

유전자알고리즘에서의 실수처리 방법 비교

장 수 현[†] · 윤 병 주^{††}

요 약

초기의 유전자알고리즘은 염색체를 이진 스트링을 사용하여 표현하였다. 그러나 이진 스트링 표현으로는 대부분 실수(real number) 표현을 사용하는 실세계 문제들을 해결하기 어려웠기 때문에, 이를 해결하기 위해 실수 값을 염색체에 표현하는 여러가지의 다양한 표현방법들과 각 표현방법의 특성을 이용한 연산자들이 제시되었다. 본 논문은 여러가지 표현방법들과 연산자들이 각각 어떤 문제에 적합한지를 실험을 통하여 알아보았으며, 또 다른 방법으로, 실수 영역을 정수부와 소수부로 분할하여 탐색하는 염색체 표현방법과 탐색전략을 제시하였다.

A Comparative Study on Real-number Processing Method in Genetic Algorithms

Suhyun Jang[†] · Byungjoo Yoon^{††}

ABSTRACT

In early research results of genetic algorithms, binary representation has been used for chromosome. However, binary representation has been recognized to have some weak points to solve real-world applications which would be represented with real number. Although many representation schemes and operators using characteristics of each scheme have been proposed in order to solve real-world problems effectively, there was no method confirmed widely in machine learning society. In this paper, we study which chromosome representation schemes and operators for real-number processing are appropriate for a specific problem. And we propose another chromosome representation scheme and search strategy to search integer part and decimal part separately, and then to recombine each chromosome to find a solution.

1. 서 론

1975년 Holland에 의해 소개된 유전자알고리즘(genetic algorithm)은 자연 생태계의 진화이론을 이용한 탐색과정이라 할 수 있다[1]. 유전자알고리즘은 탐

색공간에서 찾고자 하는 해의 후보를 나타내는 염색체와 염색체들 사이의 정보교환을 위한 연산자들, 염색체의 적합성(fitness)을 나타내는 함수 등으로 이루어져 있다. 유전자알고리즘에서 염색체는 이진 스트링(binary string)으로 표현되며, 염색체를 처리하기 위한 연산자로는 교배(crossover) 연산자와 돌연변이(mutation) 연산자 등이 있다. 연산자들은 염색체 표현방법이나 적용할 문제의 특성에 따라 변형되어 사용되기도 한다.

초기의 유전자알고리즘은 염색체를 이진 스트링으

*본 연구는 명지대학교 산업기술연구소의 '97 교내 연구비 지원으로 수행된 결과임.

† 정 회원: 명지대학교 컴퓨터공학과

†† 총신회원: 명지대학교 컴퓨터공학과 교수

논문접수: 1997년 5월 2일, 심사완료: 1997년 11월 26일

로만 표현하였으므로 공학용 수치 계산 문제나 다차원 함수의 최소값을 찾는 문제들은 해결하기가 매우 어렵다[16, 17]. 이를 해결하기 위해 실수 값을 염색체에 표현하는 여러가지 방법이 제시되었다. 실수 값을 염색체에 표현하기 위한 방법은 일반적으로 ① 실수 값을 이진수로 변환하여 표현하는 방법[3], ② 실수 값을 짧은 이진 스트링으로 변환하여 현재의 탐색공간으로 사상시키는 방법[4], ③ 실수 값을 유전자에 직접 표현하는 방법[3] 등으로 분류할 수 있다.

본 논문에서는 실수 값을 염색체에 표현하기 위한 여러가지 방법들과 염색체 표현방법에 따라 변형된 연산자들이 각각 어떤 문제에 적합한지 알아보았다. 또, 기존의 염색체 표현방법들의 단점을 보완하고 장점을 이용하여 실수 영역을 정수부와 소수부로 분할하여 탐색하는 효율적인 표현방법과 탐색전략을 제안한다.

2. 유전자알고리즘

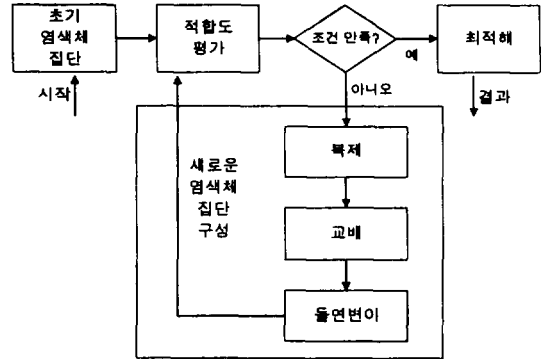
2.1 유전자알고리즘

유전자알고리즘은 자연 생태계의 자연 선택, 적자 생존(survival of the fittest)의 원리에 근거한 탐색 알고리즘이다. 즉, 유전자알고리즘은 한 세대를 구성하는 개체들 중에서 주어진 환경에 대해 적응력이 높은 개체의 유전인자 값이 다음 세대로 전해지는 진화 과정을 통하여 보다 우수한 개체를 생성하는 과정을 모델링한 것이다[1].

2.2 유전자알고리즘의 구조

(1) 탐색과정

유전자알고리즘은 탐색공간을 단일 탐색점으로 탐색하지 않고 여러 개의 탐색점으로 병행 탐색하며, 각 탐색점은 염색체 집단에 속한 하나의 염색체로 표시된다. 탐색과정 중에 각 염색체에 대하여 평가함수를 사용하여 적합도를 계산하며, 적합도가 높은 염색체는 증식되고 낮은 염색체는 도태된다. 선택된 염색체에 대해서는 교배연산과 돌연변이 연산을 수행하여 만족할 만한 수준의 적합도를 가진 염색체가 생성될 때까지 위의 과정을 반복한다. (그림 1)은 유전자알고리즘을 이용하여 염색체 집단으로부터 최적해를 얻는 과정을 나타낸다.



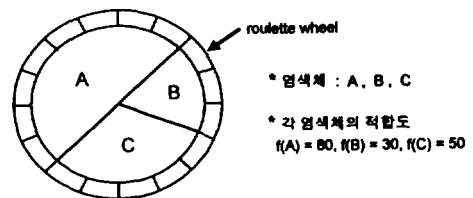
(그림 1) 유전자알고리즘의 구조
(Fig. 1) Structure of genetic algorithms

(2) 연산자

유전자알고리즘에서 주로 사용하는 3가지 기본 연산자는 복제 연산자 (reproduction operator), 교배 연산자 (crossover operator), 돌연변이 연산자 (mutation operator) 이며, 기본 연산자를 문제에 그대로 적용하거나 문제의 특성에 맞게 변형하여 적용한다.

① 복제 연산자

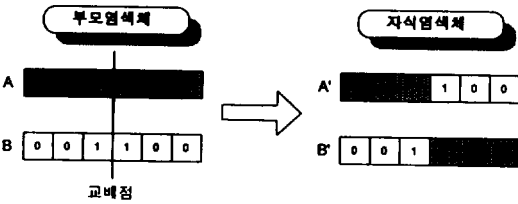
염색체의 적합도에 따라 다음 세대로 전달할 염색체를 생성하기 위한 연산자이다. 다음 세대로 전달할 염색체를 선택하기 위한 방법 중의 하나로 룰렛 휠 (roulette wheel)을 이용하는 방법이 있다. 이 방법은 각 염색체를 적합도에 비례하여 (그림 2)와 같이 룰렛 휠에 할당한다. 그러므로, 적합도가 높은 염색체가 다음 세대로 전달될 확률이 높게 된다. 그렇지만, 특정 염색체로 선택이 편중되는 것을 막기 위해, 경우에 따라서는 적합도가 높은 염색체가 도태될 수도 있는 방법이다.



(그림 2) 염색체의 적합도에 따른 룰렛 휠 할당 방법
(Fig. 2) Allocation method for roulette wheel according to chromosome's fitness

② 교배 연산자

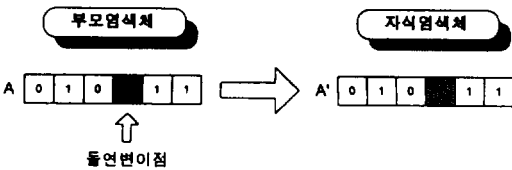
두 개의 부모 염색체를 결합하여 새로운 자식 염색체를 생성하기 위한 연산자이다. 이 연산자는 미리 정의된 교배 확률 P_c 에 따라 적용되고, 염색체의 분리 지점 (crossover site)은 같은 확률을 가지고 임의로 선택된다. 교배 연산의 예는 (그림 3)과 같다.



(그림 3) 교배 연산의 예
(Fig. 3) Example of crossover operator

③ 돌연변이 연산자

미리 정의된 돌연변이 확률 P_m 에 따라 적용되고, 임의로 선택된 유전자의 값을 바꾼다. 이 연산자는 전체 탐색공간내의 새로운 탐색점을 생성한다. 돌연변이 연산의 예는 (그림 4)와 같다.



(그림 4) 돌연변이 연산의 예
(Fig. 4) Example of mutation operator

3. 실수 값 처리를 위한 염색체 표현방법과 연산자

본장에서는 실수 값을 처리하기 위해 제시된 염색체 표현방법들과 각 표현방법들의 특성을 이용한 연산자들에 대해 설명한다. 즉, 이진수를 이용한 표현방법, 짧은 이진수를 실수 영역으로 사상시키는 방법 (이를, 동적 이진 표현이라고 한다), 실수 값 자체를 하나의 유전인으로 직접 표현하는 방법 등을 설명하고 있다. 또, 본 논문에서 제안하는 표현방법으로 정

수부와 소수부를 분할하여 탐색하는 탐색전략을 설명한다.

3.1. 이진수를 이용한 실수 표현방법

(1) 염색체 표현

이진수 표현방법은 실수 값을 이진 벡터 (binary vector)로 변환하여 염색체에 표현한다. 이 방법에서 적합도는 염색체를 실수 값으로 변환하여 평가한다. 이진수 표현방법은 염색체의 길이와 정확도가 비례하는 특징이 있다. 실수 값을 유전자(gene)에 표현하는 방법은 식 (1)을 이용하여 실수 값을 정수 값으로 변환한다. 그리고나서 식 (2)를 이용하여 유전자의 길이를 결정한다[3].

$$(U_i - L_i) \times 10^n \tag{1}$$

단, U_i : 실수의 상한(upper bound)

L_i : 실수의 하한(lower bound)

n : 실수의 소숫점 이하 자리수

$$(U_i - L_i) \times 10^n \leq 2^m - 1 \tag{2}$$

단, m : 유전자의 길이

예를들어, 이진 스트링(11001101₂)을 실수 값(3.11)으로 전환하는 과정은 (그림 5)와 같다. 여기서 실수 값의 정확도는 소수점이하 두자리까지를 요구하며, 변수의 범위는 -5.12에서 5.12까지이다. 즉, 변수의 크기가 10.24이고 소수점이하 두자리까지를 요구하기 때문에 10.24·100 으로 실수 구간을 변환해야 한다. 이 값을 이진 스트링으로 표현하기 위해서는 식 (3)과 같이 10비트가 요구된다.

$$512 = 2^9 < 1024 \leq 2^{10} = 1024 \tag{3}$$

이진수 $\langle b_9 b_8 \dots b_0 \rangle$ 을 -5.12에서 5.12까지 범위의 실수 값으로 사상시키는 과정은 2단계로 이루어 진다.

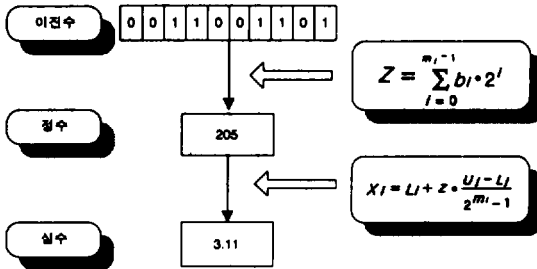
[단계 1] 이진수 $\langle b_9 b_8 \dots b_0 \rangle$ 을 정수 값으로 전환한다.

$$\langle \langle b_9 b_8 \dots b_0 \rangle \rangle_2 = \left(\sum_{i=0}^9 b_i \cdot 2^i \right)_{10} = x'$$

[단계 2] 정수 값을 실수 값으로 전환한다.

$$x = -5.12 + x' \cdot \frac{10.24}{2^{10}-1}$$

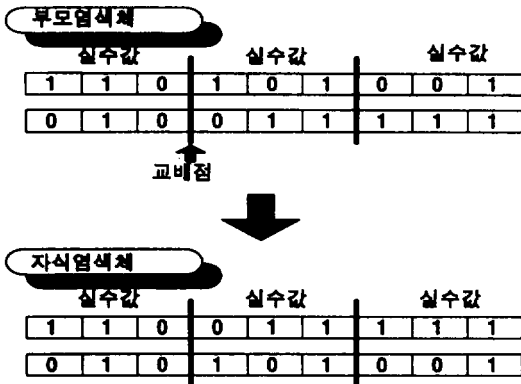
(실수 값의 하한: -5.12, 변수의 크기: 10.24일 때)



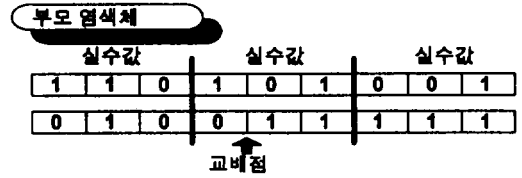
(그림 5) 11001101₂을 3.11₁₀로 전환하는 과정
 (Fig. 5) The conversion process from 11001101₂ 3.11₁₀

(2) 교배 연산자와 돌연변이 연산자

교배 연산자 (crossover operator)는 교배가 일어날 교배점 (crossover point)을 결정하는 방법에 따라, (그림 6)의 ①과 같이 실수 값 사이의 경계가 되는 부분을 교배점으로 하여 교배시키는 방법(즉, 교배할 염색체들 사이에서 실수 값 자체를 서로 바꾸어 주는 방법이다. 그러나 이 방법은 유전자에 표현된 실수 값의 범위가 한정되어 있기 때문에 돌연변이 연산자 없이는 많은 탐색점을 생성할 수 없다)과 (그림 6)의 ②와 같이 염색체에서 임의의 교배점을 발생하는 방



① 실수 값 단위로 교배점 발생



② 임의의 교배점 발생

(그림 6) 이진 표현방법의 교배연산자
 (Fig. 6) Crossover operator of binary representation

법이 있다[5]. 돌연변이 연산자 (mutation operator)는 임의의 비트를 선택하여 비트 값을 바꿈으로써 탐색 점을 바꾸는 역할을 하는 연산자이다.

3.2 동적 이진 표현방법

(1) 염색체 표현

동적 이진 표현 (dynamic parameter encoding)은 실수 값을 짧은 이진 스트링으로 변환하여 탐색공간으로 사상시키는 방법이다. 이진 스트링으로 표현된 염색체를 실수로 변환하는 방법은 식 (4)와 같다[4].

$$T(i) = L + (U - L) \frac{i + X}{2^i} \tag{4}$$

단, U: 현재 탐색공간의 상한

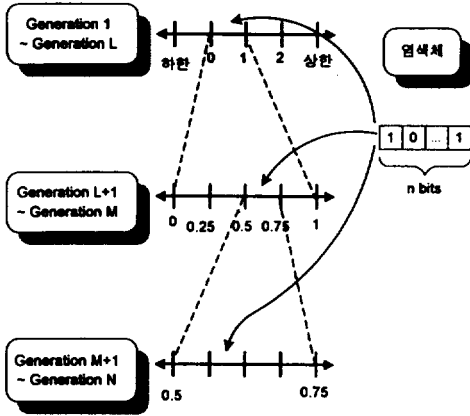
L: 현재 탐색공간의 하한

i: 실수 값을 표현하기 위한 이진 벡터의 길이

i: 이진수로 표현된 염색체를 정수로 바꾼 값

X: 0과 1사이의 임의 값

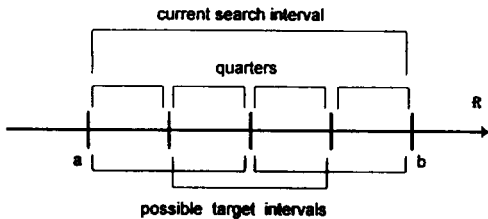
동적 이진 표현의 탐색 방법은 (그림 7)과 같이 염색체가 일정 세대를 진화함에 따라 적합도가 높은 염색체들이 집중하는 공간으로 탐색영역을 축소하면서 탐색하는 방법이다.



(그림 7) 동적 이진 표현의 탐색 과정
(Fig. 7) Search process of dynamic parameter encoding

(2) 교배 연산자와 돌연변이 연산자

동적 이진 표현에 적용될 연산자는 염색체들이 이진수로 표현되었기 때문에 3.1에서 소개된 연산자와 같은 교배 연산자와 돌연변이 연산자이다. 이 연산자 이외에 동적 이진 표현 방법은 줌 (zoom) 연산자라는 특수한 연산자를 사용한다[14]. (그림 8)과 같이 현재 탐색영역을 4구간으로 나누고 이 구간을 다시 중복된 3구간으로 나눈다. 일정 세대를 지나도록 3구간 각각에서 발생한 염색체의 수를 계산하여 미리 정해둔 기준치를 넘는 구간으로 탐색영역을 좁혀서 다음 세대부터는 그 구간에서만 탐색하는 방법이다.



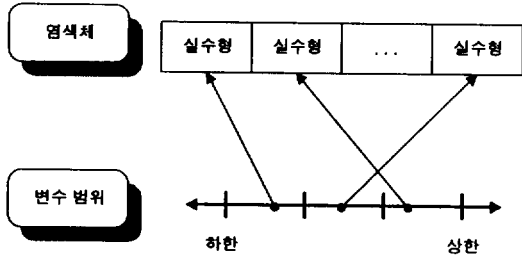
(그림 8) 동적 이진 표현의 줌 연산을 위한 간격
(Fig. 8) Interval for zoom operation of dynamic parameter encoding

3.3 실수를 직접 유전인자로 표현하는 방법

(1) 염색체 표현

실수를 직접 염색체에 표현하는 방법은 (그림 9)와 같이 유전자 하나에 하나의 실수값을 표현한다. 이

방법은 실수를 이진수로 표현하거나 이진수를 실수로 바꾸는 과정이 필요없기 때문에 값을 변환하기 위한 오버헤드가 없다.



(그림 9) 유전자에 하나의 실수 값을 직접 표현하는 방법
(Fig. 9) Direct representation of real value at gene

(2) 교배 연산자

직접 표현방법은 기존의 연산자 이외에 실수의 성질을 이용하기 위한 여러 연산자들이 있다[6, 10, 11]. 교배 연산을 위해 선택된 염색체는 $X=(x_1, \dots, x_n)$ 과 $Y=(y_1, \dots, y_n)$ 로 표기한다.

① 단순 교배 연산자

단순 교배 (simple crossover) 연산자는 이진 스트링 표현방법에서 적용된 방법으로, 1에서 $n-1$ 사이에서 임의의 교배점을 결정하여 그 교배점에서 교배하는 연산자이다. k 번째 교배점에서 교배한 염색체는 다음과 같다.

$$X' = (x_1, \dots, x_k, y_{k+1}, \dots, y_n)$$

$$Y' = (y_1, \dots, y_k, x_{k+1}, \dots, x_n)$$

② 산술 교배 연산자

산술 교배 (arithmetical crossover) 연산자는 두 염색체의 일차 조합으로 다음 세대 염색체를 생성한다 [7]. 즉, 이 연산자는 두 염색체를 두 점으로 간주하고 두 점을 연결하는 직선상의 임의의 두 점으로 다음 세대 염색체를 생성한다. 생성된 염색체는 다음과 같다.

$$X' = a \cdot X + (1-a) \cdot Y$$

$$Y' = a \cdot Y + (1-a) \cdot X$$

$a: 0$ 과 1 사이의 임의의 값으로 상수 또는 변수

③ 평균 교배 연산자

평균 교배 (average crossover) 연산자는 두 유전자 값의 평균으로 다음 세대 유전자를 생성한다[12]. 예를들어, 3, 5, 6 번째 유전자에 이 연산자가 적용될 경우, 생성된 염색체는 다음과 같다.

$$X' = \left(x_1, x_2, \frac{(x_3 + y_3)}{2}, x_4, \frac{(x_5 + y_5)}{2}, \frac{(x_6 + y_6)}{2} \right)$$

$$Y' = \left(y_1, y_2, \frac{(x_3 + y_3)}{2}, y_4, \frac{(x_5 + y_5)}{2}, \frac{(x_6 + y_6)}{2} \right)$$

④ 휴리스틱 교배 연산자

휴리스틱 교배 (heuristic crossover) 연산자는 탐색의 방향을 유도하기 위해 선택된 염색체의 적합도를 이용해서 다음 세대 염색체를 생성한다. 생성된 염색체의 값이 탐색 범위내에 들어오지 않는다면 r 값을 다시 발생하여 다른 염색체를 생성한다. X 의 적합도가 Y 의 적합도보다 높을 때 생성된 염색체는 다음과 같다.

$$X' = X + r(X - Y)$$

$$Y' = X$$

r : 0과 1사이의 임의 값 $16(l/j/5)$

(3) 돌연변이 연산자

돌연변이 연산을 위해 선택된 염색체는 $X = (x_1, \dots, x_n)$ 로 표기한다.

① 균일 돌연변이 연산자

균일 돌연변이 (uniform mutation) 연산자는 이진 스트링 표현방법에서 적용된 방법으로 1에서 $n-1$ 사이에서 임의의 돌연변이 점을 발생하여 실수 값의 범위내의 임의의 값으로 변환한다. 이 연산자는 탐색 초기에 탐색점들을 균일하게 발생하는 특징이 있다.

② 경계 돌연변이 연산자

경계 돌연변이 (boundary mutation) 연산자는 선택된 유전자를 실수 값의 상한값이나 하한값으로 변환한다. 이 연산자는 최적해가 탐색공간의 경계부분에 위치해 있을때 이용하기 위해 만들어진 연산자이다.

③ 불균일 돌연변이 연산자

불균일 돌연변이 (non-uniform mutation) 연산자는 유전자의 값이 진화 초기에는 광범위한 실수 값으로 바뀌고 세대가 지날수록 좁은 범위의 실수 값으로 바

뀐다. 그러므로 이 연산자는 미세조정 능력을 갖는 연산자이다[8]. 예를들어 k 번째 유전자를 돌연변이 시키고자 할 때, 유전자의 값을 바꾸는 방법은 다음과 같다.

$$x'_k = \begin{cases} x_k + \Delta(t, UB - x_k) & \text{if } random = 0 \\ x_k - \Delta(t, x_k - LB) & \text{if } random = 1 \end{cases}$$

$$\Delta(t, y) = y \cdot (1 - r^{(1 - \frac{t}{T})^b})$$

단, UB : 변수의 상한, t : 현재 세대,

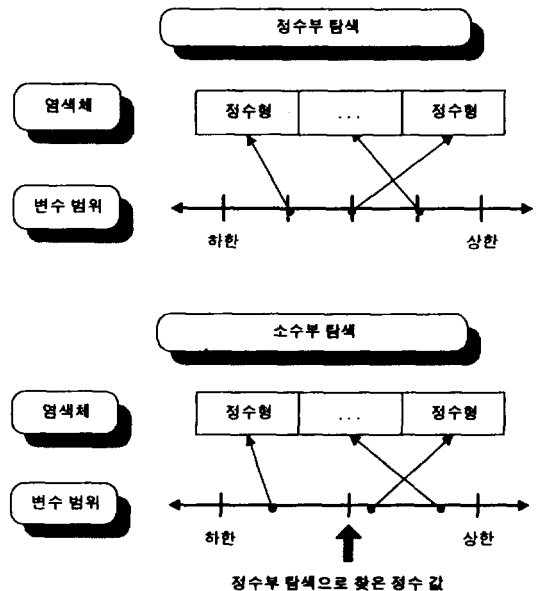
r : 0과 1사이의 임의의 값, LB : 변수의 하한,

T : 최대 세대, b : 반복수행의 의존도

3.4 정수부와 소수부 분할 탐색에 의한 실수 표현

(1) 염색체 표현

실수 공간의 탐색점들을 염색체에 표현하고나서 최적해를 찾는 것은 탐색공간이 매우 방대하다. 그래서 본 논문에서는 실수 공간을 탐색하기 위해 새로운 유전자 표현방법과 탐색전략을 제안한다. 이 방법은 탐색공간을 정수 범위로 좁히기 위해 실수 값의 정수 부만을 유전자에 표현해서 정수 값에 대한 탐색을 수



(그림 10) 정수부와 소수부 분할 탐색 (Fig. 10) Two-way searching by dividing integer and fraction part

행한다. 그리고나서 실수 값의 소수부를 유전자에 표현하여 소수 값에 대한 탐색을 수행한다. 정수부와 소수부의 탐색이 완료되면 정수부와 소수부를 결합하여 최적해로 해석한다. 염색체 표현방법과 탐색과정은 (그림 10)과 같다.

(2) 교배 연산자와 돌연변이 연산자

교배 연산자와 돌연변이 연산자는 3.3절의 직접 표현에서 사용한 방법을 이용하였다. 교배 연산자는 교배할 두 염색체 사이에서 유전자들을 교배하는 단순 교배 연산자, 일차 조합으로 다음 세대의 유전자를 생성하는 산술 교배 연산자, 평균으로 유전자를 생성하는 평균 교배 연산자, 그리고 적합도를 이용하여 유전자를 생성하는 휴리스틱 교배 연산자가 있다. 돌연변이 연산자는 선택된 유전자를 임의의 정수 값으로 변환하는 균일 돌연변이 연산자, 정수의 상한이나 하한 값으로 변환하는 경계 돌연변이 연산자, 그리고 세대 초기에는 큰 범위의 정수 값으로 변환하고 세대가 지날수록 좁은 범위의 정수 값으로 변환하여 주는 불균일 돌연변이 연산자가 있다.

4. 실험 및 토의

4.1 실험 환경

3장에서 설명한 각 표현 방법들의 효율성을 검증하기 위해 De Jong이 제시한 5가지의 서로 다른 특성을 갖는 함수를 이용하였다[1, 9, 13, 15]. 각 함수와 변수 범위는 (표 1)과 같다. 실험에 사용한 파라메타는 한 세대의 염색체 수는 30개이고, 허용 오차는 (목표값 - 실제값) ≤ 0.00001 이거나 최대 1000세대 미만으로 설정하여 실험하였다. 각 염색체 표현방법에 적용되는 교배 연산자는 0.1에서 0.8의 확률로, 돌연변이 연산자는 0.003에서 0.8의 확률로 조합하여 예비실험을 시행한 후, 그 결과 가장 좋은 연산자와 확률로 10번씩 실험하여 최상의 경우를 상호 비교하였다.

4.2 실험 결과 및 토의

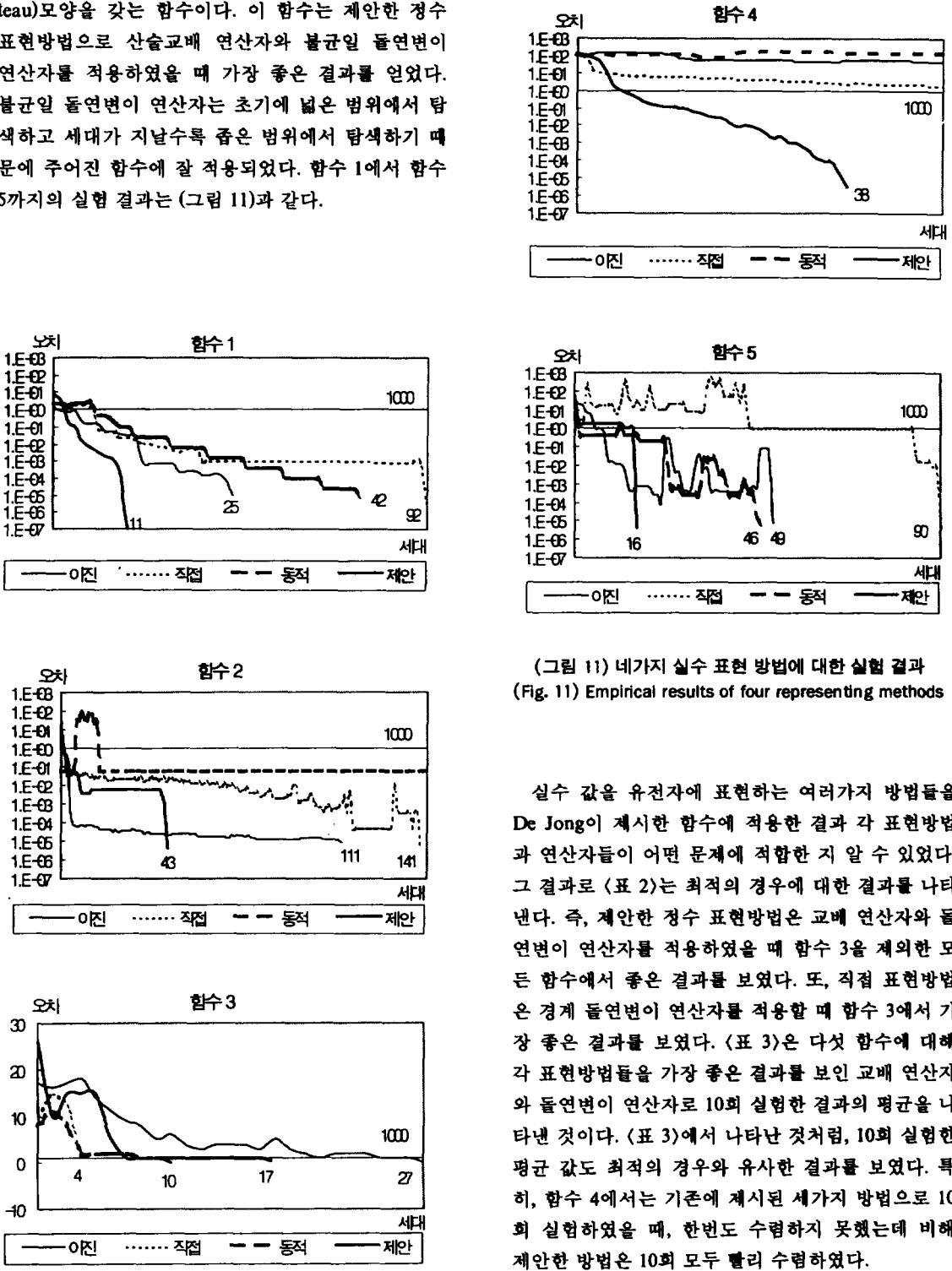
함수 1은 지역최소값이 없고 하나의 전역 최소값만 존재하는 문제이다. 이 함수는 이진 표현방법들이 직접 표현방법보다 잘 적용되었으며 제안한 정수 표현방법으로 평균 교배 연산자와 불균일 돌연변이 연

(표 1) De Jong이 제시한 다섯 가지 함수
(Table 1) Five test-bed functions suggested by De Jong

함수 1	$f_1(x_i) = \sum x_i^2$ [-5.12 ≤ x _i ≤ 5.12]
함수 2	$f_2(x_i) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$ [-2.048 ≤ x _i ≤ 2.048]
함수 3	$f_3(x_i) = \sum integer(x_i)$ [-5.12 ≤ x _i ≤ 5.12]
함수 4	$f_4(x_i) = \sum i x_i^4 + Gauss(0,1)$ [-1.28 ≤ x _i ≤ 1.28]
함수 5	$f_5(x_i) = [0.002 + \frac{1}{j + \sum_{i=1}^j (x_i - a_{ij})^6}]^{-1}$ $a_{1j} = 16(j \bmod 5 - 2), a_{2j} = 16(\lfloor j/5 \rfloor)$ [-65.536 ≤ x _i ≤ 65.536]

산자를 적용하였을 때 가장 좋은 결과를 얻었다. 평균 교배 연산자는 염색체군에 있는 우성의 실수 값들을 그들의 중심 값으로 변환하여 주기 때문에 효율적인 탐색이 가능하였다. 함수 2는 전역최소값보다 조금 큰 지역최소값이 하나 존재하는 문제이다. 이 함수는 제안한 정수 표현방법으로 산술 교배 연산자와 균일 돌연변이 연산자를 적용하였을 때 가장 좋은 결과를 얻었다. 지역 최소값은 균일 돌연변이 연산자로 극복할 수 있었다. 함수 3은 계단형으로 이루어진 함수이다. 이 함수는 직접 표현방법으로 교배 연산자에 큰 관계없이 경계 돌연변이 연산자를 큰 확률로 적용하였을 때 가장 좋은 결과를 얻었다. 찾고자 하는 값이 허용범위의 경계부분에 있었기 때문에 경계 돌연변이 연산자가 잘 적용됨을 알 수 있었다. 함수 4는 가우스 잡음(Gaussian noise)이 있는 문제이다. 이 함수는 제안한 정수 표현방법으로 평균 교배 연산자와 불균일 돌연변이 연산자를 적용하였을 때 가장 좋은 결과를 얻었다. 이 함수는 다변수 함수이기 때문에 동적 이진 표현의 경우 탐색영역의 축소에 문제가 발생하였다. 그래서 허용오차를 만족하는 최소값을 찾을 수 없었다. 그러나 제안한 정수 표현방법은 허용오차를 만족하는 최소값을 쉽게 찾을 수 있었다. 함수 5는 깊이가 다른 몇 개의 구멍이 있는 큰 평원(pla-

teau)모양을 갖는 함수이다. 이 함수는 제안한 정수 표현방법으로 산술교배 연산자와 불균일 돌연변이 연산자를 적용하였을 때 가장 좋은 결과를 얻었다. 불균일 돌연변이 연산자는 초기에 넓은 범위에서 탐색하고 세대가 지날수록 좁은 범위에서 탐색하기 때문에 주어진 함수에 잘 적용되었다. 함수 1에서 함수 5까지의 실험 결과는 (그림 11)과 같다.



(그림 11) 네가지 실수 표현 방법에 대한 실험 결과
(Fig. 11) Empirical results of four representing methods

실수 값을 유전자에 표현하는 여러가지 방법들을 De Jong이 제시한 함수에 적용한 결과 각 표현방법과 연산자들이 어떤 문제에 적합한 지 알 수 있었다. 그 결과로 <표 2>는 최적의 경우에 대한 결과를 나타낸다. 즉, 제안한 정수 표현방법은 교배 연산자와 돌연변이 연산자를 적용하였을 때 함수 3을 제외한 모든 함수에서 좋은 결과를 보였다. 또, 직접 표현방법은 경계 돌연변이 연산자를 적용할 때 함수 3에서 가장 좋은 결과를 보였다. <표 3>은 다섯 함수에 대해 각 표현방법들을 가장 좋은 결과를 보인 교배 연산자와 돌연변이 연산자로 10회 실험한 결과의 평균을 나타낸 것이다. <표 3>에서 나타난 것처럼, 10회 실험한 평균 값도 최적의 경우와 유사한 결과를 보였다. 특히, 함수 4에서는 기존에 제시된 세가지 방법으로 10회 실험하였을 때, 한번도 수렴하지 못했는데 비해 제안한 방법은 10회 모두 빨리 수렴하였다.

〈표 2〉 다섯 가지 함수에 대한 실험 결과(최적의 경우)
 〈Table 2〉 Empirical results on five test-bed functions(best case)

	표현 방법	교배 (확률)	돌연변이 (확률)	세대수	오차
함수 1	이진 표현	임의 교배(0.6)	임의의 점(0.03)	25	0.00001
	동적 이진 표현	임의 교배(0.7)	임의의 점(0.03)	42	0.00001
	직접 표현	휴리스틱 교배(0.8)	균일(0.2)	92	0.00001
	제안한 방법	평균 교배(0.7)	불균일(0.7)	7	0.0
함수 2	이진 표현	유전자 교배(0.6)	임의의 점(0.05)	111	0.00001
	동적 이진 표현	임의 교배(0.7)	임의의 점(0.005)	1000	0.06
	직접 표현	산술 교배(0.6)	불균일(0.6)	141	0.00001
	제안한 방법	산술 교배(0.7)	균일(0.3)	43	0.00001
함수 3	이진 표현	임의 교배(0.6)	임의의 점(0.02)	27	0.0
	동적 이진 표현	유전자 교배(0.7)	임의의 점(0.01)	17	0.0
	직접 표현	단순 교배(0.7)	경계(0.8)	4	0.0
	제안한 방법	단순 교배(0.7)	경계(0.8)	10	0.0
함수 4	이진 표현	유전자 교배(0.6)	임의의 점(0.005)	1000	0.07
	동적 이진 표현	임의 교배(0.6)	임의의 점(0.003)	1000	29
	직접 표현	산술 교배(0.7)	불균일(0.6)	1000	0.0001
	제안한 방법	평균 교배(0.7)	불균일(0.7)	38	0.00001
함수 5	이진 표현	임의 교배(0.6)	임의의 점(0.1)	49	0.00001
	동적 이진 표현	유전자 교배(0.6)	임의의 점(0.05)	46	0.00001
	직접 표현	평균 교배(0.7)	불균일(0.6)	90	0.00001
	제안한 방법	산술 교배(0.1)	불균일(0.5)	16	0.00001

〈표 3〉 다섯 가지 함수에 대한 실험 결과(평균)
 〈Table 3〉 Empirical results on five test-bed functions(average case)

	표현 방법	평균 세대수	표준편차	1000세대의 계산으로도 수렴되지 않은 회수
함수 1	이진 표현	49.4	19.16	-
	동적 이진 표현	47	4.08	-
	직접 표현	218	147.59	-
	제안한 방법	9.6	4.27	-
함수 2	이진 표현	515.8	340.69	5
	동적 이진 표현	1000회 이상	-	10
	직접 표현	362.87	259.80	2
	제안한 방법	162.14	153.72	3

함수 3	이진 표현	89.3	34.20	-
	동적 이진 표현	27	9.12	4
	직접 표현	5.6	1.77	-
	제안한 방법	10.5	0.70	-
함수 4	이진 표현	1000회 이상	-	10
	동적 이진 표현	1000회 이상	-	10
	직접 표현	1000회 이상	-	10
	제안한 방법	45.9	9.53	-
함수 5	이진 표현	322.11	314.74	1
	동적 이진 표현	138	173.55	3
	직접 표현	283.55	246.55	1
	제안한 방법	72.7	56.32	-

5. 결 론

본 논문에서는 보다 정확한 값을 요구하는 실세계 문제들이나 공학용 수치계산 문제 등을 해결하기 위해 제시된 실수 값을 염색체에 표현하는 여러 방법들이 각각 어떤 유형의 문제에 적합한지를 알아 보았다. 그리고 실수 영역을 정수부와 소수부로 나누어 탐색하는 방법을 제시하였다. 기존에 제시된 세가지 표현방법으로 ①이진수를 이용한 실수 표현방법, ②동적 이진 표현방법, ③실수를 직접 유전인자로 표현하는 방법과 본 논문에서 제안하는 정수부와 소수부분 탐색에 의한 실수 표현방법을 De Jong이 제시한 5가지 표준 함수에서 실험하였다. 그 결과, 연속함수이며 전역최대값만을 갖는 함수 1은 직접 표현을 사용하였을 때 최적해로 가장 늦게 수렴하였으며, 제안한 정수부와 소수부를 나누어 탐색하며 평균 교배 연산자로 우성의 염색체들을 중앙으로 모아 주었을 때 가장 빨리 수렴하였다. 산등성이 모양의 함수 2는 동적 이진 표현을 사용하였을 때 지역 최소 값에 빠졌기 때문에 최적해로 수렴하지 못하였다. 이 함수도 제안한 방법으로 탐색하였을 때 최적해로 가장 빨리 수렴하였다. 함수 3은 최적해가 탐색공간의 경계부분에 있었기 때문에 실수 값을 직접 표현하여 경계 돌연변이 연산자를 높게 적용하였을 때 가장 빨리 최적해로 수렴하였다. 함수 4는 최적해 주위가 평원 모양을 갖는 함수로서 기존의 표현방법으로는 최적해로 수렴

하지 못했는데 비해 제안한 정수부와 소수부로 나누어 탐색하는 방법은 빨리 최적해로 수렴하였다. 그리고 함수 5는 여러개의 지역 최대 값이 존재하는 문제로 실수 값을 직접 표현했을 때 가장 늦게 수렴하였고 정수부와 소수부를 나누어 탐색하면서 불균일 돌연변이 연산자를 적용했을 때 가장 빨리 수렴하였다. 이와같이 여러 가지 유형의 문제에 어떤 표현방법이 적합한지를 알아보았고, 특히 제안한 정수부와 소수부분 탐색에 의한 실수 표현방법이 대부분의 함수에서 최적해로 빨리 수렴함을 알 수 있었다.

참 고 문 헌

- [1] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
- [2] Thomas Baeck and Hans-Paul Schwefel, "An Overview of Evolutionary Algorithms for Parameter Optimization", *Evolutionary Computation*, 1(1), pp. 1-23, 1993.
- [3] Zbigniew Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Third, Extended Edition, Springer-Verlag, 1995.
- [4] Nicol N. Schraudolph and Richard K. Belew, "Dynamic Parameter Encoding for Genetic Algorithms", *Machine Learning*, Vol. 9, pp. 9-21,

Kluwer Academic, 1992.

[5] Robert Hinterding, "Representation and Self-adaptation in Genetic Algorithms", Proceedings of the 1st Korea-Australia Joint Workshop on Evolutionary Computation, pp. 77-90, 1995.

[6] Zbigniew Michalewicz, "Evolutionary Operators for Continuous Convex Parameter Spaces", Proceedings of the 3rd Annual Conference on Evolutionary Programming, eds. A. V. Sebald and L. J. Fogel, River Edge, NJ, World Scientific Publishing, pp. 84-97, 1994.

[7] L.-J. Eshelman and J. D. Schaffer, "Real-Coded Genetic Algorithms and Interval-Schemata", in D. Whitley (Ed.), Foundations of Genetic Algorithms II, Morgan Kaufmann, San Mateo, CA, 1992.

[8] Cezary Z. Janikow, Zbigniew Michalewicz, "An Experimental Comparison of Binary and Floating Point Representation in Genetic Algorithms", ICGA'91, pp. 31-36, 1991.

[9] Joachim Sprave, "Linear Neighborhood Evolution Strategy", Proceedings of the 3rd Annual Conference on Evolutionary Programming, eds. A. V. Sebald and L. J. Fogel, River Edge, NJ, World Scientific Publishing, pp. 42 - 51, 1994.

[10] Stephen Smith, "An Evolutionary Program for a Class of Continuous Optimal Control Problems", ICEC'95, pp. 418 - 422, 1995.

[11] Yujiro Miyata, Takeshi Furuhashi and Yoshiki Uchikawa, "A Proposal of a New Local Improvement Mechanism for Genetic Algorithm and A Creative Thinking Support", ICEC'95, pp. 485-490, 1995.

[12] Jean-Michel Renders and Stephane P. Flasse, "Hybrid Methods Using Genetic Algorithms for Global Optimization", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 26, No. 2, April, 1996.

[13] Lashon Booker, "Improving Search in Genetic Algorithms", Davis, L., (Editor), Genetic Algorithms and Simulated Annealing, Morgan Kaufmann

Publishers, Los Altos, CA, 1987.

[14] David B. Fogel, *Evolutionary Computation*, IEEE Press, 1995.

[15] Kenneth A. De Jong and Mitchell A. Potter, "Evolving Complex Structures via Cooperative Coevolution", Proceedings of the 4th Annual Conference on Evolutionary Programming, pp. 307-317, 1995.

[16] David E. Goldberg, "Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking", Complex Systems, 5, pp. 139-168, 1991.

[17] Lawrence Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[18] B. Yoon, D. J. Holmes, G. Langholz and A. Kandel, "Efficient Genetic Algorithms for Training Layered Feedforward Neural Networks", Information Sciences, 76, pp. 67-86, 1994.



장수현

1993년 명지대학교 전자계산학과(공학사)
 1996년 명지대학교 대학원 컴퓨터공학과(공학석사)
 1997년~현재 명지대학교 대학원 컴퓨터공학과 박사과정

관심분야: Machine Learning, Knowledge-Based System



윤병주

1975년 경북대학교 수학과(학사)
 1982년 한국과학기술원 전산학과(석사)
 1994년 Florida State University 전산학과(박사)
 1975년~1977년 KAL 시스템개발실 근무

1982년~현재 명지대학교 컴퓨터공학과 부교수
 관심분야: Machine Learning, Knowledge-Based System, Hybrid Intelligent Systems