

최적화기법으로서의 유전알고리즘과 그 응용

Genetic Algorithms as Optimisation Tools and Their Applications



진 강 규

G. G. Jin

- 1953년 10월 12일생
- 한국해양대학교 교수
- 퍼지제어, 유전알고리즘



하 주 식

J. S. Ha

- 1933년 7월 20일생
- 한국해양대학교 교수
- 최적제어, 최적추정이론

1. 서 론

유전알고리즘은 진화원리에서 발견된 몇몇 특징들을 컴퓨터 알고리즘과 결합시켜 복잡한 최적화 문제를 해결하려는 도구로서 1975년 미국의 Holland 교수에 의해 처음으로 개발되었다". 주어진 문제에서 탐색환경이 多變數 또는 多峰(multi-modal)이 되어 대단히 복잡하거나 또는 부분적으로 알려질 경우는, 勾配(gradient)에 기초한 재래식 방법을 사용하여 최적화하는 것은 매우 어렵게 되고 경우에 따라서는 불가능할 수도 있다. 이러한 이유로 유전알고리즘과 같은 강인한 탐색법이 요구된다. 유전알고리즘의 장점은 연속성(continuity), 미분가능성(differentiability), 單峰性(unimodality) 등과 같이 탐색공간에 대한 제약으로부터 자유롭다는 것이다. 다시 말하면 목적함수 외 탐색공간에 대한 사전지식을 필요로 하지 않고, 매우 크고 복잡한 공간일지라도 전역해 쪽으로 수렴

해 갈 수 있다는 것이다. 이러한 특성 때문에 유전알고리즘은 실제 환경에서 많은 복잡한 최적화 문제를 해결하는 방법으로 인정을 받고 있으며, 함수의 최적화^{2,3)}, 신경회로망의 학습⁴⁾, 동적시스템의 식별 및 제어^{5,6)}, 신호처리⁷⁾ 등 여러 분야에 성공적으로 응용되고 있다.

이러한 중요성에 비해 유전알고리즘에 대한 연구는 국내적으로는 아직 미진한 수준이나 최근 이에 대한 관심이 고조되고 있으며, 또한 그 응용분야도 점점 넓어져 이론 개발과 실질적인 응용에 확산되리라 생각된다. 따라서 본 해설기사는 유전알고리즘의 원리와 응용 사례를 살펴봄으로서 최적화 문제를 해결하려는 독자들에게 조금이나마 도움을 주고자 한다.

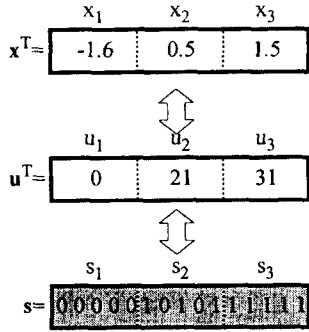


그림 2. 실수값을 가지는 벡터의 부호화

하고, 역부호화 함수는 반대동작을 한다. 이 표현 방법은 멀티 퍼래미터를 표현하는데 쉽게 응용될 수 있는데, 각각의 퍼래미터를 먼저 부호화한 후 이를 결합하여 구조체(structure) **s**로 표현할 수 있다. 그림 1은 양의 정수값 퍼래미터를 가지는 벡터 $\mathbf{x} = [11\ 5\ 3]^T$ 에서 각 퍼래미터에 4 비트가 할당될 때 부호화 결과이다.

그러나 임의의 실수값을 가지는 퍼래미터는 먼저 unsigned integer로 변환한 후 이를 다시 스트링으로 표현하는 절차가 필요하다. 다음과 같은 탐색공간 $X^M = \{\mathbf{x} \mid x_j^{(l)} \leq x_j \leq x_j^{(u)}, 1 \leq j \leq M\}$ 을 생각하자. 여기서 $\mathbf{x} = [x_1 \cdots x_M]^T$, M 은 퍼래미터 수, $x_j^{(l)}$ 과 $x_j^{(u)}$ 는 각각 j 번째 퍼래미터 x_j 의 하한치와 상한치이다. 탐색공간 내의 임의의 퍼래미터 x_j 를 길이가 l 인 스트링으로 표현한다면 x_j 는 식(1)로부터 선형적으로 $[0, 2^l - 1]$ 사이의 unsigned integer u_j 로 변환될 수 있고, 다시 u_j 는 스트링 s_j 로 부호화될 수 있다.

$$u_j = \frac{2^l - 1}{x_j^{(u)} - x_j^{(l)}} (x_j - x_j^{(l)}) \quad (1 \leq j \leq M) \quad (1)$$

반대로 복호화는 이러한 과정을 역순하면 된다. 그림 2는 퍼래미터들이 $-1.6 \leq x_1, x_2, x_3 \leq 1.5$ 에서 제한되고 5비트 스트링으로 표현될 경우의 부호화 예를 보여 준다.

3.2 초기 집단의 생성

퍼래미터의 표현 방법이 결정되면 GA가 동작되기 위해 $k=0$ 세대에서의 구조체 집단 $\mathbf{P}(k)=$

$[\mathbf{s}_1(k) \cdots \mathbf{s}_N(k)]^T$ 의 생성이 요구된다. 여기서 N 은 집단 크기이다. $\mathbf{P}(k)$ 는 무작위로 또는 경험을 기반으로 생성될 수 있다. 전자는 각 구조체의 비트값을 난수발생기에서 생성되는 2진 정수로 설정하고, 후자는 다른 탐색 알고리즘으로부터 얻어진 근사치를 이용하거나 아니면 경험에 기초한다.

3.3 기본 동작자

대부분의 GA들과 마찬가지로 SGA도 재생산, 교배, 돌연변이를 기본 동작자로 하고 있다. 그림 3은 세 동작자를 통해 현 세대 집단 $\mathbf{P}(k)$ 가 다음 세대 집단 $\mathbf{P}(k+1)$ 로 진화해가는 과정을 보여 주고 있다. 여기서 $\bar{\mathbf{P}}(k)$ 와 $\hat{\mathbf{P}}(k)$ 는 임시로 생성되는 집단으로서, 특히 전자는 交配給源 (mating pool)이라 불리워진다.

3.3.1 재생산 동작자(Reproduction operator)

재생산 동작자는 더 적합한 개체(individual)가 더 높은 생존 기회를 가지는 적자생존의 현상을 모방하려는 인위적인 메커니즘이다. 집단 $\mathbf{P}(k)$ 내의 각 개체에게는 목적함수로부터 얻어지는 적합도의 크기에 비례하는 선택확률이 부과되고, 이 확률에 따라 개체들이 선택되어 交配給源 $\bar{\mathbf{P}}(k)$ 이 생성된다. 일반적으로 적합도가 큰 개체일수록 더 많이 선택되게 된다.

재생산을 알고리즘 형태로 실현하는 방법은 여러 가지가 있으나 가장 많이 이용되는 방법은 룰렛휠 선택 (roulette wheel selection)이고 SGA도 이를 채용하고 있다. 그림 4는 재생산 동작 예를 보여 주고 있다. 각 개체는 선택확률에 비례해서

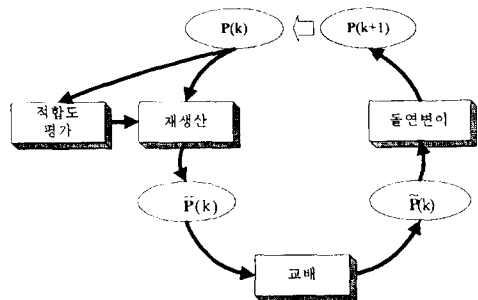
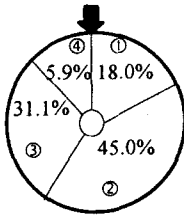


그림 3. SGA의 동작 블록선도

i	구조체 $s_i(k)$	적합도 $f_i(k)$	선택확률 $f_i(k)/\sum f_i(k)$
1	0111011	9.8	18.0
2	0111111	24.5	45.0
3	1100010	16.9	31.1
4	0100010	3.2	5.9
합 계		54.4	100.0



i	구조체 $s_i(k)$
1	0111011
2	0111111
3	1100010
4	0100010

그림 4. 룰렛휠 선택을 이용한 재생산 동작

룰렛휠의 스토면적을 할당받으며, 휠을 돌려 정지할 때 하나의 개체가 선택된다. 이러한 회전 - 정지 - 선택 동작이 4번 반복되면 동작이 완료된다.

3. 3. 2 교배 동작자(Crossover operator)

교배 동작자는 자연환경에서 생물들이 성적 교배를 통하여 자손을 생산하는 과정을 알고리즘 형태로 흉내낸 것이다. 교배 동작은 각 개체들이 보유하고 있는 단편적인 우성 인자들을 서로 교환하기 위하여 交配給源으로부터 한 쌍의 아버지 구조체를 임의로 선택하고, 교배점을 선정한 후, 이 점에서 절단된 구조체 일부를 서로 교환 결합함으로써 그들의 자손들(offsprings)을 생성한다. 교배 동작은 여러 방법으로 구현될 수 있지만, 가장 간단하면서도 많이 이용되는 것은 1점 교배(one-point crossover)이고 SGA도 이를 이용한다. 그림 5는 1점 교배의 한 예를 보여 준다.

1점 교배동작에서 아버지 쌍이 임의로 선택되면 교배확률 P_c 에 의해 교배동작의 발생유무가 테스트된다. 이를 구현하기 위해 P_c 를 입력해서 '1' 또는 '0'를 출력하는 함수 프로그램¹¹¹⁾이 이용될 수 있는데, 만약 '1'이 출력되면 하나의 교배점이 임의로 선택되고 교배점 이후를 절단해서 서로 교환한 후 두 자손이 생성되며, '0'이 출력되면 교배동작없이 선택된 아버지가 자손이 된다. 이러한 일련의 동작은 교배된 자손의 숫자가 아버지의 숫자

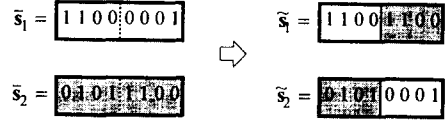


그림 5. 1점 교배 동작

와 같을 때까지 반복된다. 일반적으로 교배확률은 0.45 - 1.0 사이에서 설정된다^{2,3,12)}.

3. 3. 3 돌연변이 동작자(Mutation operator)

재생산 동작과 교배 동작은 집단이 진화해 가는 과정에서 이를 더욱 강하게 해주지만, 地域解나 死點 (dead corner)에 빠질 때는 이를 벗어나게 할 수는 없다. 이러한 원치 않는 해로부터 벗어나기 위한 한 방안으로 GA에는 돌연변이가 이용되며, 이는 자연계의 돌연변이 현상을 모방한 것이다. 돌연변이 동작자는 돌연변이 확률 P_m 에 따라 구조체의 비트 (또는 유전인자)를 임의로 변화시키므로서 초기 세대에서 특정 비트가 고정되는 것을 방지할 뿐만 아니라 또한 탐색영역을 확대해 주기도 한다.

SGA가 채용하고 있는 단순 돌연변이 동작자는 P_m 에 따라 모든 구조체 내의 각 비트에 돌연변이가 발생할 것인지를 테스트하고, 만약 발생된다면 '1'을 '0'으로 또는 '0'을 '1'로 반전한다. 그림 6은 단순 돌연변이 동작의 한 예를 보여 주며, P_m 은 0.001 - 0.01 범위 내에서 설정된다^{2,3,12)}.

3. 4 적합도 평가

한 생물이 끊임없이 투쟁하며 환경에 적응해 가는 생존 능력은 그 환경에 대한 적합도(fitness)를 반영한다. 이러한 환경에 대한 적합도는 GA에서는 개체의 적합도를 평가하는 과정을 통해서 반영되어지는데, 적합도는 목적함수로부터 계산되어진다. 자연 환경에서 적합도가 크면 클수록 더 많은 보상을 받듯이, GA에서도 적합도 함수가 최대화 문제 형태로 기술되고 음의 값을 갖지 않도록 할 필요가 있다. 따라서 최소화 또는 최대화 문제



그림 6. 단순 돌연변이 동작

중 하나로 기술되는 목적함수를 $F : X^m \rightarrow R$, 적합도 함수를 $f : S \rightarrow R$ 라 하면 적합도 함수는 다음과 같은 寫像을 통해 얻을 수 있다.

$$f(s) = F(x) - F_{\min}(\text{최대화}) \quad (2a)$$

$$= -F(x) - F_{\min}(\text{최소화}) \quad (2b)$$

여기서 F_{\min} 는 적합도가 음의 값을 갖지 않도록 설정되는 임의의 상수이다.

4. 스키머 이론

4.1 스키머란?

자연환경에서 모든 개체들은 각 세대를 통해 선택적 과정을 경험하게 되며, 그럼으로써 보다 유리한 변이를 보유한 개체는 덜 유리한 변이를 보유한 개체보다 훨씬 적응된 자손을 남기게 되는데, GA에서도 강하고 약한 유전인자들이 한 세대에서 다음 세대로 어떻게 진화해 가는 가를 설명해 줄 수 있는 매우 유익한 메커니즘이 있으며 이것이 바로 스키머(schema)이다. 스키머란 스트링(多變數에서는 구조체) 위치의 유사점을 효과적으로 표시할 수 있는 템플릿으로 2진 부호화에서는

$H = h_1 h_2 \dots h_i \dots h_l, h_i \in \{0, 1, *\}$ 로 정의된다. 여기서 l 는 스트링 길이이고 $*$ 는 이것이 있는 위치에 '1' 또는 '0' 둘 중 어느 것이 와도 상관없음을 뜻한다. 예컨대 스트링 $s_1 = 0011$ 는 스키머 $H = **11$ 에 속하지만, $s_2 = 0101$ 는 세번째 비트가 '1'이 아니어서 H 에 속하지 않는다.

4.2 스키머의 성장과 쇠퇴

k 세대에서 어떤 스키머 H 가 나타나는 頻度(frequency)를 $m(H, k)$ 라 하면 SGA 재생산, 교배, 돌연변이 동작을 거치는 동안 그 頻度は 변경되어 $k+1$ 세대에서는 식 (3)으로 표시되어 진다¹¹⁾.

$$m(H, k+1) \cong m(H, k) \frac{\bar{f}(H, k)}{f(k)} \left[1 - \frac{\delta(H, k)}{l-1} P_c - \alpha(H, k) P_m \right] \quad (3)$$

여기서 $f(k)$ 는 집단 내에 있는 전체 스트링의 평

균 적합도이고, $f(H, k)$ 는 스키머 H 에 속하는 모든 스트링의 평균 적합도이며, 거리 $\delta(H, k)$ 는 스키머 내에서 첫 특이점('0' 또는 '1')과 마지막 특이점과의 위치 차이이고, 차수 $\alpha(H, k)$ 는 스키머에 포함된 특이점의 수이다.

식 (3)은 이산형의 동적방정식으로 스키머 H 가 집단 전체의 평균보다 큰 적합도를 가지면 가질수록, 거리와 차수가 적으면 적을수록 그 頻度は 점점 증가해 가고, 반대의 경우에는 점점 소멸되어 감을 알 수 있다. 다시 말하면 강한 스키머는 약한 스키머 보다 다음 세대에서 더욱 빈번히 나타나게 된다는 사실을 설명해 주고 있다.

5. 고급 유전 동작자

대부분의 GA들은 그들의 탐색성능을 높이기 위하여 앞서 설명한 세 동작자 외에 다양한 기능의 고급 동작자를 채용하고 있는데 여기서는 주요한 몇개만을 간단히 설명한다.

5.1 스케일링 윈도우(Scaling window)

앞서 언급하였듯이 k 세대에서 구조체 $s(k)$ 의 적합도는 식 (2)로부터 계산될 수 있는데, 흔히 F_{\min} 은 사전에 알기가 어려우므로 경험과 실험을 바탕으로 충분히 적은 값을 선택하거나 아니면 현재까지의 집단에서 발견된 목적함수의 최소값으로 설정할 수 있다. 그러나 전체 해집단의 성능이 개선된 상황에서도 계속 이 값이 고정되면 개체 사이의 상대적인 적합도 차이가 감소되고 이로 인해 GA는 더 나은 개체를 찾고 보답해 주는데 어려움을 겪게 된다. 따라서 전체적인 성능이 증가하는 동안에도 일관된 選擇壓(selection pressure)을 유지하기 위하여 F_{\min} 을 변경할 필요가 있다. Grefenstette는 최근의 W_s 세대 집단 내에서 목적함수의 최소값으로 F_{\min} 을 변경하는 한 방법을 제시하였는데¹²⁾, 스케일링 윈도우 W_s 는 사용된 개체 집단의 수를 뜻한다.

5.2 엘리티즘(Elitism)

루울렛휠 선택에 기초한 재생산 동작자는 그 확률적 선택 속성 때문에 재생산 동안 최적 개체의

생존을 보장해주지 못한다. 최적 개체의 손실은 GA의 탐색능력을 저하시키는 한 요인이 되므로, 한 세대에서 다음 세대로 최적 개체의 생존을 보장하기 위하여 고안된 것이 엘리티즘이다²⁾. 엘리티즘은 k 세대 집단에서 최적 개체를 복제하고 있다가 세 동작을 거쳐 완성된 k+1 세대의 집단에서 최적 개체가 파괴되었으면 복제된 최적 개체를 이 세대에서 가장 약한 개체와 교환해준다.

5.3 세대 갭(Generation gap)

루울렛휠 선택에 기초한 재생산은 강한 개체들을 여러 번 복제할 수 있고 최적해의 인자를 보유하고 있는 약한 개체들을 초기에 제거하도록 동작할 수 있는데, 이는 초기 수렴속도를 높여주지만 반면에 유전적 다양성(genetic diversity)을 잃게 한다. 따라서 초기 세대에서 약한 개체들이 소멸되는 것을 막기 위하여 다음 세대를 구성하는 개체의 일부는 임의의 선택으로 정하고 나머지는 재생산 동작자에 의해 선택되도록 하여 현 세대의 집단과 다음 세대의 집단 사이에 비포갭(nonoverlapping)이 유지되도록 하는데 이를 세대 갭 G이라 한다²⁾ (그림 7참조).

일반적으로 G를 적게 유지하면 초기에 유전자의 다양성을 유지할 수 있는 반면에 수렴 속도를 느리게 하는 경향이 있고, 크게 유지하면 반대의 경향을 갖는다.

5.4 전위(Inversion)

轉位는 traveling salesman (TS) 문제나 VLSI의 layout task 등과 같이 순서화된 퍼라미터 집합의 순열을 최적화하는 문제를 다루는 GA에서 매우 유용하게 이용되는 동작자다. 상기 문제에 1점

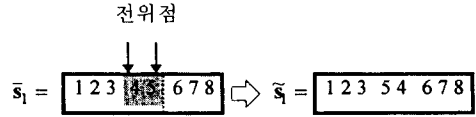


그림 8. 전위

교배가 사용되면 TS문제에서는 도착지가 두 번 나타나다가 아니면 VLSI 설계 문제에서는 한 회로의 요소가 빠진대거나 하는 불합리한 해가 생성될 수 있다. 따라서 이러한 문제점을 막기 위하여 轉位는 어버이에 있는 모든 유전인자가 자손에게 나타나도록 해준다. 원만한 轉位 동작을 보장하기 위하여 비트 구조체는 비트 대신에 퍼라미터 구조체가 되어야 한다. 그림 8은 轉位의 한 예를 보여 준다. 그림에서 퍼라미터 구조체를 자르는 두 전위점이 무작위로 선택되고 가운데 부분이 반전된다.

6. 유전알고리즘의 퍼라미터 조정

다른 최적화 알고리즘과 마찬가지로 GA도 탐색 성능에 영향을 미치는 여러 퍼라미터들이 있는데 집단 크기 N, 교배 확률 P_c , 돌연변이 확률 P_m 등이 여기에 속한다. 주어진 테스트 환경에서 최적의 성능을 얻기 위하여 이러한 퍼라미터들의 조정은 대단히 중요하다. GA의 성능 평가는 다양한 특성의 탐색공간을 가지는 테스트 함수를 필요로 한다. 대부분의 GA 사용자들은 De Jong이 제시한 함수를 벤치마크로 사용한다²⁾.

현재까지 GA 퍼라미터를 최적으로 선정하는 일의적인 방법은 없고, 경험과 실험에 의존함으로써 어느 정도 합당한 선택을 유도해 낼 수 있다. 이에 대한 광범위한 조사와 연구 결과는 참고문헌에서 찾을 수 있다^{2,3,12)}.

7. 유전알고리즘의 응용

매우 복잡한 최적화 문제를 다루는 메커니즘을 제공함으로써 GA는 그 응용 범위가 넓고 방대하며 또한 계속 증가 추세인데, 본 해설에서는 네 분야에 국한해서 응용 예를 살펴보기로 한다.

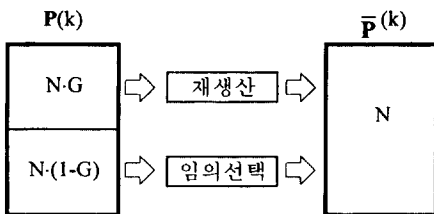


그림 7. 세대 갭

7.1 함수의 최적화

다차원 공간에서 복잡한 비선형 함수의 해를 찾는 문제는 공학분야에서 대단히 중요하다. GA는 이러한 복잡한 함수의 최적화 문제를 성공적으로 다루어 왔다²³⁾. 일반적으로 GA는 제한조건이 없는 탐색과정으로 간주되지만 이는 쉽게 제한조건이 있는 최적화 문제에 확장될 수 있다. 제한조건이 있는 문제는 제한조건에서 벗어날 때 penalty를 부과함으로써 제한조건이 없는 문제로 변환된다¹³⁾. 공학 분야에서, 예컨대 생산시간과 생산비용의 최소화 문제와 같이, 여러 목적함수를 동시에 최적화하여야 하는 경우가 종종 있다. 이와 같은 복합 목적함수의 최적화 문제는 GA를 사용함으로써 해결할 수 있다¹⁴⁾. 그리고 여러 실제 최적화 문제들은 multiple 지역해를 동시에 발견하기를 요구하는 경우가 있다. 이러한 문제를 해결하기 위하여, Goldberg와 Richardson는 탐색공간에서 멀리 떨어진 점들 간에 경쟁을 줄임으로서 GA내에 서로 다른 스트링의 안정한 부분 집단을 생성하는 sharing 함수 메커니즘을 제안하였다¹⁵⁾.

7.2 시스템 식별

전통적으로 동적시스템의 식별은 모델의 구조 선택과 선택된 모델의 파라미터를 추정하는 두 과정을 포함한다. 아주 단순한 파라미터 추정문제에서 발생하는 탐색공간도 흔히 多峰일 수가 있으며, 더우기 실제계 추정문제에 있어서 모델링 오차와 측정 잡음이 수반되므로 재래식 방법으로는 전역해를 얻기가 어렵다. 이를 극복하기 위한 한 방법으로 GA가 이용될 수 있는데 Das와 Goldberg는 이산 시스템의 파라미터를 추정하는 문제에 GA를 응용하였다¹⁶⁾. 여기서, GA는 시스템과 모델의 출력 사이의 이송오차의 합이 최소가 되도록 모델의 파라미터를 조절하는데 이용되었다. Kristinsson과 Dumont는 연속 및 이산 시스템의 식별에 GA가 이용될 수 있음을 보였다. 추정자로서의 GA는 시스템의 극과 영점을 추정하고 이 추정치들은 극 배치법으로 적응제어기를 설계하는데 이용되었다⁶⁾. 시간 지연을 가지는 연속시스템을 온라인 식별하는 한 방법은 Jin의 연구에서 찾아볼 수 있다¹⁷⁾.

이 연구에서 적응자로서의 GA는 모델의 시간 지연과 파라미터를 동시에 조절하므로써 시스템과 유사한 동적특성을 얻도록 사용되었다.

7.3 시스템 제어

최근 GA의 장점들이 기존의 제어이론과 결합되어 제어시스템의 성능을 높이려는 연구들이 이루어지고 있다. Wang과 Kwok는 GA를 이용하여 pH를 중성화하는 프로세스에서 PID 제어기의 파라미터를 최적 조정하는 문제를 다루었다¹⁸⁾. GA로 조정된 PID 제어시스템의 성능이 Ziegler-Nichols법 또는 Hooke and Jeeves 패턴 탐색법으로 조정된 PID 제어시스템보다 우수함이 입증되었다. Porter 등은 GA가 PID 제어기에 기초한 기준모델 적응제어시스템을 설계하는데 효과적임을 보였다¹⁹⁾. 여기서 GA는 기준모델과 플랜트 출력 사이의 오차가 최소가 되도록 PID 제어기의 파라미터를 반복적으로 조절한다. GA를 최적제어에 응용한 예는 Hunt의 연구에서 찾아볼 수 있는데 여기서 GA는 선형 연속시스템과 이산시스템용 LQG와 H_{∞} 최적제어기를 설계하는데 사용되었다²⁰⁾. 최근 퍼지 논리, 신경회로망 기술 등이 결합된 제어시스템들을 개발하고, 그들의 성능을 높이도록 GA를 응용하는 연구가 활발히 진행되고 있다. Karr는 역진자 시스템을 퍼지제어하는 한 방법을 제시하였고 여기서 GA는 소속함수의 중심과 폭을 최적 조정하는데 이용되었다²¹⁾. Pham과 Jin은 시간 지연을 가지는 프로세스용 적응퍼지제어기를 설계하는 한 기법을 제시하였는데⁶⁾, 프로세스 식별과 퍼지 신경회로망의 학습을 위해 두 하이브리드 GA가 이용되었다. Jones과 Porter는 기준모델 신경회로망 PID 제어시스템을 제안하였는데²²⁾, 이 시스템에서 PID제어기는 세 다층신경회로망으로 구현되고 GA는 기준모델과 플랜트 출력 사이의 오차가 최소가 되도록 회로망의 가중치들을 조절하는데 이용되었다.

7.4 신경회로망의 최적화

신경회로망 설계 시 설계자 파라미터로 여러 요소가 있는데, 은닉층의 수, 은닉 뉴런의 수와 노드 간의 연결, 그리고 역전파 학습법이 이용될 때 학

습를 그리고 모멘트를 등이 여기에 포함된다. 지금까지 이들의 선택은 주로 경험과 실험에 의존해왔고, 지역해로의 수렴은 역전과 학습법이 당연하고 있는 문제점이다.

신경회로망 설계 시 탐색공간은 매우 복잡하게 되기 때문에 체계적으로 설계하기 위해서는 GA를 사용하는 것은 바람직하다. Miller 등은 소규모 회로망의 구조를 최적화하는데 GA가 역전과 학습법과 더불어 이용될 수 있음을 보였고²³⁾ 이 문제는 Whitley and Bogart에 의해 더욱 확대되었다²⁴⁾. 신경회로망의 구조가 미리 결정될 때, 회로망의 학습법으로서의 GA의 잠재성은 가중치를 최적화하는 문제에서 밝혀졌다^{4,25)}.

8. 결 론

이상과 같이 GA의 원리와 그 응용을 SGA 중심으로 살펴보았다. GA는 복잡한 최적화 문제를 해결하는데 강인성과 안정성을 보이며, 특히 목적함수가 많은 피크치를 가질 때 재래식 방법들이 어려움을 겪게 되는데 반해, 전역해를 찾는 능력 때문에 여러 분야에서 광범위하게 응용되고 있다. 국내에서도 이 분야에 대한 체계적인 연구와 응용이 절실히 필요하다고 생각된다.

참고문헌

- 1) J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Michigan, 1975
- 2) K. A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, PhD Dissertation, The University of Michigan, Ann Arbor, Michigan, 1975
- 3) J. D. Schaffer et al., *A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization*, Proc. 3th Int. Conf. on Genetic Algorithms and Their Applications, Arlington, VA, pp. 51 - 60, 1989
- 4) R. L. Aaron et al., *Genetic Algorithm like Learning Rule for Neural Networks*, Proc. '93 IEEE Conf. on SMC, Le Touquet, France, Vol.1, pp.

- 137 - 142, 1993
- 5) K. Kristinsson and G. A. Dumont, *System Identification and Control Using Genetic Algorithms*, IEEE Trans. Syst., Man and Cybern., Vol. 22, No. 5, pp. 1033 - 1046, 1992
- 6) D. T. Pham and G. Jin, *Evolutionary Design of an Adaptive Fuzzy Logic Controller for Processes with Time Delays*, Proc. '94 IEEE Int. Conf. on SMC, San Antonio, TEX, Vol. 1, pp. 431 - 436, 1994
- 7) D. M. Etter et al., *Recursive Adaptive Filter Design Using an Adaptive Genetic Algorithm*, Proc. '82 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, France, pp. 635 - 638, 1982
- 8) K. Krishnakumar, *Micro - genetic Algorithms for Stationary and Non - stationary Function Optimization*, SPIE, *Intelligent Control and Adaptive Systems*, Vol. 1196, pp. 289 - 296, 1989
- 9) D. Whitley, *The Genitor Algorithm and Selection Pressure : Why Rank - based Allocation of Reproductive Trials is Best*, Proc. 3th Int. Conf. on Genetic Algorithms and Their Applications, Arlington, VA, pp. 116 - 121, 1989
- 10) D. T. Pham and G. Jin, *A Hybrid Genetic Algorithm*, Proc. 3rd World Cong. on Expert Systems, Seoul, Korea, Vol.2, pp. 748 - 757, 1996
- 11) D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison - Wesley Publishing Co. Inc., N.Y., 1989
- 12) J. J. Grefenstette, *Optimization of Control Parameters for Genetic Algorithms*, IEEE Trans. Syst., Man, Cybern., Vol. SMC - 16, No. 1, pp. 122 - 128, 1986
- 13) L. Davis and M. Steenstrup, *Genetic Algorithms and Simulated Annealing : An Overview*, In *Genetic Algorithms and Simulated Annealing*, L. Davis (Ed.), Pitman, 1987
- 14) J. D. Schaffer, *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*, Proc. 1st Int. Conf. on Genetic Algorithms and Their Applications, Pittsburgh, PA, pp. 93 - 100, 1985
- 15) D. E. Goldberg and J. Richardson, *Genetic Algorithms with Sharing for Multimodal Function Optimization*, Proc. 2nd Int. Conf. on Genetic

- Algorithms and Their Applications, Cambridge, MASS, pp. 41 - 49, 1987
- 16) R. Das and D. E. Goldberg, Discrete - time Parameter Estimation with Genetic Algorithms, Proc. 19th Annual Pittsburgh Conf. on Modeling and Simulation, pp. 2391 - 2395, 1988
 - 17) G. Jin, Intelligent Fuzzy Logic Control of Processes with Time Delays, PhD Thesis, University of Wales Cardiff, UK, 1996
 - 18) P. Wang and D. P. Kwok, Optimal Design of PID Process Controllers Based on Genetic Algorithms, Proc. 12th Triennial World Congress of IFAC, Sydney, Australia, Vol. 4, pp. 193 - 197, 1993
 - 19) B. A. Porter et al., Genetic Model - Reference Adaptive Control Systems Incorporating PID Controllers, Proc. IFAC Workshop on Safety, Reliability and Applications of Emerging Intelligent Control Technologies, Hong Kong, pp. 9 - 14, 1995
 - 20) K. J. Hunt, Polynomial LQG and H Controller Synthesis : A Genetic Algorithm Solution, Proc. 31st IEEE Conf. on Decision and Control, Tucson, AZ, pp. 3604 - 3609, 1992
 - 21) C. L. Karr, Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm, Proc. 4th Int. Conf. on Genetic Algorithms and Their Applications, San Diego, CAL, pp. 450 - 457, 1991
 - 22) A. H. Jones and B. Porter, Genetic Tuning of Model - Reference Neural PID Controllers, Proc. IFAC Workshop on Safety, Reliability and Applications of Emerging Intelligent Control Technologies, Hong Kong, pp. 21 - 26, 1994
 - 23) G. F. Miller et al., Designing Neural Networks Using Genetic Algorithms, Proc. 3rd Int. Conf. on Genetic Algorithms and Their Applications, Arlington, VA, pp. 379 - 384, 1989
 - 24) D. Whitley and C. Bogart, The Evolution of Connectivity : Pruning Neural Networks Using Genetic Algorithms, Proc. Int. Conf. on Neural Networks, pp. 134 - 137, 1990
 - 25) T. R. Smith et al., Calibration of Neural Networks Using Genetic Algorithms, with Application to Optimal Path Planning, Proc. '87 First Annual Workshop on Space Operations Automation and Robotics, Houston, TEX, pp. 519 - 526, 1987