

유전자알고리즘에서 단성생식과 양성생식을 혼용한 번식을 통한 개체진화 속도향상

정 성 훈*

Improvement of evolution speed of individuals through hybrid reproduction of monogenesis and gamogenesis in genetic algorithms

Sung Hoon Jung*

요 약

본 논문에서는 유전자알고리즘에서 단성생식과 양성생식을 혼용하여 개체진화 속도를 향상시키는 방법에 대하여 제안한다. 단성생식은 암수의 구분이 없는 세균이나 단세포 생물이 두 개의 개체로 분열되는 방법으로 유전적으로 지역적 탐색에 유리하며 양성생식은 암수의 구분이 있는 개체가 만나 생식하는 방법으로 유전적 다양성을 확보하는데 유리하다. 이러한 특성은 유전자알고리즘에서 개체의 진화속도를 향상시키는데 적절히 이용될 수 있다. 본 논문에서는 선택된 개체가 상대적으로 좋은 개체의 경우 진화를 위하여 지역적 탐색을 강화하는 단성생식을 하게하고 상대적으로 좋지 않은 개체의 경우 유전자의 다양성을 확보하여 전역적 탐색을 강화하는 양성생식을 하게하였다. 단성생식의 경우 지역적 탐색을 강화하기 위하여 돌연변이 확률을 기존의 유전자알고리즘 보다 낮추었으며 양성생식의 경우 유전자의 다양성 확보를 위하여 돌연변이 확률을 기존의 유전자알고리즘 보다 크게 높였다. 4가지 함수최적화 문제에 적용해본 결과 3개의 함수에서 성능이 매우 좋았으나 전역 최적해가 분산되어 있는 4번째 함수에서는 성능이 좋지 못하였다. 이는 전역최적해가 분산되어 있는 경우 안정적 진화에 혼란을 주기 때문인 것으로 판단된다.

▶ Keyword : 유전자알고리즘, 단성생식, 양성생식, 진화

Abstract

This paper proposes a method to accelerate the evolution speed of individuals through hybrid reproduction of monogenesis and gamogenesis. Monogenesis as a reproduction method that bacteria or monad without sexual distinction divide into two individuals has an advantage for local search and gamogenesis as a reproduction method that individuals with sexual distinction mate and breed

• 제1저자, 교신저자 : 정성훈

• 투고일 : 2010. 11. 22, 심사일 : 2010. 12. 28, 게재확정일 : 2011. 01. 17.

*한성대학교 정보통신공학과 (Dept. of Information and Communications Engineering, Hansung University)

※ 본 논문의 초기결과는 2010년도 전자공학회 추계학술대회에서 발표하였음. 본 연구는 한성대학교 교내연구비 지원과제 임.

the offsprings has an advantages for keeping the diversity of individuals. These properties can be properly used for improvement of evolution speed of individuals in genetic algorithms. In this paper, we made relatively good individuals among selected parents to do monogenesis for local search and forced relatively bad individuals among selected parents to do gamogenesis for global search by increasing the diversity of chromosomes. The mutation probability for monogenesis was set to a lower value than that of original genetic algorithm for local search and the mutation probability for gamogenesis was set to a higher value than that of original genetic algorithm for global search. Experimental results with four function optimization problems showed that the performances of three functions were very good, but the performances of fourth function with distributed global optima were not good. This was because distributed global optima prevented individuals from steady evolution.

▶ Keyword : genetic algorithms, monogenesis, gamogenesis, evolution

1. 서론

유전자알고리즘은 최적화 알고리즘의 하나로 많은 공학적 문제에 성공적으로 응용되어 왔다 [1-4]. 특히 수학적 최적화가 불가능한 미분 불가능한 함수나 여러 인자들의 조합으로 최적화되는 조합적 최적화에 유용하게 사용되었다. 그러나 유전자알고리즘이 기존의 최적화 방법에 비하여 많은 장점이 있음에도 몇 가지 단점으로 인하여 응용에 어려움이 있다. 첫 번째로는 유전자알고리즘은 전역 최적해를 찾는다는 것을 보장하지 못한다는 것이다. 그러므로 만약 응용문제가 반드시 전역 최적해를 찾아야하는 경우라면 유전자알고리즘 보다는 전역 최적해를 보장하는 다른 알고리즘을 사용해야한다. 유전자알고리즘은 차적의 해를 빠른 시간 내에 찾는 문제에 유용하다. 물론, 상황에 따라서 최적을 찾을 가능성도 있다.

두 번째 문제는 유전자알고리즘의 고유문제로서 조속수렴 현상에 의한 느린 진화문제 이다 [1,2]. 이 문제는 초기 개체가 지역최적해로 빠르게 빠져들어 더 이상 진화하지 못하고 해당 지역 최적해에 머물러있는 현상으로서 일단 이 현상에 빠지면 개체가 해당 지역 최적해를 빠져 나오는데 오랜 시간이 걸리기 때문에 유전자알고리즘의 성능에 심각한 지장을 초래한다 [2-5]. 지역 최적해에 빠지면 대부분의 개체가 다양성을 잃고 유사하게 되기 때문에 교배(crossover)연산을 가해도 자식이 해당 지역 최적해 근처에 생성되어 빠져나오지 못한다. 또한 새로운 유전형질을 찾기 위한 돌연변이(mutation)연산은 돌연변이 확률이 낮기 때문에 부모와 다른 곳에 자식을 생성할 확률이 매우 낮다 [2,4].

이러한 문제를 해결하고자 여러 가지 방법이 도입되었는데, 그 중에 대표적인 것은 조속수렴현상에 빠졌을 때 돌연변이

의 확률을 높이는 방법이다 [2-7]. 다만, 돌연변이의 확률을 높이면 좋은 유전정보를 지닌 개체의 유전정보도 같이 손상되기 때문에 이를 막으면서 전역 최적해로의 진화를 가속하는 것이 중요하다.

본 논문에서는 최적해로의 지속적인 진화에 유리한 단성생식 방법과 유전적 다양성을 확보할 수 있어서 지역최적화에 빠지는 것을 줄여줄 수 있는 양성생식을 혼용한 번식 방법을 사용하여 유전자알고리즘의 개체 진화 속도를 향상시키는 방법을 제안한다. 기존의 유전자알고리즘에서는 오로지 양성생식만을 사용하기 때문에 전역 최적해 근처에 있는 개체임에도 불구하고 다른 영역에 자식을 생성하는 경우가 있으며 전역 최적해에서 멀리 떨어져 있음에도 불구하고 유전적 다양성 형성에 크게 기여하지 못하는 문제가 발생하였다.

이를 해결하기 위하여 본 논문에서는 각 개체의 적합도에 따라서 등급을 매긴 후 선택된 부모 개체의 등급이 좋은 경우 단성생식을 하게하고 그렇지 않은 경우 양성생식을 하게 하였다. 또한 단성생식을 하는 개체는 세부적인 탐색을 가능하게 하기 위하여 돌연변이 확률을 낮추고 양성생식을 한 개체는 돌연변이를 높게 하여 다양성 확보에 도움이 되게 하였다. 제안한 방법을 4개의 함수최적화 문제를 적용하여 실험한 결과 본 논문에서 제안한 방법이 다양성을 확보하면서도 최적해로의 지속적인 진화가 가능하게 되어 개체 진화속도가 향상됨을 볼 수 있었다. 본 논문에서 제안한 방법은 기존의 방법에 쉽게 적용되어 성능을 높이는데 사용될 수 있다.

본 논문의 구성은 다음과 같다. 2절에서는 본 논문에서 제안한 혼용번식 방법과 장점을 설명한다. 3절에서는 4개의 함수최적화 문제에 적용한 실험 결과 및 분석을 기술한다. 4절의 결론으로 글을 맺는다.

II. 혼용번식 방법

본 논문에서 제안한 단성생식과 양성생식을 혼용한 번식 방법을 적용한 유전자알고리즘은 Algorithm 1 과 같다.

Algorithm 1. Proposed Genetic Algorithm

```

// t : time //
// n : population size //
// P : populations //
//  $r_i$  : rank of  $i$ th individual //
//  $r_m$  : rank for monogenesis //
//  $p_m^m$  : mutation probability for monogenesis //
//  $p_m^g$  : mutation probability for gamogenesis //
1 t ← 0
2 initialize  $P(t)$ 
3 evaluate  $P(t)$ 
4 while (not termination-condition)
5 do
6   t ← t + 1
7   // select  $P(t)$  from  $P(t-1)$ 
8   for  $i = 1$  to  $n$ 
9     select  $i$ th individual by roulette wheel
10    if ( $r_i < r_m$ ) then // monogenesis
11      if ( $i = \text{odd}$ ) then
12        set  $i$ th individual to  $i + 1$ th individual
13         $i ++$ 
14      else
15        set  $i$ th individual to  $i - 1$ th individual
16      end if
17    end if
18  end for
19  // recombine  $P(t)$ 
20  do crossover with  $i$ th and  $i + 1$ th individuals
21  // do mutation
22  if (monogenesis) then
23    mutate with  $p_m^m$ 
24  else
25    mutate with  $p_m^g$ 
26  end if
27  evaluate  $P(t)$ 
28 end

```

제안한 알고리즘은 기존의 유전자알고리즘에서 부모 개체를 선택하는 곳과 돌연변이 하는 곳에서 차이점을 보이는데 Algorithm 1에 이러한 차이점이 기술되어 있다. 제안한 알고리즘에서 각 개체는 적합도에 따라서 등급이 매겨져있다. 룰렛 휠 선택을 이용하여 다음 세대의 자식을 생성할 부모 개체들을 선택하는데 만약 이 선택된 개체의 등급이 미리 정해 놓은 등급 r_m 이내로 좋은 개체이면 단성생식을 하고 그렇지 않으면 양성생식을 한다. r_m 은 제안한 유전자알고리즘의 파라미터로서 실험에서 미리 선정하는 값이다. 예를 들어 10개의 개체가 있어 개체가 적합도에 따라서 1등부터 10등까지 매겨져 있고 미리 정해놓은 등급 r_m 이 4라면 1,2,3등까지의 개체만 단성생식을 수행한다. 단성생식은 하나의 개체가 분열을 통하여 두 개의 개체가 되는 것으로 교배는 없고 돌연변이만 존재한다. 교배가 없고 돌연변이만 있으므로 양성생식에서와 같은 자식 개체의 급격한 유전자변이는 없고 안정적인 진화가 일어난다. 다만, 이런 경우 기율기를 따라서 진화될 가능성이 높아 조속수렴에 빠질 확률이 높아진다. 좋은 등급의 개체를 단성생식 하면 진화를 안정적으로 할 수 있다.

Algorithm 1에서 10~17 줄은 단성생식을 위하여 선택된 개체를 두 개의 부모 개체로 분열하는 과정이다. 배열 구조에서 홀수 번째 개체가 단성생식이면 짝수 번째 개체로 분열하고 짝수 번째 개체이면 홀수 번째 개체로 분열하는 것을 보여준다. 예를 들어 설명하면 룰렛 휠로 i 번째의 부모 개체로 선택된 개체의 등급 r_i 가 미리 정한 등급 r_m 보다 좋아서 (등급은 낮을수록 좋음) 단성생식을 하는 것으로 결정된 경우, 만약 이 개체가 홀수 번째 부모개체이면 단성생식의 복제를 위하여 그 다음 짝수 번째로 복제하고 그 다음번 부모 개체는 선택할 필요가 없으므로 i 를 하나 증가한다. 만약 그 개체가 짝수 번째 부모개체이면 단성생식의 복제를 위하여 그 개체를 그 이전 홀수 번째 개체로 복제한다. 결국, 그 이전 홀수 번째 선택된 개체는 단성생식으로 선택된 개체로 대체된다.

recombine $P(t)$ 는 교배와 돌연변이로 이루어져있다. 교배는 양성생식일 경우에만 수행한다. 양성생식은 두 개의 개체가 교배 및 돌연변이를 통하여 자식을 생성하는 것으로서 두 부모 개체로부터 좋은 형질을 받은 자식 개체의 경우 두 부모 개체보다 월등히 뛰어난 개체가 생성된다. 이러한 특성은 양성생식의 장점이 되지만 형질의 조합이 적절하지 않은 경우 두 자식 개체 모두 부모와 멀리 떨어져 있는 열등한 개체가 될 수 있다. 이런 점에 있어서 양성생식은 단성생식보다 안정적인 진화가 어렵다.

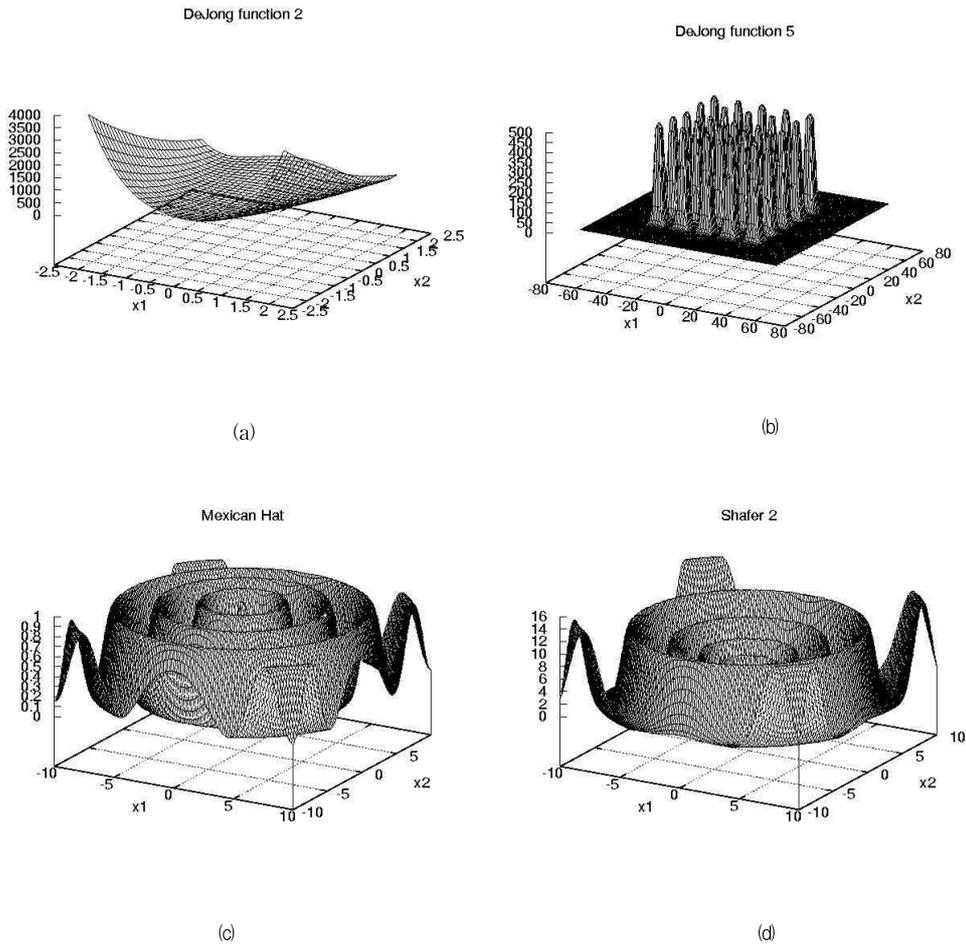


그림 1. 최적화 함수: (a) DeJong 함수 2 (b) DeJong 함수 5 (c) Mexican Hat (d) Shafer 2
 Fig. 1. Optimization functions: (a) DeJong function 2 (b) DeJong function 5 (c) Mexican Hat (d) Shafer 2

돌연변이는 단성생식과 양성생식 모두에게 발생한다. 다만 안정적 진화를 목적으로 한 단성생식은 돌연변이 확률을 낮게 주고 다양성 확보를 목적으로 한 양성생식은 돌연변이 확률을 높게 준다. 이러한 과정이 Algorithm 1 의 22~26줄에 있다. 이와 같이 단성생식과 양성생식을 혼용한 번식은 비교적 좋은 개체가 적은 돌연변이로 미세 탐색을 수행하며 비교적 나쁜 개체는 유전 형질의 교배 및 큰 돌연변이를 통하여 전역 탐색을 동시에 수행할 수 있기 때문에 지역 최적해에 빠질 가능성을 줄이면서도 안정적인 진화가 기대된다.

III. 실험 및 결과

본 논문에서 제안한 방법이 유전자알고리즘의 개체 진화에 끼치는 영향을 분석하기 위하여 여러 논문에서 많이 사용되어 온 4개의 함수 최적화 문제를 적용하여 실험하였다 [4,8,9]. 수식 1에 주어져 있는 4개의 함수 최적화 문제는 DeJong 2 함수 (f_1), DeJong 5 함수 (f_2), Mexican Hat 함수 (f_3), Schafer 2 함수(f_4)로서 그림 1은 4개 함수의 입출력관계를 보여준다. 유전자알고리즘의 개체는 이러한 네 개의 함수최적

화 상의 x_1 과 x_2 상의 한 점에 위치하며 해당 x_1 과 x_2 에서의 함수 값이 해당 개체의 적합도가 된다. 그러므로 해당 함수의 최적화란 함수의 최고값에 해당하는 x_1 과 x_2 값을 찾는 것이다. DeJong 2 함수는 비교적 단순한 함수이나 $(-2.5, -2.5)$ 영역에 전역 최적해에 대하여 $(2.5, -2.5)$ 영역에 지역 최적해를 갖기 때문에 지역 최적해에 빠지기도 쉽고 빠지면 나오기가 매우 어려운 문제이다.

$$\begin{aligned}
 f_1 &= 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \\
 f_2 &= 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a[i][j])^6} \\
 f_3 &= 0.5 - \frac{\sin(\sqrt{x_1^2 + x_2^2})\sin(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))(1.0 + 0.001(x_1^2 + x_2^2))} \\
 f_4 &= (x_1^2 + x_2^2)^{0.25} \sin(50(x_1^2 + x_2^2)^{0.1} + 1)^2
 \end{aligned}
 \tag{1}$$

표 1. 실험결과
Table 1. Experimental Results

단성생식 %		10% 내의 등급일 경우 단성생식을 함					
	비트 수	20비트			24비트		
함수	개체 수	기존방법	제안방법	기존/제안	기존방법	제안방법	기존/제안
f_1	10	6952.9	562.6	12.4	59078.5	933.0	63.3
f_2		5858.8	5256.4	1.1	37401.3	20664.5	1.8
f_3		159245.5	31613.4	5.0	776085.3	117521.8	6.6
f_4		10099.8	16603.1	$\sqrt{0.6}$	250111.6	136774.1	1.8
f_1	20	5438.7	87.4	62.2	17194.1	248.8	69.1
f_2		7321.0	821.1	8.9	18759.6	1094.0	17.1
f_3		59966.7	3157.4	19.0	521964.2	14446.2	36.1
f_4		5105.8	16153.9	$\sqrt{0.3}$	41878.4	51929.6	$\sqrt{0.8}$
단성생식 %		20% 내의 등급일 경우 단성생식을 함					
	비트 수	20비트			24비트		
함수	개체 수	기존방법	제안방법	기존/제안	기존방법	제안방법	기존/제안
f_1	10	6952.9	192.2	36.2	59078.5	235.5	250.9
f_2		5858.8	2363.9	2.5	37401.3	5697.1	6.6
f_3		159245.5	13734.0	11.6	776085.3	46003.2	16.9
f_4		10099.8	22869.2	$\sqrt{0.4}$	250111.6	78742.7	3.2
f_1	20	5438.7	531.6	10.2	17194.1	1519.8	11.3
f_2		7321.0	776.3	9.4	18759.6	1504.8	12.5
f_3		59966.7	6595.1	9.1	521964.2	6915.1	75.5
f_4		5105.8	51292.0	$\sqrt{0.1}$	41878.4	126356.3	$\sqrt{0.3}$

DeJong 5 함수는 (0,0) 에 전역 최적해를 근처에 많은 지역 최적해가 근처에 몰려있어 전역 최적해로 진화가 어려운 함수이다. Mexican Hat 함수 또한 (0,0) 에 있는 전역 최적해를 원형으로 등글게 지역 최적해가 둘러싸고 있어서 매우 어려운 문제 중의 하나로 모든 방향에서 전역 최적해로 접근하는 개체를 지역 최적해로 빠지게 할 수 있다. Shafer 2 함수는 Mexican Hat 함수와 모양은 비슷하나 전역 최적해가 가운데 있지 않고 네 모서리에 존재하는 함수로서 가운데에 많은 지역 최적해를 갖고 있다.

실험은 단순 유전자알고리즘과 비교하였다. 기존에 여러 가지 방법으로 성능을 향상시킨 유전자알고리즘이 존재하지만 본 논문에서는 단성생식과 양성생식을 혼용한 번식이 개체 속도 향상에 미치는 영향을 분석하기 위하여 단순 알고리즘과 비교하였다. 기존의 성능향상 방법은 본 논문에서 제안하는 것과 같은 번식방법을 혼용한 것이 아니기 때문에 단순 비교하기가 힘들다. 또한 제안한 방법을 기존의 방법에 같이 사용할 수 있기 때문에 기존의 성능향상 알고리즘과 비교는 하지 않았다.

표 1. 은 실험결과를 보여준다. 기존방법과 제안방법은 번식방법을 제외하고는 모두 같은 파라미터를 사용하였다. 유전자알고리즘이 전역 최적해를 찾으면 해당 세대를 기록하여 실험하였다. 즉, 유전자알고리즘의 개체 중에서 어떤 임의의 개체가 최고의 적합도를 갖는 x_1 과 x_2 의 위치를 찾아가는 경우 그 때의 세대 수를 최적화 성공 세대수로 기록하였다. 유전자알고리즘의 경우 초기 개체의 분포에 따라서 성능이 달라지는 특성을 고려하여 10개의 다른 무작위 개체로 실험한 후 평균을 구하였다. 표준편차도 구하였으나 간단히 나타내기 위하여 표에서 생략하였다. 기존방법은 $p_c = 0.6$, $p_m = 0.05$ 를 사용했으며 혼용번식에서는 단성생식의 경우 $p_m^m = 0.025$ 를 양성생식의 경우 $p_m^j = 0.5$ 를 사용하였다. 개체 수는 10개와 20개로 실험하였다. 일반적인 경우 큰 개체수를 사용하나 성능비교 시에는 작은 개체에서 성능차이를 확실히 볼 수 있으므로 본 논문에서는 작은 개체수를 사용하여 실험하였다. 염색체의 비트는 20비트와 24비트를 사용하였다. 예를 들어, 24비트를 사용했을 경우에는 x_1 , x_2 축으로 각각 12비트의 염색체로 함수공간을 분할하여 최적화를 수행한 것으로서 탐색공간은 대략 16×10^6 이 된다. 그러므로 비트 수가 클수록 탐색공간이 넓어 최적해를 찾는 데 더 어렵다. 단성생식을 할 등급은 10% 와 20%에 대하여 실험하였다. 표 1.에서 기존/제안은 기존의 방법으로 찾은 평균 세대 수를 제안한 방법으로 찾은 평균 세대 수로 나눈 것을 뜻한 것으로 1보다 크면 성

능이 향상된 것을 나타낸다. 표에서 보듯이 제안한 방법이 기존의 양성생식만을 사용한 방법에 비하여 대부분의 경우 좋은 결과를 알 수 있다. 특히, 함수 f_1 과 f_3 에서는 제안한 방법이 월등히 좋은 결과를 알 수 있었다. 단성생식을 하는 등급을 20%로 한 경우에 10개의 개체 수에서는 10%보다 성능이 좋았으나 20개의 개체 수에서는 10%보다 대부분 좋지 않았다. 이는 10개의 개체 수에서 10%는 1개의 개체로 단성생식을 하는 개체 수가 너무 작아 기인한 것으로 생각된다. 10개의 개체 수에서는 20%의 단성생식이 성능이 좋았고 20개의 개체 수에서는 10%의 단성생식이 좋은 것을 볼 때 단성생식을 수행할 개체 수는 전체 개체 수의 특정 %로 하는 것보다 2~3개 등의 특정 개체수로 하는 것이 더 좋을 것으로 판단된다.

표에서 \checkmark 표시는 제안한 방법이 기존의 방법보다 성능이 좋지 않은 경우를 표시한다. 함수 f_4 에서 성능이 안 좋은 경우가 발생하는데, 아마도 최적해가 네 곳에 흩어져 있기 때문인 것으로 생각된다. 즉, 제안한 방법은 최적해로의 안정된 진화를 가능하게 하는데 최적해가 여러 곳에 흩어진 경우 최적해로의 방향이 혼란되어 성능이 저하되는 것으로 판단된다. 결국 본 논문에서 제안한 방법은 최적해가 넓게 분포되어 있는 문제보다는 비교적 좁은 곳에 분포되어 있으며 최적해 근처에 지역 최적해가 많은 경우에 기존 방법보다 더 좋은 성능을 낼 수 있다. 본 논문의 핵심내용은 기존의 유전자알고리즘이 양성생식만을 수행하는데 비하여 단성생식과 양성생식을 혼용하여 전역 최적해로의 안정적 진화와 지역 최적해에서 쉽게 벗어나는 두 효과를 거둔 것에 있다 하겠다. 실험결과 이러한 효과가 있음을 확인할 수 있었다.

IV. 결론

본 논문에서는 단성생식방법과 양성생식 방법을 혼용하여 번식을 함으로서 단성생식 방법의 장점과 양성생식 방법의 장점을 취할 수 있는 방법을 제안하였다. 즉, 낮은 등급의 개체는 양성생식과 높은 돌연변이 확률로 넓은 영역에 걸쳐서 탐색하게 하고 좋은 등급의 개체는 단성생식과 낮은 돌연변이로 안정적인 진화를 가능하게 하였다. 이렇게 함으로서 유전자알고리즘이 조속수렴에 빠지는 확률을 줄이고 안정적으로 전역 최적해에 접근할 수 있게 되어 성능이 향상되었다. 4개의 함수최적화 문제를 이용하여 실험한 결과 본 혼용 번식방법이 매우 효과적임을 확인할 수 있었다.

참고문헌

- [1] D. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning". Addison-Wesley, 1989.
- [2] J. Andre, P. Siary, and T. Dognon, "An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization," *Advances in engineering software*, Vol. 32, No. 1, pp. 49-60, 2001.
- [3] M. Srinivas and L. M. Patnaik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithm," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, pp. 656-667, Apr. 1994.
- [4] A. Tuson, "Adapting Operator Probabilities in Genetic Algorithms," master thesis, Dept. of Artificial Intelligence, University of Edinburgh, UK, 1995.
- [5] E. Alba and B. Duroso, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 2, pp. 126-142, 2005.
- [6] A. E. Eiben, Z. Michalewicz, m. Schoenauer, and J. E. Smith, "Parameter Control in Evolutionary Algorithms", *Studies in Computational Intelligence*, Vol. 54, pp. 19-46, 2007.
- [7] Silja Meyer-Nieberg and Hans-Georg Beyer, "Self-Adaptation in Evolutionary Algorithms," *Studies in Computational Intelligence*, Vol. 54, pp. 47-75, 2007.
- [8] K. DeJong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", Ph. D. Dissertation, University of Michigan, 1975.
- [9] Marcin Molga and Czeslaw Smutnicki, "Test functions for optimization needs," <http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf>

저자 소개



정성훈

1988: 한양대학교 공학사.
 1991: 한국과학기술원 공학석사.
 1995: 한국과학기술원 공학박사.
 1996 - 현재: 한성대학교 정보통신공학과 교수
 관심분야: 지능시스템, 시스템생물학, 뇌공학
 E-mail : shjung@hansung.ac.kr