

## 퍼지제어기를 위한 유전알고리즘(Genetic Algorithms)의 적용

金容虎, 全洪兌  
中央大學校 電子工學科

### I. 서론

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

최근에는 최적화 및 탐색문제, 제어규칙 찾는문제 그리고 신경회로망 학습문제등 여러부문에 많이 응용되고 있으며 그의 적용이 확대될 전망이다. 따라서 본 논문에서는 유전알고리즘을 소개하고 함수 최적화 문제와 Fuzzy 제어규칙의 자동생성을 위해 유전알고리즘이 어떻게 적용되고 어느정도 효용성이 있는지 살펴보기로 한다.

유전알고리즘은 자연생태계의 진화메카니즘을 모방하였는데 실제로 자연계의 진화과정이 모두 밝혀져있지는 않지만 중요한 몇가지는 알려져 있어 유전알고리즘은 이러한 진화과정에서 일부 관찰된것을 사용하였다. 자연생태계의 진화과정을 간단히 살펴보면 다음과 같다.<sup>[6]</sup> 진화는 염색체(chromosome: 생명체의 구조를 부호화하기 위한 기관소자)에서 일어나며 생명체는 염색체를 부호화하는 과정을 통해 부분적으로 창조된다. 염색체의 부호화-복호화 과정의 자세한 내용은 알수 없으나 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 염색

체를 보다 자주 부호화 해주는 과정이다.

3) 재생산(reproduction)과정은 진화가 일어나는 시점이다. 돌연변이는 자손의 염색체와 어버이의 염색체를 다르게 할 수있고 재조합처리는 어버이의 염색체들을 결합하므로써 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 극복해왔듯이 어려운 문제를 해결하는 기술이 될것이라 믿었다. 그래서 염색체를 0과 1로 구성된 2진 string으로 표현하고 각각의 염색체의 집단에 대해 모의진화(simulated evolution)를 실행하였다. 자연계처럼 유전알고리즘은 훌륭한 염색체를 찾는 문제에 대해 염색체속에있는 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 평가의 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)

또한 유전알고리즘을 문제해결에 적용 시키기 위한 2가지 메카니즘이 있다.

첫째, 염색체상에 문제의 해를 부호화 하는 방법

둘째, 문제속에서 염색체의 가치 측정을 하는 평가 함수 설정 방법

첫번째의 경우 부호화 하는 기술은 문제마다 다르며 부호화는 bit string으로 수행한다. 두번째의 경우 평가함수는 유전 알고리즘과 해결해야 할 문제를 결합시켜주며 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 전개해나가 본다.

본 원고의 구성은 다음과 같다. 2 장에서는 기본적인 유전 알고리즘의 전개 및 최적화문제에 대한 적용 예를 보이고, 3 장에서는 단순유전알고리즘(SGA: Simple Genetic Algorithms)과 미소유전알고리즘( $\mu$ -GA: Micro-Genetic Algorithms)의 소개 및 TSP문제로의 적용을, 4장에서는 퍼지제어기의 규칙 자동생성에 대한 적용예를 보이고, 5장에서는 결론 및 향후 연구방향이 제시된다.

## II. 유전 알고리즘 (Genetic Algorithms)

유전알고리즘의 바탕이 되는 자연 생태계의 일반적인 진화과정은 다음과 같다.<sup>[7]</sup>

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 재생산 하도록 해준다. - 재생산(reproduction) 은 진화가 일어난 시점이다. -

세째, 돌연변이는 자손의 염색체를 아버지 염색체와 다르게 해준다.

네째, 재조합(Reproduction) - 교배(crossover) - 과정은 아버지의 염색체를 결합하여 매우 다른 염색체를 자손에게 준다.

이 유전알고리즘은 미지수 함수  $Y=G(x)$ 의 최적해를 발견하는 모의진화(Simulated evolution) 형의 탐색(Search) 알고리즘의 성격을 갖으며 John

Holland 는 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. 세째, 유전 알고리즘은 global 한 최적해의 발견을 가능케 한다.

다음은 기본적인 유전알고리즘의 처리과정을 나타내고 있으며 이후 이 처리과정을 바탕으로 평가함수로써 2진 f6함수를 최적화 시키는 과정을 전개해 본다.

### 유전알고리즘

1. 염색체의 집단을 초기화 한다
2. 집단속에 있는 각각의 염색체를 평가한다
3. 현재의 염색체들을 교배시킨후 새로운 염색체들을 만든다 : 아버지 염색체들을 결합시킬때 돌연변이와 재조합(recombination)을 적용한다.
4. 아버지 염색체들을 없앤다
5. 만들어진 염색체들을 평가한후 집단속에 넣는다
6. 시간이 되면 중지하고 최상의 염색체를 제공한다 : 그렇지 않으면 3번으로 간다

그림 1. 유전알고리즘의 처리과정

상기 처리과정의에 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 변환하기위해 염색체를 44 비트로 구성한다. 이 2개의 수가 함수에 대입이된후 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.

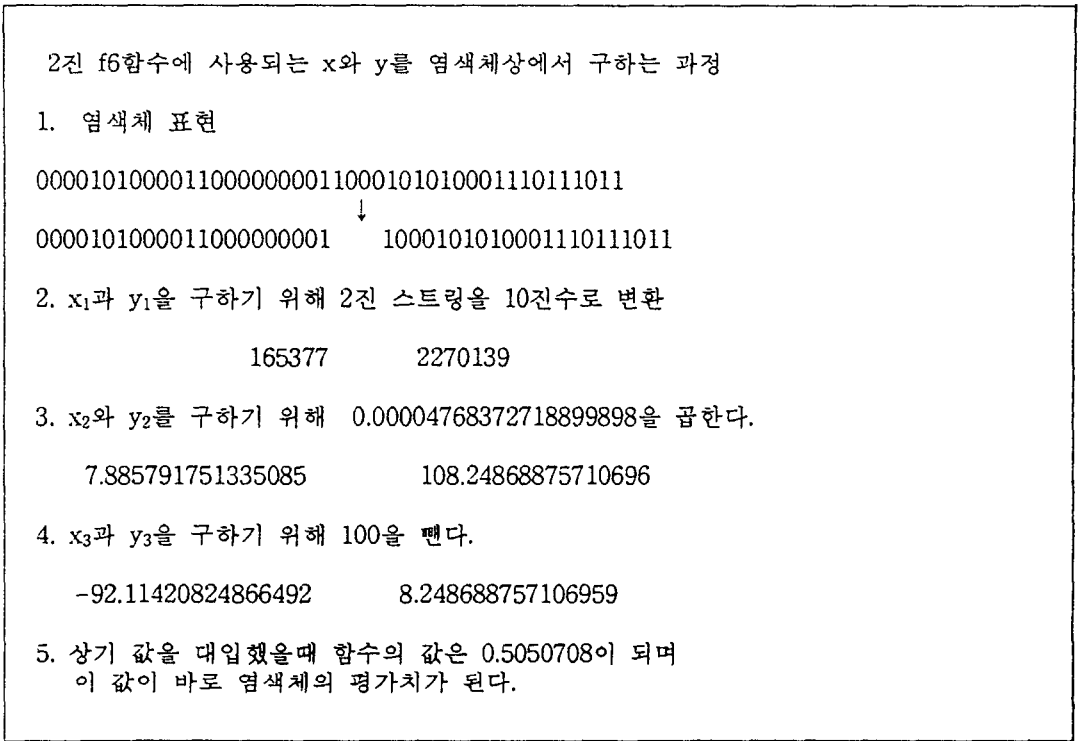


그림 2. 2진 f6 함수에 사용되는 값의 염색체표현 및 복호화 과정

세째, x<sub>3</sub>과 y<sub>3</sub>을 -100에서 100까지의 범위에 있게 하면서 구하기위해 x<sub>2</sub>와 y<sub>2</sub>에서 100을 뺀다.  
 그림 2는 상기과정을 보여주고 있다. 입력으로써 x<sub>3</sub>과 y<sub>3</sub>을 사용하는 2진 f6함수는 다음과 같다.

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

우리의 목적은 이 함수의 최대값을 구하기위한 x와 y를 결정하는 것인데 이것이 바로 '함수의 최적화'이다. 유전알고리즘의 효율성을 보기위해 2진 f6 함수를 사용하는 이유는 그림 3에서 보우지는 것과 같이 이 함수가 단 하나의 최적해를 갖고 있으며, 오실레이션을 하기때문에 기존의 함수최적화 기법으로는 해결하기 어렵기 때문이다.

그림 3은 f6함수의 궤적을 나타낸것이다. 이 그래프에서 x는 -100에서 100까지의 범위를 갖으며 x=0에서 최적값을 갖으며 x와 y에 관해 대칭인 모양을 갖는다.

2) 집단 설정  
 집단을 설정하는데에는 여러가지 방법이 있는데 그중 간단한 방법을 고찰해본다. 우선 집단 및 각각의 염색체를 2진수로 표현한다. 여기에서는 1개의 염색체를 44비트로 구성한다. 그리고 집단의 초기화는 난

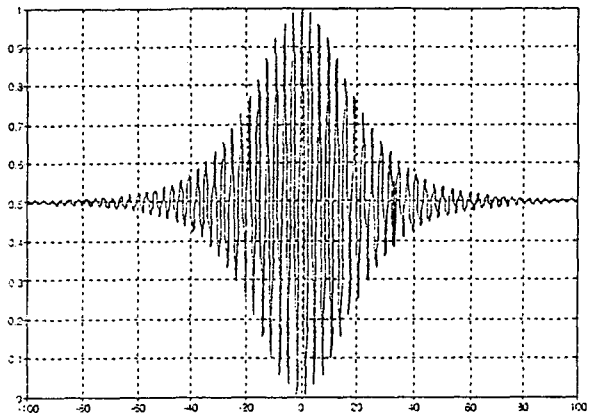
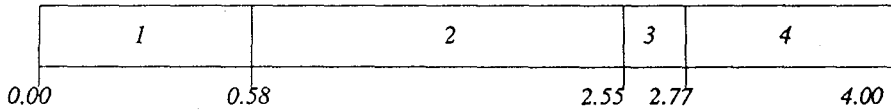


그림 3. x=0에서 최적값을 갖는 f6 함수의 그래프

Roulette wheel 방법(컴퓨터로 구현)



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

난수를 0~4까지 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	0 1 1 0 1
2	1 1 0 0 0
3	1 1 0 0 0
4	1 0 0 1 1

수발생에 의하여 알고리즘의 전개는 이 초기집단으로부터 시작된다. 다음세대를 만들었을때 이 초기집단을 전부 삭제한다. 재생산은 이전의 집단을 새로운 집단으로 대체 시키는것이며 이때 이를 '세대교체 (generation)'라 한다. 아버지의 선택은 roulette wheel방법을 사용하며 집단의 크기는 파라미터 값을 고려하여 100개로 설정한다. 세대교체는 40번 시행해본다. 다음은 아버지를 선택하는 roulette wheel 방법이다.

3) 재생산 방법

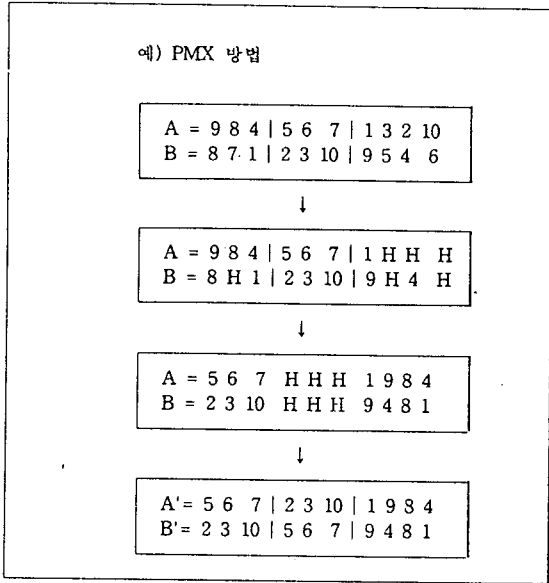
재생산방법은 스트링을 결합하는데 사용하는 기능 (operator)를 어떻게 선택하는가에 달려있다고 볼수 있는데 다음은 기능자들의 종류이며 여기에서는 단일

점 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)

이상은 함수최적화 문제를 해결하기 위한 유전알고리즘의 처리절차를 보인 것이며 다음 장에서는 유전 알고리즘 중에서 대표적으로 사용되는 방법을 소개한다.



### III. 단순유전알고리즘(SGA)과 미소유전 알고리즘( $\mu$ -GA)

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

#### 1. 단순유전알고리즘

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

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

단순유전알고리즘의 처리과정은 다음과 같다.

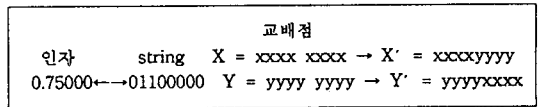
- [단계 1] 10 진수 인자 1과 0으로 구성된 string 형태로 재표현하기 위한 string의 길이를 선정함.
- [단계 2] 단계 1의 string 을 이용해 초기집단(population)을 구성함
- [단계 3] 집단의 각 string 을 10진수 인자로 변환하고 적합도를 계산
- [단계 4] 적합도에 따라 해당 string 들을 선택함.

[단계 5] String 간의 교배(crossover)와 돌연변이(mutation)등의 진화과정을 거쳐 새로운 집단을 구성함.

[단계 6] 단계 3 이하의 과정을 반복함(적합도가 수렴 할 때까지).

그림 4.(a) 는 단계 1 에서의 인자의 string 표현과 그 역 표현(즉 string 표현으로부터 인자로의 표현)을 보여주고 있으며 단계 3에서의 적합도는 함수에 대한 최적 정도를 의미한다. 그리고 단계 4에서의 string 선택은 적합도의 높은 정도에 의해서 이루어진다.

그림 4.(b)는 8비트인 두 string X와 Y의 교배를 보여주고 있다. 이때 교배점(crossing site)은 임의의 난수로 선택하게 된다. 돌연변이는 교배와 동시에 행해지며, 임의의 비트를 선택하여 비트전환으로 이루어진다(단계5). 한 예로 X = 1001 일때 임의로 선택된 비트가 3번째이면 돌연변이를 한 새로운 x 는 1011이 된다.



(a) 인자와 string (b) string X와 Y의 교배

그림 4. 인자의 string 재표현과 그 교배

단순유전알고리즘의 효율성을 보기위해 TSP문제에 적용시켜본다. TSP문제는 순서가 있기때문에 교배시 문제가 발생하므로 다음과 같은 선택들이 필요하다.

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

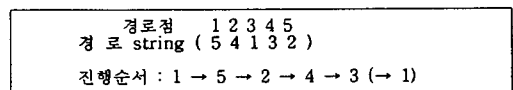


그림 5. 인접표현에의한 string 구성

둘째, 경로에 순서가 있으므로 순서를 고려한 Greedy 교배를 사용한다.

Greedy 교배의 단계는 다음과 같다.

- [단계 1] 자손의 경로의 시작점으로써 아버지의 시작경로중 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.)

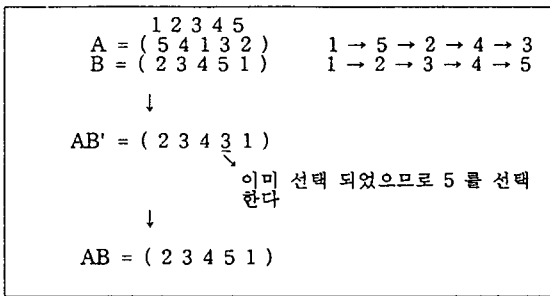


그림 6. Greedy 교배에의한 generation

그림 7은 200개의 도시에관한 초기여행경로를 나타내고있고 그림 8은 493번의 세대교체를 시킨후 나온 최적의 경로를 보여주고 있다.

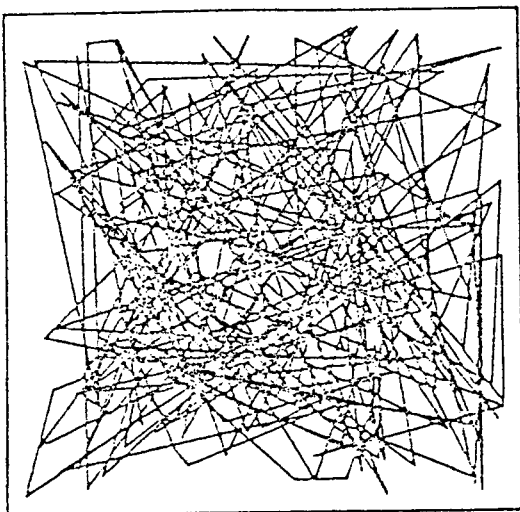


그림 7. 200개의 도시에대한 초기경로

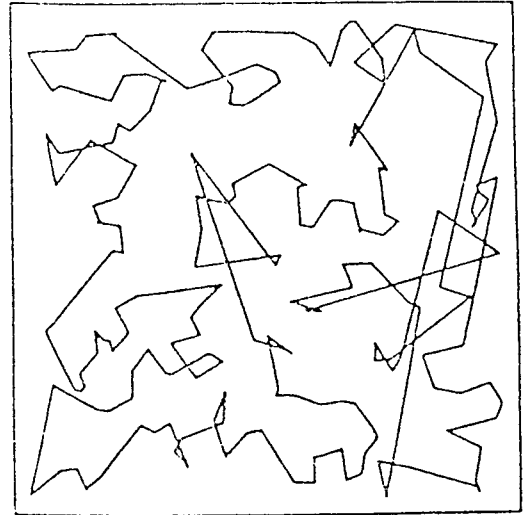


그림 8. 493번의 세대교체를 통해 찾은 최적 경로

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

상기 단순유전알고리즘에서는 '유전자 집단'의 크기가 30에서 200개 정도이다. 따라서 유전 알고리즘의 계산을 위한 시간소비량은 크다고 할 수 있다. 계산 시간은 적은 집단의 선택으로 줄일 수 있는데 이를 위해 제안된 방법이 '미소 유전알고리즘( $\mu$ -GA: Micro-Genetic Algorithms)이다.<sup>[3]</sup>

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

- [단계 1] 집단의 크기를 5개로 임의로 선택하거나 4개는 임의로 선택하고 1개는 이전의 탐색으로 부터 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 4 개의 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.

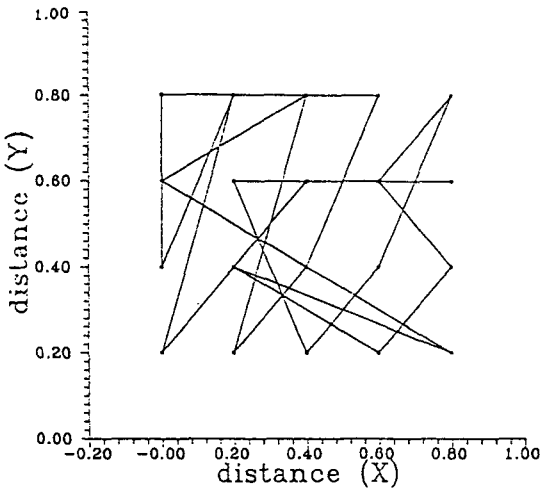


그림 9. 정방형으로 위치한 20개도시에대한 초기 경로

그림 9는 20개의 정방형 조각업의 경로를 찾기 시작할 때의 경로이고 그림 10과 그림 11은  $\mu$ -GA가 찾은 최적 경로를 나타내고 있다. 이때 최적의 경로가 1개 이상임을 알수 있다.

그림 12는 적합도값과 Generation과의 관계를 보여

주고 있는데 약 500 Generation에서 최적의 경로를 찾을 수 있다. 500 Generation이라 할지라도 집단수가 5개에 불과하므로 소요된 계산시간은 매우 작다.

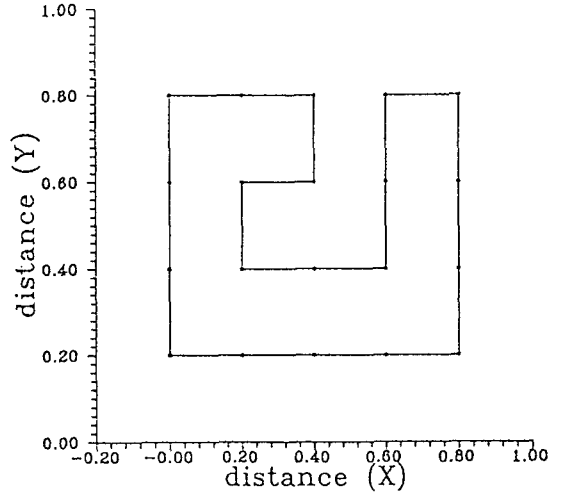


그림 10. 정방형으로 위치한 도시의 최적 경로(1)

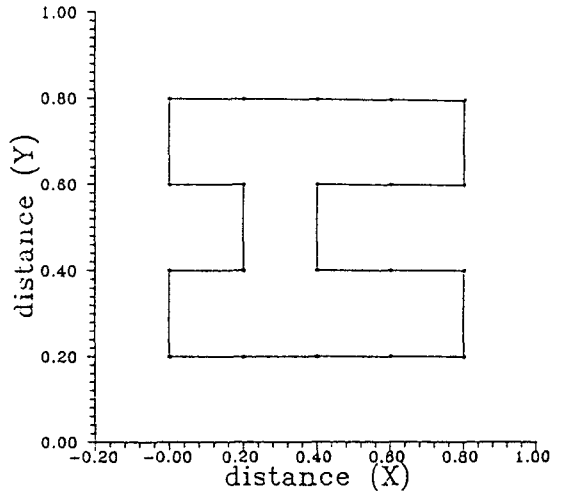


그림 11. 정방형으로 위치한 도시의 최적 경로(2)

#### IV. 퍼지제어기 규칙의 자동생성을 위한 유전알고리즘(SGA)의 적용

퍼지 논리 제어기는 구조가 간단하여 실시간 제어 가 가능하고 시스템의 구조나 동적 특성을 필요하지

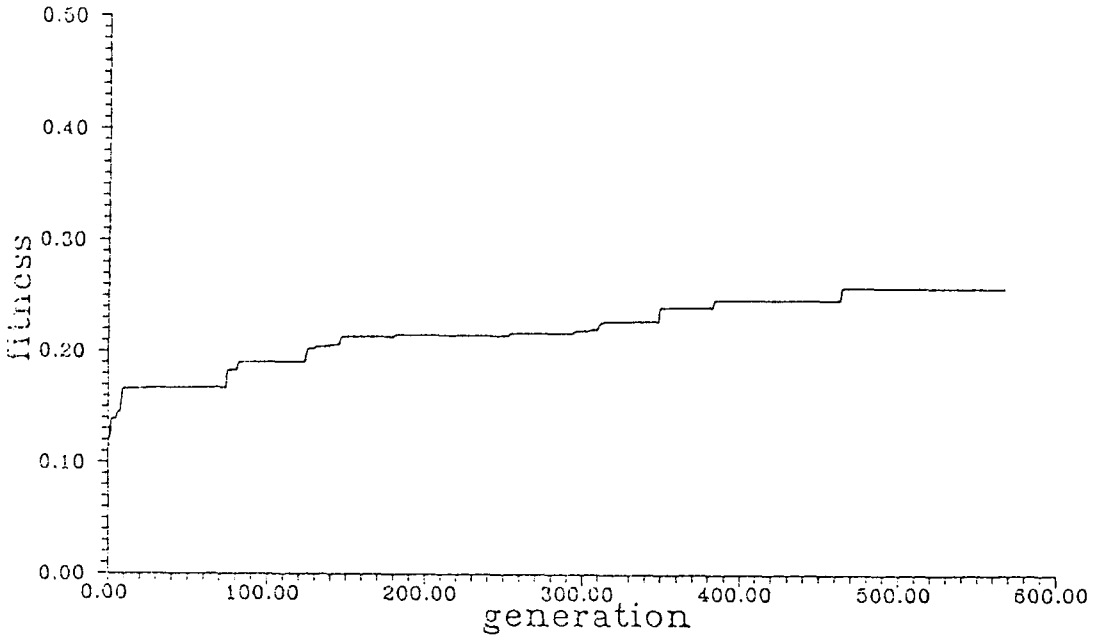


그림 12. 정방향으로 위치한 도시에대한 적합도

않는 등의 많은 잇점 때문에 최근에 그 응용 범위가 확대되고 있다.<sup>[12]</sup> 그러나 퍼지 논리 제어기의 제어 규칙은 일반적으로 전문가가 구성해 주기 때문에 제어 대상이 시변 또는 비선형 시스템인 경우 정밀한 제어 규칙을 만드는 일은 용이한 일이 아니다. 이에 본 장에서는 상기의 유전 알고리즘(Genetic Algorithm)을 이용하여 제어 목적에 적합한 퍼지 제어 규칙을 자동 생성 할 수 있는 방법을 제안한다. 그리고 2 자유도 로봇 매니퓰레이터의 궤적 제어에 적용하여 제안 된 방법의 유효성을 입증한다.

퍼지 논리 제어기(fuzzy logic controller)는 기본적으로 기존의 제어 이론에 퍼지 집합 이론과 언어적 제어(linguistic control)의 규칙 베이스를 첨가하여 요구되는 성능을 만족하도록 구성된 폐쇄한(feed-back) 제어 시스템이다. 이러한 퍼지논리 제어기는 복잡한 비 선형 시스템의 제어시 퍼지 집합을 분할하여 각 영역에 따른 규칙 베이스를 구성하면 기존의 비선형 제어기에 비해 훌륭한 성능을 얻을 수 있으며, 시스템의 동적 특성이 시변(time varing)일 경우 제어기 자체가 시스템의 응답에 대응하므로 기존의 제어기보다 비 선형-시변 시스템의 제어가 용이할 수 있다. 그러나 퍼지 논리 제어기는 제어 대상 시스템의 수학적 모델링이 필요로 하지 않기 때문에 타당

한 규칙 베이스를 구성하는 일은 매우 어려운 과제이다. 따라서 상기의 유전 알고리즘을 응용하여 퍼지 제어기를 구성할 경우 기존의 합성방식보다도 적응성이 뛰어난 훌륭한 적응 퍼지 제어기의 구현이 가능하게 된다. 일반적으로 퍼지 논리 제어기는 기본적으로 다음과 같은 4가지 구성요소로 되어 있다.

- (i) 퍼지화기(fuzzifier)
- (ii) 퍼지 규칙 베이스(fuzzy rule base)
- (iii) 퍼지 추론기(fuzzy inferencing engine)
- (iv) 비퍼지화기(defuzzifier)

상기 4가지 구성 요소중 논리 제어기를 구성하는데 어려운 점은 최적의 규칙 베이스의 구성과 퍼지화기 혹은 비퍼지화기의 환산계수 결정이다. 먼저 퍼지 규칙 베이스는 일반적으로 전문가의 지식이나 실험에 의해 규칙 베이스를 구성할 수 있다. 그러나 제어플랜트가 높은 비선형 시스템인 경우에는 규칙 베이스의 구성에 어려움 있다. 설사 규칙 베이스를 구축하였다해도 그것이 최적의 구성인지 알수없다. 따라서 상기와 같은 비선형 시스템의 경우 규칙 베이스는 다음과 같은 자기 조정능력들을 가져야 한다.

첫째, 퍼지 변수(fuzzy variable)들에 주어진 소속함수(membership function) 들을 자율적으로 생성할 수 있어야 한다.



둘째, 내외부의 작업 환경변화에 따라 적절하게 추론 과정을 조정할 수 있어야 한다.

그리고 적응적인 퍼지 규칙베이스 구성이외에 중요시 해야 하는 것은 퍼지화기(fuzzifier)와 비퍼지화기에서의 환산계수 결정이다. 환산계수들은 퍼지제어기의 전체성능에 큰 영향을 미치기 때문에 환산계수들의 자율적인 조정이 요구된다. 본 장에서는 적응적인 퍼지규칙 베이스 구성과 자율적인 환산계수들의 결정을 위해 퍼지제어기에 유전 알고리즘(genetic algorithms) 기법을 도입하여 적응 퍼지제어기를 설계한다. 한편 퍼지 논리 제어기의 구성은 그림 13과 같다. GE는 Error의 전체 집합(universe of discourse)을 정하는 환산 계수(scaling factor)이고 GC는 Change of Error의 환산 계수(scaling factor)이다. 그리고 C는 제어기의 명령이 되며 R은 제어대상의 명령에 대한 결과이다.

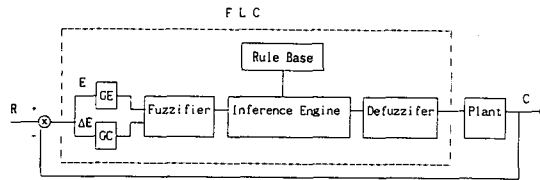


그림 13. 퍼지 제어기

그림 13의 퍼지 논리 규칙 베이스는 식 (1)과 같은 "If-Then"형식의 규칙들로 구성되어 있다. 그러나 제어 대상에 따라 적절한 규칙을 구성하는 일은 용이하지 않고 특히 제어 대상이 시변, 비선형일 경우 그 구성은 매우 어렵다.

$$\underbrace{\text{If } X \text{ is } A_i \text{ and } Y \text{ is } B_i}_{\text{전건부}} \text{ then } \underbrace{U \text{ is } C_i}_{\text{후건부}} \quad (i=1 \cdots N) \quad (1)$$

여기에서 X, Y 와 U는 각각 입력 및 출력 변수들이고, Ai, Bi, Ci는 그림 2과 3의 소속함수들을 갖는 퍼지 변수들이다.

식 (1)의 N개로 구성되어 있는 규칙 베이스를 제어 대상에 맞게 조정할 수 있는 방법은 각 규칙의 전건부(antecedent)와 후건부(consequent)의 퍼지 변수 소속함수(membership function)들을 변경하거나 입력들의 (universe of discourse상의 설정을 위한) scaling factor를 조정하는 것이다. 이에

본 연구에서는 제어 환경에 맞는 최적 퍼지 규칙 베이스를 자동 생성하기 위해 규칙 베이스의 후건부 Ci (i=1...N)의 소속함수들과 scaling factor를 주요 조정 대상 혹은 최적화 인자로 삼는다.

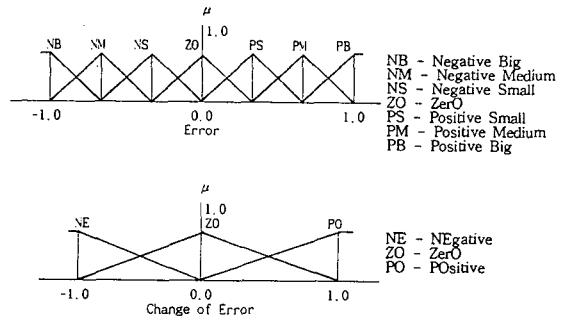


그림 14. 전건부의 소속함수

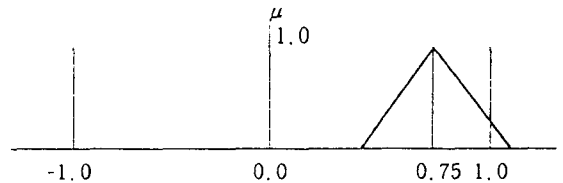


그림 15. 후건부 언어 변수의 소속 함수

### 1. 최적화 대상 인자와 적합도

환산계수(GE와 GC)는 입력변수를 전체집합 상에 설정하기위한 파라미터로 퍼지 변수의 소속함수에 못지 않게 퍼지 논리 제어기의 성능에 큰 영향을 미친다. 이에 (10진수 형의) 환산계수를 최적화 대상의 하나로 선정하고, 유전 알고리즘을 적용하기 위해 이들 값들을 unsigned binary로 표현한다.

그러나, 또 하나의 최적화 대상인자인 후건부 소속함수들을 특징 지우는일은 환산계수처럼 단순하지 않다. 일반적으로 전체집합 상에서 이등변 삼각형의 소속함수를 특징 지우는데에는 3개 꼭지점들의 위치들을 필요로 하지만, 삼각형 소속함수의 밀변 길이를 일정하게 유지하면 상단 꼭지점 만으로도 그 소속함수를 특징 지울 수 있다. 이에 전체집합 상의 상단 꼭지점 위치를 다른 하나의 최적화 대상으로 삼으며, 그 위치를 규칙 베이스의 적합도에 따라 유전 알고리즘을 이용 조정한다. 이를 위해 꼭지점의 상단 위치

들을 signed binary로 표현한다.

그림 16은 binary로 표현된 환산계수와 (한 규칙 베이스를 구성하는 후건부 퍼지 변수들의) N개 소속 함수들을 하나의 string으로 합성한 예를 보여 주고 있다. 이 string을 이용 유전 알고리즘에 의한 최적화 과정이 이루어진다.

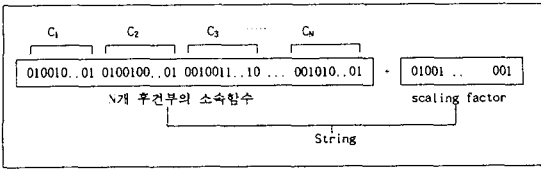


그림 16. 하나의 string으로 합성한 환산계수와 N개의 소속함수

한편, 적합도는 퍼지 논리 규칙의 성능 평가의 기준이 되는 함수이며 집단내에서 진화 대상의 요구되는 string을 선택하는 기준이 된다. 따라서 그림 2의 제어 시스템에서의 평가 함수는 다음과 같이 설정한다.

$$Fitness = \frac{K_1}{Error} + \frac{K_2}{Change\ of\ Error} + \frac{K_3}{Energy} \quad (2)$$

Error Change of Error Energy 여기에서 Energy는 plant에 가해지는 입력이며 \$K\_1, K\_2, K\_3\$는 상수이다. 식 (2)에서 높은 적합도는 Error 및 Change of Error, Energy의 감소를 의미하기 때문에 집단내에서의 높은 적합도를 갖는 string은 제어 목적에 좀더 접근한 제어 규칙들로 구성된 규칙 베이스를 의미한다.

2. 진화에 의한 규칙 베이스의 생성

최적화 대상 인자들과 이들의 string 표현법을 결정 한 후 유전 알고리즘에 한 규칙 베이스의 최적화 과정은 다음과 같이 이루어진다.

1) 초기 집단의 구성

초기 집단을 구성하기 위해서는 우선 초기 string을 결정해야한다. 초기 string은 초기에 구성된 규칙 베이스와 초기 환산계수값을 이용 구성 할 수 있고, string의 표현법(이진수 표현 및 string length)에 맞추어 난수 발생에 의해 임의로 구성 할 수도 있다. 이 초기 string과 난수 발생에 의해 구성된 M개의 string(혹은 규칙 베이스)들을 이용 하나의 초기 집

단을 구성한다. 이때 집단은 임의의 M개의 규칙 베이스들을 binary로 표현한 집단을 나타낸다. 그림 17은 집단의 한예를 보여준다.

	규칙들 후건부의 \$C_i\$	환산계수
규칙 베이스 1	001001000.....100010010100	1010100...1010
규칙 베이스 2	001001001.....010010010101	0010010...1101
규칙 베이스 3	010010010.....010001000100	0100000111011
	001010100.....000101101011	1011101...1001
~		
규칙 베이스 M	001001010.....110101000100	1010011...0101

그림 17. 초기 String 집단

2) 진화에 의한 새로운 규칙 베이스 구성

초기 string 집단의 각 행(혹은 string)은 그림 6에서 보여지는 것처럼 하나의 규칙 베이스와 환산계수들을 나타낸다. 집단의 각행(혹은 string)에 의한 적합도는 다음과 같이 구해진다.

[단계 1] String을 10진수 값으로 역변환하여 퍼지 규칙들의 후건부 변수들의 소속함수와 환산계 수를 결정한다.(그림 18 참조)

[단계 2] 구성된 규칙 베이스와 환산계수들을 이용 퍼지 논리 제어기를 구성한다.

[단계 3] 단계 2의 퍼지 논리 제어기를 이용하여 Error, Change of Error 및 Energy 그리고 식 (2)를 이용하여 한 string의 적합도를 구한다.

이와 같은 과정을 집단의 모든 행(혹은 string)에 적용하여 모든 규칙 베이스들의 적합도를 결정한다.

그림 18은 단계 1에서 한 string을 규칙 베이스의 소속함수로 역변환 하는 과정을 보여주고 있다.

모든 string(혹은 규칙 베이스)들의 적합도를 구한 후 이들 중 우성인(적합도가 높은) string(혹은 규칙 베이스)들을 이용, 다음 세대가 될 string들을 선택 한다. 즉, 높은 적합도를 갖는 string에 높은 선택 확률을 주어 선택될 가능성을 높여준다.<sup>[4]</sup> 한 예로 집단내에 4개의 string \$A\_1, A\_2, A\_3\$, 그리고 \$A\_4\$가 있고 각각의 적합도가 0.33, 0.17, 0.04, 0.46이라면 \$A\_1\$은 33%, \$A\_2\$는 17%, \$A\_3\$은 4%, 그리고 \$A\_4\$는 46%의 선택확률을 가진다. 표 1은 각각으로부터 예상 할 수 있는 선택 횟수 분포의 한 예를 보여준다.

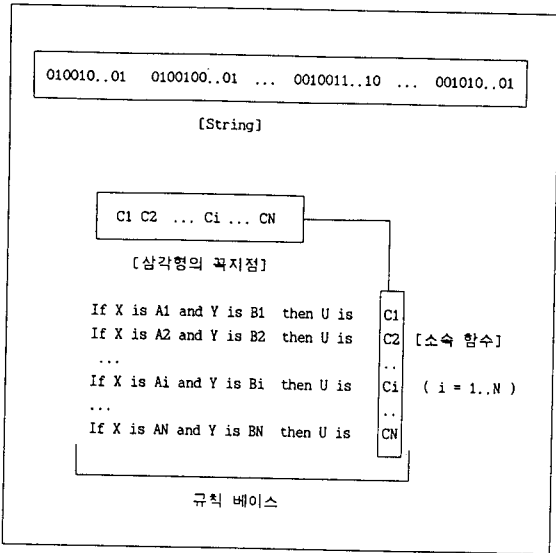


그림 18. String으로부터 후건부 소속함수의 결정

표 1. 적합도로부터의 선택

String	Fitness	선택확률	예상 분포 (1)	예상 분포 (2)
A <sub>1</sub>	0.33	33x	2	1
A <sub>2</sub>	0.17	17x	0	1
A <sub>3</sub>	0.04	4x	0	0
A <sub>4</sub>	0.46	46x	2	2
합	1.00	100x	4	4

선택된 string들로부터 교배(crossover)와 돌연변이(mutation)의 진화 과정<sup>[4]</sup>을 거쳐 다음 세대 집단을 구성한다. 교배는 임의의 두 string간에 교차로 이루어진다. 한 예로 두 string A<sub>1</sub>과 A<sub>4</sub>를 교배시키면 그림 8과 같이 A<sub>1</sub>'와 A<sub>4</sub>'로 되어 다음 집단을 구성하는 string이 된다. 여기서 교배점은 임의의 난수로 선택된다.

그리고 돌연변이는 임의로 정해진 확률로 일어나며 binary string의 임의 bit 전환으로 이루어진다. 가령 돌연변이가 일어날 확률이 0.033이고 string 길이가 184비트이면 전체 비트 중 6비트(184 x 0.033)를 비트 전환(즉 0→1 그리고 1→0)하여 돌연변이를 수행한다. 돌연변이는 진화 과정에서 초기에 local extrimal에 빠지는것을 방지하는 역할을 한다.<sup>[4] [6]</sup>

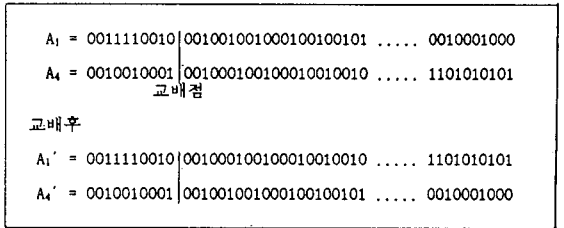


그림 19. 두 string A1, A4의 교배

이와 같은 진화를 통해 구성된 새로운 집단은 이전 집단보다 적합도가 높은 집단이 되며, 상기 과정을 반복할 경우 집단내의 어느 한 string이 최대의 적합도를 갖게 되는 상태에 이르게된다. 이때 최적의 규칙 베이스는 적합도가 가장 높은 string을 이용하여 구성하게되며 이렇게 구성된 규칙 베이스는 plant의 Error, Changer of Error 그리고 입력 Energy를 최소화 시키는 최적의 규칙을 의미하게 된다.

### 3. 로봇트 매니플레이터의 퍼지 논리 제어기에의 응용

로봇트 매니플레이터(robot manipulator)는 고도의 비선형,시변 시스템으로 이를 위한 퍼지 논리 제어 규칙을 구성하는것은 어려운 작업이다. 본 절에서는 앞서 제안한 유전 알고리즘을 이용한 퍼지 논리기를 응용하여 로봇트 매니플레이터 논리 제어기를 구성한다. 그림 20은 로봇트 매니플레이터 위치 제어를 위한 퍼지 논리 제어기의 구성을 보여주고있다. 제어기의 유전 알고리즘(혹은 GA)은 규칙 베이스를 구성하는 규칙들의 후건부 소속함수들과 (fuzzifier의 일부분인) 환산계수들을 퍼지 논리 제어기에 할당시키고 제어기가 제어를 행하는 동안 매니플레이터의 상태, 즉 error와 change of error 그리고 추론되는 토크를 이용하여 현재 규칙 베이스의 적합도를 계산한다.

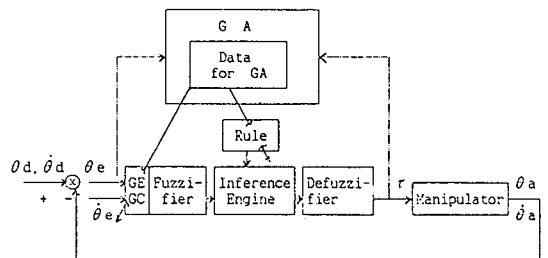


그림 20. 매니플레이터의 퍼지 논리기

퍼지화기는 singleton 기법에 의해 구성되며 crisp 한 값인 각도  $\theta_e(=\theta_d-\theta_a)$ , 속도 오차  $\theta_v(=\theta_d-\dot{\theta}_a)$ 를 퍼지 변수로 변환한다. 추론부(Inference Engine)은 Mamdani의 직접법을 채택하고 규칙 베이스는 다음의 "If - Then" 문장으로 구성한다. [7] [8]

$$\text{If } \theta_e \text{ is } A_i \text{ and } \theta_v \text{ is } B_j \text{ Then } U \text{ is } C_k \quad (3)$$

$$(i=1\cdots 7, j=1\cdots 3, k=1\cdots 21)$$

여기서  $A_i, B_j, C_k$ 는 그림 14,15의 소속함수를 갖는 퍼지 변수이다.

한편 비퍼지화기는 제어기에서 많이 쓰는 무게 중심법 (식 (4))을 사용한다.

$$U^* = \frac{\int \mu_U \times U}{\int \mu_U} \quad (4)$$

여기서  $U^*$ 는 비퍼지화된 토크이고,  $\mu_U$ 는 추론 결과의 소속함수이다.

4. 컴퓨터 시뮬레이션

1) 명령 궤적 및 입력

앞서 제안한 알고리즘의 효용성을 입증하기 위해 그림 10의 2축 평면 매니플레이터를 시뮬레이션 대상으로 삼는다.

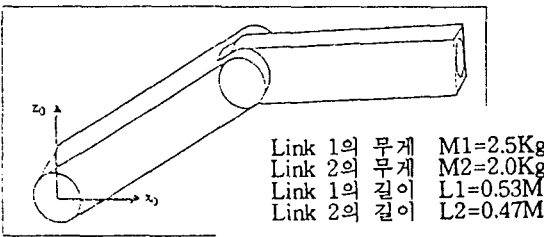


그림 21. 2축 평면 매니플레이터

그리고, 주행 경로는 다음과 같은 사각형으로 하였다.

- 사각형 한변의 길이 : 0.25m
- 사각형의 각 꼭지점 : (0.40, 0.00, 0.20),  
 (0.65, 0.00, 0.20),  
 (0.65, 0.00, 0.45),  
 (0.40, 0.00, 0.45)

또한 유전 알고리즘의 적용을 위해서 string 길이는 368비트(환산계수 32비트, 후진부 소속함수 336비트)로 하였으며 돌연변이 확률은 0.033, 그리고

string 갯수는 250개로 하였다.

그리고 사각형궤적의 주행을 통한 규칙의 적응성을 확인하기 위해 식 (5a)~(5c)와 같은 원형 명령을 입력한다.

$$\text{Position X} = 0.6 + 0.20 \cos(\theta_t) \quad (5a)$$

$$\text{Position Y} = 0.0 \quad (5b)$$

$$\text{Position Z} = 0.3 + 0.15 \sin(\theta_t) \quad (5c)$$

2) 결과

표 2는 16번 진화를 거친후에 생성된 최적의 규칙을 보여준다. 표에서의 수치는 삼각형 모양을 갖는 후진부의 소속함의 꼭지점 값을 나타낸다. 로봇트 매니플레이터 궤적 제어를 위해서는 표의 규칙들만으로도 충분히 제어가 됨을 알수 있었다. 조인트 1의 퍼지 논리 제어기에서 Error의 환산계수는 0.034 (rad)이고 Change of Error의 환산계수는 0.0353 (rad), 그리고 조인트 2의 퍼지 논리 제어기의 각 환산계수는 0.059(rad)와 0.337(rad)임을 알 수 있다.

표 2. 로봇트 매니플레이터의 최적 규칙 베이스

		NS	Error Z0	PS
Change of Error	NE		0.42	0.71
	Z0	-0.80	0.0	0.87
	PO		-0.28	0.72

Scaling factor of Error : 0.034  
 Scaling factor of Change of Error : 0.353

(a) 조인트 1의 최적 규칙 베이스

		NS	Error Z0	PS
Change of Error	NE	0.18	0.48	
	Z0	-0.87	0.0	0.26
	PO	0.06	-0.33	

Scaling factor of Error : 0.059  
 Scaling factor of Change of Error : 0.337

(b) 조인트 2의 최적 규칙 베이스

한편, 그림 22는 초기 임의로 만들어준 초기 집단중에서 적합도가 가장 높은 규칙 베이스에의한 로봇트의 주행 궤적을, 그리고 그림 23은 16번 진화 후의 사각

형 주행 궤적을 보여주고 있다. 이들 결과들로 볼때 초기 규칙 베이스에 의해서 발생한 오차들이 진화를 거듭함에 따라 오차가 거의 사라짐을 알 수 있다. 또한 16번 진화후의 규칙 베이스가 최적 상태임을 그림 24의 적합도의 변화에서 알 수 있다. 즉, 16번 진화 이후에는 적합도의 증가가 거의 없음을 알 수 있다.

또, 16번 진화후 찾은 퍼지 논리 규칙 베이스 ( 표 1의 규칙 베이스)의 적응성을 실험하기위해 식 (5)의 원형 궤적을 주행하도록 한다. 그림 25는 주행 원형 궤적을 보여주고 있다. 이 결과에서 사각형 주행으로부터 얻어진 규칙 베이스가 더 이상의 진화 없이 변화된 명령 궤도를 훌륭히 수행하는것을 알 수 있다. 그리고 그림 26에서는 제어 대상의 파라미터 변경(로봇 매니플레이터 각 링크의 질량을 30% 증가)에 대해서도 효율적인 궤적제어가 이루어짐을 알 수 있다.

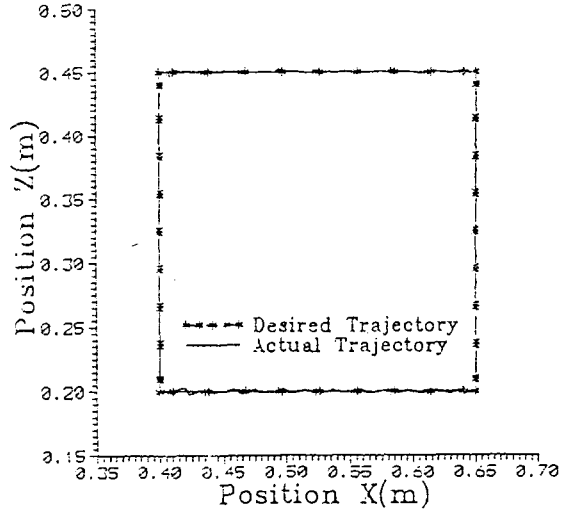


그림 23. 16번 진화후의 가장 높은 적합도를 갖는 string의 주행 결과

Ⅶ. 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.

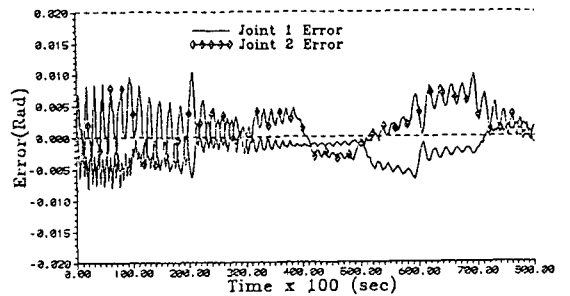


그림 24. 각 진화단계에서의 적합도

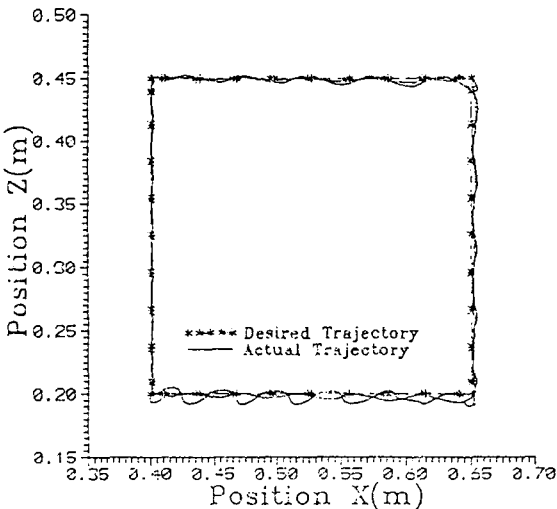


그림 22. 초기 임의의 규칙 베이스에서 가장 높은 적합도를 갖는 string에 의한 주행 결과

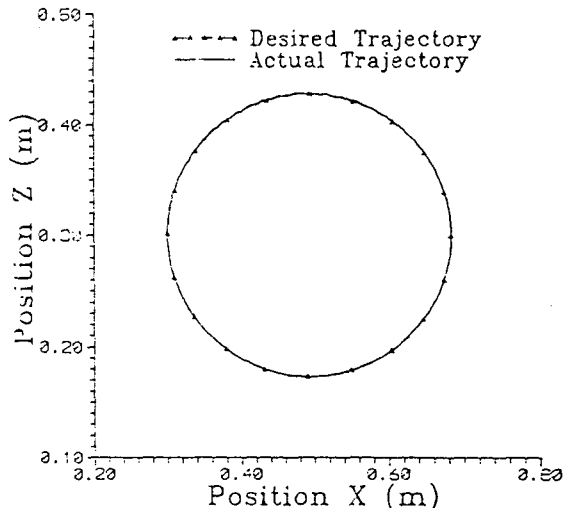


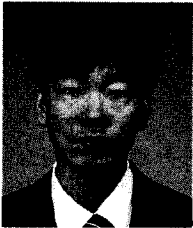
그림 25. 16번 진화후 찾은 최적 규칙에 원형 궤적의 실험

아직까지 국내에서는 유전알고리즘에 관한 연구가 미미한 실정이나 앞으로는 본격적으로 연구, 적용될 것으로 보는데 보다 획기적인 기능자의 개발로 온라인 적용까지 가능할 것으로 생각된다.

#### 參考文獻

- [ 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, NewYork, 1991.
- [ 6 ] 박 세희, 김 용호, 심 귀보, 전 흥태, "유전알고리즘을 이용한 퍼지 규칙 베이스의 자동생성", *전자공학회지*, vol.29-B, pp.60-67, 1992.
- [ 7 ] 김용호, 강훈, 전흥태, "유전알고리즘을 이용한 로봇트 매니플레이터의 최적시간 경로계획", *한국자동제어 학술회의*, pp.436-440, 1992.
- [ 8 ] Shin-ichi Horikawa et al , "A Fuzzy Controller Using A Neural Network And Its Capability To Learn Control Rules," *Proceedings of the International Conference on Fuzzy Logic & Neural Networks* , pp.103-106, 1990
- [ 9 ] Hideyuki TAKAGI, "Fusion Technology of Fuzzy Theory and Neural NetWorks - Survey and Future Directions - , " *Proceedings of International Conference on Fuzzy Logic & Neural Networks*, pp. 13 - 26, 1990
- [10] Shihuang SHAO, "Fuzzy Self-Organizing Controller And Its Application for Dynamic Processes, " *Fuzzy Sets and Systems* 26 , pp. 151-164, North-Holland , 1988
- [11] Joey K.Paker, Ahmad R. Khoogar, David E. Goldberg, "Inverse Kinematics of Redundant Robots using Genetic Algorithms, " *Proceedings 1989 IEEE International Conference on Robotics And Automation*, pp.271-276, 1989
- [12] Abraham Kandel, *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley Publishing Company, 1986
- [13] Witold Pedrycz, *Fuzzy Control and Fuzzy Systems*, Research Studies Press LTD., 1989
- [14] 박 세희, 조 현찬, 심 귀보, 전 흥태, "유전알고리즘을 이용한 퍼지 제어 규칙의 생성 , " *한국 퍼지 시스템 학회*, 1991년도 추계학술대회 학술발표 논문집, pp. 108-119, 1991
- [15] K.S. Fu, R.C. Gonzalez, C.S.G. Lee , *Robotics control, Sensing, Vision and Intelligence*, McGraw-Hill

筆者紹介



金容虎

1962年 3月 4日生

1986年 2月 광운대학교 전자공학과 졸업

1988年 2月 중앙대학교 대학원 전자공학과(석사)

1992年 2月 ~ 현재 중앙대학교 대학원 전자공학과(박사과정)

주관심분야 : Robotics, Fuzzy-Neural & Optimal Control Genetic algorithms



金洪兪

1955年 11月 27日生

1976年 2月 서울대학교 공과대학 전자공학과 졸업

1982年 2月 뉴욕 주립대 전기및 전자(석사)

1986年 2月 ~ 뉴욕 주립대 전기및 전자(박사)

1976年 11月 ~ 1979年 8月 국방과학 연구소 연구원

1976年 9月 ~ 1981年 8月 한국전자통신 연구소 연구원

1986年 9月 ~ 현재 중앙대학교 부교수

1990年 1月 ~ 1991年 12月 대한전자공학회 편집위원

1990年 1月 ~ 현재 대한전자공학회 전문위원

1991年 1月 ~ 현재 한국퍼지시스템학회 총무이사

1993年 1月 ~ 현재 대한전자공학회 제어계측연구회 전문위원장

주관심분야 : Robotics, Fuzzy logic Control, Fuzzy-neural fusion, neural network, Genetic algorithms