364     |
| computing time(sec) | 33.56          | 7.91         |

하지만 Fig. 1에서 보면 2진수형 GA는 세대가 증가함에 따라 국소최소점 영역을 벗어나 225번째 세대(A점)에 이른 후 세대가 증가하여도 수렴의 속도가 매우 느리다. 이는 여러 문헌[3][4]에서도 지적되었듯이 2진수형 GA가 지니는 한계로서 최적점 근처까지는 잘 도달하나 그 후 국부탐색의 효율성은 매우 떨어지는 경향이 있다. 2진수형 변수를 연속적인 성질을 가지게 하기 위해서는 2진수 문자열의 길이(L)를 길게 하여 주어야 하는데, 이는 설계변수가 취할 수 있는 경우의 수가  $2^L$ 개로 기하학적으로 증가하고 또한 본 문제와 같이 최적해 영역의 끝이 매우 깊은 경우는 최적해 부근에 대상 설계점의 수가 매우 적어 국부탐색에 상당한 어려움이 있다.

이에 반해 Fig. 1에 나타난 바와 같이 실수형

GA는 200번째 세대(B점)에서 전체최소점 영역에 도달한 후, 국부 탐색 능력이 매우 뛰어난을 알 수 있다. 이는 step 변수를 사용하여 설계변수의 증분량을 세대가 진행됨에 따라 감소시킴으로서 2진수형 GA에 비해 국부영역에 대해 보다 자세한 탐색이 가능하였다.

또한 각 방법의 설계상수는 Table 2의 값을 사용하고, 초기 설계집단을 변화해 가며 최적화를 수행하여 보았는데, 각 방법 모두 10개의 서로 다른 초기집단에 대해 실수형 GA는 10번 모두 전체최소점을 구하였고, 2진수형 GA는 7번만 전체최소점을 구하였고 3번은 Table 1의 2, 5, 7번의 국소최소점에 수렴하였다. 그리고 계산시간 비교에서도 실수형 GA가 평균 4배 정도 적게 소요되었다.

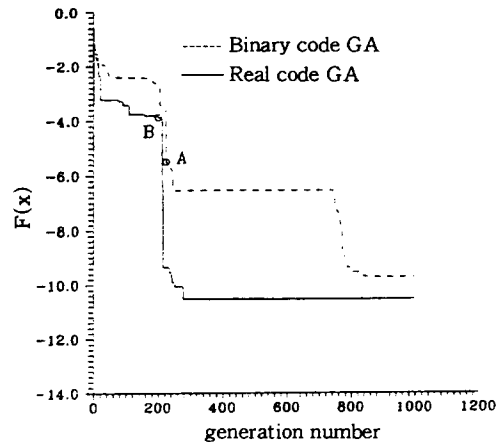


Fig. 1 Optimization of the Shekel function

이상에서 실수형 설계변수로 구성된 최적화문제의 경우 실수형 GA가 2진수형 GA에 비해 설계영역 전체에 대한 탐색 능력이 우수하고, 국부 탐색 능력도 뛰어나며, 계산시간도 효율적이었다.

#### 4.1.2 Wong's problem

다음 최적화 문제로 Wong's problem을 선택하였다. 본 문제는 설계변수 20개, 제한조건 17개로 구성된 최소화 문제로, Pierre[8]에 의해 구해진 최적해  $F=130.49$ 이다. 2진수 GA와 실수형 GA에 의한 최적화에 사용한 설계 상수는 Table 4와 같으며, 2진수형 GA에 사용된 문자열의 길이는

Table 4. Design parameters for the Wong's problem

|                         | Binary code GA | Real Code GA |
|-------------------------|----------------|--------------|
| population size         | 50             | 50           |
| chromosome length       | 10             | 1            |
| total chromosome length | 200            | 20           |
| crossover probability   | 0.9            | 0.9          |
| mutation probability    | 0.001          | 0.001        |
| max. generation No.     | 1000           | 1000         |
| step <sub>mg</sub>      | -              | 0.01         |

200으로 상당히 길게 설정되었다. 각 방법에 의한 최적화 결과는 Table 5와 Fig. 2에 나타내었다. 결과에서도 알 수 있듯이 실수형 코드에 의한 결과가 더 정확하고, 계산시간 면에서는 거의 7배 이상 효율적이었다. 4.1.1절의 Shekel 함수의 결과와 비교하여 보면, 문자열의 길이가 길수록 즉 설계 변수의 수가 많을수록 계산시간 면에서는 실수형 GA가 더욱 효율적임을 알 수 있다. 본 문제도 Fig. 2에서도 나타나 있듯이 Shekel 함수와 마찬가지로 GA는 최적해 부근까지는 빠른 속도로 접근하나, 최적해 부근에서 더 향상하는 데는 많은 노력이 요구되는 경향이 있다. 이에 대해 실수형 GA는 step이라는 변수를 통해 설계변수의 증분량을 점차로 줄여줌으로서 국부 탐색의 효율이 크게 향상되었음을 알 수 있다.

Table 5. Optimization results of the Wong's problem

|                     | Binary code GA | Real code GA |
|---------------------|----------------|--------------|
| optimum value       | 153.62         | 132.42       |
| error(%)            | 17.73          | 1.48         |
| computing time(sec) | 159.23         | 22.02        |

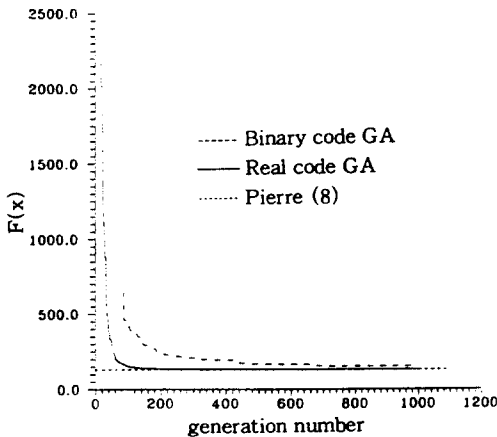
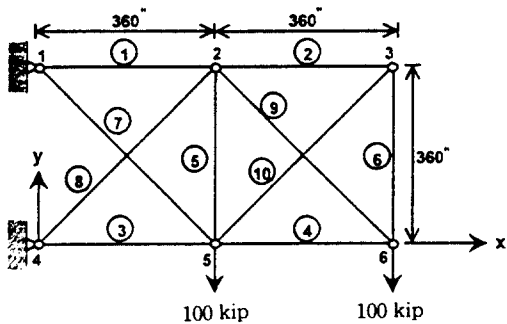


Fig. 2. Optimization of the Wong's problem

### 4.1.3 10-bar 트러스 문제

본 문제는 실제 구조물의 최적설계에 대한 예제로 Fig. 3과 같은 트러스 구조물[9]에 대해 최소 중량설계를 행하였다. 설계변수는 각 부재의 단면적( $x_i$ )이고, 제한조건은 각 부재에 작용되는 응력( $\sigma_i$ )이 허용응력( $\sigma_a$ )을 초과하지 않아야 하며, 각 절점의 y방향 변위( $\delta_j^y$ )가 허용변위( $\delta_a^y$ )를 넘지 않도록 하였다. 응력과 변위는 유한요소법에 의해 해석하였다.



Young's modulus  $E=10^7$  psi

Fig. 3 The 10-bar truss problem

$$\text{Minimize Weight} = \gamma \sum_{i=1}^{10} x_i l_i \quad (7)$$

$$\text{subject to } g_i(x) = 1 - \frac{|\sigma_i|}{\sigma_a} \geq 0,$$

$$i=1, 2, \dots, 10$$

$$g_j(x) = 1 - \frac{|\delta_j^y|}{\delta_a^y} \geq 0,$$

$$j=1, 2, \dots, 6$$

$$0.1 \leq x_i \leq 35.0, \quad i=1, 2, \dots, 10$$

여기서 비중  $\gamma=0.11b/\text{inch}^3$ 이고,  $l_i$ 는  $i$ 부재의 길이이며, 허용응력  $\sigma_a=25,000\text{psi}$ 로 하였으며, 허용변위  $\delta_a^y=2.0\text{inch}$ 로 하였다. 각 최적화 방법에 사용한 설계상수는 설계집단의 크기는 100, 교배와 돌연변이 확률은 각각 0.9와 0.01을 사용하였으며, 계산 횟수는 1000번으로 하였다. 그리고 2진수형 GA에서 각 설계변수의 문자열의 길이는 20, 실수형 GA에서 step<sub>mg</sub>는 0.01로 하였다. 최적화 결과는 Table 6과 Fig. 4에 나타내었다. 결과는 앞

의 예제에서와 마찬가지로 실수형 GA가 2진수형 GA에 비해 최적점 부근에서 국부 탐색능력이 뛰어난 것을 알 수 있고, 또한 계산 시간면에서도 1.5배 정도 효율적이었다.

Table 6. Optimization results of the 10-bar truss problem

| Member No.            | Reference[9] | Binary code GA | Real code GA |
|-----------------------|--------------|----------------|--------------|
| 1                     | 30.031       | 29.170         | 30.066       |
| 2                     | 0.100        | 0.147          | 0.104        |
| 3                     | 23.274       | 20.902         | 23.044       |
| 4                     | 15.286       | 17.814         | 15.465       |
| 5                     | 0.100        | 0.153          | 0.100        |
| 6                     | 0.557        | 0.278          | 0.593        |
| 7                     | 7.468        | 8.962          | 7.428        |
| 8                     | 21.198       | 22.352         | 20.862       |
| 9                     | 21.618       | 21.320         | 21.983       |
| 10                    | 0.100        | 0.187          | 0.100        |
| optimum value         | 5061.6       | 5153.9         | 5061.9       |
| computation time(sec) |              | 1146.67        | 767.26       |

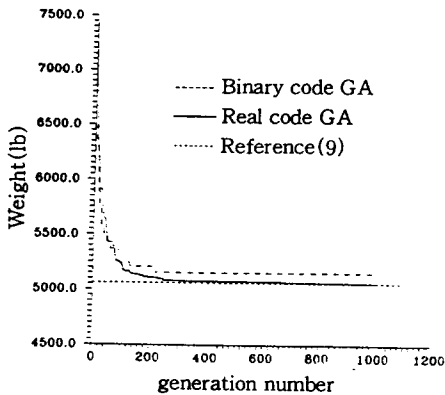


Fig. 4 Optimization results of the 10-bar truss problem

### 4.2 다목적함수 최적화

다목적함수 최적화문제는 하나의 최적해가 존재하는 것이 아니라 다수의 해가 최적해가 되며 이를 Pareto 최적해 집합이라 한다. 기존에는 다목적함수 최적해 집단을 구하기 위해 다목적함수 문제를 임의의 설계상수를 기준으로 단일 목적함수 문제화하여 설계상수를 변경하여 주어 최적해 집합을 구하였다. 그리고 다목적함수 문제를 GA에 의한 다목적함수 최적화 방법으로 MOGA

(Multicriteria Optimization by Genetic Algorithm)가 개발되었다.[6][10] MOGA는 설계변수를 2진수형 코드를 사용하였다. MOGA의 특징은 Pareto 최적해 정의를 직접 사용하고, 각 설계점의 적합성을 현 세대에서의 Pareto 최적해정의를 만족여부에 따라 인위적으로 부여하였으며, 현 세대의 최적해 집합은 다음 세대에 그대로 전달하여 준다. MOGA는 탐색이 다수의 설계점에 의해 행해지는 GA의 특성을 이용하여 한번의 최적화 계산으로 최적해 집합을 구할 수 있는 장점이 있다. 본 연구에서는 MOGA의 최적화 과정을 따라 2진수형 설계변수 대신 실수형 설계변수를 직접 사용하였으며, 그 결과를 기존의 2진수형 설계변수를 사용한 결과와 비교 검토하였다.

#### 4.2.1 Non-convex 문제

본 문제는 Pareto 최적해 집합이 non-convex한 형태를 가지며, 3부분으로 분리되어 구성되어 있으며, 다음과 같이 표현된다.

$$\begin{aligned}
 &\text{minimize} && F_1 = -x_1 + 5x_2 \\
 &&& F_2 = x_1 + x_2 \\
 &\text{subject to} && g(x) = g_1(x) \cup g_2(x) \cup g_3(x) \\
 &\text{where} && g_1(x) = 1 - x_1^2 - x_2^2 \geq 0 \\
 &&& g_2(x) = 1 - (x_1 - 1)^2 - x_2^2 \geq 0 \\
 &&& g_3(x) = 1 - (x_1 - 2)^2 - x_2^2 \geq 0 \\
 &&& 0 \leq x_1 \leq 5.0, \quad -2.0 \leq x_2 \leq 2.0
 \end{aligned}$$

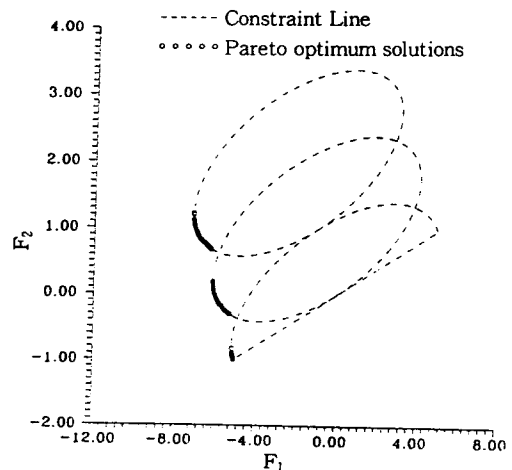


Fig. 5 Pareto optimal solutions of the non-convex problem with binary code MOGA



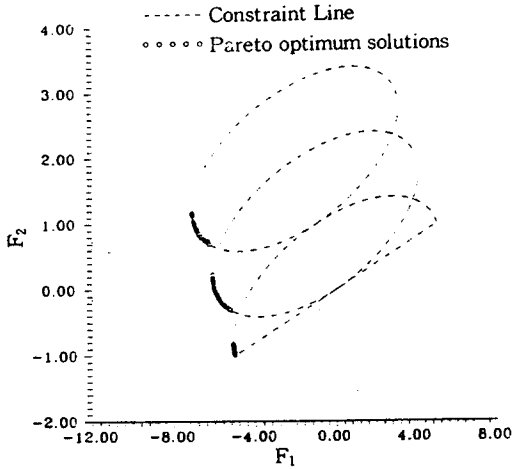


Fig. 6 Pareto optimal solutions of the non-convex problem with real code MOGA

2진수형과 실수형 MOGA, 두 방법 모두 300개의 최적해를 구하였으며, 그 결과를 각각 Fig. 5과 Fig. 6에 도시하였다. 그림에서 점선 내부가 설계 가능영역으로 두 방법 모두 매우 좋은 결과를 보여 주고 있음을 알 수 있다. 그리고 세대가 진행함에 따라 각 세대의 설계점중 Pareto 최적해의 갯수의 수렴도를 비교하여 보았는데 Fig. 7에서도 알 수 있듯이, 실수형 MOGA가 본 문제에서는 더 느린 속도로 수렴해 갔으나, 계산 시간에서도 오히려 2진수형이 20.05초, 실수형이 5.01초로 실수

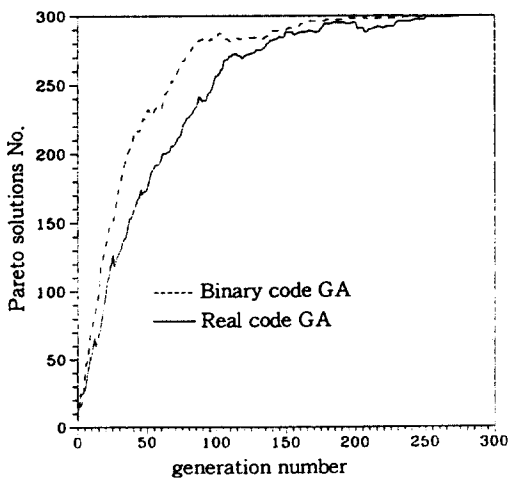


Fig. 7 Pareto solutions No. vs. generation No. of the non-convex problem

형 변수를 사용한 방법이 적게 소요되었다. 이는 2진수형 변수를 사용함으로써 해서 이전 세대의 필요한 정보를 복제하고, 실제 실수형 값으로 매핑할 때 그리고 비트 단위로 행해지는 교배과정에 요구되는 시간이 그만큼 많이 소요되었기 때문이라 사료된다.

4.2.2 10-bar 트러스 문제

4.1.3절에서 사용한 트러스 구조물(Fig. 3)에 대해 구조물의 중량과 변위를 목적함수로 하여 다목적함수 최적설계를 행하였다. 설계변수(x)는 각 부재의 단면적이며, 제한조건은 각 부재에 작용되는 응력이 허용응력을 넘지 않아야 하고, 6번 절점의 y방향 변위( $\delta_6^y$ )가 일정한도를 초과하지 않도록 하였다.

$$\begin{aligned}
 &\text{Minimize} && \text{Weight} = \gamma \sum_{i=1}^{10} x_i l_i \\
 &&& \text{Displacement} = |\delta_6^y| \quad (9) \\
 &\text{subject to} && g_1(x) = 1 - \frac{|\sigma_i|}{\sigma_a} \geq 0, \\
 &&& i = 1, 2, \dots, 10 \\
 &&& g_{11}(x) = \frac{|\delta_6^y|}{1.8} - 1 \geq 0 \\
 &&& g_{12}(x) = 1 - \frac{|\delta_6^y|}{6.0} \geq 0 \\
 &&& 0.1 \leq x_i \leq 35.0, \quad i = 1, 2, \dots, 10
 \end{aligned}$$

대상 설계점의 수를 500개로 하였으며, 중량의 최소점 부근에 국부탐색 능력을 강화하기 위해 각 세대마다 구해진 최적해 집합중 중량의 최소점에는 적합성(fitness)을 다른 설계점의 10배를 부가하였다. 두 방법에 의한 결과를 Fig. 8에 도시하였다. 중량의 최소점 부근에서 실수형 MOGA가 더 정확한 해를 구했음을 알 수 있다. 또한 두 방법에 대해 세대가 진행됨에 따라 각 세대의 설계점중 Pareto 최적해 갯수의 수렴도를 비교하였더니, Fig. 9에서와 같이 두 방법이 비슷한 정도로 수렴하였다. 계산 시간은 2진수형 MOGA가 344.55초 소요되었고, 실수형이 200.75초 소요되어, 2진수형 설계변수로 정의된 문제에서 2진수형 MOGA의 변환과정이 많은 시간을 요구함을 알 수 있다.

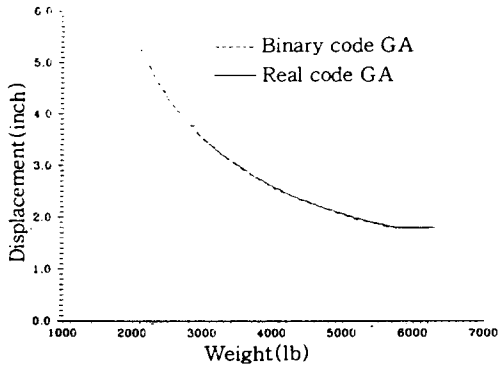


Fig. 8 Pareto optimal solutions of the 10-bar truss problem

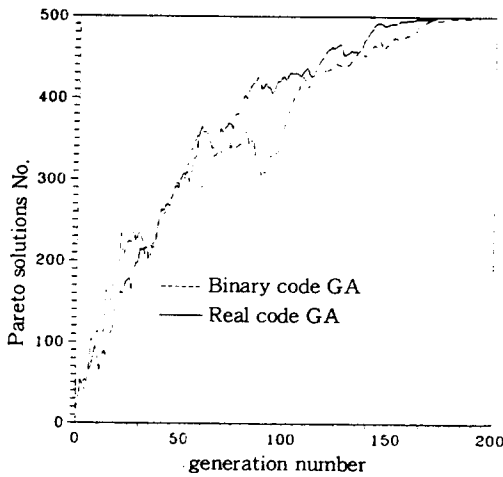


Fig. 9 Pareto solutions No. vs. generation No. of the 10-bar truss problem

### 5. 결 론

본 연구에서는 GA에 의한 최적설계시 실수형 설계변수를 직접 사용하기 위해 새로운 교배와 돌연변이 과정을 개발하였다. 그리고 여러 형태의 실수형 설계변수로 구성된 단일 및 다목적함수 최소화 문제에 적용하여 그 결과를 기존의 2진수형 설계변수를 사용한 GA에 의한 결과와 비교, 검토한 결과 다음과 같은 결론을 얻었다.

(1) 실수형 GA는 2진수형 GA 사용시 필요한 변환과정에 요구되는 시간이 매우 적어졌고, 또한 컴퓨터 주 기억용량의 사용에도 효율적이어서 계산시간 면에서 효율적이다.

(2) 2진수형 GA는 국부 탐색 능력에 매우 약한 한계를 가지나, 실수형 GA는 설계변수의 증분량을 조절할 수 있는 기능을 가져 국부 탐색 능력이 강화되어 해의 정확도가 보다 우수하다.

(3) 앞으로의 연구과제는 실수형과 이산적 형태의 설계변수가 혼합된 문제의 효과적인 처리를 위해 실수형 GA와 2진수형 GA를 동시에 고려한 GA에 대한 연구가 요망된다.

### 참 고 문 헌

1. Holland, J. H., *Adaptation in Natural and Artificial Systems*, Univ. Michigan, Ann Arbor, MI, 1975.
2. Goldberg, D. E., *Computer-Aided Gas Pipeline Operation using Genetic Algorithms and Rule Learning*, Ph. D. thesis, Dept. Civil Eng., Univ. Michigan, 1983.
3. Goldberg, D. E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, 1989.
4. 양영순, 김기화, "전체최적화를 위한 확률론적 탐색기법," 전산구조공학, 제5권, 제2호, 1992.
5. 양영순, 김기화, 유원선, "유전적 알고리즘에 의한 선체 구조물의 이산적 최적설계", 대한조선학회논문집, 제31권, 제4호, 1994.
6. 김기화, "Genetic Algorithm에 의한 다목적함수 최적구조설계", 서울대학교 조선해양공학과 박사학위논문, 1994.
7. Mockus, J., *Bayesian Approach to Global Optimization*, Kluwer Academic Publishers, 1989.
8. Pierre, D. A. and Lowe, M. J., *Mathematical Programming via Augmented Lagrangians: An introduction with Computer Programs*, Addison-Wesely, 1975.
9. Haug, E. J. and Arora, J. S., *Applied Optimal Design*, John Wiley & Sons, 1979.
10. 김기화, "유전적 알고리즘을 이용한 최적 구조설계", 대한조선학회지, 제31권, 제1호, 1994.

(접수일자 : 1995. 3. 4)