

동적인 교차 및 돌연변이 확률을 갖는 균일 교차방식 유전 알고리즘

A Genetic Algorithm with Uniform Crossover using Variable Crossover and Mutation Probabilities

김성수, 우광방
(Sung Soo Kim and Kwang Bang Woo)

Abstract : In genetic algorithms(GA), a crossover is performed only at one or two places of a chromosome, and the fixed probabilities of crossover and mutation have been used during the entire generation. A GA with dynamic mutation is known to be superior to GAs with static mutation in performance, but so far no efficient dynamic mutation method has been presented. Accordingly in this paper, a GA is proposed to perform a uniform crossover based on the nucleotide(NU) concept, where DNA and RNA consist of NUs and also a concrete way to vary the probabilities of crossover and mutation dynamically for every generation is proposed. The efficacy of the proposed GA is demonstrated by its application to the unimodal, multimodal and nonlinear control problems, respectively. Simulation results show that in the convergence speed to the optimal value, the proposed GA was superior to existing ones, and the performance of GAs with varying probabilities of the crossover and the mutation improved as compared to GAs with fixed probabilities of the crossover and mutation. And it also shows that the NUs function as the building blocks and so the improvement of the proposed algorithm is supported by the building block hypothesis.

Keywords : genetic algorithm, uniform crossover, variable crossover and mutation probabilities, building block hypothesis

I. 서론

GA에 대한 개념은 견고한 적응시스템에 대한 연구와 더불어 Holland[1] 와 Goldberg[2]에 의해 결실을 맺기 시작했다. 오늘날 많은 분야에서 GA가 최적화 기법의 하나로 쓰이고 있으나 초기에는 GA가 최적화 기법으로서 사용하기에는 제한적인 면을 갖고 있었다. 그 이유는 GA가 항상 최적해를 찾을 수는 없다는 문제점 때문이었다. 그러나 이 문제는 엘리트 선택과정(elitist selection procedure)에 의해 전역에 걸쳐 최적해를 탐색할 수 있는 최적화 기법으로 발전되었다[3].

GA 및 이의 응용과 관련된 많은 연구결과에 의하면 GA는 다른 최적화 기법들에 비해 오히려 더 뛰어난 성능을 발휘한 경우도 있다. 특히 견고한 제어문제에 있어서는 기존의 제어기법보다 더 우수한 결과를 나타낸다[4]. 이밖에 수개의 극부적인 해가 존재하는 multimodal문제, 어느 정도의 규칙성을 가진 regularity문제, 문제영역의 규칙성이 어느 정도 염색체로 표현 가능한 문제, 그리고 부분적인 해들 사이에 상대적 우위관계가 존재하는 epistatis문제 같은 경우에 GA가 다른 최적화기법들에 비해 더 효과적이다[5]. 최근엔 GA가 퍼지이론, simulated annealing, filtered beam search등과 결합하여 사용되는 예도 많다[6-8].

한편 공학분야에서 진화는 컴퓨터 상에서 시뮬레이션되는 최적화 과정을 의미한다. 이러한 인공진화에 기초한 알고리즘은 크게 GA, 진화 전략, 진화 프로그래밍의 세 가지로 분류된다. 이밖에 Scatter 탐색기법, 유전 프로그래밍을 통칭하여 진화 연산(Evolutionary Computation)[9], 진화 프로그램(Evolution Program)[10] 또는 진화 알고리즘(Evolu-

tionary Algorithms)[5]이라고 하며 이들 모두 예비해로 이루어진 초기 개체집단으로부터 시작한다는 공통점을 가지며, 새로운 해는 기존의 예비해를 임의로 변경시킴으로써 얻을 수 있다.

GA는 크게 선택 알고리즘, 교차 알고리즘 및 돌연변이 알고리즘으로 구성되는 일종의 확률적 탐색기법이다. 이들 각각의 과정을 어떻게 선택하느냐에 따라서 GA의 성능은 많은 영향을 받는다. 본 논문에서 제안하는 GA는 선택 알고리즘, NU개념에 입각한 균일 교차 알고리즘, 매세대마다 교차 및 돌연변이 확률값을 가변시켜주는 알고리즘 및 엘리트 알고리즘으로 이루어져 있다. 그리고 제안된 GA에서 교차의 균일도와 교차 및 돌연변이 확률의 가변화가 알고리즘의 성능에 미치는 영향을 검토한다.

II. 균일 교차 알고리즘

일반적으로 GA는 한 곳 또는 두 곳에서 교차가 진행되는다. 이를 한곳 교차(1-point crossover) 또는 두곳 교차(2-point crossover)라고 한다. 반면에 교차가 개체전체에 걸쳐서 여러곳에서 일어나는 것을 여러곳 교차(multipoint crossover)라고 하며, 여러곳 교차의 극단적인 예가 균일 교차(uniform crossover)다. 그림 1은 위의 여러가지 교차방법들을 나타낸다. 그림 1(a)는 교차점(굵은 선으로 표시된 곳)이 한 곳인 경우로 한곳 교차를 나타내며 두 개체는 교차점 이후의 모든 유전자 정보를 swapping 확률 $P=1$ 로써 서로 교환한다. 그림 1(b)는 교차점이 두곳인 경우로서 두곳 교차를 나타내며 두 교차점 사이에 있는 유전자의 정보가 확률 $P=1$ 로써 서로 교환된다. 그림 1(c)는 균일 교차를 행하는 경우로서 모든 유전자가 같은 확률 $P=0.5$ 로써 서로 정보를 교환한다. 그림 1(c)와 같은 균일 교차에 대한 개념은 이미 Ackley[11], Syswerda[12], Spears[13]등에 의하여 소개되었다. 이밖에 적응 교차(adaptive crossover)[14]를 비롯한

많은 교차 방법이 있다. 이들 여러가지 교차 방법들은 나름대로의 장단점을 갖고 있으며 균일 교차가 일반적으로 한곳 또는 두곳 교차보다 성능이 우수하다는 것이 알려져 있다 [12]. 그러나 문제에 따라서는 한곳 또는 두곳 교차가 더 나을 수도 있다. 이러한 경우에는 최적 교차 확률값이 상수값으로서 존재하는 경우라고 해석된다. 하지만 본 논문에서는 그림 1의 교차들에 대해서만 언급하기로 한다.

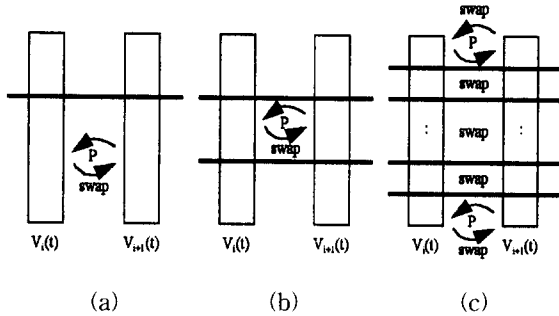


그림 1. 여러 교차방법: (a)한곳 교차, (b)두곳 교차, (c)균일 교차.

Fig. 1. Various crossover techniques: (a) 1-point crossover, (b) 2-point crossover, (c) uniform crossover.

인간의 DNA와 RNA는 복제시에 그림 1(c)와는 다소 다른 균일 교차를 행한다: RNA는 Uracil(U), Ade-nine(A), Guanine(G), Cytosine(C)이라는 4가지의 NU로 이루어져 있고 DNA의 경우에는 U대신 Thymine(T)을 포함한 네 가지의 NU로 구성되어있다. RNA와 같은 polynucleotide sequence는 복제과정에서 서로 마주 보는 두 NU가 C-G 또는 A-U인 경우에만 교차를 한다. 본 논문에서는 이 점에 착안하여 그림 2와 같은 균일 교차방식의 GA를 새로이 제안하고, 제안된 균일 교차 알고리즘은 그림 3, 그림 4를 통해 설명한다.

subprocedure PROPOSED UNIFORM CROSSOVER
begin

```

decide crossover probability randomly;
form nucleotide sequences randomly;
i = 0, j = 0
while i < cross do
begin
while j < nu do
begin
if nucleotidei,j and nucleotidei+1,j make a nucleotide pair
then swap the information between them;
else
preserve their own information;
j = j + 1;
end
i = i + 2;
end
end
end

```

where nucleotide_{i,j} : the jth nucleotide of the ith chromosome
cross : number of selected chromosomes to be crossed over
nu : number of nucleotides in a chromosome

그림 2. 제안된 균일 교차 알고리즘.

Fig. 2. A proposed uniform crossover algorithm.

제안된 균일 교차 알고리즘을 실행하기 위해서는 먼저 뉴

클레오티드의 종류가 결정되어야 한다. 그림 3, 그림 4에서는 4가지의 뉴클레오티드 A, C, G, U만을 고려했으나 일반적으로는 a가지의 뉴클레오티드를 생각할 수 있다. 그리고 제안된 알고리즘에서는 뉴클레오티드 가운데에서 마주보는 두 뉴클레오티드가 특정한 쌍을 이룰 경우 두 뉴클레오티드의 정보를 서로 교환하도록 특정한 뉴클레오티드쌍을 정의한다. 그림 3, 그림 4의 예에서는 A-U, C-G가 이러한 특정한 뉴클레오티드쌍이 된다. a = 2인 경우에는 제안된 균일 교차방식이 Ackley, Syswerda, Spears등이 제안한 균일 교차와 같아지나, 여러곳 교차에서 교차점을 한 bit 걸러 하나씩 고정한 경우와는 같지않다. 그것은 여러곳 교차에서는 swapping 확률값이 항상 1이기 때문이다. 이 가운데에서 Spears등이 제안한 균일 교차에서는 유전자들이 교차를 통해 정보를 교환할 확률값 P를 가변시켜주는 기법을 사용하였다. 그러나 이것도 역시 교차가 하나의 유전자 단위로 이루어지므로 기존 교차방식의 틀을 못 벗어났다고 볼 수 있다. 반면에 제안된 알고리즘에서는 교차확률값이 1/a로 주어지며 교차도 유전자단위가 아닌 뉴클레오티드(여러개의 유전자로 이루어짐)단위로 이루어진다는 점이 Spears 등의 교차 방식과 다르다.

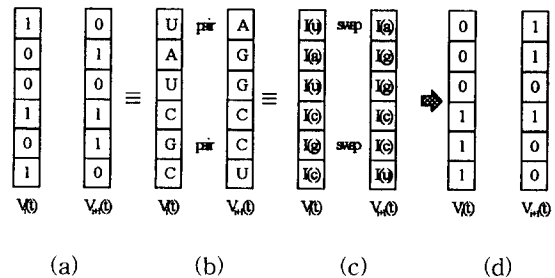


그림 3. 균일도가 1인 균일 교차.

Fig. 3. A uniform crossover with uniformity 1.

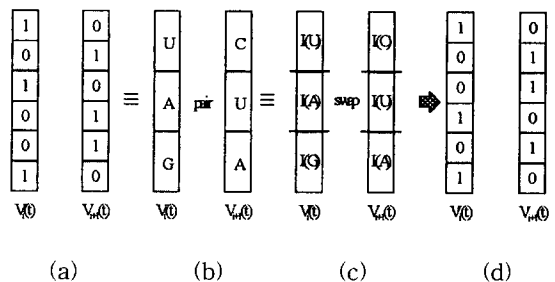


그림 4. 균일도가 1/2인 균일 교차.

Fig. 4. A uniform crossover with uniformity 1/2.

그림 2에서 cross는 교차대상으로 선택된 개체의 수를 나타내고, nu는 한 개체내의 뉴클레오티드의 갯수를 나타낸다. 개체집단으로부터 교차의 대상이 되는 개체가 선택될 확률값(일반적으로 교차 확률값이라고 함. swapping 확률값과는 구분됨.) P_c 는 매세대마다 $0 < P_c < 1$ 의 범위 안에서 임의로 가변시켜준다. 이렇게 함으로써 교차대상이 다양해져서 새로운 개체가 출현할 기회가 넓어지게 되어 최적값으로의 수렴속도에 영향을 미치게 된다. 그림 3에서는 nu=6이 되고 그림 4에서는 nu=3이 된다. 그림 3(a)는 교차를 행하기 전의 두 개체 $V_i(t)$, $V_{i+1}(t)$ 를 나타내며 이들은 통상적인 방법에 의해 P_c 의 확률로 교차대상으로 선택된 개체들이다. 그림 3(b)는 교차대상으로 선택된 두 개체의 모든 유전자에 4가지의 NU A,C,G,U가운데 하나씩을 임의로 할당한 상태를 나타낸다. 이때 4가지 NU의 발생 확률을 모두

갈게 설정하였으나 이들의 발생확률을 서로 다르게 설정할 수도 있다. 그림 3(c)는 그림 3(a)의 다른 표현으로 $I(u)$, $I(a)$, $I(g)$, $I(c)$ 는 각각 U, A, G, C의 정보 내용을 나타내며 이 예에서는 2진 값을 갖는다. 이 단계에서는 두 개체의 각각의 NU값을 순서대로 비교해서 같은 위치에 있는 두 NU값이 A-U 또는 C-G쌍을 이루면 이들 유전자가 가지고 있는 유전 정보 $I(i,j)$, $I(i+1,j)$ 를 서로 교환하고 이들이 위의 쌍을 이루지 않으면 자신의 정보를 그대로 간직한다. 이때 i 는 교차대상으로 선택된 개체의 일련 번호, j 는 각 개체 내에서의 NU의 위치를 나타낸다. 그리고 그림 3(d)는 이 결과 새로 태어난 자식개체 $V'_i(t)$, $V'_{i+1}(t)$ 를 나타낸다. 다른 개체들에 대해서도 위와 같은 방법으로 교차가 행하여진다. 그림 4에서도 같은 방법으로 교차가 일어난다.

제안된 균일 교차에서는 교차의 위치가 고정되어 있지 않고 매세대마다 임의로 정하여진다고 볼 수 있으며, 개체내의 모든 유전자는 NU가 a 가지 일 경우, $P = 1/a$ 의 확률값으로 정보를 교환하게 된다. 그리고 개체집단으로부터 교차대상이 되는 개체가 선택될 확률값 P_c 는 일반적으로 고정되어 있으나, 본 논문에서는 매세대마다 이 값을 임의로 가변시켜주기 때문에 disruption이 작아진다고 해석된다.

본 논문에서는 교차의 균일도 개념을 도입하였다. 즉 그림 3과 같이 하나의 NU가 하나의 유전자와 matching되는 경우 균일도가 '1'이라고 하고, 그림 4와 같이 두개의 유전자와 하나의 NU가 matching되는 경우에는 균일도가 '1/2'이라고 한다. 따라서 균일도가 '1/m'이면 m 개의 유전자에 하나의 NU가 matching된다. 제시된 알고리즘에서는 NU가 빌딩블록으로서 동작하게 되므로 균일도가 1인 경우보다는 균일도가 어느 정도 작은 경우에 목표값에 빨리 수렴한다. 그러나 균일도가 너무 작으면 목표값에 빨리 수렴하지 못한다. 그것은 빌딩블록 가설에 의해 설명될 수 있다. 즉 균일도가 적절히 작은 NU들은 길이가 짧고, 차수가 낮으며, 성능이 우수한 스키마로서 작용하기 때문에 이들의 결합에 의해서 원하는 스트링을 빨리 찾을 수 있게되나, 균일도가 너무 작아지면 스키마의 길이가 길어지고 차수도 높아져서 오히려 성능이 저하하게 되기 때문이다. 이러한 사실들로부터 제안된 균일 교차방식을 채용한 GA의 탐색능력은 커지게 됨을 알 수 있다. 균일 교차에 의해서 개체 $V_i(t)$ 가 개체 $V'_i(t)$ 로 바뀔 확률은 (1)과 같이된다.

$$P_1 = P^H \cdot (1 - P)^{M-H} \quad (1)$$

여기서 H 는 두 개체사이의 Hamming distance.

P 는 교차대상으로 선정된 개체의 각각의 유전자 정보가 바뀔 확률.

M 은 개체의 길이.

기존의 균일 교차에서는 균일도가 '1'이고 P 는 0.5이며, Spears등의 균일 교차에서는 균일도가 '1'이고 P 가 가변적이다. 본 논문에서 제안된 방법에 있어서는 균일도와 P 가 가변적이다. 그림 3, 그림 4의 예에서는 네가지의 NU를 도입하므로 P 는 0.25가 된다. 따라서 이 경우 교차에 의해 새로운 개체가 나타날 확률은 기존의 균일 교차방식 보다 제안된 균일 교차방식의 경우가 더 커지게 된다. 이것은 제안된 방식에 있어서 최적값에의 수렴속도가 기존의 균일 교차방식에 비해 전반적으로 빠름을 나타낸다.

GA에서 쓰인 대표적인 교차 확률값으로는 De Jong의 $P_c \approx 0.6$, Schaffer등의 $P_c \in [0.75, 0.95]$ 그리고 Grefenstette의 $P_c \approx 0.95$ 를 들 수 있다.

III. 돌연변이 알고리즘

GA에서는 교차 확률값과 같이 돌연변이 확률값도 고정

시켜 사용한다. 그러나 지금까지 이상적인 돌연변이 확률값이 알려져 있지 않기 때문에 각각의 경험에 비추어 가장 적당하다고 생각되는 값을 선택하여 사용하고 있다. 대표적인 P_m 값으로는 De Jong의 $P_m \approx 0.001$, Schaffer등의 $P_m \in [0.005, 0.01]$ 그리고 Grefenstette의 $P_m \approx 0.01$ 을 들 수 있다. 이에 반해 Mühlenbein은 최적 돌연변이 확률값을 계산하는 식으로 (2)를 제시하였다[15].

$$P_m \approx \frac{1}{M} \quad (2)$$

그러나 (2)는 적합도 함수가 unimodal pseudo-boolean function인 경우에는 최적의 돌연변이 확률값이 되지만, 적합도 함수가 multimodal pseudoboollean function인 경우에는 최적값이 되지 못한다. 그 이유는 multimodal 문제에 있어서 국부 최적값을 극복하기 위해서는 (2)이외의 다른 값이 더 우수할 수도 있기 때문이다. 그리고 (2)에서 문제가 되는 multimodality는 목적함수뿐만 아니라 encoding mechanism에 의해서도 야기될 수 있다. 하지만 세대가 지남에 따라서 돌연변이 확률값을 가변시켜주면 GA의 최적화가 가속화된다는 것이 실험적으로 입증되어 있다. Janikow[16] 등의 random mutation도 이를 뒷받침하고 있다. 그러나 random mutation은 본 논문에서 제시한 방법과 같이 돌연변이 확률값을 가변시켜주는 점이 같으나 이 기법은 실제로 표현되는 GA에 사용하기 위한 것으로서, 본 논문에서와 같이 돌연변이 확률값을 완전히 임의로 선정하는 것이 아니고, 개체에 대해 요구되는 update rate를 얻을 수 있도록 돌연변이 확률값을 가변시켜준다는 점에서 차이가 있다. 이제 돌연변이에 의해서 개체 $V'_i(t)$ 가 개체 $V''_i(t)$ 로 바뀔 확률을 구해보면 (3)이 된다.

$$P_2 = P_m^H \cdot (1 - P_m)^{M-H} \quad (3)$$

여기서 P_m 은 돌연변이 확률이다. 본 논문에서는 $0 < P_m < 1$ 의 조건이 만족되도록 돌연변이 확률값을 매세대마다 임의로 가변시켜주는 알고리즘을 제안하였다. 이에 따라 제안된 알고리즘에서는 국부 최적값을 종래보다 쉽게 벗어나서 전역 최적값으로 보다 빠르게 수렴하게 된다. 그 이유는 (3)에서 P_m 이 매세대마다 변화하므로 새로운 개체가 출현할 가능성이 종래방식보다 커지기 때문이다. 이 알고리즘은 기존의 GA들에 적용될 수 있으며, 돌연변이 확률값을 $0 < P_m < 1$ 과 같이 넓은 범위 안에서 변화시켜도 되고 또는 $0.001 < P_m < 0.1$ 과 같이 특정한 범위 내에서만 변화시켜도 된다.

위에 기술한 것처럼 교차 및 돌연변이 확률값을 가변시켜주면 이들 값을 고정시킨 경우보다 일반적으로 전역 최적값에 빨리 수렴하는 것이 사실이지만 그렇지 않은 경우도 있다. 이러한 현상은 unimodal한 문제에서 잘 나타나는데 그 이유는 특정한 문제에서는 최적 확률값이 상수값으로 존재하기 때문이다. 그러나 그 이외의 문제 특히 multimodal한 문제에 있어서는 최적의 교차 및 돌연변이 확률값이 상수값으로서 존재하지 않는다는 것이 경험상 알려져 있다. 따라서 이러한 경우 교차 및 돌연변이 확률값을 가변시켜주면 전역 최적해에 보다 빨리 수렴하게 된다. 이들 확률값을 특정한 함수값에 따라 가변시키는 기법도 있으나 이 역시 특정 함수에 구속당한다는 문제점이 있다. 따라서 본 논문에서는 교차 및 돌연변이 확률값을 임의로 가변시켜주므로써 이 문제를 해결한다.

이제 제안된 GA가 전역 최적해에 수렴하기 위한 필요충분조건을 살펴본다: $b \in IB^M = \{0, 1\}^M$, $f(b) \neq constant$ 인

모든 개체 b 에 대해 $0 < f(b) < \infty$ 를 가정하면 GA문제는 $\max \{f(b) \mid b \in IB^M\}$ 와 같은 형태의 최적화문제를 푸는 문제로 귀결된다. 상태공간 S 는 $S = IB^N = IB^{n \cdot M}$ 이고, 상태가 i 인 경우의 k 번째 개체를 $\pi_k(i)$ 로 나타낸다. 여기서 n 은 개체집단의 크기이고 M 은 개체의 길이이다. 이러한 최적화문제에 있어서 t 세대에서 상태가 i 인 경우, 개체집단내에서 최고의 적합도를 나타내는 random variable의 시퀀스를 Z_t 라고 하면, Z_t 는 (4)와 같이 나타낼 수 있다.

$$Z_t = \max \{f(\pi_k(i)) \mid k = 1, \dots, n\} \quad (4)$$

그리고 GA가 전역 최적해에 수렴하기 위한 필요충분조건은 (5)와 같이 나타낼 수 있다.

$$\lim_{t \rightarrow \infty} P\{Z_t = f^*\} = 1 \quad (5)$$

여기서 f^* 는 $\max \{f(b) \mid b \in IB^M\}$ 의 전역 최적해가 된다.

$i \in S$ 는 (6)을 만족시키는 임의의 상태를 나타내고 P_i^t 는 GA가 t 세대에서 상태가 i 일 확률을 나타낸다.

$$\max \{f(\pi_k(i)) \mid k = 1, \dots, n\} < f^* \quad (6)$$

그러면 위의 식들의 관계로부터 (7)이 성립하게 됨을 알 수 있다.

$$P\{Z_t \neq f^*\} = P_i^t \Leftrightarrow P\{Z_t = f^*\} = 1 - P_i^t \quad (7)$$

그런데 제안된 GA는 엘리트과정을 선택하므로, 이 알고리즘이 t 세대 까지의 최선해를 가지고 있을 확률은 1이 되어, $t = \infty$ 에서 GA가 상태 i 에 있을 확률은 (8)처럼 된다.

$$P_i^\infty = 0 \quad (8)$$

따라서 (7)에 (8)을 대입하면 (5)가 만족되어 주어진 GA가 전역 최적해를 가짐이 증명된다.

IV. 시뮬레이션 및 결과고찰

1. 기존 균일 교차방식과의 비교

본 논문에서 제안된 GA의 성능평가를 위하여, 여러사람이 test bed로서 자주 사용한 One Max문제, Two Max문제 및 수레 제어 문제를 선택하였다. 그리고 제안된 알고리즘과의 비교대상으로는 Holland의 GA 및 Syswerda의 균일 교차개념을 갖는 GA를 선택하였다. GA의 성능평가는 위의 문제들에 각각의 GA를 적용하여 이들이 최적값에 수렴하는 속도를 통해 이루어졌다. 그리고 제시된 알고리즘에 있어서 교차의 균일도가 알고리즘의 성능에 미치는 영향을 살펴보기 위해 균일도가 1, 1/2, 1/4, 1/8의 네 가지인 경우로 나누어서 시뮬레이션을 수행하였다. Holland와 Syswerda의 GA에 대해서는 교차 확률값이 1.0, 돌연변이 확률값이 1/M인 경우와 교차 및 돌연변이 확률을 본 논문에서 제안한 기법으로 가변시킨 경우로 나누어서 시뮬레이션을 수행하였으며 위의 모든 시뮬레이션은 IBM 486 PC상에서 이루어졌다. 시뮬레이션 내용을 요약하면 표 1과 같다. One Max, Two Max문제에서는 개체집단의 크기를 50으로 하였고, GA2, GA4, GA5 - GA8의 교차 및 돌연변이 확률값이 0과 1 사이에서 변화하도록 하였다. 그리고 최대 1000세대까지 진화를 시켰다. 그러나 수레 제어 문제에 있어서는 개체집단의 크기를 30으로 하였고, GA2, GA4, GA5 - GA8의 교차 확률값은 0과 1 사이에서, 돌연변이 확률값은 0.001과 0.1 사이에서 변화하도록 하였다. 이 경우엔 최대 300세대까지 진화를 시켰다. 그리고 위의 세 가지 문제들에 대한 모

표 1. 시뮬레이션 내용.

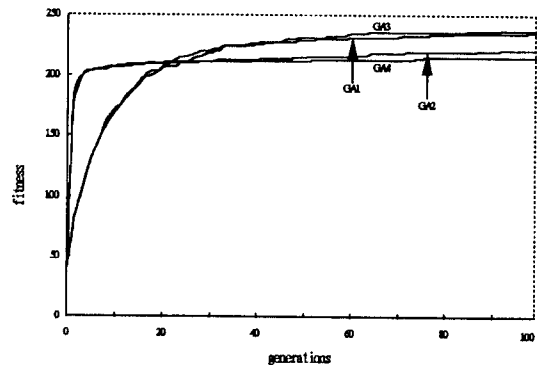
Table 1. Contents of simulation.

Genetic Algorithms	Crossover Probability	Mutation Probability	Uniformity
GA1(Holland)	1.0	1/M	-
GA2(Holland)	variable	variable	-
GA3(Syswerda)	1.0	1/M	-
GA4(Syswerda)	variable	variable	-
GA5(Proposed)	variable	variable	1
GA6(Proposed)	variable	variable	1/2
GA7(Proposed)	variable	variable	1/4
GA8(Proposed)	variable	variable	1/8

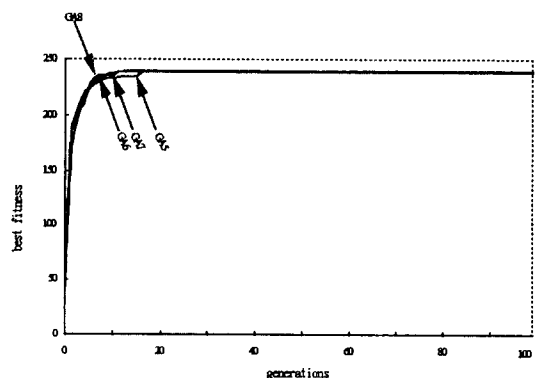
든 결과는 각각 20번씩 시뮬레이션해서 얻은 결과의 평균값이다. 그리고 매번 난수 발생기의 seed값을 바꾸어서 시뮬레이션을 수행하였다.

1.1 One Max문제

이 문제는 M 이 24이고 유전자값이 0과 1로 이루어진 개체 가운데에서 유전자값이 모두 1인 개체를 찾는 문제로서 unimodal 문제의 간단한 예가 된다. 개체내의 1의 갯수를 c 라고 하고 이 문제의 적합도값을 $fitness \equiv 10 \times c$ 로 정의한다. 그러면 이 문제의 최적해는 240이다. 그림 5(a)를 보면 One_Max문제에 있어GA1,GA3의 성능이 GA2, GA4의 성능보다 우수하였다. 그러나 교차 및 돌연변이 확률을 가변시켰을 경우 진화초기의 수렴속도가 매우 빨랐다. 그림 5(b)에서는 GA6, GA7, GA5, GA8의 순서로 성능이 향상된다. GA8의 경우 20번 가운데 한번은 목표값에 수렴하지 못하였다. 위의 시뮬레이션 결과들로부터, 제시된 알고리즘은 빌딩블록 가설을 잘 만족시키는 것으로 해석된다. 그리고



(a)



(b)

그림 5. One_Max문제에 대한 각 GA의 수렴과정.
Fig. 5. Convergence procedures of each genetic algorithm for One_Max problem.

표 2에서 보면 One_Max문제의 해결에 있어서 전체적으로 제안된 GA의 성능이 Holland와 Syswerda의 GA 성능보다 우수하였다.

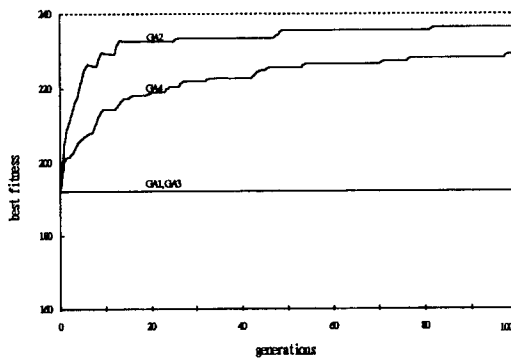
표 2. One_Max 문제에 대한 시뮬레이션 결과.
Table 2. Simulation results for One_Max problem.

Methods	Generations	cpu time[h:m:s]	Best fitnesses
GA1	190.0	00:00:03.84	240.0
GA2	715.2	00:00:15.90	234.5
GA3	84.3	00:00:01.78	240.0
GA4	959.9	00:00:19.30	229.5
GA5	8.7	00:00:00.21	240.0
GA6	6.9	00:00:00.10	240.0
GA7	7.6	00:00:00.11	240.0
GA8	55.0	00:00:01.20	239.0

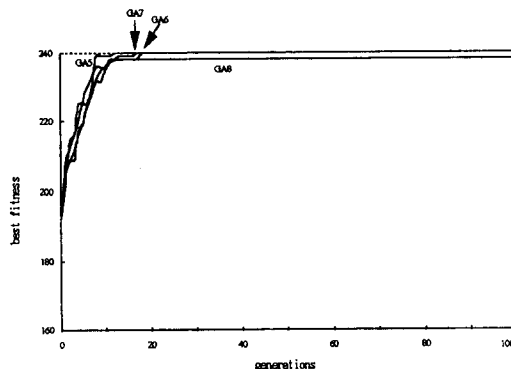
1.2 Two Max문제

이 문제는 multimodal problem의 예로서 M이 24이고 유전자수가 0과 1로 이루어진 개체가운데에서 유전자값이 모두 1인 개체를 찾는 것이다. 이 문제에서는 하나의 전역 최적해와 하나의 국부 최적해가 존재한다. 개체내의 1의 갯수를 c라고 하고 이 문제의 적합도값을 (9)와 같이 정의한다. 그러면 이 문제의 최적해는 240 이 된다.

$$fitness \equiv |18 \times c - 192| \tag{9}$$



(a)



(b)

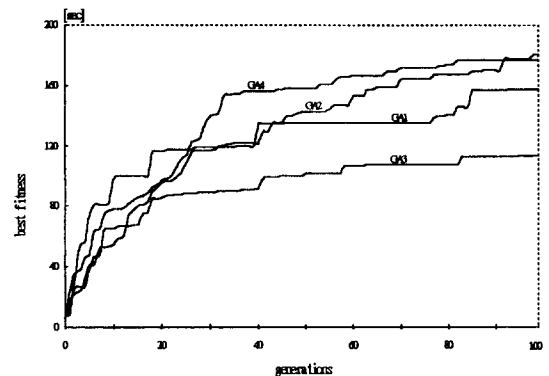
그림 6. Two_Max문제에 대한 각 GA의 수렴과정.
Fig. 6. Convergence procedures of each genetic algorithm for Two_Max problem.

그림 6(a)와 표 3을 보면 GA1, GA3에서는 주어진 기간

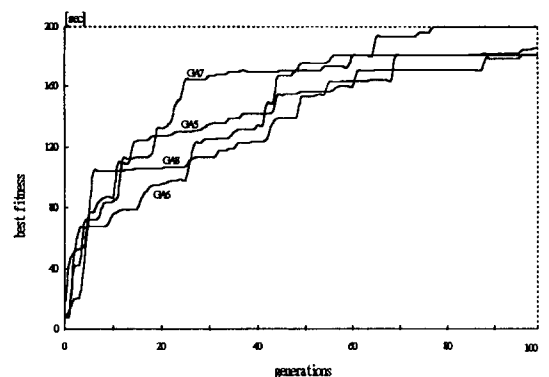
에 목표값에 한번도 수렴하지 못하였다. 그러나 이 경우에 교차 및 돌연변이 확률을 가변시켜준 경우 국부최적해인 192를 쉽게 벗어나서 전역 최적해로 수렴하였다. 그리고 그림 6(b)와 표 3을 보면, 제시된 GA에서는 네 경우 모두 전역 최적해를 쉽게 찾았다. 이처럼 Two_Max문제에 있어서 기존의 GA에 교차 및 돌연변이 확률값을 가변시켜준 경우 국부 최적해를 쉽게 벗어나서 전역 최적해로 수렴하였고, 제안된 알고리즘에 의해서는 더욱 빠른 속도록 주어진 문제를 풀었다. 이 문제에서는 성능이 GA5, GA6, GA7, GA2, GA4, GA8의 순으로 우수했다. GA8에서는 20번 가운데 한번 목표값에 수렴하지 못하였다. 이 경우에도 제안된 알고리즘이 빌딩블록가설을 잘 만족시킨 것으로 해석된다.

표 3. Two_Max 문제에 대한 시뮬레이션 결과.
Table 3. Simulation results for Two_Max problem.

Methods	Generations	cpu time[h:m:s]	Best fitnesses
GA1	1000.0	00:00:20.16	192.0
GA2	62.3	00:00:01.33	240.0
GA3	1000.0	00:00:17.42	192.0
GA4	127.8	00:00:02.54	240.0
GA5	6.2	00:00:00.10	240.0
GA6	6.7	00:00:00.11	240.0
GA7	7.6	00:00:00.10	240.0
GA8	87.3	00:00:01.13	238.2



(a)



(b)

그림 7. 수레 제어 문제에 대한 각 GA의 수렴과정.

Fig. 7. Convergence procedures of each genetic algorithm for a cart control problem.

1.3 수레 제어 문제

수레 제어 문제의 상세한 내용은 부록 A에 언급되어있다. 수레 제어 문제의 적합도값은 수레가 주어진 조건에 따라 실패 없이 제어될 수 있는 시간으로 정의하고 목표값은 200 초로 정하였다. 그리고 실험에서는 상태변수의 초기값은 x 는 $\pm 0.1[m]$ 사이, θ 는 $\pm 2.5[^\circ]$ 사이, \dot{x} 와 $\dot{\theta}$ 는 0.0의 값을 갖도록 하였고 초기 개체집단은 적합도값이 10미만인 개체만으로 구성하였다. 그림 7(a)에서 보면 GA1, GA3는 GA2, GA4보다 성능이 좋지 않았다. 이것은 비선형 제어 문제에 있어서도 교차 및 돌연변이 확률값을 가변시켜주면 성능이 향상될 수 있음을 보여준다. 그림 7(b)에서는 GA7, GA6, GA5, GA8의 순으로 GA의 성능이 우수하였다. 이로부터 비선형 제어 문제에서도 교차의 균일도가 작을 수록 알고리즘의 수렴속도가 개선됨을 확인하였다. 그것은 교차의 균일도가 너무 작아지면 스키마의 길이가 커지고 스키마의 차수가 높아지므로 빌딩블록 가설에 의해 최적값을 찾는 데 불리하게 되기 때문이다. 이 그림에서 보면 빌딩블록을 이루는 스키마의 크기가 커지면 한 세대에서 다음세대로 바뀔 때 적합도값의 변화폭이 커짐을 알 수 있다. 그러나 이처럼 적합도값의 변화폭이 크다고해서 최적값에의 수렴속도가 빠른 것은 아님을 알 수 있다. 그리고 GA4의 경우 20번의 시뮬레이션 과정에서 한번은 최적값에 수렴하지 못하였다. 반면에 GA1, GA3는 각각 4번, 12번씩 목표값에 수렴하지 못하였다.

표 4. 수레 제어 문제에 대한 시뮬레이션 결과.

Table 4. Simulation results for a cart control problem.

Methods	Generations	cpu time[h:m:s]	Best fitnesses
GA1	88.4	00:06:47.04	161.36
GA2	53.9	00:02:50.94	200.00
GA3	188.4	00:28:50.49	113.33
GA4	54.0	00:03:08.48	193.50
GA5	40.0	00:01:57.98	200.00
GA6	36.0	00:01:43.45	200.00
GA7	28.0	00:01:00.27	200.00
GA8	54.0	00:01:52.79	200.00

2. 여러곳 교차방식과의 비교

본 논문에서 제안된 GA의 성능평가를 위하여 Multi_Max 문제를 선택하였다. 그리고 제안된 알고리즘과의 비교대상으로서는 Syswerda의 균일 교차개념을 갖는 GA 및 여러곳 교차방식의 GA를 선택하였고 이들의 swapping 확률값을 0.25로 하여 이들의 통상적인 swapping 확률값 0.5 및 1.0과 다르게 하였다. GA의 성능평가는 Multi_Max 문제에 각각의 GA를 적용하여 이들이 최적값에 수렴하는 속도의 비교를 통해 이루어졌다. 그리고 제안된 알고리즘에 있어서 교차의 균일도가 알고리즘의 성능에 미치는 영향을 살펴보기 위해 균일도가 1, 1/2, 1/3, 1/4, 1/8의 다섯 가지인 경우로 나누어서 시뮬레이션을 수행하였다. Syswerda의 GA와 여러곳 교차방식의 GA에 대해서는 교차 확률값이 1.0, 돌연변이 확률값이 1/M이 되도록 하였고, Multi_Max 문제에 대한 시뮬레이션 내용을 요약하면 표 5와 같다.

시뮬레이션시 개체집단의 크기를 50으로 하였고, 제안된 알고리즘의 교차 확률값은 0과 1 사이에서 그리고 돌연변이 확률값은 0.001과 0.1사이에서 변화하도록 하였다. 그리고 최대 500세대까지 진화를 시켰다.

시뮬레이션에서 채택한 Multi_Max문제는 (10)과 같다. 이 문제에서는 주어진 구간에서 (10)의 최대값을 찾는 것으로, 여러개의 국부 최적해가 존재한다.

$$f(x) = x \sin(10\pi x) + 1.0 \text{ for } x = [0..4] \quad (10)$$

표 5. 시뮬레이션 내용.

Table 5. Contents of simulation.

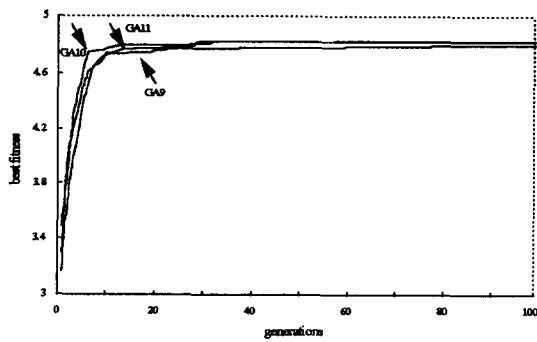
Genetic Algorithms	Crossover Probability	Mutation Probability	Swap Probability	Uniformity
GA9(Syswerda)	1.0	1/M	0.25	-
GA10(Multipoint)	1.0	1/M	0.25	-
GA11(Multipoint)	1.0	1/M	0.25	-
GA12(Proposed)	variable	variable	0.25	1
GA13(Proposed)	variable	variable	0.25	1/2
GA14(Proposed)	variable	variable	0.25	1/3
GA15(Proposed)	variable	variable	0.25	1/4
GA16(Proposed)	variable	variable	0.25	1/8

Multi_Max문제에 대한 시뮬레이션 결과는 그림 8 및 표 6과 같다. 그림 8에서는 각 알고리즘의 우열을 가리기가 힘들다. 그러나 표 6에 의해서는 각 알고리즘의 성능이 비교적 명확히 구분된다. 이 문제에서 각각 20번의 시뮬레이션을 통해 20번 모두 최적값을 찾은 경우는 제안된 알고리즘에서 균일도를 1/3으로 하였을 때 뿐이었고(GA14), cpu time과 목표값에 도달한 평균세대도 GA14가 가장 빨랐다. 그러나 제안된 알고리즘에 있어서 균일도가 1/8인 경우는 성능이 많이 떨어지는 것을 알 수 있다. 이것은 제안된 알고리즘에 있어서 뉴클레오티드 즉 빌딩블록의 크기가 문제의 성능에 많은 영향을 미침을 나타낸다. 특히 제안된 알고리즘에서 균일도가 1/2인 경우(GA13)와 GA10, GA11과 비교해보면 GA13이 GA10, GA11보다 수렴속도가 빠름을 알 수 있다. 비록 검토된 문제가 비교적 간단한 문제여서 제안된 알고리즘과 비교대상 알고리즘의 성능이 많은 차이가 나지는 않았지만, 제안된 알고리즘이 우위임을 나타내었다. 이 결과는 제안된 알고리즘에 있어서 뉴클레오티드의 역할이 중요함을 나타낸다. 그리고 제안된 알고리즘에서는 뉴클레오티드의 크기는 swapping 확률과 아무런 관련이 없다. 제안된 알고리즘에 있어서 swapping 확률에 영향을 미치는 것은 단지 뉴클레오티드의 가지수 뿐이다.

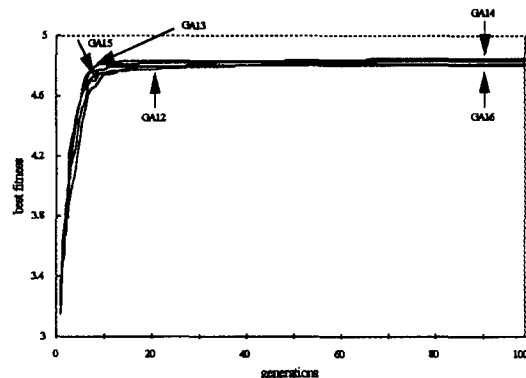
표 6. Multi_Max 문제에 대한 시뮬레이션 결과.

Table 6. Simulation results for a Multi_Max problem.

Methods	Generations	cpu time[h:m:s]	ave. best fitnesses	success/trial
GA9	193.4	00:00:05.04	4.8201324	16/20
GA10	92.0	00:00:02.51	4.8401319	19/20
GA11	88.4	00:00:02.68	4.8401319	19/20
GA12	148.8	00:00:04.55	4.8301274	17/20
GA13	75.0	00:00:02.15	4.8401319	19/20
GA14	58.9	00:00:01.68	4.8501315	20/20
GA15	97.4	00:00:02.74	4.8401319	19/20
GA16	161.55	00:00:04.56	4.8001334	15/20



(a)



(b)

그림 8. Multi_Max 문제에 대한 각 GA의 수렴 과정.

Fig. 8. Convergence procedures of each genetic algorithm for a Multi_Max problem.

V. 결론

본 논문에서는 DNA와 RNA를 구성하는 NU개념에 기초한 균일 교차방식의 GA를 제안하였으며, 이에 는 교차 및 돌연변이 확률값을 매세대마다 변화시켜주는 알고리즘이 포함되어 있다. Yoshikawa등이 사용한 DNA Coding Method도 실제 DNA의 메카니즘으로부터 아이디어를 얻었으나 이 방법에서는 한곳 교차와 일반적인 돌연변이를 사용하여 본 논문의 교차 및 돌연변이 기법과는 근본적으로 다름을 알 수 있다[17].

본 논문에서 제안된 교차방법은 실제의 DNA, RNA에서 일어나고 있는 교차 메카니즘과 같다. 즉 생물학적 RNA는 A,C,G,U의 뉴클레오티드열로 이루어져 있고, 이러한 RNA가 다른 RNA와 교차를 수행할 때에는 마주 보는 두 뉴클레오티드들이 A-U, C-G가 될 때에는 서로 교차를 행하고 A-C, G-U, A-G, C-U, A-A, C-C, G-G, U-U와 같은 경우에는 교차를 수행하지 않는 방식으로 새로운 RNA가 복제되게 된다. 본 논문에서 제안된 교차방법에서도 이와 같은 방법으로 교차가 이루어진다. 예로서 이진수로 이루어진 두 개체(인조 RNA)에 있어서 이들 RNA의 각 유전자(또는 유전자들)에 임의로 A, C, G, U(4가지의 뉴클레오티드만을 정의한 경우)를 할당하고, 두 개체의 같은 위치에 있는 뉴클레오티드를 차례대로 서로 비교해서 그들이 A-U, G-C쌍을 이루면 두 뉴클레오티드내의 정보를 서로 교환하는 방식으로 교차가 이루어진다. 이때 A, C, G, U와 같이 a개 뉴클레오티드의 발생확률을 같게 한 경우에는 뉴클레오티드의 정보를 서로 교환할 확률값이 $1/a$ 이 된다. 그리고 기존 방법에서는 교차대상이 되는 개체의 선정에 고정된 교차 확률값

을 써서 수행하지만, 본 논문에서는 교차대상을 선정하는 데 쓰이는 교차확률값을 매세대마다 임의로 선정하여 준다. 또한 돌연변이 확률값도 매세대마다 임의로 선정하여 준다. 이러한 점들이 본 논문에서 새로이 제안된 내용이다.

제안된 기법에서는, 교차 메카니즘이 뉴클레오티드 개념에 바탕을 두고 있어서 전역 최적값을 찾는 능력이 향상되었다. 즉 제안된 알고리즘에서는, 균일 교차가 유전자 단위가 아닌 빌딩블록 단위로 이루어지므로, 알고리즘의 성능개선이 빌딩블록 가설에 의해 뒷받침 된다. 그리고 교차 및 돌연변이 확률값을 매세대마다 임의로 가변시켜주는 기법의 도입에 의해 전역 최적값으로의 수렴속도가 더욱 향상된다고 할 수 있다.

제안된 GA를 One_Max, Two_Max, Multi_Max 및 수레 제어 문제에 적용하여 시뮬레이션을 수행한 결과로부터 얻은 결론은 다음과 같다.

첫째, 교차 및 돌연변이 확률값을 모든 세대에 대해 고정시킨 경우보다 매세대마다 가변시킨 경우에 알고리즘의 성능이 더 향상되었다. 특히 이들 확률값을 가변시킨 경우 진화의 초기에 적합도값의 개선이 빠르게 진행되었다.

둘째, NU가 빌딩블록으로서 동작하고, 교차가 빌딩블록 단위로 이루어지므로 NU의 크기와 차수가 작은 경우 빌딩블록 가설에 따라 GA의 성능이 향상되었다.

셋째, 교차 및 돌연변이 확률의 가변화는 기존의 GA에 적용가능하다.

넷째, 제안된 알고리즘에 있어서 균일도가 알고리즘의 성능에 많은 영향을 미친다. 특히 교차의 균일도가 작은 경우 전반적으로 GA의 성능이 향상되었다.

다섯째, 제안된 알고리즘은 최적값으로의 수렴속도가 빨라서 신속한 처리속도를 요구하는 문제의 해결에 적합하다.

참고문헌

- [1] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.
- [2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [3] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE Trans. on Neural Networks*, vol. 5, no. 1, pp. 96-101, 1994.
- [4] C. Sammut, "Experimental results from an evaluation of algorithms that learn to control dynamic systems," *Proc. of 5th Int. Conf. on Machine Learning*, pp. 437-443, 1988.
- [5] 장병탁, "인공진화에 의한 학습 및 최적화," 제어·자동화·시스템공학회지 제 1 권, 제 3 호, pp. 52-61, 1995.
- [6] B. Carse and T. C. Fogarty and A. Munro, "Adaptive distributed routing using evolutionary fuzzy control," *Proc. of 6th Int. Conf. on Genetic Algorithms*, pp. 389-396, 1995.
- [7] P. P.C. Yip and Y. -H. Pao, "Combinatorial optimization with use of guided evolutionary simulated annealing," *IEEE Trans. on Neural Networks*, vol. 6, no. 2, pp. 290-295, 1995.
- [8] C. W. Holsapple, V. S. Jacob, R. Pakath and J. S. Zaveri, "A genetics based hybrid scheduler for

generating static schedules in flexible manufacturing contexts," *IEEE Trans. on SMC*, vol. 23, no. 4, pp. 953-972, 1993.

[9] D. B. Fogel and L. J. Fogel, "Guest editorial evolutionary computation," *IEEE Trans. on Neural Networks*, vol. 5, no. 1 pp. 1-2, 1994.

[10] Z. Michalewicz, *GENETIC ALGORITHMS + DATA STRUCTURES = EVOLUTION PROGRAMS*, Springer Verlag, Berlin, 1992.

[11] D. Ackley, *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers, 1987.

[12] G. Syswerda, "Uniform crossover in genetic algorithm," *Proc. of 3rd Int. Conf. on Genetic Algorithms*, pp. 2-9, 1989.

[13] W. M. Spears, K. A. De Jong, "On the virtues of parameterized uniform crossover," *Proc. of 4th Int. Conf. on Genetic Algorithms*, pp. 230-236, 1991.

[14] W. M. Spears, "Adaptive crossover in evolutionary algorithm," *Proc. of 4th Evolutionary Programming*, pp. 367-384, 1995.

[15] T. Bäck, "Optimal mutation rates in genetic search," *Proc. of 5th Int. Conf. on Genetic Algorithms*, pp. 2-8, 1993.

[16] C. Z. Janikow and Z. Michalewicz, "An experimental comparison of binary and floating point representation," *Proc. of 4th Int. Conf. on Genetic Algorithms*, pp. 31-36, 1991.

[17] T. Yoshikawa, T. Furuhashi and Y. Uchikawa, "An emergence of fuzzy control rules for mobile robots using DNA coding method," *Proc. of Int. Symposium on Artificial Life and Robots*, Beppu, Japan, pp. 162-165, 1996.

[18] R. H. Cannon Jr., *Dynamics of Physical Systems*, McGraw-Hill, 1967.

부록

A. 수레 제어 시스템

수레 제어 문제는 크게 두개의 과제를 갖고있다. 하나는 수레의 좌우측으로 일정한 시간간격(T)마다 일정한 크기의 힘(F)을 가하여 수레가 주어진 트랙의 범위를 벗어나지 않도록 하는 것이고, 또 하나는 수레위에 힌지에 의해 결합되어있는 막대가 수직선상과 이루는 각도가 주어진 범위 내에 들도록 하는 것이다. 시뮬레이션 과정에서 트랙의 길이 x 는 $\pm 2.4[m]$, 막대가 수직선과 이루는 허용각도 θ 는 $\pm 12.0^\circ$ 로 설정하였다. 그리고 힌지에서 막대의 무게중심까지의 길이 l 은 $0.5[m]$, 중력가속도 g 는 $9.8[m/sec^2]$, 막대의 질량 m_p 는 $0.1[kg]$, 수레의 질량 m_c 는 $1.0[kg]$, F 는 $10[N]$, T 는 $0.02[sec]$ 이다. 실험에 사용된 수레 제어 시스템의 구성은 그림 A-1과 같으며, 수레의 수학적 모델식으로는 케논이 제시한 식을 사용하였다[18].

$$\ddot{\theta} = \frac{g \sin \theta - \frac{\cos \theta (F + m_p l \dot{\theta}^2 \sin \theta)}{m_p + m_c}}{\frac{4l}{3} - \frac{l m_p \cos^2 \theta}{m_p + m_c}} \quad (A-1)$$

$$\ddot{x} = \frac{F + m_p l \dot{\theta}^2 \sin \theta - \dot{\theta} \cos \theta}{m_p + m_c} \quad (A-2)$$

(A-1), (A-2)를 오일러변환하면 (A-3) - (A-6)이 된다.

$$\theta(t+1) = \theta(t) + \dot{\theta}(t)T \quad (A-3)$$

$$\dot{\theta}(t+1) = \dot{\theta}(t) + \ddot{\theta}(t)T \quad (A-4)$$

$$x(t+1) = x(t) + \dot{x}(t)T \quad (A-5)$$

$$\dot{x}(t+1) = \dot{x}(t) + \ddot{x}(t)T \quad (A-6)$$

수레의 상태공간은 4차원공간이 되며, 상태변수는 하나의 차원을 나타낸다. 상태공간의 각 점에 있어서, 수레는 다음 상태에서 힘을 수레의 왼쪽으로 가할 것인지 또는 오른쪽으로 가할 것인지를 결정해야한다. 하지만 상태공간안에는 무수한 점들이 존재한다. 그래서 우리는 이들 무한한 탐색공간을 512개의 유한한 탐색공간으로 줄였다. 이때 탐색공간의 분할은 그림 A-2에 따라서 하였다.

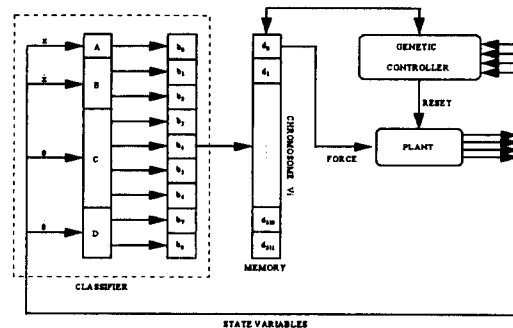


그림 A-1. 막대가 위에 달린 수레 제어 시스템의 구성.

Fig. A-1. A configuration of a cart control system with an inverted pole.

- x : -2.4 to 0 , 0 to $2.4[m]$ (two partitions)
- \dot{x} : $-\infty$ to -0.25 , -0.25 to 0 , 0 to 0.25 , 0.25 to $+\infty$ [m/sec] (four partitions)
- θ : -12 to -10 , -10 to -5 , -5 to -2.5 , -2.5 to -1.5 , -1.5 to -1 , -1 to -0.5 , -0.5 to -0.25 , -0.25 to 0 , 0 to 0.25 , 0.25 to 0.5 , 0.5 to 1 , 1 to 1.5 , 1.5 to 2.5 , 2.5 to 5 , 5 to 10 , 10 to $12[^\circ]$ (sixteen partitions)
- $\dot{\theta}$: -50 to -5 , -5 to 0 , 0 to 5 , 5 to $50[^\circ/sec]$ (four partitions)

그림 A-2. 상태변수의 분류영역.

Fig. A-2. Partitions for classifying state variables.

그림 A-1의 메모리는 어느 한 개체 $V_i(t) = \{d_0 d_1 d_2 \dots d_{511}\}$ 의 유전정보를 갖고있고 유전정보는 플랜트에 가해질 힘의 방향을 나타낸다. 본 논문의 수레 제어 시스템은 일반적인 제어 시스템과는 달리 제어를 위한 어떠한 선형적인 지식도 필요로하지않는다. 제어는 임의의 초기 상태변수값을 갖고 시작한다. 이 값들에 의하여 플랜트는 다음상태의 상태변수를 분류기, 유전제어기에게 출력한다. 유전제어기는 상기한 상태변수값들을 받으면 수레의 위치와 막대의 각도가 주어진 범위내에 있는지의 여부를 판정해서, 주어진 범위를 벗어났으면 유전제어기는 RESET신호를 플랜트에 보내고 메모리내의 정보는 새로운 개체 $V_j(t) = \{d_0 d_1 d_2 \dots d_{511}\}$ 의 정보로 바뀌게되고 플랜트는 처음부터 다시 제어를 시작한다.

이때 분류기로부터 메모리에 입력되는 값은 메모리의 번지를 나타낸다. 예로서 번지값이 26인 경우 메모리의 26번지에 저장되어있는 이진값($d_j \in \{0,1\}$, $j = 0,1,\dots,511$)이 플랜트에 출력되는데 이 값은 다음 상태에 플랜트에 가해질 힘을 나타낸다. 이진수 0은 $10[N]$ 이, 이진수 1은 $-10[N]$ 이 수레에 가해짐을 뜻한다. 플랜트는 이 힘에 의하여 새로운 상태

변수값을 출력한다. 그리고 만약 플랜트가 RESET신호를 받지 않으면 시스템은 제어를 계속한다. 이 과정은 목표로 설정한 값에 도달하는 개체를 찾을 때까지 개체집단내의 모든 개체에 대해서 반복된다. 만약 설정한 세대가 다 지나도 목표값을 갖는 개체를 못 찾으면 프로그램은 실행을 멈춘다.



김 성 수

1955년 12월 30일생. 1981년 홍익대 전기공학과졸(학사), 1986년 연세대 전기공학과졸(석사), 1993년 4월이래 독일 Bochum 대 전자공학과 박사과정(Institut für Neuroinformatik), 1996년 연세대학교 전기공학과졸(박사). 1981년 포항제철 동력부 기술원, 1982년 현대건설 국내전기부 기사, 1983년 제19회 기술고등고시 합격(전기직), 1984년~1987년 문교부 시설국 사무관, 1987년~1989년 광주시교육위원회 설비계장, 1989년 4월~현재 특허청 심사4국 정보심사담당관실 심사관(공업서기관). 주요 관심 분야는 유전 알고리즘, 신경회로망, 퍼지 제어, 분산 제어, FMS, 패턴인식.



우 광 방

1934년 1월 25일생. 1957년 연세대 전기공학과졸(학사), 1959년 연세대 전기공학과졸(석사), 1962년 및 1964년 오레곤주립대 전기공학과에서 석사 및 박사학위 취득, 1966년~1971년 워싱턴대 전기공학과 조교수, 1971년 미 국립암연구소 책임연구원, 1980년 재미한국과학기술자협회장, 1982년~현재 연세대학교 전기공학과 교수, 1992년~현재 연세대학교 자동화기술연구소 소장, 1995년 당학회회장. 주요 관심 분야는 유전 알고리즘, 신경회로망, 퍼지 제어, 분산 제어, FMS, 패턴인식.