

지역적으로 경쟁하는 유전자 알고리즘 (A Genetic Algorithm with Local Competing)

강 태 원 ^{*}

(TaeWon Kang)

요 약 한 개의 모집단으로 구성되는 단순 유전자 알고리즘은 일반적으로 하나의 최적해를 찾는 경우에만 효과적이다. 그러나, 많은 문제들은 여러 개의 최적해를 가질 수 있으며, 그것들 모두를 찾는 것이 중요한 경우가 많다. 이 논문에서는 모집단 내 개체들에 지리적인 이웃의 개념을 부여하여, 각 개체들이 지역적으로 경쟁하면서도 전역적으로 유전자를 교환할 수 있도록 하여, 하나의 모집단이 여러 개의 최적해를 포함하도록 하는 유전자 알고리즘을 제안한다. 또한, 30비트, 6차 바이폴라-디셉티브 함수(bipolar-deceptive function)를 비롯한 여러 개의 최적해를 갖는 다양한 문제들에 적용하여 성능을 평가한다. 마지막으로 제안한 알고리즘에 대한 몇 가지 개선 방향을 제시하였다.

키워드 : 유전자 알고리즘, 다수의 최적해, 이웃, 세포자동자, 바이폴라-디셉티브 함수

Abstract On the whole, the simple GAs with just one population set is effective in finding one optimal solution. However, many real world problems have a lot of optimal solutions, and often it is important to find all of them. In this paper, we propose a GA that has a population set containing multiple optimal solutions. In the proposed GA, each of the individuals in population set has its own geological neighbors, and they exchange their genes globally as well as compete with others locally. The paper then evaluates the proposed GA along with many multimodal problems including a 30bit, order-six bipolar-deceptive function. Finally, we present some improvement directions of the proposed GA.

Key words : Genetic Algorithms, Multiple Solution, Neighborhood, Cellular Automata, Bipolar-deceptive Function

1. 서 론

한 개의 모집단으로 구성되는 단순 유전자 알고리즘은 일반적으로 하나의 최적해를 찾는 경우에만 효과적이다. 실제로 단순 유전자 알고리즘을 적용하면 세대가 거듭되면서 모집단 내의 개체들은 거의 유사한 유전자를 갖게된다[1]. 단순 유전자 알고리즘은 한 개의 모집단을 구성하는 것으로 시작된다. 여기서 모집단 내의 개체는 문제에 대한 잠정적인 해를 나타내는 것이며, 초기 모집단 내의 개체는 임의로 정한다. 초기 모집단이 생성되면 각 개체를 평가하여 그들의 적합도를 계산하고, 그 값에 근거하여 다음 세대에 자손을 남길(즉, 자신의 유전자를 다음 세대에 넘길) 개체를 선택한다. 선택된 개체들은 자신의 유전자를

모두 전달하거나, 교차를 통하여 다른 개체와 부분적으로 혼합된 유전자를 전달하거나, 돌연변이를 통하여 자신의 것과 다른 유전자를 포함하도록 전달한다. 그렇게 만들어진 개체들로 구성된 모집단이 다음 세대를 형성하는 것이다. 결국 이러한 진화 과정을 되풀이하면 모집단내 개체들은 거의 유사한 유전자를 갖게되며, 결과적으로 단순 유전자 알고리즘에서 모집단은 하나의 최적해로 수렴하게 된다.

그러나, 실세계의 많은 문제들은 여러 개의 최적해를 가질 수 있으며, 동시에 그것들 모두를 찾는 것이 중요한 경우가 많다[1, 2, 3]. 단순 유전자 알고리즘에서 모집단이 하나의 최적해로 수렴하는 이유는 각 개체들이 전역적으로 유전자를 교환하기 때문이다. 이 논문에서는 모집단 내 개체들에 이웃의 개념을 부여하여 그것들이 지역적으로 경쟁하고 결과적으로 지역적으로 유전자를 교환하도록 하여 하나의 모집단이 여러 개의 최적해를 포함하며 수렴하도록 한다. 이웃이라는 개념을 모델링하

^{*} 정 회 원 : 강릉대학교 컴퓨터공학과 교수

twkang@kangnung.ac.kr

논문접수 : 2002년 1월 18일

심사완료 : 2002년 3월 12일

는 방법은 여러 가지가 있을 수 있다. 예를 들어 유전자 알고리즘 분야에서 비트 스트링으로 표현된 개체에 대하여 1~2개의 비트만이 다른 것을 이웃으로 한다던가, 특정한 꼬리표를 붙인 후 동일한 꼬리표를 가지면 이웃(혹은 형제)이라고 간주하는 경우 등이 그것이다. 그러나 이 논문에서는 보다 지리적인 이웃의 개념을 사용한다. 이 논문에서는 모집단내의 개체들을 원환체(Torus) 형태로 구성한다. 즉, 도너츠 형태의 격자공간이 개체들의 거주 공간이 되며, 각 개체는 이 공간상에서 이웃이라는 관계를 가지며 지역적으로 상호 경쟁하며 진화한다. 즉, Conway의 생명게임(The Game of Life)[4]에서 격자공간상의 각 셀들이 이웃 셀들의 상태에 의하여 자신의 생과 사가 결정되는 것과 유사한 방식을 사용한다. 지역적인 단순한 규칙을 사용하는 생명게임에서 각 셀들이 지역적으로 상호작용하며 전역적으로 충분히 복잡한 동작을 보이는 것을 이용하려는 것이다[5,6,7]. 이 논문에서는 이렇게 지역적으로 경쟁하는 유전자 알고리즘을 제시하고, 30비트, 6차 바이폴라-디셉티브 함수(bipolar-deceptive function)를 비롯한 여러 개의 최적해를 갖는 다양한 문제들에 적용하여 성능을 평가한다.

이 논문의 구성은 다음과 같다. 우선 2장에서는 여러 개의 해를 동시에 찾기 위한 기존의 연구들을 살펴보고, 3장에서는 원환체형 격자공간 모집단으로 구성된 지역적으로 경쟁하는 유전자 알고리즘을 제안하고, 4장에서는 여러 개의 최적해를 갖는 다양한 문제들에 제안한 알고리즘을 적용하여 성능을 평가·분석해보고, 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

여기서는 진화 알고리즘에 기반 하여 여러 개의 최적해를 동시에 찾는 알고리즘에 대한 연구 사례를 소개한다. 가장 처음으로 이 문제의 해결을 시도한 것은 반복에 기반한 방법이다[2]. 이 방법은 동일한 문제에 대하여 유전자 알고리즘을 독립적으로 여러 번 적용하여, 여러 최적해를 찾는다는 매우 단순한 것이며, 다수의 최적해 각각이 찾아질 확률이 거의 같다는 가정을 전제로 하지만, 실제로 그렇지 않기 때문에 효과적이지 못하다. 이 방법은 일단 최적해가 찾아지면 그것을 배제하도록 적합도 함수를 수정하는 것으로 Beasley, Bull, 그리고 Martin에 의하여 수정되어 사용되고있다[8, 9].

다음으로 De Jong에 의하여 제안된 밀집 인자(crowding factor) 모델이 있다[9, 10]. 이 방법은 새로이 생성된 개체가 모집단내의 가장 유사한 개체를 대체한다는 개념에 바탕을 둔 것이다. 실험적으로 이 방법은

여러 개의 최적해를 갖는 문제에 대하여 2개의 최적해만을 찾는 것으로 알려졌다[9]. 그러나, 현재는 Mahfoud에 의한 결정적 밀집(deterministic crowding) 알고리즘과 Harik에 의한 RTS(restricted tournament selection) 등으로 개량되어 사용되고 있다[2,3,9,11].

다른 방법으로는 Goldberg와 Richardson이 제안한 것으로 공유에 기반한 방법이 있다[9, 12]. 이 방법은 모집단 내의 각 개체에 대하여 그것들 사이의(유전형 혹은 표현형 수준에서의) 유사성을 계산한 후, 유사한 집단에 속하는 것끼리 적합도를 감소시키도록 하여, 특정한 형태의 유전자형을 갖는 개체가 모집단의 대부분을 차지하는 것을 제한함으로써 모집단이 여러 개의 최적해로 수렴하도록 유도하는 방법이다. 결국 모집단은 유사한 정도에 의하여 (유전형 혹은 표현형 측면에서) 독자적인 부 모집단(subpopulation)을 구성하게 되며, 다양한 문제에 적용한 결과 상당히 효과적인 것으로 알려져 있다. 이 방법의 단점은 개체들 사이의 유사성을 평가하는 추가 시간이 필요하다는 것이다.

최근에 Spears에 의하여 제안된 것으로 공유와 제한된 짝짓기를 기반으로 하는 방법이 있다[13]. 이 방법 역시 Goldberg 등의 방법과 유사하며, 여기서는 각 개체에 (임의로) 꼬리표를 붙여서 같은 꼬리표를 갖는 것들은 같은 부 모집단에 속하는 것으로 취급한다. 그리고, 적합도를 부 모집단 단위로 평가함으로써, 여러 개의 최적해를 동시에 찾을 수 있다. 제한된 짝짓기는 상이한 유전자를 갖는 최적 개체간의 교차를 금지시키는 것을 말한다. 일반적으로 최적 개체간의 교차의 결과가 나쁘기 때문이다.

기존의 연구들은 나름대로의 근거를 가지고 제안된 방법들이며 특히, 그것들은 공통적으로 적합도를 제어하여 특정 개체가 모집단 전체를 차지하는 것을 제한한다는 특징을 갖는다. 이와는 다른 것으로 병렬 유전자 알고리즘 분야에서는 단절된 평형(punctuated equilibria)에 기반하여 다수의 모집단을 독립적으로 진화하도록 하되 주기적으로 개체들을 이주(migration)하는 기법을 사용한다. 그러나 이 방법이 여러 개의 해를 찾는 데 때로 유용하기는 하지만 근본적으로 이것은 더 나은 해를 더 빨리 찾는 것에 중점을 둔 것이며 이주주기(migration intervals), 이주율(migration rates)등을 제어해야 하는 복잡함을 수반한다[14]. 가장 최근에는 공유에 기반한 것으로 지역 탐색(local search)과 GA를 결합한 혼합 GA에 대한 연구 결과가 발표되었다[15]. 이 논문에서는 이러한 것들과는 다르게 모집단 내 개체들을 원환체형의 격자공간에 거주시켜 보다 지리적인

이웃의 개념을 부여하고, 각 개체들이 지역적으로 경쟁하면서도 전역적으로 유전자를 교환할 수 있도록 하여, 하나의 모집단이 여러 개의 최적해를 포함하도록 하는 유전자 알고리즘을 제안한다.

3. 지역적으로 경쟁하는 유전자 알고리즘

어떤 문제에 유전자 알고리즘(GA)을 적용하기 위해서는 개체의 표현방법, 모집단 구성 방법과 적용할 진화 연산자 및 유전 연산자를 정하여야 한다. 제안하는 유전자 알고리즘 역시 동일한 요소를 정하여야 한다. 여기서는 지역적으로 경쟁하는 유전자 알고리즘(LCGA로 표기)에서 사용하는 원환체형 격자공간 모집단과 거기에서의 이웃의 개념을 설명하고, LCGA에서 사용하는 진화 및 유전 연산자를 설명한다.

3.1 원환체형 격자공간 모집단과 이웃

유전자 알고리즘에서 개체의 표현법이 정해지면 그러한 개체들의 모임인 모집단을 구성하게 된다. 단순 유전자 알고리즘에서 모집단은 일정한 개수(모집단의 크기)의 임의의 개체들의 모임이며, 모집단 내의 개체들은 단지 같은 세대 구성원이라는 것 이외의 어떠한 관계도 가지고 있지 않다.

LCGA에서는 모집단 내 개체들 사이에 지리적인 이웃의 개념을 부여하기 위하여 임의로 생성된 개체들을 다음 그림 1(a)와 같은 원환체 구조로 배열되도록 한다. 즉, 격자가 그려진 도너츠 모양의 공간에서 각 격자가 하나의 개체들이 된다. 이러한 원환체형 격자공간은 한 단면을 자르면 원통형이 되고, 그것을 다시 세로로 잘라 펼치면 그림 1(b)와 같은 2차원 테이블 구조가 된다. 물론 모집단의 이러한 구성은 논리적인 것이다. 즉, 실제 구현은 어떤든 간에 개체들이 논리적으로 그렇게 구성되면 된다.

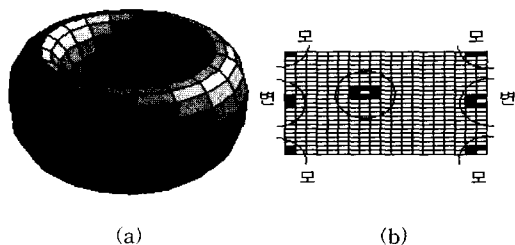


그림 1 원환체와 이웃

원환체형으로 구성된 모집단에서 개체의 이웃은 여러 가지 방법으로 정의될 수 있다. 그림 2는 가능한 3가지

경우를 나타내며, 그림 1의 (b)에는 그림 2의 (a)와 같이 인접한 8개의 이웃을 갖는 개체를 내부, 변, 모서리로 구분하여 나타낸 것이다.

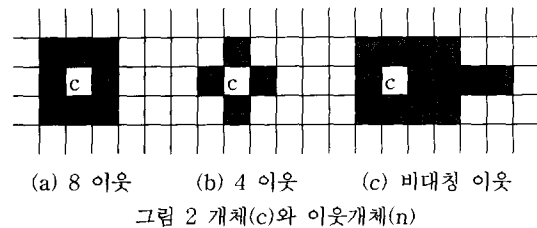


그림 2 개체(c)와 이웃개체(n)

3.2 진화 및 유전 연산자

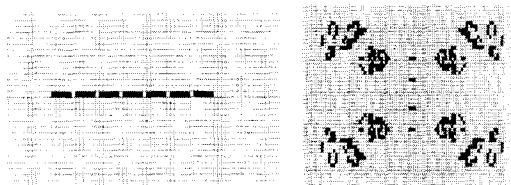
유전자 알고리즘을 적용할 때 사용하는 진화 연산자(evolutionary operator) 즉, 선택 연산자 및 교차와 돌연변이로 대표되는 유전 연산자(genetic operator)의 종류는 매우 다양하며, 각각의 방법은 나름대로의 특징을 갖는다[16, 17]. 이 논문에서 제안하는 지역적으로 경쟁하는 유전자 알고리즘 역시 다양한 진화 및 유전 연산자를 사용할 수 있다. 그러나, LCGA에서 중요한 것은 개체들이 이웃의 개념을 가지고 있으며, 따라서 모집단 내의 개체들이 내부적인 지역성을 가지고 있다는 것이다. 따라서, 진화 및 유전 연산자를 고안함에 있어서 지역성이 반드시 적용되도록 해야한다. 극단적으로, 모집단을 원환체형 격자공간 형태로 구성하였다고 하더라도, 이웃의 개념이 없이 보통의 연산자들을 사용한다면 보통의 단순 유전자 알고리즘과 다를 것이 없다. 지역성을 적용하면 모집단내의 개체들은 지역 중심으로 진화함으로써 전체 모집단에 여러 개의 최적해가 존재할 수 있을 것이다. 반면에 지역이 병렬 유전자 알고리즘에서와 같이 분리된 일종의 섬(island)을 구성하는 경우에는 적절한 이주정책을 사용하지 않으면 양질의 최적해를 찾기가 어려울 수 있다. 따라서 지역적으로 경쟁하되 각 개체들이 유전자를 전역적으로 전파하는 수단이 제공되어야 할 것이다.

이 논문에서는 이러한 두 가지 성질을 만족하는 가능한 여러 가지 방법 중에 "생명게임(The Game of Life)"[4]에 사용된 규칙을 응용한 진화 및 유전 연산자를 제안한다. 생명게임은 1970년 수학자인 John Conway가 고안한 것으로, 무한히 확장 가능한 바둑판 모양의 격자공간에서 진행되는 게임이다. 각 격자 상에는 셀들이 거주하며 각 셀들은 "생" 아니면 "사"의 상태에 있다. "생" 상태의 셀이 있는 격자는 특정한 표시를 하고, "사"인 경우는 빈 격자로 표시하며, 각 셀은 대각선

방향을 포함한 8개의 이웃 셀을 갖게된다(그림 1(a)). 이러한 상황에서 초기에 임의로 셀들을 배열한 후 다음과 같은 3가지 규칙을 사용하여 격자공간상의 셀들이 “생”과 “사”를 되풀이한다.

- 규칙1: 3개의 “생”인 이웃을 갖는 “사”인 셀은 “생”이 된다.
- 규칙2: 2개 혹은 3개의 “생”인 이웃을 갖는 “생”인 셀은 현 상태를 유지한다.
- 규칙3: 그 이외의 경우 “생”인 셀은 “사”가되고, “사”인 셀은 현 상태를 유지한다.

복잡계(complex systems)에 대한 모델링 수단인 세포자동자(cellular automaton)의 한 예인 생명게임은 “창발적 복잡성(emergent complexity)” 혹은 “자기 조직계(self-organizing systems)”라고 불리기도 하는데, 이렇게 지역적인 정보만을 사용하는 간단한 규칙으로 진행되는 생명게임이 매우 복잡한 전역적인 동작을 보인다는 것은 널리 알려진 사실이다[5, 7]. 실제로 Paul Rendell은 초기 셀들을 적절하게 배치하여 만능 튜링머신을 만들기도 하였다[18]. 다음 그림 3은 왼쪽(a)과 같은 초기상태에서 출발하여 수십 단계가 지난 후의 패턴(b)을 나타낸다.



(a) 초기 상태 (b) 수십 단계 진화 상태
그림 3 생명 게임의 복잡성

생명게임이 이 논문의 주제는 아니므로 더 이상 언급하지는 않겠으며, 이 논문에서는 지역적으로 상호작용하는 단순한 규칙을 사용하는 계에서 매우 복잡한 전역적인 현상이 일어날 수 있다는 사실에 기초하여, 진화 및 유전 연산자를 만든다.

제안하는 LCGA에서는 먼저 각 개체의 적합도를 계산한 후, 각 개체에 대하여 지역 적합도를 계산한다. 지역 적합도는 자신의 이웃 중에서 자신보다 적합도가 큰 이웃의 개수를 의미한다. 다음으로 진화 및 유전 연산은 개체의 지역 적합도가 특정한 값 보다 작으면 개체는 변하지 않고, 그렇지 않은 경우 유전 연산을 적용한다.

제안하는 지역적으로 경쟁하는 유전자 알고리즘(LCGA)의 세부적인 처리과정은 다음과 같다.

LCGA의 절차:

1. 문제 정의

문제를 분석하여, 목적 함수 및 개체의 표현법을 정한다.

2. 모집단 구성 및 이웃의 유형 결정

모집단의 크기 |P|는 일반성을 잃지 않고 $n \times m$ ($m, n \geq 2$, 개체의 중복 가능)으로 정할 수 있으며, 이 경우 모집단을 크기가 $n \times m$ 인 원환체형 격자공간으로 구성한다. 이때, 사용할 이웃의 유형도 정한다. 여기서, n, m 은 원환체를 평면으로 펼쳤을 때 행과 열의 수를 각각 나타낸다.

3. 적합도 및 지역 적합도 계산

3.1 각 개체의 적합도 fit를 계산한다.

3.2 각 개체에 대하여, 지역 적합도 lof를 계산한다. lof는 자신의 이웃 중에서 자신보다 적합도가 큰 이웃의 개수를 말한다.

4. 진화 및 유전 연산

각 개체에 대하여 진화 및 유전 연산은 다음의 규칙을 따른다.

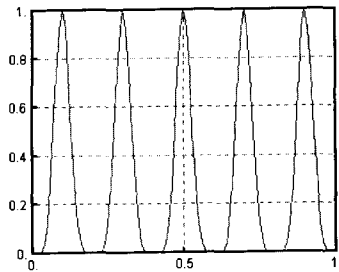
- ① lof가 s보다 작으면 개체는 변하지 않는다.
- ② lof가 s보다 작지 않으면 유전 연산을 적용한다.

5. 3번으로 이동 또는 종료

결국, LCGA에서는 각 개체가 지리적으로 인접한 개체와 경쟁하여 자신의 “생·사”가 결정되며 따라서, 특정 개체가 모집단 전체로 급속히 퍼지는 것을 제한하게 된다. 반면에 전체 모집단 수준에서는 지역을 구분하는 경계가 정해져있는 것이 아니므로 (원환체형 모집단은 사방으로 끝없이 연결되어 있다) 자연스럽게 지역 간의 유전자 교환이 일어나며 서서히 전파된다. 또한, LCGA는 지역적인 정보를 전역적으로 전파한다는 점에서 기존의 방법들과는 다르다. 기존의 방법들은 모집단 내 모든 개체들의 적합도 정보를 사용하여 지역적인 진화를 유도하는 반면, LCGA는 이웃의 적합도 정보를 사용하여 다수의 최적해를 찾는다는 전역적인 진화를 유도한다.

4. 실험 및 분석

제안한 LCGA 알고리즘이 여러 개의 해를 동시에 찾는 것을 확인하기 위하여, 이 논문에서는 크게 3가지 경우로 구분하여 실험을 수행하였다. 첫 번째 실험은 1차원 목적함수를 대상으로 여러 가지 최적해가 다양한 형태를 취하거나 여러 가지 최적해들이 정의역 상에 다양하게 분포하는 경우이고, 두 번째 실험은 2차원 목적함수를 대상으로 하였으며, 세 번째는 목적 함수가 약 5백만 개의 지역최적해와 32개의 전역최적해를 갖는 30비트, 6차 바이폴라-디셉티브 문제이다.

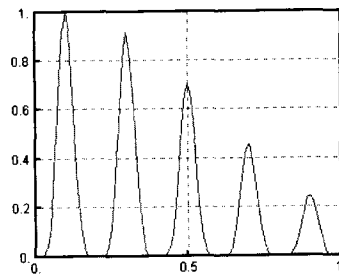


(a) $f_1(x) = \sin^6(5\pi x)$

0.953:0.092	0.660:0.290	<u>0.500:1.000</u>	<u>0.300:1.000</u>	<u>0.500:1.000</u>	<u>0.100:1.000</u>
<u>0.700:1.000</u>	<u>0.500:1.000</u>	0.087:0.880	0.369:0.010	<u>0.700:1.000</u>	<u>0.500:1.000</u>
0.115:0.840	<u>0.700:1.000</u>	<u>0.701:0.999</u>	0.524:0.649	0.234:0.017	0.041:0.046
<u>0.300:1.000</u>	<u>0.300:1.000</u>	<u>0.300:1.000</u>	<u>0.300:1.000</u>	<u>0.900:1.000</u>	0.244:0.066
<u>0.300:1.000</u>	<u>0.700:1.000</u>	<u>0.300:1.000</u>	<u>0.300:1.000</u>	0.326:0.609	<u>0.100:1.000</u>
<u>0.501:0.999</u>	0.589:0.000	<u>0.900:1.000</u>	0.354:0.082	0.418:0.000	0.615:0.000

(b) 수행결과(해:적합도)

그림 4 함수 1과 LCGA 수행후의 모집단



(a) $f_2(x) = e^{-2 \ln^2((x-0.1)/0.8)^2} \sin^6(5\pi x)$

<u>0.099:1.000</u>	0.129:0.532	0.350:0.114	0.115:0.845	0.666:0.200	0.279:0.663
<u>0.100:1.000</u>	<u>0.100:1.000</u>	<u>0.100:1.000</u>	0.961:0.007	<u>0.100:1.000</u>	0.641:0.025
0.396:0.000	0.489:0.660	<u>0.100:1.000</u>	<u>0.100:1.000</u>	<u>0.299:0.917</u>	<u>0.100:1.000</u>
0.256:0.201	<u>0.100:1.000</u>	0.253:0.158	0.876:0.177	<u>0.100:1.000</u>	0.785:0.000
<u>0.903:0.246</u>	0.053:0.158	0.094:0.975	0.512:0.622	0.509:0.657	<u>0.098:0.998</u>
<u>0.099:1.000</u>	<u>0.102:0.997</u>	0.458:0.185	<u>0.101:0.999</u>	<u>0.100:1.000</u>	<u>0.100:1.000</u>

(b) 수행결과(해:적합도)

그림 5 함수 2와 LCGA 수행후의 모집단

4.1 1차 목적함수에 대한 실험 및 분석

여기서는 4개의 1차 목적함수를 대상으로 제안한 LCGA의 성능을 평가한다. 여기에 사용한 목적함수들은 Goldberg, Spears, De Jong, Mahfoud, Harik 등이 여러 개의 최적해를 갖는 문제를 해결하는 알고리즘을 제안하고 그것의 성능을 평가하는데 사용한 것이다[9]. 실험은 펜티엄 933MHz CPU, 램 256 MByte, Windows me 운영체제 하에서 C 언어를 사용하였다. 공통적으로 모집단의 크기는 36으로 하였으며, 세대수는 2000으로 하였다. LCGA의 경우 이웃은 8 이웃을 사용하였으며, 3절의 LCGA 수행과정 중 단계 4는 구체적으로 다음을 적용하였다. 교차 확률은 0.75, 돌연변이 확률은 0.05로 하였다. 교차 시에는 8개의 이웃 중에서 1개를 임의로 선택하도록 하였다. 여기서 lof는 자신의 이웃 중에서 자신보다 적합도가 큰 이웃의 개수를 나타내는 지역 적합도이고, s와 d는 개체의 상대적 우열을 구분하기 위하여 임의로 정한 값이다. 이 값 역시 해결하려는 문제에 따라 적절한 값이 있을 것으로 추정되지만, 이 논문에서는 고정된 값을 사용하였으며 별도의 연구가 뒤따라야 할 것이다. 특별히 다른 매개변수나 규칙을 사용하는 경우 별도로 언급하였다.

- ① lof가 s(=3)보다 작으면 개체는 변하지 않는다.
- ② lof가 d(=6)보다 크면 일점(1-point) 교차연산을 수행한다.

①, ② 이 외의 경우는 돌연변이 연산을 적용한다.

이러한 규칙은 주위의 개체들에 비하여 자신의 적합도가 상대적으로 낮지 않으면 ①) 살아남고, 자신의 적합도가 상대적으로 낮으면 ②) 주위에 있는 우수한 개체들의 유전자를 취할 수 있는 교차연산을 수행하며, 그렇지 않은 경우는 자신의 유전자를 임의로 교체하는 돌연변이 연산을 적용한다는 생각에 바탕을 둔다.

1) 함수 1(f1)

이 함수는 값이 같은 5개의 전역최적해를 가지며, 최적해들이 정의역상에 균등하게 분포하는 경우이다. 최적해는 0.1, 0.3, 0.5, 0.7, 0.9이며 이때의 함수값은 모두 1.0이다.

그림 4(b)에서 알 수 있듯이 LCGA가 5개의 전역최적해를 모두 찾았다는 것을 알 수 있다. 또한, 이들 최적해들이 모집단의 56%를 차지하며 0.1 2개, 0.3 5개, 0.5 5개, 0.7 5개, 0.9 3개로 비교적 고르게 분포함을 알 수 있다.

2) 함수 2(f2)

이 함수는 한 개의 전역최적해와 4개의 지역최적해를 가지며, 최적해들이 정의역상에 균등하게 분포하는 경우이다. 전역최적해는 0.1 이고, 지역최적해는 0.3, 0.5, 0.7, 0.9이며 이때의 함수값은 각각 1.0, 0.917, 0.707, 0.458, 0.250이다.

그림 5(b)에서 알 수 있듯이 최적해들은 모집단의 50%를 차지하지만 0.1 16개, 0.3 1개, 0.5 0개, 0.7 0개, 0.9 1개이므로 그 중에서 89%가 전역최적해이다. 결국, LCGA가 5개의 최적해 중에서 전역최적해를 찾는다는 것을 알 수 있다. 그러나, 0.3과 0.9 두 개를 제외하고, 다른 지역최적해들을 모집단내에 유지하지 못한다는 것을 알 수 있다. 이 점에 대해서는 4.4절 종합적인 분석에서 논의한다.

3) 함수 3(f_3)

이 함수는 f_1 과 마찬가지로 5개의 전역최적해를 가지지만, 최적해들이 정의역상에 균등하게 분포하지 않는 경우이다. 최적해는 0.08, 0.246, 0.45, 0.681, 0.934이며 이때의 함수 값은 모두 1.0이다.

그림 6(b)에서 알 수 있듯이 최적해들이 정의역상에 균등하게 분포하지 않는 경우에도 LCGA가 5개의 전역 최적해를 모두 찾는다는 것을 알 수 있다. 또한, 이들

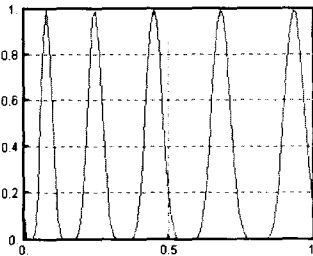
최적해들이 모집단의 47%를 차지하며 0.08 2개, 0.246 4개, 0.45 3개, 0.681 5개, 0.934 3개로 비교적 정의역의 크기에 맞게 분포함을 알 수 있다.

4) 함수 4(f_4)

이 함수는 f_2 와 마찬가지로 한 개의 전역최적해와 4개의 지역최적해를 가지지만, 최적해들이 정의역상에 균등하게 분포하지 않는 경우이다. 전역최적해는 0.08 이고, 지역최적해는 0.246, 0.45, 0.681, 0.934이며 이때의 함수 값은 각각 1.0, 0.917, 0.707, 0.458, 0.250이다.

그림 7(b)에서 알 수 있듯이 최적해들이 모집단의 42%를 차지하지만 모두 0.08이므로, 이 경우에도 LCGA가 5개의 최적해 중에서 전역최적해를 찾는다는 것을 알 수 있다. 하지만 f_2 와 마찬가지로 지역최적해들을 모집단내에 유지하지 못한다. 특히, 이 경우는 지역 최적해가 넓은 탐색공간을 차지하기 때문에 2000세대 후의 최종 모집단에 지역 최적해가 전혀 없다.

결론적으로 실험에 사용한 1차 목적함수에 대하여 제안한 LCGA가 여러 개의 전역최적해를 찾아내기는 하지만, 지역최적해를 유지하지는 못한다는 것을 알 수 있다. 이러한 특성에 대해서는 4.4절 종합적인 분석에서 논의한다.

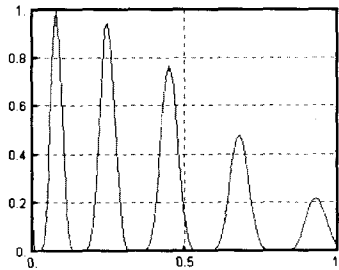


(a) $f_3(x) = \sin^6(5\pi(x^{3/4} - 0.05))$

0.779:0.001	<u>0.451:1.000</u>	0.573:0.000	<u>0.247:1.000</u>	0.823:0.000	0.949:0.903
0.653:0.653	0.441:0.946	0.203:0.154	0.555:0.000	<u>0.247:1.000</u>	<u>0.934:1.000</u>
0.536:0.002	<u>0.080:1.000</u>	0.612:0.052	<u>0.934:1.000</u>	0.947:0.934	<u>0.681:1.000</u>
<u>0.247:1.000</u>	0.556:0.000	<u>0.682:1.000</u>	0.472:0.752	<u>0.451:1.000</u>	0.239:0.951
0.483:0.526	<u>0.934:1.000</u>	<u>0.961:1.000</u>	0.961:0.721	0.803:0.000	<u>0.451:1.000</u>
<u>0.682:1.000</u>	<u>0.681:1.000</u>	<u>0.080:1.000</u>	0.833:0.002	<u>0.247:1.000</u>	0.039:0.032

(b) 수행결과(해:적합도)

그림 6 함수 3과 LCGA 수행후의 모집단



(a) $f_4(x) = e^{-2\ln 2((x-0.1)/0.8)^2} \sin^6(5\pi(x^{3/4} - 0.05))$

0.293:0.135	0.078:0.995	0.662:0.414	<u>0.081:0.997</u>	0.243:0.947	0.387:0.037
0.078:0.996	0.965:0.128	0.544:0.000	0.678:0.482	<u>0.081:0.998</u>	<u>0.079:0.998</u>
0.245:0.954	<u>0.081:0.996</u>	<u>0.080:0.999</u>	<u>0.080:0.999</u>	0.850:0.006	0.191:0.033
0.822:0.000	<u>0.079:0.999</u>	0.267:0.658	0.179:0.003	<u>0.079:0.999</u>	0.523:0.014
<u>0.080:0.999</u>	0.166:0.000	<u>0.080:0.999</u>	0.633:0.145	<u>0.079:0.999</u>	<u>0.080:0.999</u>
<u>0.080:0.999</u>	<u>0.081:0.997</u>	0.407:0.219	<u>0.080:0.999</u>	0.938:0.217	0.222:0.570

(b) 수행결과(해:적합도)

그림 7 함수 4와 LCGA 수행후의 모집단

-4.60:-4.93:30.786	-4.61: 4.93:30.891	<u>4.63:-4.92:31.013</u>	-1.10:-1.84:23.679	-4.59: 4.83:30.375
4.63: 4.92:30.761	-4.63: 4.83:30.822	-3.89: 4.19:14.454	-4.62: 4.62:30.671	-1.27: 1.87:19.263
1.60:-4.19:20.235	2.55:-0.39:22.861	-4.62:-4.62:30.647	-3.04:-0.74:23.344	-0.61: 1.13:23.214
1.05: 3.82:25.457	<u>-4.63:-4.92:31.044</u>	4.61:-4.93:30.810	0.29: 3.78:18.249	-4.63:-4.72:30.748
4.64: 4.83:30.842	-4.62:-4.82:30.907	-3.50:-2.68:18.930	-2.60: 0.86:23.453	-4.60: 4.93:30.803
-1.55: 4.96:20.186	-3.05: 3.64:25.474	3.09: 4.51:26.453	4.63: 4.83:30.897	-4.53: 4.56:19.661
2.59:-2.66:22.239	4.63: 4.83:30.947	4.60:-4.73:30.642	4.63:-4.73:30.818	4.62:-4.82:30.894
-4.64: 4.73:30.791	1.03: 2.57:19.635	-4.69:-1.94:25.736	3.16: 4.49:21.126	4.15:-3.12:28.369
4.61: 4.93:30.928	3.10:-2.77:21.669	2.17: 0.85:23.576	4.64:-4.73:30.794	-4.55:-0.29:23.858
4.63:-4.72:30.850	4.15:-3.43:28.749	4.64:-4.93:30.905	-4.64: 4.82:30.758	-0.80:-0.08:20.912
-4.61: 4.93:30.846	-3.44: 3.83:22.744	4.64: 4.72:30.776	1.26:-1.85:21.288	-4.63:-4.92:30.712
3.54: 2.05:23.150	-1.65:-2.54:24.195	-4.62: 4.83:30.665	-4.60:-4.83:30.635	2.26:-2.10:21.341
4.60: 4.92:30.821	-1.19:-1.14:23.020	2.35:-1.10:19.474	-3.95:-1.64:20.410	-4.62:-4.92:30.953
4.15:-4.53:29.922	2.45: 1.81:21.578	-3.53: 0.88:22.065	-4.65:-4.62:30.479	2.92: 0.84:19.239
-4.62:-4.82:30.925	-4.63: 4.72:30.800	4.63: 4.63:30.742	-4.61: 4.93:30.828	-0.21: 0.52:22.112
-4.62: 4.92:30.895	<u>4.62: 4.92:31.011</u>	1.80: 2.87:17.513	-4.12: 4.93:30.520	1.83:-1.38:18.634
-3.97: 2.93:22.811	4.63: 4.63:30.697	2.78:-0.36:20.349	4.12:-2.42:28.002	-0.46: 0.15:21.327
-4.63:-4.82:30.947	-4.64: 4.83:30.864	0.82:-4.95:22.330	-4.61: 4.92:30.931	4.64:-4.82:30.896
2.71:-3.94:25.584	1.46:-1.76:19.619	0.79: 3.82:24.954	-4.64: 4.82:30.890	<u>-4.62: 4.93:31.002</u>
3.42:-1.18:17.457	2.32: 0.40:19.828	-4.63:-4.92:30.883	-4.32:-4.18:13.939	2.00: 1.61:22.395

4.2 2차 목적함수에 대한 실험 및 분석

여기서는 3개의 2차 목적함수를 대상으로 제안한 LCGA의 성능을 평가한다. 실험 환경은 4.1과 같으나 다만, 모집단의 크기만을 문제에 따라 다르게 하였다.

1) 함수 5(f_5)

이 함수는 무수히 많은 지역최적해를 갖는 함수로 $(-5,5) \times (-5,5)$ 의 범위에서 4개의 전역최적해 (4.626, 4.925), (-4.626, 4.925), (-4.626, -4.925), (4.626, -4.925)를 가지며 그때의 함수값은 31.051 이다.

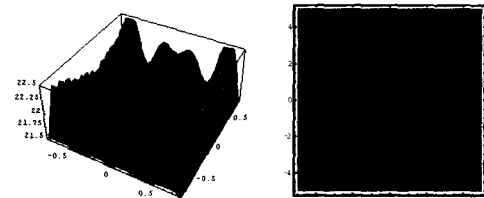


그림 8 $f_5(x, y) = 21.5 + x \times \sin(4x) + y \times \sin(20xy)$

모집단의 크기를 100으로 했을 때의 실험결과는 위와 같다. 소수점 이하 둘째 자리까지의 정밀도로 4개의 전역최적해를 모두 찾음을 알 수 있다. 또한, 최적 적합도 31.051의 97%에 해당하는 30.0 이상의 함수값을 갖는 개체들이 전체 모집단의 49%를 차지하며, 각각의 4분면에 각각 10, 17, 12, 10개의 개체들이 분산되어 있음을 알 수 있다. 결국 비교적 고르게 높은 적합도를 갖는 개체들이 분산되어 있음을 알 수 있다.

2) 함수 6(f_6)

이 함수는 지역최적해는 없으며 4개의 전역최적해를

갖는 함수로 $(-5,5) \times (-5,5)$ 의 범위에서 4개의 전역최적해 (1, 1), (-1, 1), (-1, -1), (1, -1)를 가지며 그때의 함수값은 0.135335 이다.

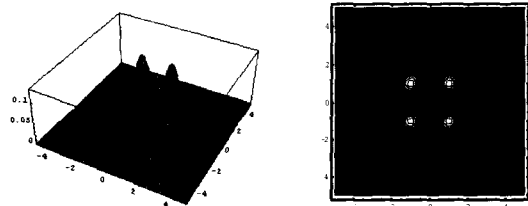


그림 9 $f_6(x, y) = x^2 \times y^2 \times e^{-(x^2 + y^2)}$

모집단의 크기를 36으로 했을 때의 실험결과는 아래와 같다. 전체 36 개체 중에서 16개가 전역최적해에 해당한다고 볼 수 있으며, 4 영역의 최적해를 모두 찾음을 알 수 있다. 각각의 4분면에 각각 1, 6, 5, 4개씩 분산되었다는 것을 알 수 있다. 이 경우는 1사분면의 최적해가 1개만 포함되었음을 알 수 있다. 그러나, 이러한 편차는 실험을 되풀이 해본 결과 특별한 경우이고 평균적으로 고르게 분포됨을 알 수 있었다.

3.04:-3.26:0.000	-0.58: 4.36:0.000	<u>-1.00:-0.98:0.135</u>	<u>-1.02: 0.99:0.135</u>
-2.45:-3.17:0.000	-4.39: 2.69:0.000	<u>-0.98: 0.99:0.135</u>	-1.39: 3.58:0.000
4.63: 2.48:0.000	-1.52: 2.77:0.001	<u>1.02:-1.02:0.135</u>	<u>-1.01: 0.99:0.135</u>
-3.25:-4.57:0.000	<u>-1.03: 1.01:0.135</u>	<u>-1.02:-0.99:0.135</u>	-4.05:-2.60:0.000
-2.40: 1.65:0.003	<u>-1.01: 1.00:0.135</u>	2.54:-0.48:0.002	3.78: 3.97:0.000
<u>0.99:-1.01:0.135</u>	-0.35: 1.43:0.028	<u>-0.99:-0.97:0.135</u>	<u>1.00:-0.99:0.135</u>
1.12:-1.92:0.033	1.25:-1.04:0.120	<u>1.03: 1.02:0.135</u>	-3.16: 2.55:0.000
-4.49: 1.12:0.000	<u>-1.01:-0.99:0.135</u>	<u>-0.98: 1.00:0.135</u>	<u>1.03:-1.00:0.135</u>
4.80:-1.49:0.000	0.55: 3.73:0.000	<u>-1.02:-1.00:0.135</u>	-2.14: 3.13:0.000

3) 함수 7(f_7)

이 함수는 $(-5,5) \times (-5,5)$ 의 범위에서 4개의 전역최적해 A, B, C, D를 가지며, f_2 와 유사하게 최적해들이 정의역상에 비대칭적으로 분포하는 경우이다. 전역최적해는 각각 (3.00, 2.00), (-2.81, 3.13), (-3.78, -3.28), (3.58, -1.85)이고, 이때의 함수값은 모두 0 이다.

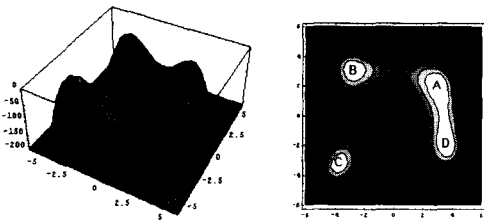


그림 10 $f(x, y) = -(x^2 + y - 11)^2 + (x + y^2 - 7)^2$

모집단의 크기를 36으로 했을 때의 실험결과는 아래와 같다. 2개의 전역최적해를 찾았으며, 오차가 0.05보다 작은 개체는 17개로 전체 모집단의 약 50%이고, 각 4분 면에 각각 4, 5, 3, 5개씩 분포함을 알 수 있다. 또한, 여기서는 개체와 적합도를 나열하지는 않았지만 모집단의 크기를 100으로 하여 4개의 전역최적해를 모두 찾을 수 있었다.

결론적으로 실험에 사용한 2차 목적함수에 대하여 제안한 LCGA가 여러 개의 전역최적해를 찾아낸다는 것을 알 수 있다.

4.3 30비트, 6차 바이폴라-디셀티브 문제에 대한 실험 및 분석

이 문제는 Goldberg, Deb 및 Horn 등이 제안한 것으로 탐색공간의 크기는 2^{30} 이고, 5,153,632개의 최적해를 가지며, 그 중에서 32개만이 전역최적해인 다루기 어려운 문제이다[19]. 이 문제는 6비트가 모두 0(000000)이거나 모두 1(111111)일 때 적합도 값 1을 갖는 5개의 부분함수로 구성된다. 따라서 최적해는 000000 000000

000000 000000 000000, 000000 000000 000000 000000 111111, ... , 111111 111111 111111 111111 111111 111111 32개이며 적합도 값은 모두 5이다.

다음 표 1은 제안한 LCGA에 이 문제를 적용한 실험 결과를 나타낸다. 실험 환경은 앞과 동일하며, 모집단의 크기와 세대수를 달리해가며 실험하였다. 또한, 개체의 적합도가 상대적으로 작은 경우(LCGA 절차 4의 ②)에는 교차 후 돌연변이 연산을 적용하였다. 이는 단순 유전자 알고리즘에서와 마찬가지로 우수 개체의 유전자를 물려받은 후, 약간의 변형을 가하기 위한 전략을 바탕으로 한 것이다. 즉, 이웃에 비하여 아주 열등한 경우 일방적으로 주위의 우수한 개체의 유전자를 그대로 가져옴으로서 새로운 유전자의 유입이 없다는 점을 고려한 것이다. 교차 및 돌연변이 확률은 각각 0.02와 0.01을 사용하였다.

실험 결과 모집단의 크기가 약 5000인 경우 10000세대 이내에 32개의 최적해를 모두 찾는다는 것을 알 수 있다. 그러나, 모집단이 1600을 넘는 경우들에 대하여 세대수를 크게 증가시켜도 발견한 최적해의 개수가 그다지 늘어나지 않는다는 것을 알 수 있다. 물론 그 이유는 교차 및 돌연변이 확률을 작게 한 때문일 것이다. 또한, 모집단의 크기가 커지면서, 수행 종료 후 모집단내에 포함되어있는 최적해의 개수(B)와 발견한 최적해의 개수(A)의 비율이 작아진다는 점도 주목할 만하다. 모집단의 크기가 커지면서 이 비율이 작아지는 것은 당연한 현상이기는 하지만, 중복이 심하다는 것은 탐색을 비효율적으로 만드는 원인이 된다. 따라서, 여러 개의 최적해가 존재하며 그것을 동시에 찾는 경우를 다루기 위해서, 모집단내 개체들이 중복되지 않도록 하는 절차를 개발해야할 것이다.

4.4 종합적인 분석

여기서는 실험에 사용한 문제들에 대하여 기존의 연구 결과들과 비교하여 제안한 LCGA의 장·단점과 특성을 분석한다. 여러 가지 최적해를 동시에 찾는 알고리

2.69: 0.86:-21.251	-2.38:-3.49:-85.799	-2.31:2.93:-7.935	2.97: 1.90:-0.235
-3.69:-0.23:-118.890	3.60:-1.83:-0.010	-2.78: 3.15:-0.040	3.02: 1.94:-0.057
3.59:-1.87:-0.010	0.52: 4.26:-178.426	<u>3.00: 2.00:-0.001</u>	3.01: 1.97:-0.013
-3.60:-4.52:-103.956	-1.38: 1.36:-102.570	-1.32:-4.78:-408.545	<u>-2.81: 3.13:-0.001</u>
1.75: 4.00:-131.510	3.57:-1.86:-0.014	0.19:-2.48:-181.208	-3.79:-3.26:-0.049
3.58:-1.87:-0.008	-2.81: 3.14:-0.007	-2.80: 3.11:-0.013	-0.66:-0.38:-176.354
-3.77:-3.25:-0.046	-4.63: 1.22:-237.967	3.60:-1.86:-0.022	-4.35: 3.39:-127.007
-2.78: 3.14:-0.020	2.99: 2.05:-0.046	-1.87:-1.87:-116.495	3.62:-1.87:-0.072
-3.75:-3.27:-0.040	3.01: 2.02:-0.012	0.21:-3.55:-243.944	-0.54:-0.81:-180.069

표 1 30비트, 6차 바이폴라-디셉티브 문제에 대한 실험 결과

모집단의 크기	세대수	발견한 최적해의 개수(A)	종료시 모집단내 최적해의 개수(B)	찾은 최적해의 비율 (A/32*100%)	A/B(%)
400	30000	7	14	21.9	50
	20000	5	6	15.6	83.3
	10000	0	0	0	-
900	30000	12	52	37.5	23.1
	20000	12	51	37.5	23.5
	10000	10	27	31.3	37.0
1600	30000	18	83	56.3	21.7
	20000	18	83	56.3	21.7
	10000	18	82	56.3	22.0
2500	30000	23	142	71.9	16.2
	20000	23	141	71.9	16.3
	10000	23	139	71.9	16.5
3600	30000	28	274	87.5	10.2
	20000	28	259	87.5	10.8
	10000	28	236	87.5	11.9
4900	30000	32	306	100	10.5
	20000	32	299	100	10.7
	10000	32	290	100	11.0

즘은 일반적으로 1) 모집단내에 전역최적해를 유지해야 하며 동시에 2) 모집단내에 지역최적해들도 유지해야 하는 것으로 알려져 있다[9].

먼저 밀집(crowding)에 기반한 방법의 경우, De Jong의 방법을 단독으로 사용하였을 때, 함수 f_1 , f_2 에 대하여 두 개의 최적해만을 찾는 것으로 알려졌다[9,11]. 이후 De Jong, Goldberg, Harik, Mahfoud 등에 의하여 5개의 최적해를 모두 찾을 수 있도록 개선된 방법들이 제안되었다. 그러나, 이 방식들도 세대수가 커지면서 평균 적합도는 커지지만 전역최적해를 유지하지 못하는 등의 문제점을 지니고 있으며, 결정적 밀집 알고리즘이 가장 안정적인 것으로 연구되었다[11]. 30비트, 6차 바이폴라-디셉티브(이하 디셉티브 문제로 표기)의 경우 결정적 밀집 알고리즘으로 해결되었다. Harik 역시 밀집에 기반한 RTS(restricted tournament selection) 알고리즘을 이용하여 이 문제의 최적해를 동시에 구한바 있다[2, 3].

다음으로 공유에 기반한 방법의 경우, Goldberg 등의 연구결과 f_1 는 5개의 최적해를 모두 찾을 수 있으며, f_2 는 적합도에 비례하여 찾을 수 있으나, 적합도의 차이가 큰 유사한 함수에 대하여는 지역최적해를 찾지 못하는

것으로 연구되었다[12]. Spears는 함수 f_1 , f_2 에 공유에 기반한 제한된 짝짓기를 적용하여 적합도가 낮은 1,2개를 제외하고 최적해를 찾을 수 있었으며, 6개의 최적해를 갖는 두 변수 함수의 경우 5개의 최적해를 찾을 수 있었다[13]. 공유에 기반한 방법들의 경우 일반적으로 디셉티브 문제에 매우 약하다. 이는 공유에 기반한 방법들이 자신과 유전형 혹은 표현형에 있어서 유사한(지역 최적인 디셉티브)개체들과 적합도를 공유하기 때문이다. Goldberg 등은 30비트, 6차 바이폴라-디셉티브 문제에 대하여, 단순 공유방법을 사용할 경우 모집단의 크기가 3,500,000이하일 때, 단 한 개의 최적해도 찾을 수 없을 수 있다는 것을 이론적으로 보였으며, 지수 적합도 축척(exponential fitness scaling)을 적용하고서야 32개의 최적해를 찾을 수 있었다[9,19].

과거의 연구들과 비교하였을 때, 제안한 LCGA의 단점은 두 가지로 요약된다. 첫째는 함수 f_2 , f_4 에 대한 실험결과에서 알 수 있듯이 지역최적해를 모집단내에 유지하지 못한다는 것이다. 기존의 방법들도 일부는 이러한 경향을 가지고 있지만 LCGA의 경우, 모집단의 크기가 작을 때, 이웃 개체들에 비하여 높은 적합도를 갖는 것은 지속적으로 살아남으며, 장기적으로 그 개체의 유전형이 모집단내

에 빠르게 퍼지기 때문에 세대수가 증가된 후에는 낮은 적합도를 갖는 개체들이 살아남기 어렵기 때문에 더욱 심할 것으로 보인다. 이러한 문제점은 확률, 불쓰만 분포 등을 적용하여 개선할 수 있을 것으로 보여진다. 두 번째는 모집단의 크기가 큰 경우에, 전역적인 정보를 사용하지 않기 때문에, 멀리 떨어진 지역에서 불필요한 중복 진화가 일어날 수 있다는 것이다. 이러한 문제점은 높은 적합도를 갖는 개체들의 유사성을 파악하여 상이한 진화를 유도하는 방법 등으로 개선해야할 과제이다.

LCGA가 갖는 장점과 특징은 세 가지로 요약할 수 있다. 첫째는 LCGA가 단순하다는 것이다. 즉, 전역최적해를 찾는다는 점에서 기존의 방법들에 뒤지지 않으면서, 문제에 종속되는 새로운 매개변수나 추가의 계산비용이 없다. 기존의 방법들은, 세대 간격(generation gap), 밀집 인자(crowding factor), 유사성, 공유적합도, 공유거리, 개체간 거리등을 정하거나 계산해야한다. 다음 표 2는 단순 GA와 LCGA의 수행시간을 비교한 것이다. 실험에 사용한 함수는 두 변수함수 f_6 이고 10회 반복한 평균이며, 일반적인 환경은 앞과 같다.

표 2 수행시간(초) 비교

모집단 크기	세대수	단순GA	LCGA
50	1000	2	1
	5000	8	6
	10000	15	12
100	1000	4	3
	5000	15	13
	10000	29	24

즉, 수행시간에 있어서 단순 GA보다 오래 걸리지 않는다는 것을 알 수 있다. 엄밀한 효율의 평가를 위해서는 단순 GA가 아니라 다른 여러 가지 최적해를 찾는 방법들에 대하여 동일한 정밀도의 해를 구하는데 필요한 적합도 평가 횟수 등을 계산해야할 것이다. 하지만 기존의 방법들 역시 매우 다양한 변형들이 존재하기 때문에, 그러한 비교는 단순히 몇 가지 프로그램을 작성한 후 비교해서는 안될 것이나, 효율에 대한 후속 연구가 따라야할 것이다. 두 번째 장점은 LCGA 자체가 병렬처리에 적합하다는 것이다. LCGA에서 개체들은 지역적으로 경쟁하기 때문에 멀리 떨어진 개체들의 적합도를 필요로 하지 않는다. 따라서 병렬로 구현하는 경우, 인접한 부분 모집단내 경계상의 개체들만 적합도 정보를 교환하면서도 전역적인 유전자 전파가 가능하다. 결국 통신비용, 이주(migration) 등을 고려하지 않고 쉽게

병렬로 구현할 수 있다.

세 번째, 기존의 방법들이 전역적인 정보를 사용하여 지역적인 진화를 유도하는 방식인 반면에, LCGA는 지역적인 정보를 사용하여 전역적인 진화를 유도한다는 특징을 갖는다. LCGA에서 개체들은 이웃의 개체들과 상호 작용한다. 실제 생태계에 있어서 지역적으로 멀리 떨어진 개체들 사이의 상호작용을 기대하기는 어렵다. 따라서, LCGA는 보다 자연스럽게 진화를 모방한다고 볼 수 있다. 물론 효율적인 최적화라는 측면에서 이러한 특징이 항상 바람직하다고 볼 수는 없다. 하지만 이러한 방식이 진화에 기반한 최적화 과정을 보다 잘 이해하는 수단으로 사용될 수 있을 것이며, 더 효과적인 기법을 개발하는 수단으로도 사용될 수 있을 것이다.

5. 결론

이 논문에서는 모집단 내 개체들을 원환체형의 격자 공간에 거주시켜 지리적인 이웃의 개념을 부여하고, 각 개체들이 지역적으로 경쟁하면서도 전역적으로 유전자를 교환할 수 있도록 하여, 모집단이 여러 개의 최적해를 포함하도록 하는 유전자 알고리즘을 제안하고 평가하였다. 개체들이 지역적으로 경쟁하되 각 개체들이 유전자를 전역적으로 전파할 수 있도록 하기 위하여 이 논문에서는 가능한 여러 가지 방법 중에 "생명게임(The Game of Life)"에 사용된 규칙을 응용한 진화 및 유전 연산자를 사용하였다. 제안한 알고리즘을 여러 개의 최적해를 갖는 4개의 1차 목적함수, 3개의 2차 목적함수 및 30비트, 6차 바이폴라-디셉티브 함수(bipolar-deceptive function)와 같은 다양한 문제들에 적용하여 성능을 평가하였다.

평가 결과는 다음과 같이 요약된다.

첫째, LCGA가 단순하다는 것이다. 즉, 전역최적해를 찾는다는 점에서 기존의 방법들에 뒤지지 않으면서, 문제에 종속되는 새로운 매개변수나 추가의 계산비용이 없다.

둘째, LCGA를 병렬로 구현하는 경우 인접한 부분 모집단내 경계상의 개체들만 적합도 정보를 교환하면 되므로 통신비용, 이주(migration) 등을 고려하지 않고 쉽게 병렬로 구현할 수 있다.

셋째, 기존의 방법들이 전역적인 정보를 사용하여 지역적인 진화를 유도하는 방식인 반면에, LCGA는 지역적인 정보를 사용하여 전역적인 진화를 유도한다는 특징을 갖는다. 따라서, LCGA는 보다 자연스럽게 진화를 모방한다고 볼 수 있다.

넷째, LCGA는 모집단의 크기가 작을 때, 이웃 개체들에 비하여 높은 적합도를 갖는 것이 지속적으로 살아

남으며, 장기적으로 그 개체의 유전형이 모집단내에 빠르게 퍼지기 때문에 지역최적해를 모집단내에 유지하지 못한다는 것이다. 이러한 문제점은 확률, 볼츠만 분포 등을 적용하여 개선할 수 있을 것으로 보여진다.

다섯째, 모집단의 크기가 큰 경우에, 전역적인 정보를 사용하지 않기 때문에, 멀리 떨어진 지역에서 불필요한 중복 진화가 일어날 수 있다는 것이다. 이러한 문제점은 높은 적합도를 갖는 개체들의 유사성을 과약하여 상이한 진화를 유도하는 방법 등으로 개선해야 할 것이다.

추후 연구과제는 분석의 결과를 토대로 LCGA의 장점과 특징을 최대한으로 유지하면서 문제점을 해결하는 방안을 개발해야 하는 것이며, 기존의 방식들과 엄밀한 비교 평가 역시 뒤따라야 할 것이다. 이러한 과정에서 LCGA의 동작을 정형적으로 다루는 수단 역시 개발해야 한다.

제안한 LCGA는 지역적으로 적용되는 단순한 규칙에 의한 전역적인 현상의 창발(emergence)이라는 원리를 이용함으로써, 개체사이에 이웃이라는 개념을 부여하여 다수의 해를 동시에 찾는다는 본래의 목적을 넘어서, 다양한 변형 유전자 알고리즘에 대한 새로운 모델이 될 것으로 기대된다.

참 고 문 헌

- [1] Z. Michalewicz, "Genetic Algorithms+Data Structures=Evolutionary Programs," Springer, pp. 169-172, 1999.
- [2] Georges Harik, Finding Multiple Solutions In Problems Of Bounded Difficulty, IlliGAL Report No. 94002, University of Illinois, The Illinois Genetic Algorithms Laboratory, 1994.
- [3] Georges Harik, Finding Multimodal Solutions Using Restricted Tournament Selection, University of Illinois, The Illinois Genetic Algorithms Laboratory, 1994.
- [4] Poundstone, *The Recursive Universe*, New York: Morrow, 1985.
- [5] Langton CG, *Artificial Life*, Addison-Wesley, 1989.
- [6] Gardner Martin, The fantastic combinations of John Conway's new solitaire game "life," *Scientific American*, pp. 120-123, October 1970.
- [7] Wolfram, S., *Cellular automata and complexity*, Reading, Mass. Addison-Wesley, 1994.
- [8] Beasley, et. all, A sequential Niche Technique for Multimodal Function Optimization, *Evolutionary Computation*, Vol.1, No.2, pp. 101-125, 1993.
- [9] Mahfoud, Niching Methods for Genetic Algorithms, IlliGAL Report No. 95001, University of Illinois, The Illinois Genetic Algorithms

Laboratory, 1995.

- [10] De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, (Doctoral dissertation, University of Michigan), Dissertation Abstract International, 36(10), 5140B, (University Microfilms No 76-9381), 1975.
- [11] Mahfoud, A Comparison of Parallel and Sequential Niching Methods, Proc. Sixth International Conference on Genetic Algorithms, pp. 136-143, 1995.
- [12] Goldberg, Richardson, *Genetic Algorithms with Sharing for Multimodal Function Optimization*, Proc. 2nd International Conference on Genetic Algorithms, pp. 41-49, 1987.
- [13] Spears, Simple Subpopulation Schemes, Proc. 3rd Annual Conference on Evolutionary Programming, San Diego, CA, World Scientific, pp. 367-384, 1994.
- [14] Erick Cantu-Paz, A Summary of Research on Parallel Genetic Algorithms, IlliGAL Report No. 95007, University of Illinois, The Illinois Genetic Algorithms Laboratory July 1995.
- [15] Prasanna V. Parthasarathy, David E. Goldberg and Scott A. Burns, Tackling Multimodal Problems in Hybrid Genetic Algorithms, University of Illinois, The Illinois Genetic Algorithms Laboratory, 2001.
- [16] Hancock, Selection Methods for Evolutionary Algorithms, in *Practical Handbook of Genetic Algorithms*, CRC Press, Vol. 2, pp. 67-92, 1995.
- [17] Pawlowsky, Crossover Operator, in *Practical Handbook of Genetic Algorithms*, CRC Press, Vol. 1, pp. 101-114, 1995.
- [18] Paul Rendell, Turing Machine implemented in Conway's Game Life, <http://www.rendell.uk.co/gol/tm.htm>, April 2000.
- [19] David E. Goldberg, Kalyanmoy Deb, Jeffrey Horn, Massive Multimodality, Deception, and Genetic Algorithms, *Parallel Problem Solving from Nature*, 2, North-Holland, pp. 37-46, 1992.



강 태 원

1985년 2월 연세대학교 수학과 이학사.
1988년 2월 고려대학교 전산과학과 이학사.
1991년 2월 고려대학교 대학원 수학과 이학석사.
1996년 8월 고려대학교 대학원 전산과학과 이학박사.
1997년 3월 ~ 현재 국립강릉대학교 컴퓨터공학과 교수. 번역서 "컴퓨터, 장자의 꿈" 등. 관심분야는 인공생명, 인공지능, 신경망, 진화 알고리즘, 퍼지이론, 혼돈이론, 프랙탈, 컴퓨터 게임