Genetic Algorithms 과 Evolutionary Strategy 의 상호 비교

유 원 연*
양 영 순*

1. 서 론

Genetic Algorithms(GA) 와 Evolutionary Strategy(ES)는 대서양을 사이에 두고 GA는 미국의 Michigan 대학의 Holland 교수(1975)를 중심으로 Goldberg(1989)에 의해, ES는 독일의 Rechenberg 가 1973년 시작하여 Schwefel (1981) 교수에 의해 널리 알려지기 시작한 최적화 방법으로서 최근에 와서야 각광을 받기 시작한 방법으로서, 아직 그들의 특성과 장 단점이 널리 인식 되지 못해 부분적으로 활용되고 있는 방법이기도 하다. 사실 이들 방법은 공히 자연계의 진화 적응성을 최적화 방법에 적용하려는 노력에서 고안된 것으로서 기존의 최적화 방법과는 달리 확률론적 최적화 방법을 사용하고 있는 특색을 가지면서, GA는 미국을 중심으로, ES는 독일을 중심으로 각기 개발되어 온 방법이나 1990년 PPSN Conference(제1차)를 통해 그간의 발전 정도를 함께 공유하고자 하는 노력에서 최근에는 제2차의 PPSN Conference가 이미 열려 그 발전 속도를 점차 가속화 시켜 가는 실정이다.

GA는 일본 학자들에 의해서 이미 유전적 알고리즘이라는 용어로 사용되기 시작하여, 특히 1994년 일본 기계학회 산하의 설계 최적화 분과 중심으로 이 GA에 의한 새로운 설계 최적화 방법에 대한 연구가 매우 활성화 되어 가는 것을 감안해 보면, 머지 않은 장래에 이러한 최적화 방법론에 대한 활용은 매우 자명한 것이 아닌가 하는 느낌이 든다. 국내에서 이용되고 있는 실적은 아직 미비하나 GA는 전산구조공학회지(1992)를 시초로 하여 국내 학회지에 한 두편 언급되는 실정이고, ES는 전기공학회지(1993)를 시초로 하여 주로 전기공학의 전자장 설계분야에서 활용되고 있는 실정이다. 최근에는 한국 과학기술원의 전기 전자공학과에서 "진화 프로그래밍을 이용한 최적화 기법"을 개발하여 50개 도시 순회 세일즈 문제에 적용한 바 있다.

* 서울대학교 조선해양공학과

2. Genetic Algorithms(GA, 유전적 알고리즘)

2.1 Genetic Algorithm의 특징

Genetic Algorithms(GA)은 다아윈이 주장한 자연진화의 법칙인 적자생존(survival of fittest)과 자연도태(natural selection)의 원리를 토대로 하여 정립된 최적화 알고리즘이다. 자연진화의 법칙에 의하면 일정한 환경에 무리를 지어 살고 있는 생명체들은 그 환경에 적합한 형질을 가진 개체가 생존할 확률을 크게 가지며, 교배(crossver)와 돌연변이(mutation)의 과정을 통해 보다 좋은 방향으로 진화해 나가고, 부적합한 형질의 개체들은 진화의 과정에서 점차 도태되어 간다. 이와 같은 진화의 과정을 되풀이 하면 결국 주어진 환경

에 가장 적합한 형질의 개체들로 형성될 것이다. GA는 이러한 생명체의 자연진화법칙을 일반 최적화 문제에 적용한 방법으로, 설계영역에 다수의 설계점을 분포시켜 목적함수 값과 제한조건의 위반 정도에 따라 각 설계점에 적합성(fitness)을 부여한다. 적합성이 클수록 다음 단계인 교배와 돌연변이의 과정에 참여할 확률을 크게하여(reproduction) 적합성이 좋은 설계점에 비슷한 형질을 가진 설계점들이 다음 단계에 보다 많이 형성되어 계산이 진행 될수록 전체 설계점들은 좋은 방향으로 탐색이 진행된다.

GA는 기존의 최적화 방법과는 다른 다음과 같은 특징이 있다.

(1) 설계변수로 코드를 사용

GA는 설계변수를 실제값을 직접 사용하는 대신(2진수)코드 형태의 문자열(string)을 사용한다.(예 ; 100101) 이러한 문자열 형태의 설계변수의 사용은 염색체의 유전인자가 나열해 있는 것과 같은 형태를 지니고 있어 교배와 돌연변이의 변환과정을 수행하기가 매우 단순하고 편리하다. 또한 이러한 문자열은 이산적인 성질을 지니고 있어 정수 또는 이산적 설계변수를 포함하는 혼합형 최적화 문제에 효과적으로 사용할 수 있다.

(2) 다수의 설계점들이 집단 탐색(multipoints search)

대부분의 최적화 방법이 한점에서 한점으로의 이동에 의한 국부적인 탐색과정을 가지는 반면에, GA는 여러개의 설계점들이 집단(population)을 이루어 동시에 탐색을 행하여 보다 넓은 설계영역에 대한 정보를 활용함으로써, 전체 또는 절대 최적점(Global Optimum)에 수렴할 확률이 기존의 방법에 비해 상대적으로 매우 크다.

(3) 직접 탐색방법

GA는 목적함수와 제한조건의 값만을 사용하고 미분값이나 그 외의 다른 정보를 필요로 하지 않는 직접 탐색방법을 이용하므로 실제 구조물의 설계와 같은 복잡하고 다양한 환경의 최적화 문제에 적합하며 또한 기본 모델의 변경으로 인한 수정이 용이하다.

2.2 Genetic Algorithm의 변환과정

이와 같이 GA는 자체의 기본 성질에서 타 방법과는 다른 특성을 가지고 있다. 이러한 특성을 적절히 활용하기 위해 다음과 같은 변환과정을 거친다.

도태(Reproduction)

각 설계점의 적합성의 크기에 따라 교배와 돌연변이의 과정에서 선택될 확률을 부여한다. 즉 적합성이 큰 설계점은 많이 선택 되도록 하고 나쁜 설계점은 적게 선택 되도록 한다. 이를 위해 여러가지 방법이 있지만 대개는 가장 단순한 방법으로 각 설계점이 선택될 확률을 $f_i/\sum f_i$로 하였다. 여기서 f_i는 i 설계점의 적합성의 크기를 뜻한다. 이러한 과정을통해 설계점들의 집단은 적합성의 크기에 따라 새로이 구성되고, 집단 전체의 적합성의 평균은 상승되는 효과가 있다.

교배(Crossover)

도태의 과정을 통해 좋은 설계점들은 강조되고, 나쁜 설계점들은 도태 되었지만 더 좋은 방향으로 진 일보 하지는 못하였다. 따라서 설계점들의 향상을 위해 생명체의 유전인자 교환과 같은 교배과정을 거친다. (0, 1) 사이의 임의의 수가 확률 P_c 보다 작을 경우, 임의의 2개의 설계점을 도태의 과정을 통해 재구성된 집단 속에서 선택한다. 이때 설계점들은 문자열 형태로 구성되어 있다. 문자열의 길이 보다 작은 임의의 수 k를 선택하여, 짝지은 2개의 문자열의 k 아래쪽 부분을 서로 교환한다. 예를 들면 2진수로 구성된 문자열의 초기 상태를 P_1, P_2 라 하고, 교배가 일어난 후의 상태를 각각 C_1, C_2 라 하고, k=4 일때 다음과 같이 변환된다.

$P_1 = 0000 \mid 000 \qquad C_1 = 0000 \mid 111$

$P_2 = 1111 \mid 111 \qquad C_2 = 1111 \mid 000$

이와 같이 교배과정은 적합성이 높은 설계점과 비슷한 형질의 설계점이 많이 생성되어 계산이 되풀이 될수록 가장 좋은 설계점 주위에 다수의 설계점이 집중하게 된다.

돌연변이(Mutation)

교배에 의해 설계점들이 너무 한쪽 방향으로만 치우치는 것을 보완하기 위해 아주 작은 확률(P_m)로 각 문자열의 bit 단위로 돌연변이을 시켜준다. 즉 2진수 문자열일 경우 0은 1로, 1은 0으로 바꾸어 준다. 만일 00101 의 문자열에 돌연변이이 일어날 위치가 2라면 01101로 된다.

이와 같이 GA의 전체 전개 과정은 매우 단순하다. 하지만 다수의 설계집단과 도태, 교배 그리고 돌연변이의 과정이 합쳐진 결과는 매우 강력하다. 따라서 GA의 가장 큰 장점은 설계모델에 제약성(변수의 연속성, 미분값의 존재, unimodality 등)이 매우 적어 다양한 종류의 문제에 대해 범용성이 매우 넓다는 것이다.

3. Evolutionary Strategy(ES)

ES는 확률개념의 최적화 기법중의 하나로.Rechenberg에 의해 처음 제안되었다. 이 방법은 자연의 진화 과정을 모사한 Genetic Algorithm과 금속공학에서 금속의 구조가 안정화 되는 과정을 모사한 Simulated Annealing을 결합한 것이다. 이 방법을 구성하는 세가지 중요한 과정은 Genetic Algorithm의 재생산(regeneration) 및 돌연변이(mutation)와 Simulated annealing의 annealing이다. 재생산은 자연현상에서 부모세대로 부터 자식세대가 형성되는 과정을 모사한 것이며, annealing은 주어진 온도(환경)에서 가장 안정된 분자구조를 형성하는 과정을 모사한 것이라 할 수 있다.

이러한 과정을 수치적으로 모사하여 최적화 알고리즘에 도입하면, 초기값을 적절한 값으로 선정된 μ개의 모델변수 집합들이 부모세대를 구성하고, 재생산의 과정에 의해 λ개의 모델변수 집합들이 형성되면 이들이 자식세대를 형성한다. μ개의 보모세대 및 λ개의 자식세대의 모델변수 집합들로부터 적자생존의 원리에 입각하여 최적화 목표를 잘 만족하는 μ개의 모델변수 집합들을 선택하여 새로운 부모세대를 구성한다. 이러한 방법을 ($\mu+\lambda$)Evolution Strategy라 한다. 보통은 부모세대와 자식세대의 모델변수 집합이 각각 1개씩인 (1+1)Evolution Strategy를 사용하므로 각 세대에서 계산량이 적고 간단하다.

적당히 선정된 부모세대의 모델변수 벡터 x_p로부터 자식세대의 모델변수 벡터 x_c가 생성되는 과정은 다음과 같다.

$$x_{ci}=x_{pi}+\alpha_i \cdot R_i$$

여기서 R_i는 평균값이 0이고 (-1,1)에서 균등분포를 갖는 확률밀도 함수에 의해 발생되는 난수이며, α_i는 x_{pi}를 중심으로 한 변화 가능크기(step size)를 나타낸다.

부모세대의 x_p와 자식세대의 x_c에 해당하는 목적함수 값을 각각 F_p와 F_c라 하면, 최소 최적화 문제에선 다음 부모세대 x_p는 다음과 같이 정해진다.

$$x_p=\begin{cases} x_c & \text{if } F_c<F_p \\ x_p & \text{if } F_c \geq F_p \end{cases}$$

한편 annealing의 기능은 최적화 과정에서 변화 가능크기 α를 적절히 조절 함으로써 모사되는데, 대개는 현 세대로 부터 이전 10N세대 동안의 재생산 과정에서 돌연변이가 일어난 횟수($F_c<F_p$ 가 발생한 횟수)를 계산하여 다음과 같이 조절하였다.

$$\alpha=\begin{cases} \alpha/0.85 & \text{if 돌연변이 횟수}>10N/5 \\ \alpha \cdot 0.85 & \text{otherwise} \end{cases}$$

여기서 N은 설계변수의 갯수이다.

4. 알고리즘의 비교

근본적으로 두 알고리즘이 생물의 자연 진화론을 모사 하였다는 데서는 동일하므로 개념적인 차원에서의 본질적인 차이는 없다고 보나, 진화과정에 필요한 수준, 즉 돌연변이, 교배 및 도태 등과 같은 조작을 하는 순서에 따라서 방법이 약간 다른 결과를 보여 주는 면도 있다. 그 중, 가장 큰 차이를 보여 주는 부분은 무엇보다도 GA는 유전인자를 이용하는 관계로 binary 코드(Genotype)를 이용하는 데 반해, ES는 실수(Phenotype)를 그

대로 이용한다는 것이다.

그리고 GA에서는 상당수의 설계집단(약 50 내지는 100 개)을 각 세대마다 생성하여 세대교차를 행해 가는데 반해, ES는 (1+1) 또는 ($\mu+\lambda$) 정도의 설계집단을 이용하므로 GA에 비해 계산량이 대단히 적다. 또한 GA에서는 교배(recombination)가 중요한 수행 인자이고, 추가 변동사항을 돌연변이가 수행토록 하나, EA에서는 돌연변이를 중요한 수행 인자로 보아 먼저 돌연변이를 시키고, 그후 교배 조작으로 추가의 변화를 시도하는 것이 다른다고 볼 수 있다.

5. 수치 결과의 비교

두 방법상의 차이를 비교하기 위해 수학적인 함수인 Test 함수와 Goldstein & Price 함수에 대하여 최적화를 수행하였다.

5.1 TEST 함수

$$\min F(X)=x_1+2x_2$$

subject to

$$g_{1(x)}=1-\frac{x_1^2+(x_2-5)^2}{25} \le 0$$

$$g_{2(x)}=1-\frac{(x_1-5)^2+x_2^2}{25} \le 0$$

$$g_{3(x)}=1-x_1 \le 0$$

$$g_{4(x)}=1-x_2 \le 0$$

이 Test 함수의 국부 최소점의 위치 및 목적함수 값은 다음과 같다.

(1.00,9.90)　F=20.80
(5.00,5.00)　F=15.00
(9.90,1.00)　F=11.90　(전체 최적점)

최적화를 수행한 결과는 Table 1과 Fig. 1과 같다.

Table 1. Optimization Results of Test function

	ES	GA
x[1]	9.898985	9.895901
x[2]	1.000002	1.01566
Object Value	11.898988	11.927214
Time(sec)	0.390000	17.030001

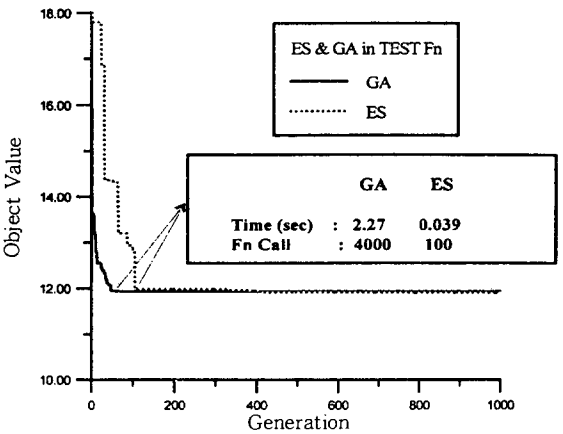

Fig. 1 Trajectory of the optimum result of Test fuction

상기의 그림에서 GA의 경우 100 개의 population을 이용하여 40번의 generation을 수행한 결과이므로 function call은 4000번(400×10)이 되고, ES 경우는 (1+1)-ES를 이용하였으므로 1개의 population에 100번의 generation 결과 최적치에 수렴하므로 function call은 100번이 된다. Table 1에 있는 시간은 1000번의 generation 동안 걸리는 시간이고, Fig. 1에 있는 시간은 그림상에서 처음 수렴되었다고 본 GA 방법(40번)과 ES 방법(100번)의 소요시간을 의미한다.

5.2 Goldstein & Price(GP) 함수

$$F = [1+(x_1+x_2+1)^2 (19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)] \times [30+(2x_1-3x_2)^2 (18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$$

$$-2 \le x_i \le 2,\ i=1,\ 2$$

여러 개의 국부 최소점이 존재하며 그 중에서 대

표적인 점의 위치와 목적함수값은 다음과 같다.

(1.2, 0.8)　F=840.0
(-0.6, -0.4)　F=30.0
(1.8, 0.2)　F=84.0
(0.0, -1.0)　F=3.0(global minimum)

이 함수에 대한 최적화 수행 결과를 보면 Table 2와 Fig. 2와 같다.

Table 2. Optimization results of Gp function

	ES	GA
x[1]	−0.000009	−0.00001
x[2]	−1.000001	−1.00014
Object Value	3.000000	3.000008
Time(sec)	0.550000	7.030000

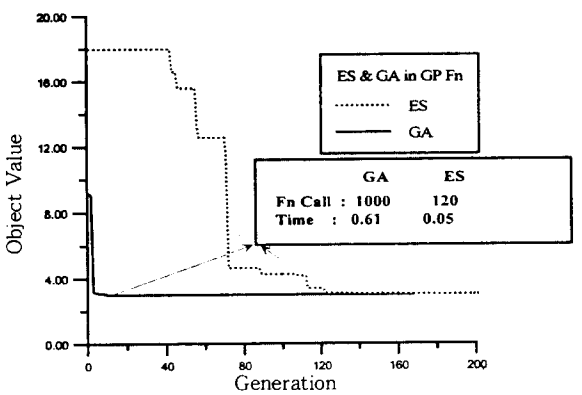

Fig. 2 Trajectory of the Optimum result of GP function

본 예제의 Function call과 시간의 의미는 예제 1의 경우와 동일하므로 Fig. 1을 참조하기 바라며, 상기의 두 예로 부터 얻어진 사실을 정리하면 다음과 같다.

(1) GA와 ES 모두 다 전체 최적점에 도달하는 것을 확인 할 수 있다.

(2) 전체 최적점에 도달하기 위한 시간은 ES가 확실히 우수함을 알 수 있다.

(3) ES는 수치최적화 기법을 이용하기 때문에 local seach 면에서 GA보다 우수하다.

그러나 GA는 離散的 변수를 다룰 수 있는 기능을 갖고 있으나, ES는 현재 그렇지 않다고 본다.

5. 결　론

최적화 방법이 구조설계 분야에 사용된지 근 30년이 지난 오늘 이론적 측면에서 보면 상당한 발전이 있어 왔다고 해도 과언이 아니나, 실무 입장에서 볼때 과연 얼마만큼 최적화 방법이 현실의 설계업무 속에 자리 잡혀 있는가를 곰곰히 되새겨 볼 필요가 있다고 본다. 사실 실제와 이론 사이의 괴리를 줄여 보려는 노력에서, 최적화 기술 분야에서도 기존의 확정론적 최적화 방법만이 아니라 확률론적 최적화 방법에 대한 연구도 시작 되었으리라 본다.

본문에서 언급한 Genetic Algorithm과 Evolutionary Strategy도 기존의 최적화 방법과 마찬가지 이유에서 복잡한 현실문제의 최적해를 추구하기 위한 또 하나의 방법으로 인식 될 필요가 있다고 보아 이 두 방법에 대한 개략적인 내용을 적어 보았으나, 충분치 못하을 많이 느껴 말미에 관련된 문헌을 수록하오니 참고하시기를 바란다.

參 考 文 獻

1. Schwefel H-P, and Manner R., "Parallel Problem Solving from Nature, 1", 1990, Dortmunt, FRG, Springer-Verlag, Lecture Notes in Computer Science 496.
2. Manner R., and Manderick B., "Parallel Problem Solving from Nature, 2", 1992, North-Holland.
3. Holland, J., H. "Adaptation in Natural and Artificial System", Univ. Michigal, Ann Arbor, MI, 1975.
4. Goldberg, D.E., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, 1989.
5. 양영순, 김기화, "전체 최적화를 위한 확률론적 탐색기법", 전산구조공학회지, 5권 2호, 1992.
6. Rechenberg, I., "Evolutionstrategie Optimierung technischer System nach Prinzipien der biologischen Evolution", Frommann-Holzboog, 1973.
7. Schwefel, H.,P., "Numerical Optimization of Computer Models.", Wiley, 1981.
8. 이홍배, 고영변 외 4명, "자계교란을 이용한 자성체 탐지를 위한 새로운 알고리즘", 전기공학회논문지, 42권 8호, 1993.