

# Genetic Algorithm 응용기술

김 용 호 · 전 홍 태  
(중앙대학교 전자공학과)

## ■ 차 례 ■

- I. 서 론
- II. 유전 알고리즘

- III. 단순유전알고리즘(SGA)과 미소유전알고리즘( $\mu$ -GA)
- IV. 결론및 앞으로의 연구방향

### I. 서 론

1970년대초 John Holland에 의해 본격적으로 연구되기 시작한 유전 알고리즘은 자연생태계의 진화과정에서 관찰된 몇가지 처리과정 중에서 '적자생존(Survival of the fittest)'의 원리를 컴퓨터 알고리즘과 결합시켜 정립된 최적화(optimization) 알고리즘이다. [2][5]

최근에는 최적화 및 탐색문제 그리고 제어규칙 찾는 문제 등 여러 부문에 많이 응용되고 있으며 적용이 확대될 전망이다. 따라서 본 논문에서는 유전알고리즘을 소개하고 실제문제에 대해 적용하여 그 효율성을 고찰해 본다.

유전알고리즘은 자연생태계의 진화메카니즘을 모방하였는데 실제로 자연계의 진화과정이 모두 밝혀져 있지는 않지만 중요한 몇가지는 알려져 있어 유전알고리즘은 이러한 진화과정에서 일부 관찰된 것을 사용하였다. 자연생태계의 진화과정을 간단히 살펴보면 다음과 같다[6].

진화는 염색체(chromosome)-생명체의 구조를 부호화하기 위한 기관소자-에서 일어나며 생명체는 염색체를 복호화하는 과정을 통해 부분적으로 창조된다. 염색체의 부호화-복호화 과정의 자세한 것은 알 수 없어도 일반적으로 인정되고 있는 몇가지 과정은 다음과 같다.

- 1) 진화는 부호화되는 생명체가 아닌 염색체에 대해 작용하는 과정이다.
- 2) 자연도태(natural selection)는 염색체와 복호화된 구조의 수행과의 관계이며, 훌륭한 구조의 염색체를 보다 자주 부호화 해주는 과정이다.
- 3) 재생산(reproduction) 과정은 진화가 일어나는 시점이다. 돌연변이는 자손의 염색체와 아버지의 염색체를 다르게 할 수 있고 재조합처리는 아버지의 염색체들을 결합함으로써 매우 다른 염색체의 자손을 만들어 준다.
- 4) 진화는 기억을 갖고 있지 않으며 훌륭한 염색체를 생산하는 정보는 '유전자 웅덩이(Gene pool)'와 염색체 복호화기의 구조속에 담겨져 있다.

John Holland는 자연계의 진화에 관한 상기의 특징들이 컴퓨터 알고리즘에 적절히 조화되면 자연계가 진화를 통해 극복해 왔듯이 어려운 문제를 해결하는 기술이 될 것이라 믿었다. 그래서 염색체를 0과 1로 구성된 2진 string으로 표현하고 각각의 염색체의 집단에 대해 모의진화(simulated evolution)를 실행하였다. 자연계처럼 유전알고리즘은 훌륭한 염색체를 찾는 문제를 염색체 속에 있는 자료를 '알지 못하는 상태(blindly)'로 조작하므로써 해결하였다. 자연계에서와 마찬가지로 해결하고 있었던 문제의 형태를 알지 못했으며 제공된 유일한 정보는 생산된 각각의 염색체에 대한 평가이고 좋은 평가의 염색체가 나쁜 평가

의 염색체보다 자주 재생산 되도록 염색체의 선택을 치우치게 하는 것이다. 이러한 유전알고리즘은 다음과 같이 크게 3분야로 나누어질 수 있다.

- 첫째, 기본적인 유전알고리즘에 관한 연구
- 둘째, 유전알고리즘을 이용한 최적화
- 셋째, 분류체계(classifier system)를 갖는 기계학습(machine learning)

여기에서는 처음 두 가지에 대해 살펴보기로 한다. 또한 유전알고리즘을 문제해결에 적용시키는 문제는 다음과 같이 2가지 매카니즘이 있다.

- 첫째, 염색체상에 문제의 해를 부호화 하는 방법
- 둘째, 문제속에서 염색체의 가치 측정을 하는 평가 함수

첫번째의 경우 부호화하는 기술은 문제마다 다르고 부호화는 bit string으로 수행한다. 두번째의 경우 평가함수는 유전 알고리즘과 해결해야 할 문제를 결합시켜 주며 또한 염색체를 입력으로 받아 해결될 문제에 대한 염색체의 수행 측정이 되는 값을 제공한다. 평가함수는 환경이 자연진화에 하는 것과 같은 역할을 유전알고리즘에 주며 환경과 개개의 상호작용은 적합도의 측정이고 염색체와 평가함수의 상호작용은 재생산시 유전알고리즘이 사용하는 적합도의 측정을 제공한다. 이상을 바탕으로 유전알고리즘을 전개해 나가 본다.

본 논문의 구성은 다음과 같다. 2장에서는 기본적인 유전 알고리즘의 전개를, 3장에서는 단순유전알고리즘(SGA : Simple Genetic Algorithms)과 미소유전알고리즘( $\mu$ -GA : Micro-Genetic Algorithms)의 설명과 적용예를, 4장에서는 결론 및 향후 연구방향이 제시된다.

## II. 유전 알고리즘(Genetic Algorithm)

유전 알고리즘의 바탕이 되는 자연 생태계의 일반적인 진화과정은 다음과 같다.[7]

- 첫째, 진화는 염색체(Chromosome) - 생물의 구조를 부호화(encoding)하기 위한 기관소자(organic device)에서 일어난다.
- 둘째, 자연도태(Natural Selection)는 훌륭한 구조를 부호화한 염색체를 더 자주 재생산하도록 해준다. 재생산(reproduction)은 진화가 일어난 시점이다.
- 셋째, 돌연변이는 사촌의 염색체를 아버지 염색체와 다르게 해준다.
- 넷째, 재조합(Reproduction) 교배(crossover) - 과정은 아버지의 염색체를 결합하여 매우 다른 염색체를 사촌에게 준다.

이 유전알고리즘은 미지수 함수  $Y=G(x)$ 의 최적해를 발견하는 Simulated evolution형의 탐색(Search) 알고리즘의 성격을 갖으며 John Holland는 상기과정을 어려운 문제 해결의 기법으로써 컴퓨터 알고리즘과 결합시켜 유전알고리즘을 탄생시켰다.

이와 같은 유전 알고리즘의 특성은 다음과 같다.

- 첫째, 유전 알고리즘은 해공간(solution space)에서 단일 해를 사용하는 것이 아니라 해집단(population of solution space)을 이용한다.
- 둘째, 유전이론은 'blind'하다. 즉 최적화 함수 정보(미분가능성, 연속성 등)를 필요치 않는다.
- 셋째, 유전 알고리즘은 global한 최적해의 발견을 가 능케 한다.

다음은 기본적인 유전알고리즘의 처리과정을 나타내고 있으며 이후 이 처리과정을 바탕으로 평가함수로써 2진 fitness를 최적화 시키는 과정을 전개해 본다.

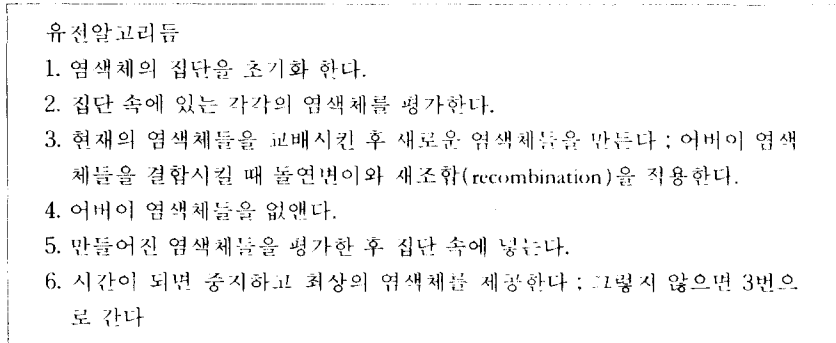


그림 1.1 유전알고리즘의 처리과정  
Fig. 1.1 process of genetic algorithm  
(810)

상기 처리과정에 의해 유전알고리즘을 전개하기 위해 다음 세가지 사항을 준비한다.

1) 평가함수 설정

평가함수는 수학적 함수인 2진 f6함수를 선택한다 (유전알고리즘에 관한 문헌에 자주 사용됨). 이 함수를 2개의 실수로 변환하기 위해 염색체를 44비트로 구성한다. 이 2개의 수가 함수에 대입이 된 후 계산된 결과가 염색체의 평가값이 된다.

44비트로 구성된 염색체는 -100에서 100까지의 범위를 갖는데 다음 세가지의 단계를 갖는다.

첫째, 44비트중 처음 22비트는  $x_1$ 의 값이고 남은 22비트는  $y_1$ 의 값을 나타낸다.

둘째,  $x_1$ 과  $y_1$ 을 0.00004768372718899898로 나누어  $x_2$ 와  $y_2$ 를 구한다.

셋째,  $x_3$ 과  $y_3$ 을 -100에서 100까지의 범위에 있게 하면서 구하기 위해  $x_2$ 와  $y_2$ 에서 100을 뺀다.

그림 1.2는 상기과정을 보여주고 있다. 입력으로써  $x_3$ 과  $y_3$ 을 사용하는 2진 f6함수는 다음과 같다.

$$0.5 - \frac{(\sin\sqrt{x^2+y^2})^2}{1.0+0.001(x^2+y^2)^2}$$

우리의 목적은 이 함수의 최대값을 구하기 위한  $x$ 와  $y$ 를 결정하는 것인데 이것이 바로 ‘함수의 최적화’이다. 유전알고리즘의 효율성을 보기 위해 2진 f6함수를 사용하는 이유는 그림 1.3에서 보여지는 것과 같이 이 함수가 단 하나의 최적해를 갖고 있으며, 오실레이션을 하기 때문에 기존의 함수최적화 기법으로는 해결하기 어렵기 때문이다.

그림 1.3은 f6함수의 궤적을 나타낸 것이다. 이 그래프에서  $x$ 는 -100에서 100까지의 범위를 가지며  $x=0$ 에서 최적값을 가지며  $x$ 와  $y$ 에 대해 대칭인 모양을 갖는다.

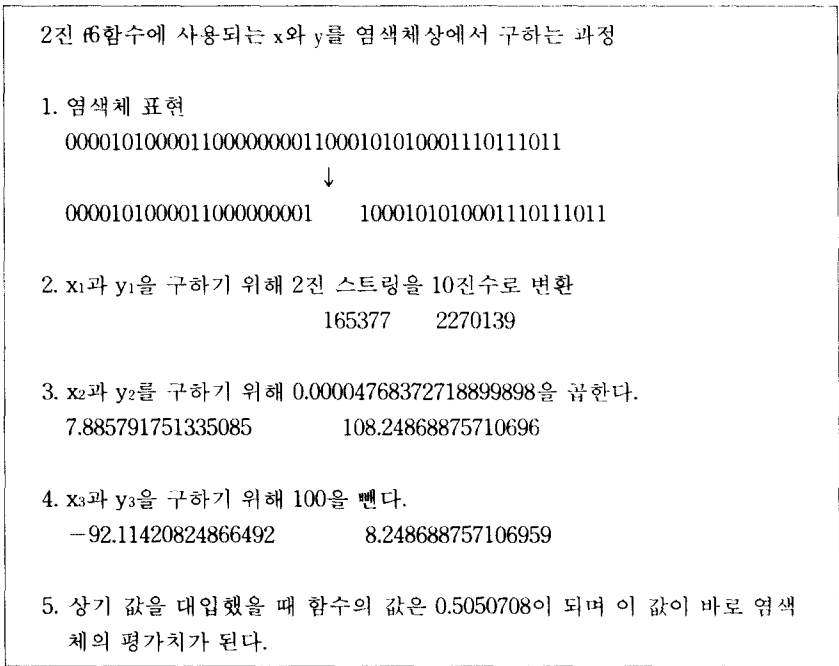


그림 1.2 2진 f6 함수에 사용되는 값의 염색체표현 및 복호화 과정  
 Fig. 1.2 chromosome representation and decoding process of values used in binary f6

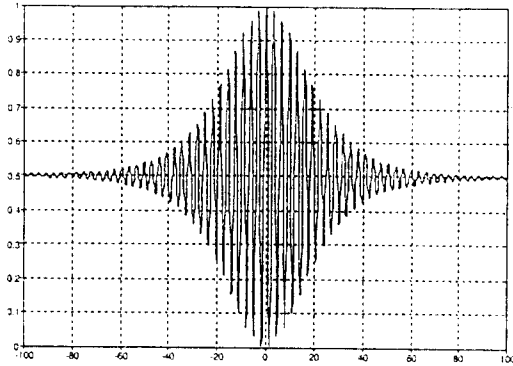


그림 1.3  $x=0$ 에서 최적값을 갖는  $f(x)$  함수의 그래프  
 Fig. 1.3 Graph of  $f(x)$  which has optimal value at  $x=0$

2) 집단 설정

집단을 설정하는데에는 여러가지 방법이 있는데 그 중 간단한 방법을 고찰해본다.

집단 및 각각의 염색체를 2진수로 표현한다. 여기에서는 1개의 염색체를 44비트로 구성한다. 집단의 초기화는 난수발생에 의하여 알고리즘의 전개를 이 초기 집단으로부터 시작한다. 그리고 다음 세대를 만들었을 때 이 초기집단을 전부 삭제한다. 재생산은 이전의 집단을 새로운 집단으로 대체시키는 것이며 이때 이를 '세대교체(generation)'라 한다. 어버이의 선택은 roulette wheel 방법을 사용하며 집단의 크기는 파라미터 값을 고려하여 100개로 설정한다. 세대교체는 40번 시행해 본다. 다음은 어버이를 선택하는 roulette wheel 방법이다.

· Roulette wheel 방법 (컴퓨터로 구현)

여기에서 1, 2, 3, 4는 스트링 번호, 0.00 ~ 4.00은 적합도값에서 계산된 선택 확률

난수를 0~4 까지 발생시켜 스트링을 선택한다

예) 난수      스트링  
 0.11 →      1  
 0.66 →      2  
 0.98 →      2  
 2.78 →      4

상기과정에서 재생산된 스트링(Gene Pool) :

스트링 번호	재생산된 스트링 (Gene Pool)
1	01101
2	11000
3	11000
4	10011

3) 재생산 방법

재생산방법은 스트링을 결합하는데 사용하는 기능(operator)를 어떻게 선택하는가에 달려있다고 볼 수 있는데 다음은 기능자들의 종류이며 여기에서는 단일점 교배와 돌연변이를 사용한다. 이때 교배율은 0.65이고 돌연변이율은 0.008로 한다.

- 교배기능자(crossover operator)의 종류

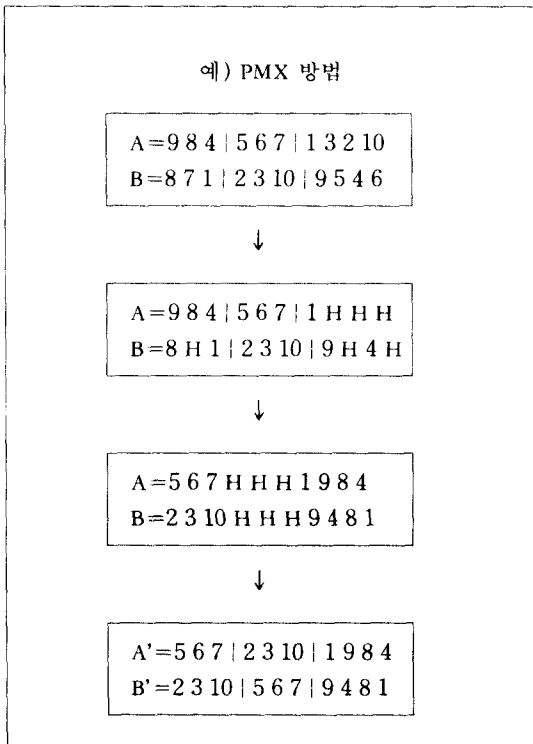
- 단순교배
- 순서교배(order based crossover) :
  - ① partially matched crossover(PMX)
  - ② order crossover(OX),
  - ③ cycle crossover(CX)

III - 1. 단순유전알고리즘

단순유전알고리즘은 스트링을 복제하고 스트링을 부분적으로 교체하는 것 이외에 어떠한 복잡한 과정을 거치지 않는 알고리즘으로써 사용되는 기능자는 다음과 같다[2].

- 재생산(reproduction)
- 교배(crossover)
- 돌연변이(mutation)

단순유전알고리즘의 처리과정은 다음과 같다.

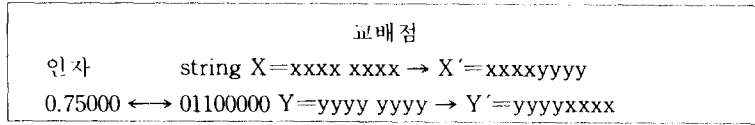


- [단계 1] 10 진수 인자 1과 0으로 구성된 string형으로 재표현하기 위한 string의 길이를 선정함.
- [단계 2] 단계 1의 string을 이용해 초기집단(population)을 구성함
- [단계 3] 집단의 각 string을 10진수 인자로 변환하고 적합도를 계산
- [단계 4] 적합도에 따라 해당 string들을 선택함.
- [단계 5] String간의 교배(crossover)와 돌연변이(mutation) 등의 진화과정을 거쳐 새로운 집단을 구성함.
- [단계 6] 단계 3이하의 과정을 반복함(적합도가 수렴할 때까지).

그림 1.4.(a)는 단계 1에서의 인자가 string 표현과 그 역표현(즉 string 표현으로부터 인자로의 표현)을 보여주고 있으며 단계 3에서의 적합도는 함수에 대한 최적 정도를 의미한다. 그리고 단계 4에서의 string선택은 적합도의 높은 정도에 의해서 이루어진다. 그림 1.4.(b)는 8비트인 두 string X와 Y의 교배를 보여주고 있다. 이때 교배점(crossing site)은 임의 난수로 선택하게 된다. 돌연변이는 교배와 동시 행해지며, 임의 비트를 선택하여 비트전환으로 이루어진다(단계 5). 한 예로 X=1001 일 때 임의로 선택된 비트가 3번째이면 돌연변이를 한 새로운 x''는 1011이 된다.

III. 단순유전알고리즘(SGA)과 미소유전알고리즘( $\mu$ -GA)

본 장에서는 현재 많이 사용하고 있는 단순유전알고리즘과 집단수를 줄여 실행속도를 빠르게 하는 미소유전알고리즘에 대해 설명한다.



(a) 인자와 string                      (b) string X와 Y의 교배

그림 1.4. 인자의 string 재표현과 그 교배  
Fig 1.4. string representation of parameter and crossover

단순유전알고리즘의 효율성을 보기위해 TSP문제에 적용시켜 본다. TSP문제는 순서가 있기 때문에 교배시 문제가 발생하므로 다음과 같은 선택들이 필요하다.

첫째, TSP문제 해결을 위한 string 표현은 다음과 같은 '인접표현(adjacency representation)'과 10진수 string을 사용한다.

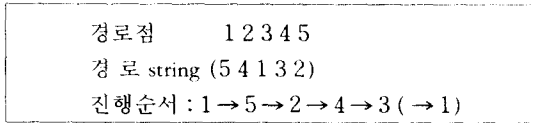


그림 1.5. 인접표현에 의한 string 구성  
Fig. 1.5. string composition by adjacent representation

둘째, 경로에 순서가 있으므로 순서를 고려한 Greedy 교배를 사용한다. Greedy 교배의 단계는 다음과 같다.

- [단계 1] 자손의 경로의 시작점으로써 아버지의 시작 경로 중 임의로 1개를 선택한다.
- [단계 2] 자손의 2번째 경로로써 두 아버지의 두번째 경로중 짧은 경로를 선택한다.
- [단계 3] 자손의 경로를 선택해가는 도중에 이미 선택한 경로만 남아있는 경우에 임의의 경로를 선택한다.
- [단계 4] 자손의 경로를 다 선택할 때까지 위 과정을 반복한다.(교배후 자손의 수는 1개가 된다.)

그림 1.7은 200개의 도시에 관한 초기여행경로를 나타내고 있고 그림 7은 493번의 세대교체를 시킨 후 나온 최적의 경로를 보여주고 있다.

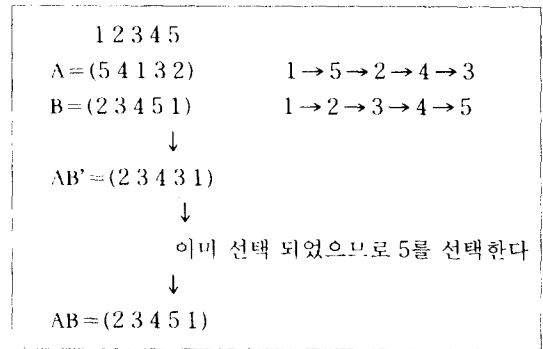


그림 1.6. Greedy 교배에 의한 generation  
Fig. 1.6. Generation through the Greedy crossover

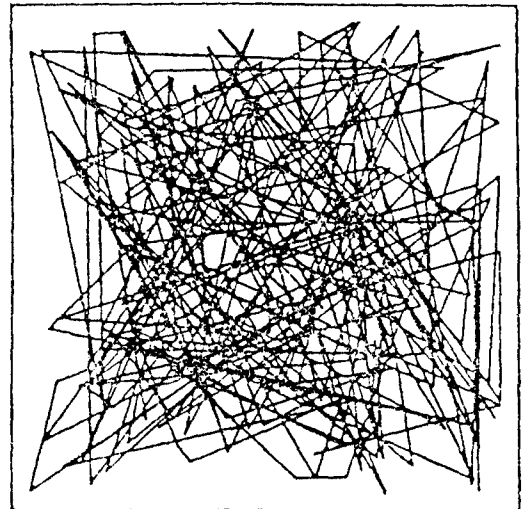


그림 1.7. 200개의 도시에 대한 초기경로  
Fig. 1.7. Initial path for 200-city

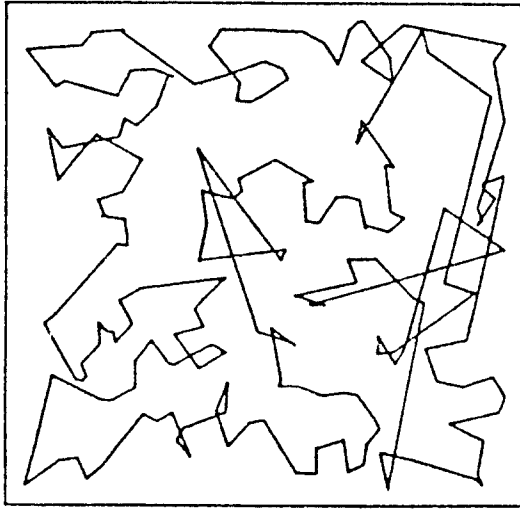


그림 1.8. 493번의 세대교체를 통해 찾은 최적 경로

Fig. 1.8. Optimal path obtained through 493 generations

### III -2. 미소유전알고리즘( $\mu$ -GA : Micro-Genetic Algorithms)

상기 단순유전알고리즘에서는 유전자 집단 크기 30에서 200개 정도이다. 따라서 유전 알고리즘의 계산을 위한 시간소비는 크다고 할 수 있다. 계산시간은 적은 집단의 선택으로 줄일 수 있는데 이를 위해 제안된 방법이 '미소 유전알고리즘( $\mu$ -GA : Micro-Genetic Algorithms)이다[3].

유전 알고리즘에서 집단수를 작게 하면 수행능력이 감소하는 경향이 있으므로 집단수를 작게하고도 수행능력에는 영향을 미치지 않는 특성을 갖는  $\mu$ -GA는 다음과 같은 처리과정을 갖는다.

[단계 1] 집단의 크기를 5개로 임의로 선택하거나 4개는 임의로 선택하고 1개는 이전의 탐색으로부터 나온 우수한 것으로 선택한다.

[단계 2] 적합도를 계산한 후 제일 우수한 것을 구한 다음 이 스트링을 5번째 스트링으로 두고 다음 세대로 넘긴다.(엘리트 규칙)

[단계 3] 남은 4개의 스트링을 재생산하기 위해 임의로 배열하여 짝을 지운다.(이때 같은 스트링끼리의 짝짓기는 피한다.)

[단계 4] 교배를 시킨다.

[단계 5] 수렴성을 검사한 후 수렴하면 [단계 1]로 간다.

[단계 6] [단계 2]로 간다.

상기 과정에서 돌연변이는 시키지 않는데 이는 [단계 1]에서 4개의 string이 임의로 선택되기 때문이다.

미소유전알고리즘의 효율성을 찾기위해 단순유전알고리즘의 경우처럼 TSP문제에 적용시켜 본다. 스트링표현 및 교배방법은 동일하며 문제해결을 위해 수정 적용된  $\mu$ -GA의 단계는 다음과 같다[7] [8].

[단계 1] 초기 집단의 크기를 임의로 5개를 선택하고 이후에는 4개는 임의로 선택하고 1개는 이전의 탐색으로부터 나온 우수한 것으로 선택한다.(이때 스트링은 10진수로 표현하며 한개의 스트링내에 중복된 숫자가 나오지 않도록 선택한다.)

[단계 2] 적합도를 계산한 후 제일 우수한 것을 구한 다음 이 스트링을 5번째 스트링으로 두고 다음 세대로 넘긴다.(엘리트 규칙)

[단계 3] 남은 4개의 스트링을 재생산하기 위해 임의로 배열하여 8쌍의 짝을 지운다.(이때 같은 스트링끼리의 짝짓기는 피한다.)

[단계 4] 교배를 시킨다.(Greedy crossover 사용)

[단계 5] 수렴성을 검사한 후 수렴하면 [단계 1]로 간다.

[단계 6] [단계 2]로 간다.

그림 1.9는 20개의 정방형 조각의 경로를 찾기 시작할때의 경로이고 그림 8과 그림 9는  $\mu$ -GA가 찾은 최적 경로를 나타내고 있다. 이때 최적의 경로가 1개 이상임을 알 수 있다.

그림 1.12는 적합도값과 Generation과의 관계를 보여주고 있는데 약 500 Generation에서 최적의 경로를 찾음을 알 수 있다. 500 Generation이라 할지라도 집단수가 5개에 불과하므로 소요된 계산시간은 매우 작다. 그림 1.13은 임의의 30개 조각의 경로를 찾기 시작할때의 경로이며 그림 1.14는  $\mu$ -GA가 찾은 최적의 경로를 보여주고 있다. 직관적으로도 최적임을 알 수 있다. 그림 1.15는 적합도값과 Generation과의 관계를 나타내고 있으며 약 800번 정도에서 최적의 경로를 찾았음을 알 수 있다. 상기 과정에서 적합도값이 최적 주행시간이다[8].

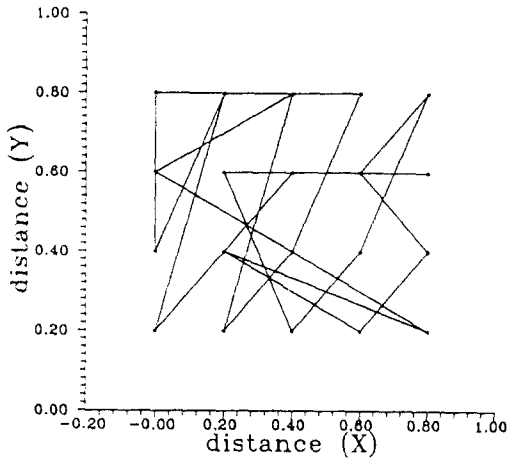


그림 1. 9. 정방형으로 위치한 20개 도시에 대한 초기 경로  
 Fig. 1. 9. Initial path of 20-cubic-city

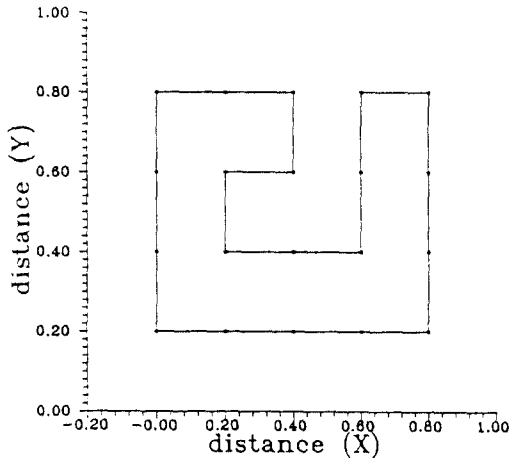


그림 1. 10. 정방형으로 위치한 도시의 최적 경로(1)  
 Fig. 1. 10. Optimal path(1) of 20-cubic-city(1)

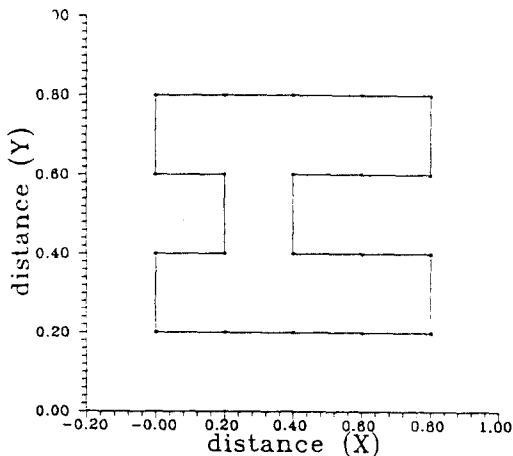


그림 1. 11. 정방형으로 위치한 도시의 최적 경로(2)  
 Fig. 1. 11. Optimal path(2) of 20-cubic-city(2)

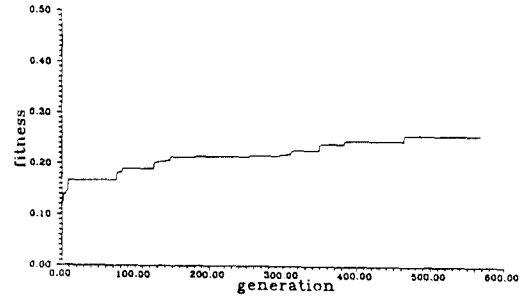


그림 1. 12. 정방형으로 위치한 도시에 대한 적합도  
 Fig. 1. 12. Fitness of cubic-city

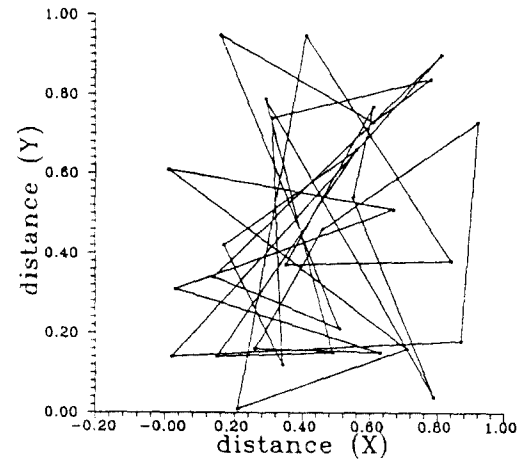


그림 1. 13. 임의로 위치한 도시 30개에 대한 초기 경로  
 Fig. 1. 13. Initial path of randomly placed 30-city

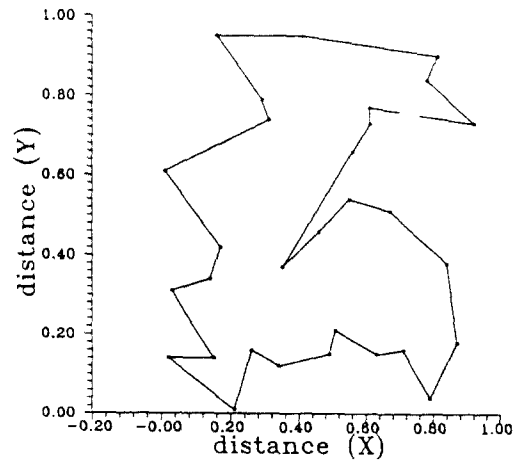


그림 1. 14. 임의로 위치한 도시 30개에 대한 최적 경로  
 Fig. 1. 14. Optimal path of randomly placed 30-city



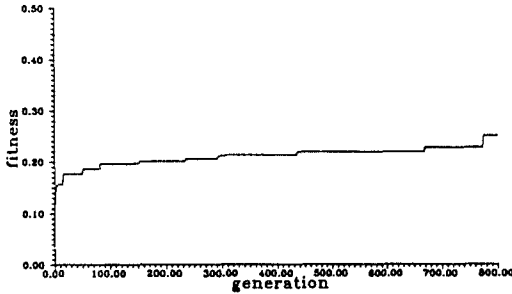


그림 1. 15. 임의로 위치한 도시에 대한 적합도  
Fig. 1. 15. Fitness of randomly placed city

IV. 결론 및 앞으로의 연구방향

최근에 유전알고리즘은 적용범위가 넓어지고 있고 그 효율성이 입증되고 있다. 현재까지 개발된 기능자(operator)들로써 문제를 훌륭히 해결해 나가고 있으며 또한 문제의 특성에 따라 기능자의 개발이 이루어 지리라 생각된다. 아직까지 국내에서는 유전알고리즘에 관한 연구가 미미한 실정이나 앞으로는 본격적으로 연구, 적용될 것으로 보는데 보다 획기적인 기능자의 개발로 온라인 적용까지 가능할 것으로 생각된다.

참 고 문 헌

1. J.Y.S.Luh and C.S.Lin, "Optimum path Planning for Mechanical Manipulators," Trans. of the ASME, vol. 102, June, 1981, pp. 142-151.
2. D.E.Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley Publishing Company, 1989.
3. Krishnakumar.K, "Micro-Genetic Algorithms for stationary and non-stationary function optnmization," Proc. SPIE Intell. Cont. adapt. Syst., 1196, 289-296, Nov. 7-8, Philadelphia, Pa., 1989.
4. Yuval Davidor, Genetic Algorithms and Robotics, A Heuristic Strategy for Optimization, World Scientific, 1991
5. L.Davis, Handbook of Genetic Algorithms, Van Nostrand ReinHold, New York. 1991.
6. 박세희, 김용호, 심귀보, 전홍태, "유전알고리즘을 이용한 퍼지 규칙 베이스의 자동생성", 전자공학회

지, vol.29-B, pp.60-67. 1992.

7. 김용호, 강훈, 전홍태, "유전알고리즘을 이용한 로봇 매니플레이터의 최적시간 경로계획", 한국자동제어 학술회의, pp. 436-440. 1992.
8. 김용호, 강훈, 전홍태, "다작업 로봇매니플레이터의 최적시간 경로 계획을 위한 미소유전알고리즘의 적용", 전자공학회지 심사중.



김 용 호

- 1986년 : 광운대학교 전자공학과 졸업
- 1988년 : 중앙대학교 전자공학과(석사)
- 1992년~현재 : 중앙대학교 전자공학과 박사과정
- 관심분야 : Robotics, Fuzzy Logic Control, Genetic Algorithms 등



전 홍 태

- 1976년 : 서울대학교 전자공학과 졸업
- 1983년 : 뉴욕주립대 전기 및 전자공학과(석사)
- 1986년 : 뉴욕주립대 전기 및 전자공학과(박사)
- 1986년~현재 : 중앙대학교 전자공학과 부교수
- 관심분야 : Robotics, Fuzzy-Neural Fusion, Genetic Algorithms 등