

유전자 알고리즘을 이용한 자가복제자와 하이퍼사이클의 구성

곽초화⁰, 위규범

한국정보과학회

{jminary⁰, kbwee}@ajou.ac.kr

Construction of Auto-replicators and Hyper-cycles Using Genetic Algorithms

Chohwa Gwac⁰, Kyubum Wee
Ajou University

요약

활자유전학(typogenetics)은 인공생명(artificial life) 연구에 사용되는 형식 시스템으로서, 자가복제자와 하이퍼사이클의 출현에 관한 연구에 효과적인 모델이다. 본 연구에서는 하이퍼사이클에 추가될 복제자의 차이점과 유사점을 측정하기 위하여 편집거리(edit distance)를 사용하여, 기존의 연구에서 생성된 하이퍼사이클보다 더 큰 크기의 다양한 하이퍼사이클들을 생성하였다.

1. 서론

생명의 기원에 대한 많은 가설이 존재한다. 이 중 영기의 우연한 합성으로 생명체가 시작되었다고 하는 가설이 있는데, 매우 낮은 확률에도 불구하고 생명체의 특성을 지닌 구조가 생성될 수 있음을 보이고자 하는 것이 활자유전학(typogenetics)의 목적 중 하나이다. 인공 생명(artificial life)의 주요한 연구 분야인 typogenetics는 Douglas Hofstadter에 의해 최초로 제안된 형식 체계이다[1]. 활자유전학은 인공생명 연구에 사용되는 형식 시스템으로서, 자가복제자와 하이퍼사이클의 출현에 관한 연구에 효과적인 모델이다[2,3].

자가복제자(auto-replicator)는 생명의 가장 큰 특징인 전파(propagation)를 모사(modeling)하였다[4]. 이것은 자신과 같은 개체를 생성해 내는 능력을 가지며, 상보적인 두 개의 문자열로 이루어져 있다. 각 문자열은 DNA의 영기를 의미하는 A, C, G, T 로 구성되어 있으며, 이 네 개의 영기를 두 개씩(doublet) 조합하여 명령어 기능을 하는 16개의 효소(enzyme)라고 정의한다. 자가복제자의 두 상보적 문자열 각각을 doublet 단위로 자르게 되면 일련의 효소들을 얻게 된다. 한 문자열에서 나온 효소들을 다시 자신에게 적용하면 그 결과 하나의 문자열이 생성되는데, 이러한 과정을 통해 두 상보적 문자열이 서로 상대방과 똑같은 문자열을 생성하는 경우 이 문자열 쌍을 자가복제자라고 부른다.

하이퍼사이클(hyper-cycle)은 한 개 이상의 복제자(replicator)로 구성되어 있으며, 각 복제자들은 이전 복제자의 효소를 현재의 복제자에게 적용하여 현재의 복제자와 같은 개체를 만들어 낸다. 이러한 작용이 하이퍼사이클을 구성하는 복제자들 간에 차례로 일어나고, 마지막 복제자의 경우 자신의 효소를 첫 번째 복제자에게 적용하여 이 첫 번째 복제자와 같은 개체를 얻게 된다.

하이퍼사이클을 이루는 복제자의 개수가 하이퍼사이클의 크기이다. 크기 n인 하이퍼사이클을 이루는 모든 복제자를 한 번

에 찾기가 쉽다. 이 문제에 대한 해결책으로 Kvasnicka는 크기 n인 하이퍼사이클을 찾기 위해 하이퍼사이클의 크기를 하나씩 늘려가는 방법을 사용하여 크기 3인 하이퍼사이클을 생성하였다[5]. 또한 Kvasnicka는 문자열들의 차이점과 유사점을 측정하기 위하여 해밍 거리(Hamming distance)를 사용하였다. 본 연구에서는 편집 거리(edit distance)를 사용하여 크기 10까지의 하이퍼사이클들을 생성하였다[6].

2. 자가복제자의 생성

본 논문에서 사용한 doublet과 해당 규칙은 다음과 같다.

[표 1] doublet과 해당 규칙

No.	doublet	instruct.	Inclin.	No.	doublet	instruct.	Inclin.
1	AA	mvr	l	9	GA	rpy	s
2	AC	mvl	s	10	GC	rpu	r
3	AG	mvr	s	11	GG	lpy	r
4	AT	mvl	r	12	GT	lpu	l
5	CA	mvr	s	13	TA	rpy	r
6	CC	mvl	s	14	TC	rpu	l
7	CG	cop	r	15	TG	lpy	l
8	CT	off	l	16	TT	lpu	l

[표 1]은 각 doublet에 대한 명령어(instruction)와 시작점인 바인딩 사이트를 찾기 위해 사용되는 경사(inclination)를 보여주고 있다. [표 2]는 [표 1]에 나온 명령어들에 대한 설명이다.

[표 2] 명령어 의미

No.	Instruction	Description
1	cop	복제 모드의 시작
2	off	복제 모드의 끝
3	mvr	오른쪽으로 한 베이스 옮김
4	mvl	왼쪽으로 한 베이스 옮김
5	rpy	오른쪽 가장 가까운 피리미딘을 찾음
6	rpu	오른쪽 가장 가까운 퓨린을 찾음
7	lpy	왼쪽 가장 가까운 피리미딘을 찾음
8	lpu	왼쪽 가장 가까운 퓨린을 찾음

(A와 G는 퓨린, C와 T는 피리미딘)

[표 3] 바인딩 사이트 결정을 위한 표

No.	1st inclin.	Last arrow	Binding site
1	s	east	A
2	s	north	C
3	s	south	G
4	s	west	T
5	l	south	A
6	l	east	C
7	l	west	G
8	l	north	T
9	r	north	A
10	r	west	C
11	r	east	G
12	r	south	T

자가복제자가 DNA와 같은 삼차원 구조를 갖는다는 가정 하에 [표 3]에서 보이는 것처럼 첫 번째 inclination과 일련의 inclination의 적용으로 인해 결정되는 마지막 방향성의 조합으로 자가복제자의 바인딩 사이트를 결정한다.

자가복제자의 3차원 구조 결정에서 모든 시작 방향은 동쪽으로 정하고, 일련의 doublet에 해당하는 inclination이 s이면 같은 방향을 유지하고, r이면 오른쪽으로, l이면 왼쪽으로 방향을 바꾼다. CGACGTCCA라는 길이 9의 스트링을 예로 들어보자.

[표 4] doublet과 적용되는 inclination과 명령어

Doublet	CG	AC	TA	CA	A
Instruction	cop	mvl	rpy	mvr	해석하지 않음
Inclination	r	s	r	s	
east ->	south->	south->	west->	west	

처음 시작 inclination이 r이고 마지막 방향이 서쪽이므로 [표 3]의 10번째 줄에서 볼 수 있듯 바인딩 사이트는 C이다. 따라서 복제자 CGACGTCCA는 첫 번째 영기 C부터 명령어를 적용한다. 영기 아래의 밑줄은 현재 명령어가 적용될 곳을 지칭하는 일종의 커서이다. 명령어 cop는 현재 처리되어야 할 문자의 상보적 문자를 생성한다. A와 C는 상보적 관계이고, C와 G도 상보적 관계이다. [표 5]는 CGACGTCCA에 효소가 적용되는 모습을 해당 명령어들과 함께 순서대로 보여주고 있다.

[표 5] CGACGTCCA에 효소가 적용되는 모습

No.	원본 스트링	명령어
1	<u>C</u> G A C T A C A A G	cop
2	<u>C</u> G A C T A C A A G	mvl
3	C G A C <u>T</u> A C A A G C T G	rpy
4	C G A C <u>T</u> A C A A G C T G A	mvr

위와 같이 원래의 문자열의 부분 문자열 GCTGA가 생성되었다. 자가복제자의 경우 생성된 문자열이 원래의 문자열과 완전히 상보적이며, 상보적 문자열 또한 원래의 문자열을 완벽히 생성한다. [표 6]은 자가복제자의 예를 보여주고 있다. 자가복제자는 제 3장에서 크기 1인 하이퍼사이클로 사용된다.

[표 6] 자가복제자의 예

S	G C C C T T C T T C T C C G T T T T
R	C G G G A A G A A G A G G C A A A A

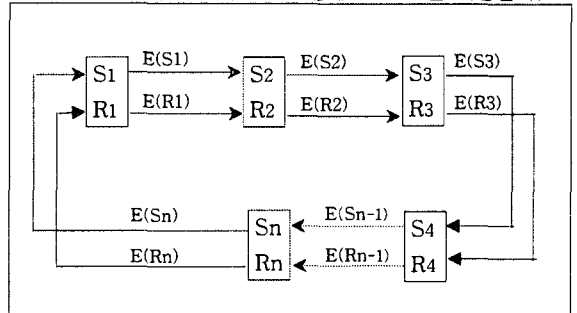
유전자 알고리즘에서 적합도를 정의하기 위해 생성된 문자열과 원래의 문자열 사이의 편집 거리(edit distance)를 계산하였다. 이 값이 작을수록 문자열의 적합도가 높은 것이라고 간주하고 아래의 식을 사용하였다. 아래의 식에서 S와 R은 자가복제자를 형성하는 상보적인 두 문자열을 가리키고, S', R'는 각각 자신의 효소를 적용하여 생성된 결과 문자열을 의미한다. 또 D는 0부터 1 사이의 값으로서 두 문자열 사이의 편집거리를 나타낸다.

$$fitness(S, R) = \frac{1}{2} * \{2 - D(S', R) - D(R', S)\}$$

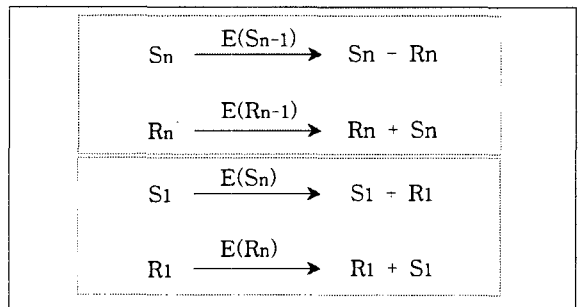
유전자 알고리즘에서 한 세대를 이루는 개체수는 2000, 돌연변이 확률은 0.01, 세대수는 2000으로 하였다.

3. 하이퍼사이클의 생성

[그림 1]은 크기 n인 하이퍼사이클의 개략적인 모습을 나타내고 있다. 여기서 E(Sx), E(Rx)는 괄호 안 문자열의 효소를 의미하고 화살표가 가리키는 문자열에 그 효소를 적용한다.



[그림 1] 크기 n인 하이퍼사이클의 모습



[그림 2] n번째 복제자의 추가 과정

[그림 2]에서와 같이 추가될 복제자가 하이퍼사이클의 새로운 구성요소가 되기 위해서는 문자열 Sn에 Sn-1의 효소를 적용했을 때 Sn의 상보적 문자열인 Rn이 나와야 하고, Rn 역시 Rn-1의 효소를 적용하였을 때, 그 결과로 Sn이 나와야 한다. 또 Sn과 Rn의 효소를 S1, R1에 적용하였을 때, 각각 R1, S1이 결과

로 나와야 복제자 (Sn,Rn)이 하이퍼사이클의 새로운 구성요소가 될 수 있다.

서론에서도 설명한 바와 같이 하이퍼사이클은 이전 복제자의 효소를 현재의 복제자에 적용하여 현재의 복제자와 같은 문자열을 만들어낸다. 크기 n인 하이퍼사이클의 복제자들을 한꺼번에 찾는 것의 시간 복잡도가 매우 높기 때문에 Kvasnicka가 사용했던 방법처럼 그 크기를 하나씩 증가시키는 방법을 이용하여 하이퍼사이클을 구하였다. 자가복제자의 경우와 마찬가지로 유전자 알고리즘을 사용하였고, 사용한 변수들도 자가복제자의 변수들과 함께 설정하였다. 추가될 복제자의 적합도는 [그림 2]에서와 같이 총 4번의 문자열의 비교가 일어나므로 다음 식을 이용하였다.

$$\text{fitness}(S_n, R_n) = \frac{1}{4} * \{4 - D(S_0, R_0) - D(S_0, \bar{R}_0) - D(S_n, R_n) - D(S_n, \bar{R}_n)\}$$

S₀와 R₀는 S_n과 R_n의 효소를 각각 S₀와 R₀에 적용하여 나온 문자열이고, S_n과 R_n은 S_{n-1}과 R_{n-1}의 효소를 S_n, R_n에 적용시켜 나온 문자열이다. D는 자가복제자에서와 같이 편집거리를 의미한다.

자가복제자를 찾는데 걸린 시간을 k라 한다면, 크기 n인 하이퍼사이클을 찾는 데에는 대략 n * k 만큼의 시간이 소요된다. 하이퍼사이클의 사이즈를 하나씩 늘려가면서 찾았기 때문에 가능한 결과이다.

Kvasnicka는 그의 논문에서 크기가 3인 하이퍼사이클을 한 개 제시하였으나, 본 연구에서는 하이퍼사이클의 크기를 계속해서 늘려 얼마든지 원하는 크기의 하이퍼사이클을 찾을 수 있음을 알게 되었다. [그림 3]은 본 연구의 결과로 얻은 크기 10인 하이퍼사이클 중 하나이다.

S1	G C C C T T C T T C T C C G T T T T
R1	C G G G A A G A A G A G G C A A A A
S2	G C C T C T C C T T C G T C T C G
R2	C G G A G A G G A A G C A G A G C
S3	G C C C C T T T C T C G T C T C C
R3	C G G G G A A A G A G C A G A G G
S4	G C C T C G T T T C T C C T C C
R4	C G G A G C A A A G A G G A G G
S5	G C C T T T C T C C C G T C T C A
R5	C G G A A A G A G G G C A G A G T
S6	G C C T C T C C C G T T T T T C T C C G
R6	C G G A G A G G G C A A A A A G A G G C
S7	G C C G C C T C T C T T C C T T C T C T T
R7	C G G C G G A G A G A A G G A A G A G A A
S8	G C C T T C T C C G T C C C C C T T T T
R8	C G G A A G A G G C A G G G G G A A A A
S9	G C C C C T T T C G T C T C
R9	C G G G G A A A G C A G A G
S10	G C C C T C C T T T C G T C T C
R10	C G G G A G G A A A G C A G A G

[그림 3] 크기 10인 하이퍼사이클의 예

자가복제자의 경우와 하이퍼사이클에서 발견할 수 있는 특이한 점은 초기 생성 당시 무작위로 생성되었음에도 불구하고, 생성된 자가복제자와 하이퍼사이클들의 구성 요소들이 비슷한 형태를 취하고 있다는 것이다. 이는 지금까지 생성된 생명체들의 DNA가 유사하다는 사실에 부합하는 흥미있는 현상이다.

4. 결론 및 앞으로의 연구 방향

생명의 기본적 특징인 전파를 모사한 자가복제자는 영기의 무작위적인 배열에도 불구하고 생명의 특징인 전파 능력을 가진 개체가 나타날 수 있다는 가능성을 시사한다. 이러한 자가복제자로 구성된 하이퍼사이클을 생성하기 위하여 유전자 알고리즘을 사용하였고, 문자열의 유사점과 차이점을 특정하기 위해 편집거리를 사용하였다. 그 결과 본 연구에서는 Kvasnicka가 제시한 크기 3인 하이퍼사이클보다 더 큰 임의의 원하는 크기의 하이퍼사이클을 얻을 수 있었다.

현재는 단순히 문자열의 적합도를 계산하는 방법으로 진화 방향을 외부에서 결정해 주었으나, 지역에 따른 한정된 자원 (limited resources)에 따라 개체의 진화 방향이 결정되는 한 단계 더 발전된 유전자 알고리즘을 적용할 계획이다. 이 알고리즘의 적용에 있어 지역적 개념과 자원의 한계를 도입하는 것은 이 방법이 외부의 개입 없이 개체들의 진화를 볼 수 있다는 장점 때문이다. 이를 통해 지역적으로 다르게 진화하는 개체들을 볼 수 있을 것으로 기대된다.

5. 참고문헌

- [1] D. Hofstadter, *Gödel, Escher, Bach: An eternal golden braid*, Basic Books, New York, 1979.
- [2] L. Varetto, *Typogenetics: An artificial genetic system*. Journal of Theoretical Biology 160, pp. 185-205, 1993.
- [3] L. Varetto, Studying artificial life with a molecular automaton, Journal of Theoretical Biology 193, pp. 257-285, 1998.
- [4] H. C. Morris, *Typogenetics: A logic for artificial life*, In C. Langton (Ed.), ALIFE I, pp. 369-395, Addison-Wesley, 1989.
- [5] V. Kvasnicka, J. Pospichal and T. Kalab, *A Study of Replicators and Hypercycles by Typogenetics*, ECAL VI, pp. 37-54, LNCS 2159, Springer, 2001.
- [6] Kyubum Wee and Woosuk Lee, *Extensions and variations on construction of autoreplicators in typogenetics*, ECAL VIII, pp. 685-694, LNAI 3630, Springer, 2005.