

다중 여왕벌 진화를 통한 여왕벌 유전자알고리즘의 성능향상

정 성 훈*

Performance Improvement of Queen-bee Genetic Algorithms through Multiple Queen-bee Evolution

Sung Hoon Jung*

요 약

여왕벌의 생식방식을 모방하여 만든 여왕벌 유전자알고리즘은 유전자알고리즘의 성능을 대폭 향상시켰다. 그러나 여왕벌 유전자알고리즘에서는 여왕벌을 하나만 사용하여 진화를 수행함으로써 개체들이 지나치게 해당 여왕벌이 있는 쪽으로 몰리는 문제를 발생하였으며 이는 결국 유전자 알고리즘의 성능저하를 가져왔다. 본 논문에서는 이러한 문제를 해결하고자 각 세대에서 가장 적합도가 좋은 여왕벌과 더불어 개체의 적합도가 부모 개체에 비하여 가장 크게 증가한 두 번째 여왕벌을 도입한 다중 여왕벌 진화 알고리즘을 제안한다. 다중 여왕벌을 도입함으로써 개체가 지역 최적해에 빠질 가능성이 줄어들고 지역 최적해에 빠진 경우에도 보다 쉽게 지역 최적해를 빠져나올 수 있게 되어 성능향상이 가능하였다. 4개의 함수최적화 문제에 적용시켜본 결과 본 논문에서 제안한 방법이 기존의 방법보다 대부분의 경우에서 성능이 향상됨을 볼 수 있었다.

▶ Keyword : 최적화, 여왕벌 진화, 다중 여왕벌 진화, 조속수렴현상

Abstract

The queen-bee genetic algorithm that we made by mimicking of the reproduction of queen-bee has considerably improved the performances of genetic algorithm. However, since we used only one queen-bee in the queen-bee genetic algorithm, a problem that individuals of genetic algorithm were driven to one place where the queen-bee existed occurred. This made the performances of the queen-bee genetic algorithm degrade. In order to solve this problem, we introduce a multiple

• 제1저자 : 정성훈 • 교신저자 : 정성훈

• 투고일 : 2012. 01. 26, 심사일 : 2012. 02. 16, 게재확정일 : 2012. 03. 12.

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

※ 본 연구는 한성대학교 연구장려금 지원과제 임.

queen-bee evolution method by employing another queen-bee whose fitness is the most significantly increased than its parents as well as the original queen-bee that is the best individual in a generation. This multiple queen-bee evolution makes the probability of falling into local optimum areas decrease and allows the individuals to easily get out of the local optimum areas even if the individuals fall into a local optimum area. This results in increasing the performances of the genetic algorithm. Experimental results with four function optimization problems showed that the performances of the proposed method were better than those of the existing method in the most cases.

▶ Keyword : Optimization, Queen-bee evolution, Multiple queen-bee evolution, Premature convergence phenomenon

I. 서론

많은 공학문제에서 발생하는 것이 최적화문제이다 [1-5]. 즉, 공학적 문제의 해결을 위해서는 특정한 목적을 위하여 주어진 파라미터의 최적의 값을 찾아야하는 경우가 많다. 이러한 공학적 최적화를 위하여 그동안 많은 알고리즘이 개발되어 왔으나 최근 들어서는 생물이나 자연에서 영감을 얻어서 개발된 알고리즘들이 많은 각광을 받고 있다 [1,4,5]. 그 중 대표적인 것이 유전자알고리즘 (GAs: Genetic Algorithms), 입자군집최적화 (PSO: Particle Swarm Optimization), 개미군집최적화 (ACO: Ant Colony Optimization) 등과 같은 것이 있다.

그 중에서도 현재 여러 분야에서 가장 많이 사용되고 있는 것이 유전자알고리즘이다. 유전자알고리즘은 다윈의 자연도태 및 적자생존의 원리를 응용한 것으로 문제의 해를 이진 비트열이나 실수벡터로 표현하여 진화시킴으로서 해를 찾는다 [1]. 유전자알고리즘의 가장 큰 장점으로서는 다양한 문제에 응용가능하다는 것과 수학적 최적화에서와 같은 전제조건 (예를 들면 미분가능 등의)이 필요 없다는 것이다. 다만, 최적해를 보장하지 못한다는 것과 언제 탐색을 종료해야하는지가 애매 모호하다는 등의 단점이 있다 [1].

유전자알고리즘을 실제 문제에 응용하는데 있어서 중요한 것은 문제가 많은 지역 최적해를 갖는 경우 유전자알고리즘의 개체가 이 영역에 빠져서 성능이 급격히 저하될 수 있다는 것이다. 이를 조속수렴현상(premature convergence phenomenon)이라고 하는데, 이를 적절히 해결해야 제한된 시간 내에 양질의 해를 얻을 수 있다 [1,8].

우리는 이러한 조속수렴현상을 극복하고 전역 최적해로의 빠른 진화를 위하여 여왕벌의 생식 메커니즘을 모사한 여왕벌 유전자알고리즘을 제안하였다 [6]. 여왕벌 유전자 알고리즘

에서는 각 세대의 최고 적합도를 갖는 여왕벌 개체와 적합도에 비례하여 선택된 개체가 번식을 하여 자식을 생성한다. 이렇게 함으로서 개체가 빠르게 최적해로의 진화가 가능하게 되었다. 그러나 개체들이 진화한 곳이 지역최적해일 경우에는 오히려 이 지역 최적해에서 빠져 나오지 못하는 문제가 발생하여 성능을 급격히 저하시킬 수 있다. 논문 [6]에서는 이러한 문제를 해결하기 위하여 개체의 특정 퍼센트를 강 돌연변이를 수행하여 일부 해결하였다. 그러나 강 돌연변이를 통하여 이러한 문제를 일정부분 완화시킬 수는 있으나 근본적인 해결책이 되지는 못한다. 그 이유는 강 돌연변이는 지역 최적해를 벗어나는 것에는 도움을 주지만 무작위적인 변위로서 기존에 찾은 좋은 개체들의 유전형질을 깨버릴 가능성이 높기 때문이다.

본 논문에서는 기존 여왕벌에 새로운 여왕벌을 도입하여 이러한 문제를 일정부분 완화함으로서 유전자 알고리즘의 성능을 향상시킨다. 새로 도입한 여왕벌은 이전 세대에 비하여 적합도가 가장 크게 좋아진 것을 선택한다. 이렇게 함으로서 비록 현재 적합도가 높지는 않아도 급격하게 적합도가 좋아지고 있는 개체 근처에 많은 자식을 생성할 수 있다. 만약 대부분의 개체가 지역 최적해에 근처에 있다고 한다면 이러한 개체들은 적합도가 급격히 좋아지기 어렵다. 결국 적합도가 급격히 좋아지고 있는 개체는 지역 최적해 외부에 있는 개체일 가능성이 높으며 이 개체 주변에 새로운 개체를 만든다는 것은 해당 지역 최적해를 벗어 날 수 있는 가능성을 높여준다. 기존 방법에서는 지역 최적해 영역 밖에 있는 개체가 생성된다고 하더라도 이런 개체의 적합도가 아주 높지 않은 경우 다음 세대에 선택되지 않을 가능성이 높아서 지역 최적해를 벗어나기 힘들다.

본 논문에서 제안한 방법에서는 두 개의 여왕벌이 있기 때문에 적합도에 비례하여 선택된 개체가 두 개의 여왕벌 중에서 하나를 선택하여 번식을 해야 한다. 두 개의 여왕벌 중에

서 어떤 여왕벌과 번식을 할지는 선택된 개체와 두 여왕벌 사이에 해밍거리를 측정하여 거리가 큰 것을 선택하게 하였다. 이렇게 함으로서 유전적 다양성을 더 높여줄 수 있다. 본 논문에서 제안한 방법을 4개의 함수최적화에 적용하여 실험한 결과 기존에 방법에 비하여 대부분의 경우에서 성능이 좋게 나오는 것을 관찰할 수 있었다. 특히 elitism을 추가로 적용한 것은 성능이 더욱 좋았으나, 지역 최적해가 아주 많으며 전역 최적해를 감싸고 있는 어려운 문제에서는 그 효과가 그렇게 크지 않았다.

II. 다중 여왕벌 진화 알고리즘

1. 여왕벌진화 알고리즘

유전자알고리즘은 다윈의 자연도태(natural selection)와 적자생존(survival of the fittest)을 응용하여 만든 최적화 알고리즘이다. 즉, 자연에서 개체들이 자연선택과 적자생존에 의하여 진화되듯이 어떤 문제의 해를 개체화하고 이 개체에 자연선택과 적자생존을 적용하여 진화시켜나감으로써 해답을 찾아가는 방법이다. 이러한 유전자 알고리즘은 부모개체 중에서 다음 세대의 개체를 생성할 부모를 선택하는 선택(selection) 연산과 선택된 부모개체를 교배(crossover)와 돌연변이(mutation)를 통하여 자식개체를 생성하는 생식(reproduction)으로 구성되어 있으며 생성된 개체가 얼마나 주어진 문제에 적합한 해를 갖는지를 평가하는 평가(evaluation)로 구성된다. 교배와 돌연변이는 모든 개체에 대하여 모두 수행하는 것은 아니고 보통 교배는 0.6의 확률로 수행하며 돌연변이는 0.05의 확률로 수행한다. 이렇게 함으로서 좋은 형질을 보존하면서 더 좋은 형질을 획득해 나가며 개체가 진화한다.

교배는 부모 세대가 지금까지 찾은 좋은 형질을 결합하여 더 좋은 개체를 찾고자 하는 연산으로 최적화연산에서 보수적 연산인 개발(exploitation)에 해당하며 돌연변이는 새로운 좋은 형질을 찾기 위하여 새로운 영역을 탐색하는 진보적 연산인 탐험(exploration)에 해당한다. 개발이 강조되면 지역 최적해에 빠져서 나오기 힘들고 탐험이 강조되면 방향성을 잃고 헤매는 무작위적 탐색이 되어 이 두 힘을 적절히 조화시켜야 좋은 성능을 낼 수 있다. 일반적으로 돌연변이는 기존에 찾은 정보를 이용하지 않고 무작위적으로 염색체 비트에 변화를 주는 것으로 그 확률이 적절하면 지역 최적해를 빠져나오는 데에는 도움이 되지만 너무 강하면 전역 최적해를 향해가

는 데는 방해가 된다.

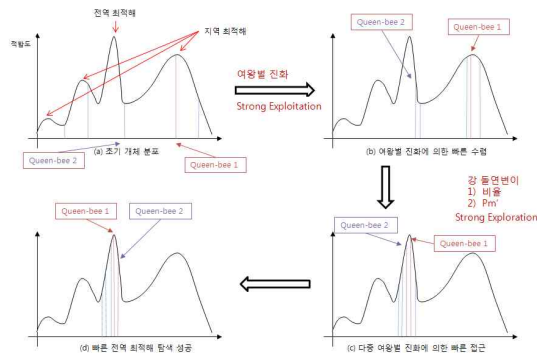


그림 1 다중 여왕벌 진화 알고리즘에서 개체진화

기존의 유전자알고리즘에서는 개체의 적합도에 비례하여 선택한 개체들을 쌍을 이루어 부모 개체로 사용한다. 이에 반하여 여왕벌 유전자알고리즘에서는 부모 두 개체 중 하나는 적합도에 비례하여 선택한 개체를 사용하나 나머지 개체는 해당 세대에서 가장 좋은 적합도를 갖는 개체(여왕벌)를 사용한다. 이렇게 함으로서 지식 개체는 여왕벌 개체 근처로 빠르게 접근한다. 이 때 만약 여왕벌 개체가 전역 최적해 영역에 있으면 대부분의 개체가 빠르게 전역 최적해 영역으로 접근하므로 빨리 전역 최적해를 찾을 수 있게 된다. 그러나 만약 여왕벌 개체가 있는 곳이 지역 최적해 영역이면 대부분의 개체가 지역 최적해로 빠져들게 되며 이 지역 최적해에서 빠져 나오는 것은 매우 어렵다. 그러므로 특히 지역최적해가 넓은 영역에 걸쳐서 많이 존재하는 문제의 경우 전역 최적해를 찾기가 어려워진다. 우리는 이러한 문제를 해결하고자 일정 퍼센트의 개체에 강한 돌연변이를 일으키게 하였다. 이렇게 하면 지역 최적해에 빠졌다하더라도 강 돌연변이에 의하여 지역 최적해를 빠져나올 수 있게 된다. 그러나 돌연변이 확률을 높이거나 돌연변이를 수행할 퍼센트를 높이는 것은 지역 최적해를 빠져나올 수 있겠지만 많은 좋은 형질을 파괴하는 문제가 있다.

2. 다중 여왕벌진화 알고리즘

우리는 본 논문에서 돌연변이 확률을 더 높이거나 돌연변이를 수행할 개체의 퍼센트를 높이는 것이 아니라 여러 개의 여왕벌을 두어 이러한 문제점을 완화하는 방법을 제안한다. 즉 기존의 여왕벌에 새로운 여왕벌을 추가하여 다중 여왕벌을 통해 진화하게 만든다. 기존의 여왕벌은 각 세대에서 가장 적합도가 좋은 개체로서 현재 상황에 좋은 개체이다. 그러나 새로운 여왕벌은 비록 현재의 적합도는 낮더라도 발전할 가능성

이 최고로 높은 여왕벌을 선택한다. 가능성이 높은 것을 판단하는 척도는 여러 가지가 가능하나 본 논문에서는 부모의 적합도에 비하여 적합도가 급격히 상승한 개체가 가능성이 높다고 보고 이를 새로운 여왕벌로 선택한다. 지역 최적해에 빠진 대부분의 개체는 자식 개체가 부모 개체 근처에 생성되기 때문에 부모 개체의 적합도와 거의 유사한 적합도를 갖게 된다. 결국 새로운 여왕벌을 도입하고 적합도에 비례하여 선택된 다른 개체와 생식을 하게한다면 개체들이 지역 최적해에서 벗어나는데 도움을 주며 성능향상에 기여할 것이다. 그림 1은 이러한 상황을 보여준다. 여왕벌이 두 개 존재하면 적합도에 비례하여 선택된 개체가 두 여왕벌 중에 어떤 개체와 생식을 할 것인지를 정해야한다. 여기에도 많은 방법이 있겠지만 본 논문에서는 개체의 다양성 유지가 중요하다는 관점에서 두 여왕벌과의 유전적 차이를 계산하고 유전적 차이가 큰 여왕벌과 생식을 하는 방법을 선택하였다.

이와 같이 각 세대에서 적합도가 가장 좋은 여왕벌 개체와 적합도가 가장 많이 커진 새로운 여왕벌 개체를 이용하여 진화를 시키면 보다 더 안정적으로 전역 최적해를 향해 진화할 수 있다. 알고리즘 1은 본 논문에서 제안한 다중 여왕벌진화 알고리즘을 보여준다.

Algorithm 1 Proposed genetic algorithm

```
// t : time //
// n : population size //
// P : populations //
//  $\xi$  (1- $\xi$ ) : normal (strong) mutation rate //
//  $p_m$  : normal mutation probability //
//  $p'_m$  : strong mutation probability //
//  $I_q$  : queen-bee individuals //
//  $I_m$  : selected individuals //
// D : Hamming distance //
// Q : queen-bee individuals //
1 t  $\leftarrow$  0
2 initialize P(t)
3 evaluate P(t)
4 while (not termination-condition)
5 do
6   t  $\leftarrow$  t + 1
7   select P(t) from P(t-1)
8   for i=1 to n // i = i+2
9     select  $I_m^i(t)$  by a selection method
```

```
10   select  $I_q^{i+1}(t)$  from Q
11   for j=1 to 2
12     set  $D_j$  from  $I_m^i(t)$  and  $I_q^{i+1}(t)$ 
13   end for
14   find max  $D_j$  and set the index k
15   set  $I_q^{i+1}(t) = Q_k(t)$ 
16   end for
17   P(t) = ( $I_m^i(t), I_q^{i+1}(t)$ )
18 recombine P(t)
19 do crossover
20 do mutation
21   for i=1 to n
22     if  $i \leq (\xi \times n)$  then
23       do mutation with  $p_m$ 
24     else
25       do mutation with  $p'_m$ 
26     end if
27   end for
28 evaluate P(t)
29 assign normal fitness to P(t)
30 assign improved fitness to P(t)
31 insert Q(t)
32 find the best individual  $I_b^1$  from P(t)
33 set first queen,  $Q_1(t) = I_b^1$ 
34 find the best improved individual  $I_b^2$  from P(t)
35 set second queen,  $Q_2(t) = I_b^2$ 
36 end do
```

기존의 여왕벌진화 유전자알고리즘에서 추가된 부분은 줄번호 8~16과 30~35에 해당한다. 먼저 기존의 여왕벌진화 유전자알고리즘을 설명하면 다음과 같다. 기존의 여왕벌진화 알고리즘에서는 부모개체를 선정할 때 줄번호 17에서 보듯이 이전 개체군에서 가장 좋은 개체인 여왕벌과 룰렛 휠 선택과 같이 선택알고리즘으로 선택한 개체가 쌍을 이루어 부모개체로 선정된다. 기존의 방법이 여왕벌을 하나만 선정해서 사용하는데 반하여 본 논문에서 제안한 알고리즘에서는 하나의 여왕벌을 추가하여 사용하여 진화한다. 새로운 여왕벌은 이전에 설명한 것처럼 부모개체로부터 가장 진화가 많이 된 개체로서 t 세대에서 기존의 여왕벌을 $Q_1(t)$ 로 나타내고 새로운 여왕벌을 $Q_2(t)$ 로 나타낸다. 줄번호 8~16의 알고리즘은 기존의 여왕

벌과 새로 추가한 여왕벌이 존재할 때 부모개체를 선정하는 과정을 보여준다. 먼저 줄번호 9에 있듯이 기존의 방법과 같이 선택방법을 이용하여 부모개체를 하나 선정한다. 그 다음 이 선택된 개체와 두 개의 여왕벌 사이에 해밍거리를 계산하고 (줄 12) 두 여왕벌 중 해밍거리가 큰 여왕벌을 이 개체와 함께 부모 개체로 한다 (줄 14~15). 해밍거리가 가장 큰 개체로 부모 쌍을 만드는 것은 이렇게 하면 자식개체의 다양성이 유지되어서 한 곳으로 몰리는 경향을 줄여주기 때문이다.

교배과정은 전통적인 단일 포인트 교배방법으로 이전 것과 동일하다. 돌연변이 과정은 기존의 여왕벌진화방법과 동일하게 ξ 퍼센트의 개체는 정상적인 돌연변이 확률 p_m 으로 돌연변이하며 나머지 개체는 강돌연변이 확률 p'_m 를 적용하여 돌연변이 시킨다. 기존의 여왕벌 진화방법은 돌연변이 후에 평가를 거쳐서 위의 과정이 반복된다. 본 논문에서 제안한 방법에서는 기존의 적합도에 추가하여 부모개체로부터 가장 진화가 많이 된 개체를 선정하기 위하여 개선 적합도(improved fitness)를 추가적으로 평가한다 (줄 30). 개선 적합도는 자식개체의 적합도가 부모개체의 적합도보다 얼마나 향상되었는지를 나타내는 적합도로서 부모개체 적합도에서 자식개체의 적합도를 뺀 것으로 계산한다. 적합도는 음수가 될 수 없으므로 음수가 되는 경우는 0으로 한다. 부모는 두 개의 개체로 이루어져있기 때문에 부모개체의 적합도를 나타내기 위해서는 하나의 적합도로 나타내야한다. 여러 가지 방법이 사용될 수 있으나 본 논문에서는 부모 두 개체의 적합도를 평균한 것을 부모의 적합도로 한다. 개선 적합도를 구한 후에는 새로운 여왕벌을 선정하여 여왕벌 군에 삽입해야한다 (줄 32~35). 최고의 기존 적합도를 갖는 개체를 선정하여 $Q_1(t)$ 로 하고 최고의 개선 적합도를 갖는 개체를 선정하여 $Q_2(t)$ 로 한다. 종료 조건을 만족할 때까지 이와 같은 과정이 반복된다. 종료조건은 일반적으로 전역 최적해를 찾는 조건으로 하나 시간이 아주 오래 걸릴 수가 있기 때문에 보통 특정 세대수가 지나면 종료되는 것으로도 한다. 본 논문에서는 종료조건을 전역 최적해를 찾는 것으로 하였으며 너무 많은 세대가 걸리는 것을 방지하기 위하여 특정 세대수가 되면 실패한 것으로 종료하게 하였다.

III. 실험 및 결과

제안한 알고리즘을 컴퓨터에서 구현하여 실험하였다. 알고리즘은 리눅스 운영체제 상에서 C언어로 구현하였으며 펜티엄4 듀얼코어 CPU상에서 실행하였다. 여러 번의 실험을 위

하여 배치파일을 이용해 수행하고 그 결과를 파일로 저장하여 실험결과를 분석하였다. 그림 2는 리눅스의 한글터미널 상에서 백스칸모자 문제에서 20비트로 10번의 실험을 하고 그 결과를 얻는 실험화면을 보여준다.

본 논문에서 제안한 다중 여왕벌 유전자알고리즘을 테스트 하기 위하여 4개의 최적화 문제에 적용하여 실험하였다.

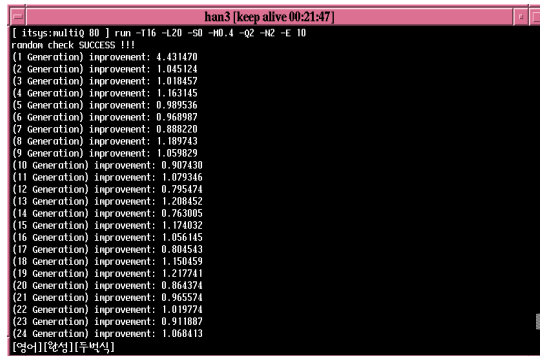
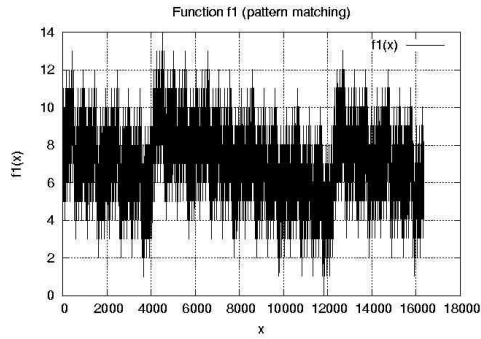


그림 2. 제안한 알고리즘의 실험화면

$$\begin{aligned}
 f_1 &= \sum_{j=1}^h m_j \begin{cases} m_j = 1 & \text{if } T_j = I_j \\ m_j = 0 & \text{if } T_j \neq I_j \end{cases} \\
 f_2 &= 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \\
 f_3 &= 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a[i][j])^6} \\
 f_4 &= 0.5 - \frac{\sin(\sqrt{x_1^2 + x_2^2})\sin(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))(1.0 + 0.001(x_1^2 + x_2^2))}
 \end{aligned} \tag{1}$$

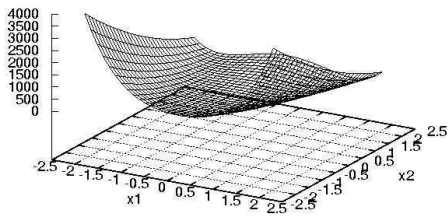
4개의 최적화문제는 다음 식 (1)과 같으며 각 문제의 적합도가 그림 3에 그려져 있다. 첫 번째 문제는 패턴매칭 문제로서 주어진 목표패턴 T 와 같은 패턴을 찾는 것이다. 그러므로 최고 적합도는 패턴의 비트수 h 가된다. 두 번째 문제는 DeJong 함수 2로서 한 개의 지역 최적해 영역과 한 개의 전역 최적해 영역으로 이루어져 있다. 전역 최적해 영역으로 진화가 일어난 경우에는 단봉함수와 유사해서 비교적 쉽게 전역 최적해를 찾을 수 있으나 지역 최적해 영역으로 진화가 일어나면 전역최적해로의 진화가 어려운 문제이다. 세 번째 문제는 DeJong 함수 5로서 $(x_1, x_2) = (0, 0)$ 인 곳에 전역최적해가 있고 그 주변에 많은 지역해가 있는 문제로서 지역 최적해에 빠지면 벗어나기 상당히 어려운 문제이다. 네 번째 문제는 맥스칸모자 문제로 세 번째 문제처럼 $(x_1, x_2) = (0, 0)$ 인 곳에 전역 최적해가 있으며 주변에 지역최적해가 많은 문제이다. 다만 세 번째 문제는 격자로 구성된 곳에 따로 지역최적해가 있는 반면에 네 번째 문제는 지역최적해가 전역 최적해를 둘

러싸고 있다.



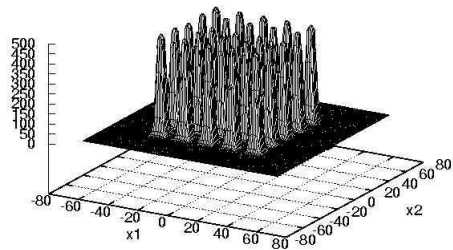
(a)

DeJong function 2



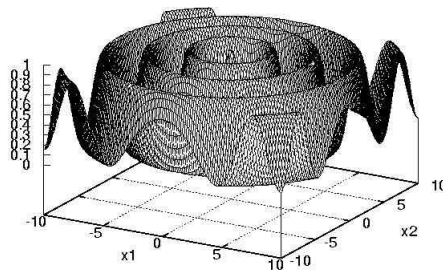
b)

DeJong function 5



(c)

Mexican Hat



(d)

그림 3. 최적화 함수 (a) 패턴매칭 (b) DeJong 함수 2 (c) DeJong 함수 5 (d) Mexican Hat

표 2 실험결과

SGA		640.9		5505.8		2021.4		43728.8	
ξ	p_m	f_1		f_2		f_3		f_4	
		QGA	PGA	QGA	PGA	QGA	PGA	QGA	PGA
0.4	0.6	21.8	11.1	209.7	97.1	123.6	53.9	1658.1	714.9
	0.8	25.5	11.5	121.3	80.5	96.6	53.6	1243.1	742.3
	1.0	21.4	14.2	27.7	26.4	99.0	43.7	25107.4	11620.0
0.6	0.6	15.6	9.7	2125.8	362.1	126.2	111.2	2316.7	1363.4
	0.8	13.4	9.4	118.7	116.7	73.9	53.5	794.0	765.4
	1.0	12.6	10.7	56.0	19.1	64.8	24.3	11292.7	3671.6
0.8	0.6	14.3	9.5	1094.4	218.8	102.5	41.7	2449.9	3827.1
	0.8	12.7	11.5	242.3	114.4	51.1	54.8	1210.1	1002.2
	1.0	10.2	10.4	103.5	39.4	52.4	53.3	12681.5	2346.1
1.0		9.2	8.0	85889.5 (4/6)	20319.2 (5/5)	36205.2	6925.1	788464.6 (3/7)	515132.1 (6/4)

표 1: 유전자알고리즘의 파라미터

파라미터	값
교배확률 (p_c)	0.6
정상 돌연변이 확률 (p_m)	0.05
개체 수	30
개체비트	20 bits
정상 돌연변이 비율 (ξ)	0.4, 0.6, 0.8, 1.0
강돌연변이 확률 (p'_m)	0.6, 0.8, 1.0

이렇게 둘러싸고 있는 경우는 일반적으로 세 번째 보다 최적화하기가 더 어렵다. 그 이유는 네 번째 문제는 개체가 전역 최적해로 접근하는 모든 주위에 지역최적해가 겹겹이 걸쳐있기 때문이다. 위와 같은 네 개의 문제에 기존의 방법과 본 논문에서 제안한 방법을 적용하여 실험하였다.

표 1은 유전자알고리즘의 설정된 파라미터로서 논문 [6]과 똑같이 설정하였다. 교배확률 및 돌연변이 확률은 일반적으로 유전자알고리즘에서 적용하는 대표적인 파라미터를 사용하였다. 개체수가 많으면 전역 최적해를 찾는 세대수는 작아지나 한 세대에 다루어야 할 개체수가 많아서 시간이 오래 걸린다. 개체수와 최적해를 찾는 세대수는 대체적으로 반비례한다. 그러나 개체수가 너무 적은 경우에는 지역 최적해를 빠져나오기가 더욱 어려워져 특히 지역 최적해가 많은 문제에서는 성능이 급격히 저하된다. 반면 세대수가 너무 많은 경우에는 많은 개체를 처리하느라 각 개체의 진화가 빨리 진행되지 못하는 단점이 있다. 그러므로 적당한 개체수를 선정해야하나

최적화 문제별로 적당한 개체수가 몇 개인지를 정확히 정하는 방법은 현재로서는 없다. 본 논문에서는 너무 작지도 크지도 않은 30개를 사용하였다. 개체비트 수는 함수최적화의 경우 찾는 해의 정밀도에 관계가 있다. 전역 최적해를 찾는 탐색영역을 해당 비트수로 표현하기 때문이다. 본 논문에서는 20비트를 사용하였는데, 이는 x_1, x_2 의 2차원으로 이루어진 문제에서 각 차원별로 10비트를 사용했다는 것을 의미한다. 예를 들어 함수 f_4 라면 x_1 및 x_2 각 축으로 $2^{10} = 1024$ 로 영역을 나누었다는 것을 뜻한다. 즉 -10은 0000000000이되고 10은 1111111111가 된다. 정상 돌연변이비율과 강돌연변이 확률은 논문 [6]과 동일하게 설정하였다.

실험은 주어진 파라미터와 무작위 초기개체를 이용하여 전역 최적해를 찾는 실험을 하였다. 개체 중 어떤 하나의 개체가 전역 최적해를 찾았을 때의 세대 수를 기록하고 이 기록을 평균하여 실험결과로 제시하였다. 각 파라미터에 대하여 다른 무작위 초기 개체로 10번의 실험을 하였으며 그 값을 평균한 실험결과가 표 2에 있다 (표준편차도 구했지만 간단하게 나타내기 위하여 생략하였다). 각 방법별로는 똑같은 초기개체를 사용하였으며 모든 파라미터는 동일하기 때문에 방법 이외에서의 차이점은 없다. 표 2에서 SGA, QGA, PGA는 각각 기존의 유전자알고리즘, 여왕벌 유전자알고리즘 그리고 본 논문에서 제안한 다중 여왕벌 유전자알고리즘을 표시한다. 표 2를 보면 몇 개의 예외를 제외하고 대부분의 경우 QGA보다 PGA가 성능이 우수함을 볼 수 있다 (똑같은 파라미터에서 더 성능이 좋은 것에 밑줄을 표기하여 나타내었다). 특히 지역 최적해가 큰 영역에 걸쳐있어 한번 빠지면 빠져나오기 힘

표 3 실험결과 (elitism 적용)

SGA		6409		5505.8		2021.4		43728.8	
ξ	p'_m	f_1		f_2		f_3		f_4	
		PGA	PGA-E	PGA	PGA-E	PGA	PGA-E	PGA	PGA-E
0.4	0.6	<u>11.1</u>	126	97.1	<u>14.7</u>	53.9	<u>17.9</u>	<u>714.9</u>	1918.3
	0.8	11.5	<u>11.4</u>	80.5	<u>11.5</u>	53.6	<u>11.9</u>	742.3	<u>382.8</u>
	1.0	14.2	<u>12.3</u>	26.4	<u>7.7</u>	43.7	<u>13.6</u>	11620.0	<u>8619.5</u>
0.6	0.6	9.7	<u>7.6</u>	352.1	<u>69.8</u>	111.2	<u>19.1</u>	<u>1363.4</u>	4099.1
	0.8	9.4	<u>9.4</u>	116.7	<u>22.8</u>	53.5	<u>14.0</u>	765.4	<u>698.4</u>
	1.0	10.7	<u>10.5</u>	19.1	<u>8.6</u>	24.3	<u>20.7</u>	<u>3671.6</u>	5687.8
0.8	0.6	9.5	<u>8.1</u>	218.8	<u>87.6</u>	41.7	<u>10.9</u>	<u>3827.1</u>	<u>1470.7</u>
	0.8	11.5	<u>8.1</u>	114.4	<u>22.9</u>	54.8	<u>8.3</u>	1002.2	<u>461.5</u>
	1.0	10.4	<u>9.8</u>	39.4	<u>22.7</u>	53.3	<u>15.1</u>	<u>2346.1</u>	5164.3
1.0		8.0	<u>7.7</u>	<u>20319.2</u> (5)	316127.2 (5)	6925.1	<u>10.4</u>	<u>515132.1</u> (6/4)	634218.0 (6/4)

든 f_2 함수에서 성능이 크게 좋은 것으로 보았을 때 우리가 본 논문의 제안 알고리즘에서 의도한 대로 효과는 나타나지 않는 것으로 판단된다. 함수 f_3 의 경우에는 비록 제안한 방법이 전반적으로 우수하였으나 개선된 정도가 크지 않았는데, 그 이유는 최대값이 유사한 지역해가 아주 많기 때문인 것으로 판단된다. 즉, 제안한 방법이 비록 다른 지역 최적해로 이동하는 것은 더 빠르게 가능하지만 그러한 지역 최적해가 많아서 그 지역 최적해가 전역 최적해일 가능성이 낮기 때문이라고 생각된다. 이러한 경향성은 지역 최적해가 전역 최적해를 감싸고 있는 함수 f_4 에서도 발생한다. 그렇지만 대부분의 경우 제안한 방법이 더 우수함을 볼 수 있다. 표에서 괄호부분은 (성공 횟수/실패횟수)를 의미한다. 괄호가 없는 것은 실험횟수 10번 모두 성공한 것이다. 실패란 아주 큰 세대동안 유전자알고리즘이 전역 최적해를 찾지 못한 경우로서 언제 찾을지 모르기 때문에 특정 세대수를 넘어가면 강제로 종료한 경우를 의미한다. 본 논문에서는 세대수가 2,000,000을 넘어가면 실패로 보고 종료하였다. ξ 가 1.0인 것은 모든 개체를 정상 돌연변이 한다는 의미로서 함수 f_1 처럼 단순한 문제는 오히려 정상 돌연변이 한 경우가 더 성능이 좋았다. 그러나 나머지 함수 f_2 에서 f_4 는 정상 돌연변이의 경우 아주 많은 세대가 걸렸으며 특히 f_2 와 f_4 의 경우에는 실패 확률도 높았다. 그 이유는 위에서 언급한 것처럼 f_2 는 지역 최적해가 넓은 영역에 걸쳐 있기 때문에 벗어나기가 힘든 문제점이 있으며 f_4 의 경우에는 전역 최적해를 감싸고 있어서 그러한 문제가 발생한다. f_2 와 f_4 에서 실패하는 횟수도 기존 방법보다 제안한 방법이 더 작았다.

표 3은 다중 여왕벌을 사용한 것에 추가하여 elitism을 적용하여 실험한 결과이다. elitism은 이전 세대에서 가장 좋은 개체를 보존하는 방법으로서 일반적으로 유전자알고리즘에 많이 사용된다. 그러나 elitism을 사용한다고 성능이 무조건 좋아지는 것은 아니다. elitism을 사용하면 지역 최적해에 더 빨리 빠지고 빠져나오기도 더 어렵다. 본 논문에서 제안한 다중 여왕벌 방법은 기존의 여왕벌 방법보다 지역 최적해 문제가 완화된 방법으로 elitism이 부정적 효과보다는 긍정적 효과를 기대할 수 있다. 실제로 표 3에서 보면 f_1 에서 f_3 에서는 elitism을 적용한 방법(PGA-E)이 대부분 더 성능이 좋음을 알 수 있다. 그러나 지역 최적해가 전역 최적해를 둘러싸고 있는 함수 f_4 에서는 elitism의 효과가 항상 발생하는 것은 아니었다.

IV. 결론

본 논문에서는 기존의 여왕벌 알고리즘이 하나의 여왕벌을 사용하여 지역 최적해에 빠진 경우 벗어나기 힘든 문제가 발생하는 것을 완화하기 위하여 여러 개의 여왕벌을 사용하는 방법을 제안하였다. 이전 세대의 최고 적합도를 갖는 여왕벌 개체에 추가적으로 부모의 적합도에 비하여 가장 크게 적합도가 좋아진 개체를 새로운 여왕벌로 삼아 두 개의 여왕벌을 이용하여 진화하는 알고리즘을 제안하였다. 부모의 적합도에 비하여 자식개체의 적합도가 얼마나 개선되었는지를 계산하기 위하여 두 부모 개체의 적합도를 평균한 것을 부모의 적합도로 사용하였다. 교배연산에서는 기존의 유전자알고리즘에서 사용하는 선택 알고리즘 (예로, 룰렛휠 선택)으로 선택된 개체와 여왕벌 개체를 교배시키기 위하여 두 개의 여왕벌 개체와의 해밍거리를 계산하고 해밍거리가 큰 여왕벌 개체와 교배하게 하였다. 돌연변이는 기존의 방법과 동일하게 하여 실험하였다. 4개의 함수최적화 문제를 이용하여 대량의 실험을 수행한 결과 기존의 하나의 여왕벌만 사용하는 유전자알고리즘보다 성능이 대부분의 경우에서 크게 향상됨을 볼 수 있었다.

참고문헌

- [1] D. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley.
- [2] C. Xudong, Q. Jingen, N. Guangzheng, Y. Shiyu, and Z. Mingliu, "An Improved Genetic Algorithm for Global Optimization of Electromagnetic Problems," IEEE Trans. on Magnetics, vol. 37, pp. 3579-3583, Sep. 2001.
- [3] Zihua Tang, Youtuan Zhu, Guo Wei, and Jinkang Zhu, "An Elitist Selection Adaptive Genetic Algorithm for Resource Allocation in Multiuser Packet-based OFDM Systems," Journal of Communications, vol. 3, no. 3, pp. 27-32, Jul 2008.
- [4] R. Poli, J. Kennedy, and T. Blackwell, "Particle

swarm optimization: An overview," Swarm Intelligence, vol. 1, pp. 33-57, Aug. 2007.

[5] M. Dorigo and T. Stutzle, Ant Colony Optimization The MIT Press, 2004.

[6] S. H. Jung, "Queen-bee Evolution for Genetic Algorithms," Electronics Letters, vol. 39, no. 6, pp. 575-576, Mar 2003.

[7] Zhang Jinhua, Zhuang Jian, Du Haifeng, and Wang Sun'an, "Self-organizing genetic algorithm based tuning of PID controllers," Information Sciences, vol. 179, pp. 1007-1018, 2009.

[8] J. Andre, P. Siarry, 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.

저 자 소 개



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