

유전 알고리즘

문병로

서울대 전산학과

1. 서론

유전 알고리즘은 집단 유전학의 개체 진화 원리를 문제 풀이에 이용하는 진화 연산의 한 갈래로 약 40여 년의 역사를 갖고 있다. 유전 알고리즘은 1960년 경부터 연구되기 시작했고 90년대 들어 폭발적이라 할만한 양적 팽창을 보이고 있다.

유전 알고리즘 연구는 60, 70년대에는 다소 모호한 경향으로 컴퓨터 과학 분야에서 소외를 받아 80년대 이후에야 제대로 평가를 받기 시작했다. 이렇게 된 데는 과학은 환원주의적이고 결정론적이어야 한다는 뉴우튼적 세계관의 영향과, 이 분야의 이론적 대부격인 John Holland가 자신의 학문을 대외적으로 전파하는 데 별 관심이 없는 스타일이어서 몇몇 단순 응용 연구자들이 설익은 논문들을 발표하면서 이 분야에 대한 부정적인 이미지를 구축하였다는 사실이 큰 이유가 되었던 것으로 보인다. 1975년에 홀랜드가 유전 알고리즘을 이론적으로 집대성한 저서 「Adaptation in Natural and Artificial Systems」 [9] 를 발표하면서 이 분야가 이론적 기반이 제대로 알려지기 시작했고 이로부터 유전 알고리즘 전문 학술대회가 열리는데 10 년이 걸렸다 (International Conference on Genetic Algorithms). 재미있게도 Holland는 미국 최초의 컴퓨터 과학 분야의 박사 학위 취득자이다.

문제의 복잡성에 대한 연구가 발전하면서 결정론적인 (deterministic) 알고리즘으로 해결할 수 있는 문제가 아주 제한되어 있다는 사실이 컴퓨터 과학자들에게 널리 알려진 것도 유전 알고리즘과 같은 통계적인 (stochastic) 알고리즘에 대한 관심을 고조시켰다. 80년대 이후 몇몇 석학들이 적응적 복잡계에 대한 연구를 시작한 것도 유전 알고리즘의 발전을 가속시켰다. 예를 들어 노벨 경제학상을 받은 스탠포드 대학의 Kenneth Arrow는 경제학을 적응적 시스템상에서 모델링하기 시작했고, 환원주의적 과학의 극치인 쿼크 이론을 정립하여 노벨 물리학상을 받은 캘리포니아 공대의 Murray Gell-Mann은 80년대 중반 이후 그의 행적과 배치되는 적응적 복잡계 연구로 방향을 거의 선회하였다. 이러한 움직임은 산타페 연구소를 중심으로 진행되고 있고 그 가운데는 John Holland가 선구자적인 위치에 우뚝 서 있다. 1997년에는 전기전자, 컴퓨터 분야의 대표적 학술지인 IEEE 트랜잭션에 유전 알고리즘으로 대표되는 진화 연산 (Evolutionary Computation)을 위한 *IEEE Transactions on Evolutionary Computation*이 개설되면서 이 분야의 역사에 한 획을 그었다 (참고로 반도체 CAD에 관한 트랜잭션이 생긴 것은 1982년, 신경망에 관한 트랜잭션이 생긴 것은 1990년이다).

2. 기초 사항들

2.1 문제 공간 탐색

모든 프로그램에 대해서 해당 프로그램이 궁극적으로 정지하는지를 판단해줄 수 있는 프로그램은 존재할 수 없다.

이것이 그 유명한 정지 문제 (Halting Problem)이다. 이것은 아주 간단해 보이지만 우리의 논리 체계 내에서는 그러한 프로그램을 가질 수 없는 것이다. 우리가 접하는 문제들은 크게 풀 수 있는 문제와 아무리 시간을 써도 풀 수 없는 문제가 있다. 위의 정지 문제와 같은 것이 풀 수 없는 문제의 한 예이고, 풀 수 있는 문제들은 현실적인 시간 내에 풀 수 있는 문제(tractable problems)와 그렇지 않은 문제(intractable problems)로 나뉜다. 예를 들어 풀 수는 있으나 현존하는 가장 빠른 컴퓨터로 1 조년이 소요된다면 누구도 시도하려 하지 않을 것이다. 현실적인 시간 내에 풀 수 있는 문제들은 문제의 크기를 n 이라 할 때 소요 시간이 n^k (k 는 상수)의 증가율을 넘지 않는 문제들을 말한다. 예를 들어 전형적인 $n \times n$ 행렬의 곱셈은 n^3 의 증가율을 가지며, n 개의 수를 크기순으로 정렬하는 것은 일반적으로 $n \log n$ 의 증가율을 가진다. 워드 프로세서에서 n 개의 단어로 이루어진 한 문단의 모양을 가장 예쁘게 만들도록 단어들의 위치를 선정하는 작업은 n 의 증가율을 가진다. 현실적인 시간 내에 최적의 답을 구할 수 없는 문제들은 현실적인 시간 내에 최적의 답에 근사하는 답을 구하는 것이 유일한 해결책인데 이를 근사 또는 approximation이라 한다. 유전 알고리즘은 이와 같이 현실적인 시간 내에 최적의 답을 구하기 힘든 경우에 품질 좋은 근사해를 구하는 효율적인 접근법들 중의 하나이다.

어떤 문제에 수많은 해가 있다면 (물론 해들의 품질/매력은 제각각이겠지만) 이들을 수학적으로 또는 알고리즘이 이해할 수 있는 형태로 표현해 놓고 나면 각각의 해는 n 차원 공간에서의 ($n \geq 1$) 점 하나에 해당한다. 이러한 해들의 집합으로 이루어지는 공간을 문제 공간 (problem space)이라 한다. 수많은 해들 중에 최적의 해를 구하는 (또는 최적에 근사한 해를 구하는) 과정은 n 차원 문제 공간에서의 탐색으로 비유할 수 있다. 우리가 행하는 대부분의 문제 풀이 과정, 유전 정보의 진화 등은 문제 공간 탐색 과정의 좋은 예들이다. 임의성은 떨어지지만 인간의 사고 과정, 이익 집단의 형성 등도 이의 예들로 볼 수 있을 것이다. 근사 알고리즘이란 이 문제 공간이 매우 복잡하고 방대해서 현실적인 시간 내에 풀 수 없는 문제에 대하여 주어진 시간에 최대한 효율적으로 이 공간의 부분 집합을 탐색하여 근사해를 찾아내는 알고리즘이다. 예를 들어 전체 문제 공간의 극히 일부분만 (예를 들어 수 조분의 일 미만) 탐색한 다음 최적치 또는 최적치에 아주 가까운 해를 찾을 수 있다면 매우 매력적인 알고리즘이 될 것이다.

2.2 유전 알고리즘

유전 알고리즘은 집단 유전학의 진화 원리를 이용하여 문제를 푸는 방법론들을 총칭하는 진화 연산 (evolutionary computation)의 대표적인 한 갈래이다. 어떤 문제가 주어지면 유전 알고리즘은 우선 해답들을 크로모솨름으로 표현하는 방법을 찾는다. 일단 표현이 되고 나면 임의로 일정 수(예를 들면 100 개)의 크로모솨름(임의로 만들어지므로 매우 매력이 떨어지는 해답들이 될 것이다)을 생성한다. 이 일군의 크로모솨름들로부터 교차와 돌연변이를 통해 새로운 크로모솨름들을 만들어 낸 다음 기존의 크로모솨름들을 대체해 간다. 매력의 정도가 높은 크로모솨름들은 그렇지 않은 것들에 비해 교차에 참여할 기회를 많이 갖지만 열등한 크로모솨름들도 기회를 전혀 박탈당하지는 않는다. 이러한 간단한 과정의 반복을 통해 크로모솨름들의 집합은 계속 평균적 매력이 높아지면서 유사한 크로모솨름들로 수렴하는 경향을 갖는다. 어느 정도 시간이 흐르면 크로모솨름들이 유사한 모양으로 수렴하여 더 이상 개선이 힘든 상태가 되는데 그 쯤에서 가장 매력적인 크로모솨름을 해답으로 제시하게 된다. 위의 과정은 유전 알고리즘을 아주 간략하게 설명한 것이다. 그림 1은 유전 알고리즘의 간단한 한 예를 들어본 것이다. 그림에서 직사각형으로 나타낸 것은 각각 하나의 해를 의미한다. 타원 속의 해들의 집합은 교차와 돌연변이를 통해 새로운 해를 만들어내는데 이러한 해들을 한 개 이상 만들어낸 다음 기존의 해들을 대체해가는 과정을 그림으로 나타낸 것이다. 교차는 그림 중 두 해를 결합해서 하나의 새로운 해를 만들어내는 과정을 일컫는데 각각의 해를 임의의 경계선을 중심으로 두 부분으로 나눈 다음 이들을 부분적으로 결합하여 새로운 해를 만들어낸다. 그림과 같이 한 점을 이용하여 교차하는 것을 일점 교차(single-point crossover)라 하고 복수 개의 점을 이용하는 교차를 다점 교차(multi-point crossover)라 한다. 그림 2는 유전 알고리즘의 전형적 구조를 알고리즘 형태로 보여준다.

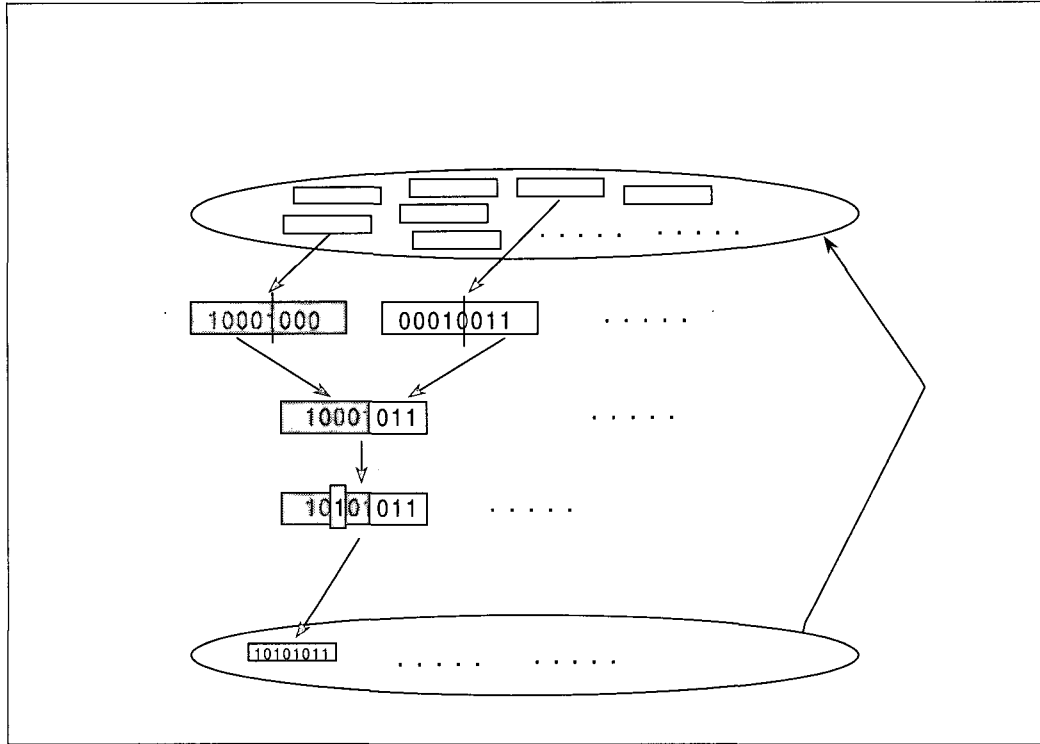


그림 1 : 유전 알고리즘 동작의 한 예

앞에서 보인 예와 같이, 많은 경우 유전 알고리즘의 크로모솜은 일차원적인 띠(string)의 형태로 표현되지만 이 외에도 다양한 표현 방법들이 있다. 예를 들어 2 차원 행렬을 크로모솜으로 쓰기도 하고, 트리(tree)라고 하는 그래프 구조를 그대로 크로모솜으로 쓰기도 한다. 크로모솜의 표현 방법이 결정되면 그에 맞는 교차 방법이 제공되어야 한다.

```

create initial population (with a given # of initial solutions);
repeat
    for  $i = 1$  to  $k$  {
        select two solutions,  $s_1$  and  $s_2$ , from the population;
        offspring $_i$  = crossover( $s_1$ ,  $s_2$ );
        offspring $_i$  = mutation(offspring $_i$ );
    }
    replace a subset of the population with offspring $_1$ , ..., offspring $_k$ ;
until (stopping criterion satisfies);
return the best solution in the population;
    
```

그림 2 : 유전 알고리즘의 전형적 구조

진화의 원리를 이용하여 문제를 해결하는 접근법들을 통칭해서 진화 연산이라고 하는데 유전 알고리즘, 유전 프로그래밍 (Genetic Programming), 진화 전략 (Evolutionary Strategy), 진화 프로그래밍 (Evolutionary Programming) 등 네 분야로 크게 나누어진다. 각각 개별적인 특징이 있으나 개략적으로 말해서 유사하다고 보면 되고 다만 유전 프로그래밍이 프로그램 자체를 진화시킨다는 점, 진화 프로그래밍만이 교배를 전혀 쓰지 않는다는 것 정도가 비교적 큰 차이점이라 볼 수 있다. 이들의 가장 주목할 공통적인 특징은 간단한 규칙으로부터 그보다 복잡한 규칙 또는 행동이 자발적으로 발생하는 “창발성(Emergence)”을 일으킬 수 있다는 것이다.

2.3 스키마 정리 (Schema Theorem)

크로모솜에는 수많은 소패턴들이 포함되어 있다. 길이 n 인 이진 크로모솜 하나에는 총 2^n 개의 소패턴들이 포함되어 있다. 이러한 소패턴들을 스키마라고 하는데, 다음은 길이 15 인 이진 크로모솜과 이에 포함된 스키마의 예를 보인다.

크로모솜의 한 예: 110100011010011
 위의 크로모솜에 포함된 스키마의 한 예: ***10001**100**

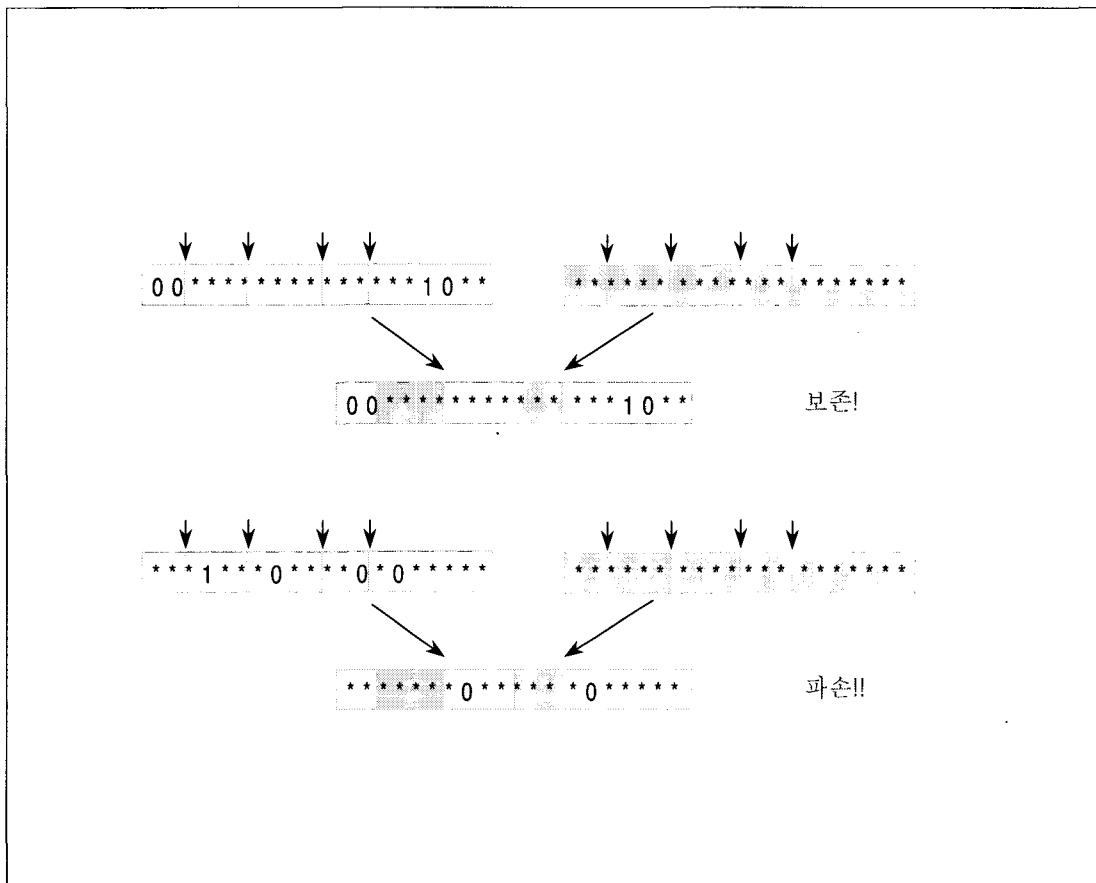


그림 3

스키마에서 *로 표시된 것은 패턴에 기여하지 않는 심볼들을 나타내고, 그 이외의 심볼들은 해당 스키마의 패턴을 규정하는데 쓰이므로 이를 유효 심볼(specific symbol)이라 한다. 유전 알고리즘은 초기의 크로모솜들에 포함되어 있던 작은 고품질 스키마들이 결합되어 점점 더 큰 고품질의 스키마로 만들어져 가는 과정이다. 일점 교차에서 스키마의 크기가 차세대에서 그 스키마를 가진 크로모솜들의 개수에 미치는 영향을 공식화한 것이 유명한 Holland의 스키마 정리[10]이고 이는 그 중요성으로 인해 유전 알고리즘의 기본 정리라 불리기도 한다. 이 정리의 핵심은 첫 번째 유효 심볼로부터 마지막 유효 심볼까지의 길이가 절대적인 영향을 미친다는 것이다. 이 길이를 스키마의 정의 길이(defining length)라 한다.

일점 교차의 경우는 스키마의 정의 길이가 생존에 절대적 영향을 미치나 다점 교차의 경우는 그렇지 않다. 그림 3은 4점 교차의 경우를 예로 보인다. 여기서는 스키마의 정의 길이보다 스키마의 유효 심볼들의 분포가 큰 영향을 미친다. 그림에서 첫 번째 스키마는 길이가 길지만 유효 심볼들이 균집된 분포를 하여 생존하기에 유리한 반면 두 번째 스키마는 유효 심볼들이 산만한 분포를 하여 생존이 상대적으로 어렵다. 이에 착안하여 [2,3,21]에서는 고품질 스키마들에서 유효 심볼들의 분포를 그래프 이론에 바탕하여 조작함으로써 상당한 유전 알고리즘의 성능 향상을 이루었다.

2.4 혼합형 유전 알고리즘

유전 알고리즘은 매우 넓은 문제 공간 탐색 기능을 갖고 있으나 국소 최적점 근처에서의 튜닝 기능이 약한 단점을 갖고 있다. 교차의 임의성으로 인하여 국소 최적점 근처의 두 점을 결합하여 국소 최적점에 이르는 데는 상당한 시간이 소요될 것이라는 것은 쉽게 상상할 수 있다. 이러한 단점을 개선한 것이 혼합형 유전 알고리즘이다. 혼합형 유전 알고리즘에서는 교차와 돌연변이로 생성된 크로모솜을 국소 최적화 알고리즘을 사용하여 국소 최적점으로 끌어 올려 (또는 내려) 준다. 이렇게 되면 유전 알고리즘은 국소 최적화 알고리즘을 위해 다양한 초기 해를 제공해 주는 역할을 하게 된다. 이 방법의 단점은 국소 최적화 알고리즘에 의해 해의 방향이 왜곡되어 이를 사용하지 않았을 때 나타날 수 있는 좋은 해에 이르지 못하게 될 수도 있다. 그러나 일반적으로 장난감 크기 이상의 문제에 대해서는 대부분 혼합형의 장점이 단점을 압도하는 경향이 있다. 대체로 수렴 시간을 단축하면서 더 좋은 해를 찾아낸다.

혼합형 유전 알고리즘에는 라마르크형과 볼드윈형이 있다. 라마르크형은 국소 최적화된 크로모솜을 그대로 사용하고 볼드윈형에서는 국소 최적화를 하되 최적화 직전 크로모솜의 품질을 평가하기 위해서만 사용된다. 일반적으로 라마르크형이 볼드윈형보다 빠른 수렴성을 가지며 더 많이 쓰이고 있다. 이처럼 유전 알고리즘에서는 생물학에서 폐기된 이론이라 하더라도 결과에 좋은 영향을 미친다면 채택한다. 궁극적인 목적이 진화나 생태계의 사실성있는 재현이 아니고 품질이 좋은 해를 찾는 것이 주목적이기 때문이다.

3. 관련 연구 및 응용 분야

3.1 함수 최적화 (Function Optimization)

함수 최적화는 임의의 함수가 주어지고 이 함수값을 최대화/최소화하는 변수값(들)을 찾는 문제이다. 예를 들면 $f = \sum_{i=1}^n -x_i \cdot \sin(\sqrt{|x_i|})$, $-500 \leq x_i \leq 500$ 에 대해 함수 f 를 최소화하는 x_i 값들을 찾는 문제와 같은 것이다. 함수 최적화는 유전 알고리즘 연구 분야에서 가장 오랜 역사를 가진 주제들 중의 하나이다 [11,5,8,29]. 유전 알고리즘을 이용한 함수 최적화를 다룬 최초의 연구자는 Hollstein이다 [11]. DeJong은 다양한 특성을 갖는 함수들을 고안하여 이

들을 이용하여 유전 알고리즘의 연산자들이 성능에 어떠한 영향을 미치는가를 연구하였다 [5]. 초기에는 새로운 아이디어들이 나오면 주로 함수 최적화 문제에 적용하여 그 효용성을 가늠하는 것이 일반적이었다. 미래에도 이 주제의 중요성은 감소하지 않을 것으로 보인다. 다만 지금까지의 연구들은 대부분 소수의 변수들만으로 이루어진 함수들을 다루었는데 점점 함수의 복잡도를 증가시켜 가는 과정에서 새로운 유전 알고리즘이 많이 출현할 것으로 예상된다.

3.2 함수 근사 (Function Approximation)

함수 최적화가 주어진 함수를 가지고 최적의 변수값(들)을 찾는 데 반해서 함수 근사는 변수값들의 집합과 이들을 이용한 측정(실험) 함수값들이 주어진다. 목적은 이 측정치들을 가장 잘 설명하는 함수 자체를 찾아내는 것이다. 함수 근사를 위한 유전 알고리즘은 91년 Rogers에 의해 발표된 이후 간헐적으로 발표되고 있으나 [24-26,18] 아직 초창기에 속한다.

전문적으로 표현하면 입출력 데이터 쌍들로부터 알려지지 않은 시스템의 수학적 모델을 찾아내는 것이다 [13]. 다른 말로 시스템 모델링이라고도 한다. 일반적으로 모델을 복잡하게 잡으면 보다 실험치를 더 잘 설명할 수 있을 개연성이 있지만 지나치게 복잡한 모델링은 실험에 포함되지 않은 데이터에 대한 적응력이 떨어지는 경향이 있는데 이를 오버피팅(overfitting)이라 한다. n 개의 변수를 사용해서 만들 수 있는 함수는 무한히 많으므로 실제 응용에서는 어느 정도 해당 분야 전문가의 직관이나 사전 지식을 사용하여 모델의 범위를 제한해 줄 필요가 있다. 회귀 분석에서도 이런 연구가 많이 이루어져 왔는데 다만 회귀 분석에서는 함수의 모양은 고정되고 파라미터(계수, 지수 등)들을 최적화하는데 반하여 진정한 함수 근사는 원칙적으로 함수 모양에 대한 사전 지식을 필요로 하지 않는다. 그러나 현재의 유전 알고리즘 기술로는 어느 정도는 함수 모양에 대한 제한은 주어야 가능한 수준이다. 원자력 공학, 화학 공학, 인공 지능 등에서 많이 발생하는 문제로 차후에 가능성이 많은 응용 분야이다.

3.3 그래프 최적화 (Graph Optimization)

그래프는 보통 $G = (V, E)$ 와 같이 나타내는데 여기서 V 는 점들의 집합이고 E 는 점들을 연결하는 선분들의 집합이다. 실세계에서의 많은 문제들이 그래프로 표현될 수 있고 그래프로 표현되는 문제의 최적해를 구하는 문제이다. 그래프 최적화는 컴퓨터를 이용하여 푸는 문제들 중 가장 광범위한 응용예를 갖고 있는 것 중의 하나이다. 가장 관심을 끄는 연구 분야는 NP-Hard[6]라 불리는 컴퓨터 과학의 어려운 문제군인데 광범위한 영역의 문제들을 포함하고 있다. n 개의 도시를 방문하고 원점으로 돌아오되 가장 짧은 여행 거리를 갖는 경로를 찾는 순회 세일즈맨 문제, 반도체 회로의 소자들을 최적의 위치에 배치하는 문제, 컴퓨터 네트워크에서의 최적 배치 문제, 복수 개 배달 트럭의 최적 스케줄링 문제, 그래프의 최적 분할 문제, 격자상에 모델링된 단백질 최적 형상 찾기 문제 등 한정된 공간에 나열하기 힘들 정도로 많다. 이 중 그래프와 반도체 회로의 분할에 있어서는 유전 알고리즘이 기존의 어떤 알고리즘보다 좋은 결과를 보유하고 있다 [2,3,21].

4. 몇 가지 연구 소주제 및 응용 예

본 단원에서는 생물학자들이 흥미를 가질만한 몇 가지 유전 알고리즘 및 인공 생명 관련 연구 주제들과 응용 예를 보인다. 아울러 아래에 소개하는 주제들은 아직은 유전 알고리즘 분야에서 주력하고 있는 연구 주제들은 아님을 밝혀 둔다. 크게 두 가지 관점의 연구자들이 있는데 하나는 진화 또는 발생의 원리를 이는데 목적이 있는 그룹이고 하나는 실제와의 일관성보다는 품질 좋은 해를 찾는 데 관심이 있는 그룹이다.

4.1 NK-Landscape

NK-Landscape는 Stuart Kauffman에 의해 제안된 유전자의 그물망 모델로 [16], 유전자들간의 연결 관계의 복잡도와 문제 공간 탐색의 특성에 관심을 갖고 만든 모델이다. 여기서 N 은 유전자의 총 개수이고 K 는 임의의 유전자가 연결을 갖고 있는 다른 유전자들의 개수를 나타낸다. 유전자 하나를 전구 하나로 가정하고 임의의 초기 상태에서부터 이 그물망이 전구들의 점멸에 의해 패턴을 바꾸어 가는 전구 쇼를 연상하면 될 것이다. 카우프만은 K 가 작을 때 주기적 끌개(periodic attractor)가 발생하는 것을 확인하고 생물체의 유전 정보도 이렇게 느슨한 그물망을 이룰 것이라고 주장한다. 이 끌개의 개념은 문제 공간 탐색에서 국소 최적점과 같은 의미를 갖는다. 카우프만은 $K=2$ 인 경우 이 끌개들의 총 수는 대략적으로 \sqrt{N} 에 비례하는 것을 관찰하였고 이것은 생물체들이 가지는 서로 다른 세포의 유형과 대략적으로 일치한다고 주장한다. 이것은 그래프 이론에서도 밀집하게 연결된 그래프보다는 느슨하게 연결된 그래프들이 복잡한 특성들을 많이 갖고 있는 것을 자연스럽게 연상시킨다.

이러한 관찰들로부터 우리는 관계가 있으면 필연적으로 끌개가 출현하기 마련이라는 “가설”을 세울 수 있다. 이러한 관계의 모델링은 다양한 시스템에서 가능한데, 끌개에 해당하는 것으로는 앞에서 이야기한 발생 시스템에서의 세포의 종류, 생태계에서의 종(species), 화학 시스템에서의 평형, 인류사에서의 국가나 이의 집단, 인간 사고 시스템에서 창출되는 개념 등을 들 수 있다. 이 모든 것은 문제 공간 탐색의 관점에서 볼 수 있으며, 다양한 끌개들은 해당 문제에서의 국소 최적점들에 해당한다. 국소 최적점들은 약한 탐색 기법으로는 좀처럼 벗어나기 힘든 유인력을 갖는다.

4.2 공진화

유전 알고리즘에서 공진화 모델의 대표적인 것으로는 Daniel Hillis의 작업을 들 수 있다 [9]. Hillis는 유전 알고리즘과 인공 생명 연구의 가장 강력한 도구가 된 커넥션 머신을 만든 사람이다 (커넥션 머신의 원형은 재미있게도 일찌기 Holland가 제안하였다).

Hillis는 커넥션 머신을 이용해서 정렬 네트워크를 찾아내는 문제에 도전하였는데 정렬 네트워크에 대한 소개는 지면상 생략한다. Hillis는 일반적으로 사용하는 population에 더하여 하나의 보조 population을 더 사용하였다. 주 population이 해들의 집합이고 보조 population은 훈련 데이터들의 부분 집합이다. 주 population에 있는 해들의 적합도는 보조 population에 있는 데이터들을 잘 만족시키느냐에 따라 결정된다. 이에 반하여 보조 population에 있는 훈련 데이터의 적합도는 얼마나 주 population에 있는 해들을 곤란하게 만드느냐에 따라 결정된다. 이런 대립 구조를 설정함으로써 두 population이 경쟁적으로 진화하는 것이 관찰되었고 이것은 진화를 괄목할 정도로 가속시켰다. 이것은 Richard Dawkins가 그의 「The Blind Watchmaker」에서 “진화적 군비 확장 경쟁 (evolutionary arms race)”이라고 부른 현상의 한 좋은 예이다. 공진화를 이용한 유전 알고리즘은 최근에 매우 활발한 연구가 이루어지고 있다 [9,27,22,1].

4.3 셀룰라 오토마타

셀룰라 오토마타는 다차원의 격자 모양의 유한자동기계(finite state automaton)로 n 개의 이웃 격자들의 상태에 따라 다음 상태가 결정되도록 만든 기계이다. 설명을 위해 가장 간단한 일차원 셀룰라 오토마타를 예로 들어 보자. 격자 N 개로 이루어진 일차원의 띠를 생각해 보자. 각 격자가 가질 수 있는 상태의 총 수는 K 라 하자. 임의의 격자는 n 개의 이웃 격자들의 상태를 참조한 다음 자신의 다음 상태를 결정한다. N 개의 격자 각각에 K 상태 중 하나씩을

초기치로 할당된 다음, 시간이 한 스텝씩 진행됨에 따라 어떻게 변하는지 관찰하는 것이다.

Steven Wolfram은 셀룰라 오토마타의 특성을 다음의 네 가지로 분류하였다 [28].

- Class I: 한 가지 상태로 수렴함
- Class II: 제한된 주기성을 보임
- Class III: 주기성없는 혼돈을 보임
- Class IV: 국소적 구조를 가진 복잡한 패턴을 보임

Christopher Langton은 일차원 셀룰라 오토마톤의 특성을 관찰하기 위해 λ 라는 매개 변수를 도입하였다 [17]. 임의의 상태 하나를 지정하여 모든 가능한 주변 상태의 조합 중 상태 q 로 가는 비율을 λ 라고 놓는다. 그의 일차원 셀룰라 오토마타에서 λ 를 변화시킴에 따라 Wolfram의 네 가지 클래스가 차례로 나타나는 것을 관찰하였다. 셀룰라 오토마타는 단순하지만 많은 응용 가치를 지니고 있다. 따져 보면 앞에서 다룬 Kauffman의 유전자 그물망도 일차원 셀룰라 오토마타의 한 예일 뿐임을 알 수 있을 것이다. 당연히 Kauffman도 그의 NK-Landscape 모델로부터 Wolfram의 클래스들을 발견할 수 있었다. 랭턴의 λ 변수에 의해 관찰된 것은 “복잡한” 패턴은 Class II와 가장 무질서한(chaotic) 패턴을 보이는 Class III의 사이에 있다는 사실이다. 이것이 최근에 유행어처럼 쓰이는 “질서와 혼돈의 가장자리”이다. 이것은 물리학에서의 두 상태간의 상전이 현상과 자주 비유되곤 한다. 다소 과감하기는 하지만, 진화적 균형 상태에 있는 생명체도 Class IV와 같은 상태에 있을 것이라고 Langton은 주장하기도 한다.

4.4 띠에라

Tom Ray는 가상적인 컴퓨터 상에서 컴퓨터 어셈블리 프로그램들이 서로 경쟁하는 시스템을 개발하였다 [23]. Tierra라고 명명된 이 시스템에서 개체(어셈블리 프로그램)는 복제의 기능을 가지고, 시스템은 개체의 나이가 들면 높은 확률로 죽도록 하는 정도의 인위적 제어만 한다. 다른 진화 기반의 기법들과는 달리 진화상에서 생존의 열쇠가 될 적합성 조차도 외부에서 미리 주지 않는다. 시스템은 프로그램이 너무 길면 증식하는데 불리하고 너무 짧으면 자기 복제에 필요한 코드를 포함하기 힘들어 생존에 불리하게 작용하도록 되어 있다. Tierra는 숙주와 기생충의 관계와 같이 자기 복제 능력은 없지만 숙주의 복제 코드를 빌어 사용할 수 있는 능력을 갖춘 아주 짧은 기생충을 만들어 내는데 성공하였고, 숙주는 기생충에 대한 면역 기능을 갖도록 진화되었다. Ray는 컴퓨터 과학자가 아니고 생태학자여서 인공 생명에 대해 생물학자들이 무관심한 또는 회의적인 태도로부터 적극적인 관심을 갖도록 하는 데 크게 기여하였다.

Tierra는 일정 기간의 “외형적” 평형 후에 오는 폭발적인 변화를 자주 보여 주었다. 이것은 진화에서 말하는 단속적 평형(punctuated equilibrium)과 일치한다. Daniel Hillis는 이보다 앞서 공진화를 사용한 유전 알고리즘을 이용하여 가장 효과적인 정렬 네트워크(sorting network)를 찾는 연구를 하는 과정에서 역시 이러한 현상을 관찰하였는데 외형적으로 안정된 시기에도 유전 정보들은 활발히 변하고 있음을 실험적으로 확인하였다. 적어도 진화의 컴퓨터 시뮬레이션만을 놓고 볼 때, 평형 상태는 진화적 폭발을 준비하고 있는 역동적 균형의 시기임이 밝혀진 셈이다. 컴퓨터 시뮬레이션은 언제라도 현상을 재현시킬 수 있고 그 내부를 원하는 각도에서 속속들이 들여다 볼 수 있다는 큰 장점을 갖고 있다.

4.5 기타

UCLA의 David Jefferson 팀은 존 뮤어 퀘적이라고 이름붙인 개미의 페로몬의 퀘적을 32×32 격자상에 흩어놓고

이 궤적을 가장 효율적으로 추적하는 프로그램(인공 개미)를 만들어 내는 작업에 유전 알고리즘을 이용하였다 [14]. 유전 알고리즘은 상상하기 힘들 정도로 지적인 전략을 가진 개미를 만들어 내었다. 인공 개미에 관한 연구는 이후에도 계속되어 John Koza는 유전 프로그래밍 기법을 이용해 존 뮤어 궤적과 같은 32×32 격자상에 좀 더 난해한 형태로 페로몬을 뿌려 놓은 산타페 궤적으로 실험을 하여 최적의 추적을 해내는 개미를 찾아내었다 [15]. 이러한 시도는 해당 자체를 찾아내는 것이 아니고 전략이나 프로그램 자체를 찾아낸다는 점에서 큰 의미가 있다. 미래에 인간이 만들기 힘든 아주 복잡한 프로그램을 자동 생산할 수 있는 첫 번째 도구가 될 가능성이 크기 때문이다 (현재에도 어느 정도 수준까지는 이미 찾아내고 있다).

Richard Dawkins는 한 점으로부터 시작하여 곤충과 유사한 인공물을 탄생시키는 시스템을 개발하였다 [4]. 이 인공물들은 돌연변이를 일으킬 수 있으며 선택에 의해 진화를 한다. 다만 여기서 선택은 사람에게 의해 이루어진다. 인공물의 모양은 9 개의 파라미터에 의해 결정되고 이 파라미터 값들의 집합을 진화시키는 것이 이 작업의 주 골격이다. Dawkins는 이러한 단순한 작업에 의해 전혀 예측하지 못했던 복잡한 모양의 인공물을 만들어 내는 것을 관찰하였다. 그는 이 인공물들을 바이오모프라고 명명하였다.

Lindenmayer는 일찍이 60년대 말에 촘스키의 문맥자유문법(context-free grammar)을 이용하여 식물의 성장 과정을 기술하는 시스템을 제안했는데 이를 린덴마이어 시스템 또는 줄여서 L-계(L-system)라고 한다 [19]. 최근에 이 시스템에 기초하여 진화 연산 기법을 이용하여 생장의 법칙을 진화시키는 시도가 이루어지고 있다 [12,20].

5. 결어

유전 알고리즘의 역사와 개요, 이의 응용 분야, 생물학과 관련된 약간의 주제들을 살펴 보았다. 선택한 주제들은 본 강연의 참가자들이 흥미를 느낄 만한 것으로 골라본 것일 뿐 이들의 대표성에 대한 판단은 유보하고 싶다. 유전 알고리즘의 목적은 크게 문제 풀이와 시뮬레이션으로 나눌 수도 있는데 아무래도 생물학자들은 시뮬레이션에 의한 정확한 진화 현상이나 생태계, 촉매 고리 등의 재현에 관심이 더 클 것으로 생각된다. 그러나 최적의 중 분류표를 만드는 일 등은 문제 풀이에 속하는 예가 될 것이다. 이에 반해 비생물학자들은 정확한 재현보다는 목적에 부합되는 좋은 해를 찾는 것이 주 관심사이다. 그래서 생물학에서 이미 폐기된 이론이라도 결과에 도움이 된다면 흔쾌히 사용한다.

고대로부터 근 2천년간 세계를 지배해 오던 아리스토텔레스의 의지론적 세계관이 18세기에 데카르트-뉴우튼으로 대표되는 결정론적인 세계관으로 대체된 후 200 년 이상 세계를 지배해 왔다. 이 결정론적인 세계관은 20세기 초 하이젠 베르크에 의해 깨진 이후 지금까지 새로운 세계관으로의 긴 이동 과정에 있다. 새로운 세계관의 중요한 특징중의 하나는 주체와 객체의 구별이 모호해진다는 것인데 유전 알고리즘을 비롯한 진화 연산은 궁극적으로 프로그램 스스로가 외부 프로그램의 개입없이 진화의 대상이 되는 방향으로 발전되어 이 새로운 세계관을 이끄는 한 축을 이루게 될 것으로 예상된다. 또한 21세기에는 보다 다양한 분야에서 적용적 모델이 정립될 것으로 예상된다. 생물학, 물리학, 컴퓨터 과학, 제 공학들, 경제학 등이 다양하게 결합하는 학제적 연구가 필요한 분야이기도 하다.

참고 문헌

1. H. Barbosa, "A Coevolutionary Genetic Algorithm for a Game Approach to Structural Optimization," *International Conference on Genetic Algorithms*, pp. 545-552, 1997.
2. T. Bui and B. Moon, "Genetic Algorithm and Graph Partitioning," *IEEE Transactions on computers*, 45(7): 841-855, 1996.
3. T. Bui and B. Moon, "GRCA: A Hybrid Genetic Algorithm for Circuit Ratio-Cut Partitoning," *IEEE Transactions on CAD*, 17(3): 193-204, 1998.
4. R. Dawkins, "The Evolution of Evolvability," *Artificial Life*, C. Langton (Ed.), pp. 201-220, 1989.
5. K. DeJong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," Doctoral dissertation, University of Michigan, 1975 (*Dissertation Abstract International* 36(10), 5140B).
6. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, 1979.
7. D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, 1989.
8. D. Goldberg, K. Deb, H. Kargupta, and G. Harik, "Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms," *International Conference on Genetic Algorithms*, pp. 56-64, 1993.
9. D. Hillis, "Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure," *Artificial Life II*, C. Langton et al. (Eds.), pp. 313-324, 1992.
10. J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
11. R. Hollstein, "Artificial Genetic Adaptation in Computer Control Systems," Doctoral dissertation, University of Michigan, 1971 (*Dissertation Abstract International* 32(3), 1510B).
12. C. Jacob, "Evolution Programs Evolved," *International Conference on Evolutionary Computation*, pp. 42-51, 1996.
13. J. Jang, C. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, 1997.
14. D. Jefferson et al, "Evolution as a Theme in Artificial Life: The Genesys/Tracker System," *Artificial Life II*, C. Langton et al. (Eds.), pp. 549-578, 1992.
15. J. Koza, "Genetic Evolution and Co-Evolution of Computer Programs," *Artificial Life II*, C. Langton et al. (Eds.), pp. 603-629, 1992.
16. S. Kauffman, "Adaptation on Rugged Fitness Landscapes," *Lectures in the Science of Complexity*, D. Stein (Ed.), pp. 527-618, 1989.
17. C. Langton, "Life at the Edge of Chaos," *Artificial Life II*, C. Langton et al. (Eds.), pp. 41-91, 1992.
18. R. Leardi, R. Boggia, and M. Terrile, "Genetic Algorithms as a Strategy for Feature Selection," *Journal of Chemometrics* 6, pp. 267-281, 1992.
19. A. Lindenmayer, "Mathematical Models for Cellular Interaction in Development, I & II," *Journal of Theoretical Biology* 18: 280-315, 1968.
20. K. Mock, "Wildwood: The Evolution of L-system Plants for Virtual Environments," *IEEE International Conference on Evolutionary Computation*, pp. 476-480, 1998.
21. B. Moon, Y. Lee and C. Kim, "GEORG: VLSI Circuit Partitioner with a New Genetic Algorithm Framework", *Journal of Intelligent Manufacturing*, Accepted for publication, 1998.
22. J. Paredis, "Coevolving Cellular Automata: Be Aware of the Red Queen!," *International Conference*

on *Genetic Algorithms*, pp. 393, 1997.

23. T. Ray, "An Approach to the Synthesis of Life," *Artificial Life II*, C. Langton et al. (Eds.), pp. 371-408, 1992.

24. D. Rogers, "G/SPLINES: A Hybrid of Friedman's Multivariate Adaptive Regression Splines (MARS) Algorithm with Holland's Genetic Algorithm," *International Conference on Genetic Algorithms*, pp. 383-391, 1991.

25. D. Rogers, "Development of the Genetic Function Approximation Algorithm," *International Conference on Genetic Algorithms*, pp. 589-596, 1995.

26. D. Rogers, "Evolutionary Statistics: Using a Genetic Algorithms and Model Reduction to Isolate Alternative Statistical Hypothesis of Experimental Data," *International Conference on Genetic Algorithms*, pp. 735-742, 1997.

27. C. Rosin and R. Belew, "Methods for Competitive Coevolution: Finding Opponents Worth Beating," *International Conference on Genetic Algorithms*, pp. 373-380, 1995.

28. S. Wolfram, "Universality and Complexity in Cellular Automata," *Reviews of Modern Physics* 55, pp. 601-644, 1983.

29. J. Yang, J. Horng, and C. Kao, "A Continuous Genetic Algorithm for Global Optimization," *International Conference on Genetic Algorithms*, pp. 230-237, 1997.