Journal of Korean Institute of Intelligent Systems
Vol. 25, No. 1, February 2015, pp. 022-028
http://dx.doi.org/10.5391/JKIIS.2015.25.1.022

ISSN(Print) 1976-9172
ISSN(Online) 2288-2324

셀 수준의 진화 프레임워크를 통한 인공개체의 행동로직 진화

Evolution of Behavioral Logic of Artificial Individuals Using Cell-level Evolution Framework

정보선, 정성훈†

Bo-Sun Jung, and Sung Hoon Jung†

한성대학교 정보통신공학과

Department of Information and Communications Engineering, Hansung University

요 약

본 논문에서는 셀 수준의 진화 프레임워크를 이용하여 인공개체의 행동로직을 진화하는 연구를 수행하였다. 이를 위하여 셀 수준의 진화 프레임을 구현하였으며 이 프레임 상에서 인공개체가 먹이를 먹기 위해 행동로직을 진화하는 것을 살펴보았다. 인공개체의 행동로직 진화를 관찰하기 위하여 행동결정 로직 프레임을 제안하여 적용하였다. 테스트결과 인공개체가 빠른 세대 내에 먹이를 잘 먹는 로직으로 진화하는 것을 관찰할 수 있었다. 또한 여러 번의 실험을 통하여 대부분의 실험에서 거의 동일한 행동양식을 보이는 것으로 진화하는 것을 확인할 수 있었다. 본 논문에서 제안한 방법은 기존의 진화 알고리즘을 이용한 알고리즘이나 하드웨어의 진화와는 다른 방법으로서 기본적으로 접근 방법에 차이점이 있다. 이런 결과로 보았을 때 본 논문에서 제안한 프레임워크가 셀 수준의 진화를 관찰해볼 수 있는 좋은 도구가 될 수 있음을 알 수 있다.

키워드 : 진화 프레임워크, 인공개체, 행동로직

Abstract

In this paper, we studied the evolution of behavioral logic of artificial individuals using cell-level evolution framework. We first implemented cell-level evolution framework and then investigated the evolution of behavioral logic that artificial individuals ate foods on the framework. A logic frame for behavioral decisions of artificial individuals was devised and applied to the framework. From extensive tests, we found that most artificial individuals could evolve the behavioral logic that they could eat food in a short generation. It was also confirmed that most behavioral logics showed nearly same behaviors of artificial individuals in most tests. Our method has the differences from existing algorithms using evolutionary algorithms and evolvable hardwares in that it is a basically different approach. These results showed that our framework could be a good tool for investigating the evolution of artificial individuals in a cell-level.

Key Words : Evolution framework, Artificial individuals, Behavioral logic

1. 서 론

40억년의 지구 역사에서 30억년이 걸려서 단세포가 발생했고 10억년이 걸려서 다세포가 출현했으며 그 이후 수 천만 년에 걸쳐서 저등동물에서 고등동물로의 진화를 통해 인류가 지구상에 존재하게 되었다고 한다[1,2]. 초기 지구의 열악한 환경에서 현재와 같이 수천만 종이 조화롭게 어울려 살아가는 아름다운 지구가 만들어진 그 모든 과정이 밝혀지지는 않았으나 그 과정에서 중요한 역할을 한 것으로 인류가 밝혀낸 것 중에 하나가 다아윈이 주장한 진화이다[3]. 진화란 환경에 적합한 개체가 살아남는다는 적자생존의 원리로서 지구상에 존재하는 개체가 점점 더 자연에 적응하는 존재로 변화해가는 원리를 설명해준다. 즉, 생물의 생식과정에서 발생한 돌연변이로 인하여 부모와 다른 개체가 출현하게 되고 이러한 돌연변이 개체가 부모개체보다 더 환경에 적합한 경우 부모개체보다 더 잘 살아남게 되어 진화가 일어난다. 물론, 진화만으로 이렇게 복잡하고 정교한 고등동물들이 존재하게 되었는지는 확실하지 않으며 진화 이외에 다른 메커니즘에 대해서도 연구되고 있는 상황이다. 그렇다하더라고 진화라는 메커니즘은 단순한 물질에서 복잡한 생물로의 탄생을 가능하게 하는 주된 요소임은 틀림없다.

Received: Nov. 6, 2014
Revised : Jan. 20, 2015
Accepted: Feb. 4, 2015
† Corresponding author(shjung@hansung.ac.kr)
본 논문의 초기결과는 전자공학회 2014년도 추계학술대회에서 발표하였습니다.
본 연구는 한성대학교 교내학술연구비 지원과제임
This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

본 논문은 이와 같은 가정 하에 세포수준이나 생물의 조직수준 혹은 개체수준에서 단순한 시스템이 복잡한 시스템으로 진화하는 과정에서 일어나는 여러 가지 현상이나 내부적으로 생성되는 회로 등을 쉽게 모사해볼 수 있는 진화 모델링 및 시뮬레이션 프레임워크를 제안한다. 셀 우주(CC: Cellular Cosmos) 라고 불리는 이러한 모델링 및 시뮬레이션 프레임워크는 실제 생물의 진화과정을 Boolean 모델링 수준에서 모사해볼 수 있는 기능을 제공하며 또한 공학적 회로 설계에도 응용할 수 있다.

본 논문에서 제안하는 프레임워크와 유사한 기존연구로는 폰노이만이 제안한 세포 자동자(CA: Cellular Automata)가 있다 [4]. 세포 자동자는 셀이 일정한 격자 모양으로 구성된 것으로 각 셀은 유한한 상태를 갖고 있으며 다음 시간에서의 상태는 이웃하는 8개의 셀의 상태에 따라서 결정된다. 이러한 상태변화는 8개의 이웃 셀의 상태에 따라서 특정한 규칙으로 지속적으로 일어난다. 이러한 과정에서 특정한 패턴이나 현상이 생성될 수 있다. 본 논문에서 제안한 셀 우주는 세포 자동자와 유사한 셀 격자로 구성되어 있다. 그러나 가장 큰 차이점은 각각의 셀이 유한한 상태를 갖고 시간에 따라서 상태가 변화하는 것이 아니라 셀에 모델링 및 시뮬레이션하려는 대상 개체가 존재한다는 것이다. 이러한 개체의 입/출력 로직은 미리 주어지는 것이 아니라 초기에는 임의의 값을 가지며 시간이 지나감에 따라서 환경에 따라 진화해간다. 다만 입/출력 로직은 모델링 및 시뮬레이션하려는 대상에 따라서 미리 특정한 구조로 구성되어 있어야한다.

셀 상에 존재하는 개체는 개체수준에서 독립적으로 존재하며 셀 공간상에서 이동하며 번식할 수 있다. 개체가 번식할 경우 일정한 비율로 돌연변이를 발생시켜 환경에 더 적합한 개체가 살아남게 하여 진화를 일으킬 수 있다. 일정한 횟수를 수행하고 난 후에 충분히 진화된 개체의 내부 로직이나 행동양식을 살펴봄으로서 모델링된 개체의 진화적 특성을 확인하고 분석할 수 있다. 특히, 개체의 특성이나 여러 가지 환경적 요소를 변화시켜 가면서 개체가 어떤 로직으로 진화하는지 살펴볼 수 있다. 이러한 실험을 통하여 개체가 왜 그러한 내부 로직을 갖도록 진화했는지 분석하는데 도움을 받을 수 있다. 또한 실험설계에 따라서 공학적 응용회로 설계로도 응용할 수 있다.

본 논문에서는 제안한 셀 우주를 인공개체가 먹이를 먹는 행동양식으로 진화하는 응용에 적용하였다. 이를 위하여 인공개체의 행동결정 로직 프레임을 특정한 입/출력을 갖는 디지털로직 형태로 주었다. 해당 로직 프레임의 연결은 초기에는 무작위로 주어졌기 때문에 초기 인공개체의 행동은 먹이의 위치와 상관없는 행동을 보인다. 먹이 위치와 상관없는 행동을 보이는 개체는 세대가 지나감에 따라서 굶어죽게 되며 살아남은 개체는 번식을 통하여 우월한 유전자를 자식에게 물려준다. 인공개체는 번식 시에 일정한 확률로 돌연변이가 일어나는데 부모 개체보다 진화된 개체는 더 잘 살아남아 많은 자식개체를 생성하므로 점점 더 먹이를 잘 먹는 개체가 살아남게 된다. 다양한 실험을 통하여 인공개체의 행동결정 로직이 먹이를 점점 더 잘 먹는 로직으로 진화됨을 확인하였으며 대부분의 진화된 개체는 유사한 행동양식을 보이는 것으로 진화됨을 확인하였다.

본 논문의 구성은 다음과 같다. 먼저 2절에서 본 논문에서 적용한 응용과 유사한 기존의 연구에 대하여 소개한다. 이를 통하여 본 논문에서 제안한 방법이 기존의 연구와의 차이가 있음을 설명한다. 3절에서 본 논문에서 제안한 셀 우주의 전체적인 구조와 동작을 설명한다. 4절에서 시뮬레이션 결과와 그 의미를 설명하며 5절의 결론으로 끝을 맺는다.

2. 기존의 유사연구

본 논문에서 제안한 것은 시뮬레이션 프레임워크로서 첫 번째로 적용한 인공개체-먹이 문제는 하나의 응용이다. 이러한 응용과 실험적으로 유사한 기존의 연구들이 있다. 그러한 것으로 대표적인 것이 진화 하드웨어(Evolvable Hardware) 이다 [8-13]. 진화 하드웨어는 진화 알고리즘을 이용하여 하드웨어를 자동으로 설계하려는 연구로서 아날로그 회로나 디지털회로 진화에 성공적으로 응용된 예가 많이 발표되었다 [9-12]. 진화 하드웨어에서 주로 사용하는 진화 알고리즘은 유전자 알고리즘(Genetic Algorithm)이 많이 사용되며 또한 최근에는 유전자 프로그래밍(Genetic Programming)을 이용하기도 한다[8,9]. 진화 알고리즘을 이용하여 진화적으로 하드웨어를 설계하기 위하여 먼저 설계할 대상 하드웨어에 따라서 진화 알고리즘의 개체를 설계해야한다. 이는 일반적인 진화 알고리즘 응용에 있어서 개체 인코딩에 해당하는 것으로서 개체 인코딩을 효율적으로 설계해야 진화가 빨리 일어나 성공적인 설계 결과를 얻을 수 있다. 특히 진화 하드웨어 문제에서 가장 큰 문제 중의 하나로 지적되는 확장성(scalability)을 해결하기 위해서 매우 중요하다.

확장성이란 설계하려는 하드웨어의 크기를 크게 한 경우에도 잘 진화되어야하는 특성을 말한다. 예를 들면, 4비트 가산기 설계를 잘 하는 알고리즘이 8비트나 16비트 가산기 설계도 잘 할 수 있도록 하는 것이다. 기본적으로 진화 알고리즘을 이용하여 하드웨어를 설계하는 것은 탐색문제로서 인코딩한 방법에 따른 전체 해 공간에서 해당 하드웨어의 입/출력을 만족하는 진리표를 생성하는 해를 찾는 과정이다. 그러므로 비트 수가 커지면 전체 탐색공간이 급격하게 커지는 조합 폭발(combinational explosion) 문제가 발생한다. 이를 위하여 밀러는 카티젼 유전자 프로그래밍(CGP: Catesian Genetic Programming)을 제안하였다[14]. 기본적으로 확장성 문제를 해결하기 위해서는 분할정복 알고리즘이 많이 사용된다[10].

또한 유전자 알고리즘을 이용하여 개체의 자체 알고리즘이나 협동 알고리즘을 진화시키려는 시도가 있다[15,16]. 이러한 연구에서도 기본적으로 개체의 전략적 행동을 유전자 알고리즘의 염색체 형태로 표현해 유전자 알고리즘의 연산을 적용해서 진화시키는 방법이다[16]. 이러한 알고리즘에서는 유전자 알고리즘의 연산으로 일반적으로 선택, 교배, 돌연변이, 적합도 평가의 과정을 해를 찾을 때까지 반복한다[15]. 그리고 인공생명(AL: Artificial Life)연구에서도 개체의 진화를 모사하기 위하여 진화 알고리즘을 사용한다.

이러한 기존의 유사연구와 본 연구에서 제안한 방법 간의 가장 근본적인 차이점은 연구목적과 접근방법의 차이로서 우리가 제안한 방법은 기본적으로 셀 공간에 존재하는 개체가 주어진 환경에 따라서 어떤 특징으로 진화해 나가는지를 살피는데 근본적인 목적이 있다. 즉 2차원 3차원 공간상에 존재하는 어떤 개체를 대상으로 한 것으로서 연속공간이 아닌 격자 형태의 셀 공간에서 환경에 다른 진화적 생태적 특징 생성을 살펴보려는 것이다. 이러한 측면으로 보았을 때

는 세포 자동자와 유사한 측면이 있다. 다만 세포 자동자는 각 셀이 개체가 아닌 상태이며 각 셀의 상태변화는 이웃하는 8개의 상태에 따라 변화하는데 그 변화 규칙이 미리 주어진 것이다. 그러나 우리의 방법에서는 각 셀에 개체가 존재하며 그 개체는 제한된 범위의 이웃 셀들을 관찰할 수 있으며 그 관찰된 결과로 자신의 행동을 결정한다. 다만 관찰된 결과와 행동결정 사이에는 미리 어떤 규칙도 주지 않고 시뮬레이션 목적에 따라서 다양한 형태로 모델링 될 수 있다. 예를 들면 본 논문에서의 응용에서처럼 디지털 회로로 모델링 될 수도 있으며 퍼지입출력 형태로 모델링될 수도 있다. 또한 미분방정식 형태로 모델링될 수 있다. 인공개체들은 모델링된 형태로 셀 공간 안에서 생존하며 자식을 번성하며 목적에 따라서 죽고 사는 과정도 도입될 수 있다. 이러한 시뮬레이션을 통하여 셀 공간수준에서 입/출력 로직으로 모델링된 개체의 다양한 진화적 특성을 알아볼 수 있다. 그러므로 기존의 진화 알고리즘 응용에서와 같이 염색체로 표현된 개체나 이러한 개체를 모아놓은 개체군도 없으며 더불어 대부분의 진화 알고리즘에서 사용하는 부모선택, 교배, 적합도 평가 등이 없고 따로 구분되어 있지 않다.

3. 셀 우주(Cellular Cosmos)

셀 우주 (CC: Cellular Cosmos)는 목적에 따라서 2차원 혹은 3차원의 셀 격자로 구성된다. 해당 셀 격자는 2차원 혹은 3차원의 공간을 의미하는 것으로 CC 안에서 시뮬레이션 되는 공간의 최소단위이다. 그러므로 시뮬레이션 목적에 따라서 하나의 셀은 하나의 개체가 차지하는 공간이 될 수 있으며 또한 하나의 개체가 다수의 공간을 차지할 경우 다수의 공간 중에 하나로 시뮬레이션 될 수 있다. 그러므로 CC를 이용하여 셀 수준의 진화를 모델링 및 시뮬레이션하려면 CC의 2차원 혹은 3차원 공간상에 어떤 대상이 어떻게 존재하는지를 결정해야한다. 또한 모델링 및 시뮬레이션 목적에 따라서 대상 이외에 어떤 다른 환경적 요소가 필요한지를 판단해서 적용해야한다. 무엇보다 중요한 것은 CC를 통하여 확인하고 싶은 진화적 요소가 무엇인지를 명확하게 결정하고 해당 요소가 드러나게끔 실험을 설계해야한다. 이를 위하여 진화적 실험대상의 기본적 구조를 설계하여 적용해야한다. 본 논문에서 실험한 인공개체-먹이 모델을 예로 들어 설명하면 다음과 같다.

- 시뮬레이션 목적: 2차원 공간상에 존재하는 개체가 먹이를 찾아서 먹는 로직을 Boolean 수준의 로직으로 진화적으로 찾음
- 대상: 개체
- 환경: 개체와 먹이
- 진화요소: 개체의 입/출력 로직을 특정 로직구조로 설계

본 절에서는 본 논문에서 실험한 인공개체-먹이 모델을 예로 들어 CC를 설명한다.

3.1 CC의 구성

CC는 그림 1과 같이 2차원 셀 격자공간으로 구성된다. 셀 격자공간에는 개체(삼각형)와 먹이(십자형)이 존재한다. 개체는 자신의 주변 공간을 인식하고 주변 공간 정보에 따라서 자신의 행동을 결정한다. 개체는 동/서/남/북 네 방향으로 이동할 수 있는데 이동하는 방향을 기준으로 8개의 이웃 셀 중에서 5개의 셀만을 인식할 수 있다고 가정한다. 그

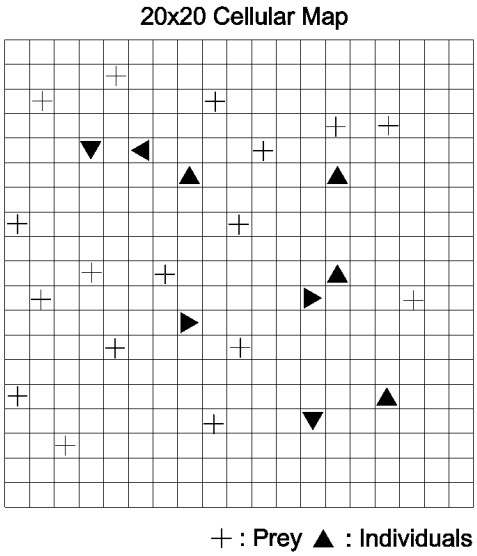

그림 1. CC의 구성
Fig. 1. Composition of CC

림 2는 개체의 이동방향에 따른 인식범위를 보여준다. 즉, 그림 2에서 개체가 화살표 방향으로 움직인다고 가정할 때 개체가 인식할 수 있는 범위는 N1~N5이 된다. 시뮬레이션 목적에 따라서 인식 범위를 설정할 수 있다. 예를 들면, 목적에 따라서 그림 2에서 그 다음 이웃의 9개의 셀까지 인식범위를 확대할 수 있다. 즉, N1, N4 왼쪽으로 두 개, N1, N2, N3 위로 3개, N3, N5 오른쪽으로 두 개 총 7개에다가 대각선 방향으로 두 개 까지 인식범위를 확대할 수 있다. 개체가 인지 가능한 N1~N5의 셀에는 먹이가 존재할 수 있고 또한 다른 개체가 존재할 수 있다. 하나의 셀에는 하나의 먹이만 존재가능하며 개체가 먹이 셀에 있으면 먹이를 먹은 것으로 간주한다. 개체는 하나의 셀에 여러 개가 존재 가능하지만 먹이가 있는 경우 하나의 개체만 먹이를 먹은 것으로 간주한다.

개체의 행동은 현재 진행방향 기준으로 직진, 좌회전, 우회전, 정지의 네 가지 행동을 할 수 있다. 개체는 N1~N5의 상황을 고려하여 자신의 행동을 결정해서 시행해야한다. 본 실험의 목적이 개체가 외부환경 즉 먹이에 따라서 어떻게 자신의 행동로직을 진화시키는가를 관찰하는 데에 있으므로 환경에 따른 행동결정 로직에 대한 프레임을 설계해서 제공해야한다. 이 때 개체는 오로지 먹이에만 관심을 가지며 셀 공간상에 여러 개의 개체가 존재할 수 있으므로 다른 개체에 대한 정보는 행동결정에 영향을 미치지 않게 설계했다. 결국, 개체 행동결정 로직은 5개의 셀 공간상에 먹이가 있는지에 대한 정보를 바탕으로 자신의 네 가지 행동을 결정하므로 입력은 5비트 출력은 2비트(4가지 행동을 encoding 해서 갖고 있음)가 된다. 이러한 행동결정 로직에 대한 로직 프레임은 그림 3과 같다. 즉 5개의 셀의 먹이 상태를 입력받아서 4가지 행동 중에 하나를 결정하여 움직인다.

3.2 행동결정 로직 프레임

본 응용에서 진화적 요소는 먹이에 대한 개체의 움직임이다. 이를 위하여 개체는 CC상에서 그림 2와 같은 관찰영역을 가지며 관찰된 결과에 따라서 자신의 행동을 결정한다. 이때 관찰된 결과를 입력으로 자신의 행동을 결정짓는 내부

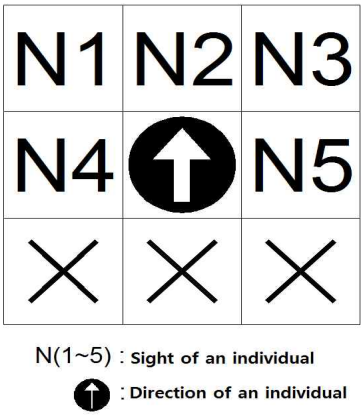

그림 2. 개체의 인식범위
Fig. 2. Recognition range of individuals

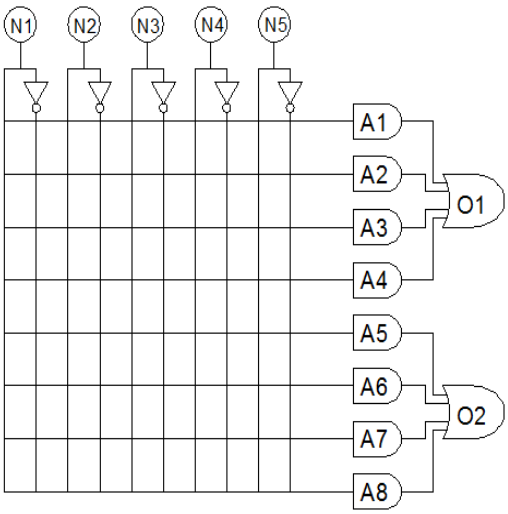

그림 3. 행동결정 로직 프레임
Fig. 3. Logic frame of behavioral decision

로직 프레임이 주어져야한다. 이러한 로직 프레임은 모델링 및 시뮬레이션 목적에 따라서 완전히 자유롭게 주어질 수 있으나 본 논문에서는 진화된 결과분석을 원활히 하기 위하여 그림 3과 같은 특정한 로직 프레임을 사용한다.

표 1. 출력에 대한 개체의 행동
Table 1. Behaviors of individuals according to outputs

O1	O2	Behaviors
0	0	Stop
0	1	Turn left
1	0	Turn right
1	1	Go straight

개체가 N1~N5까지 다섯 셀을 관찰하므로 5개의 입력이 존재하며 네 가지 행동을 수행해야하므로 출력은 O1, O2 두 개가 출력된다. 개체별로 로직을 스스로 진화시켜 나가기 위하여 출력별로 네 개의 AND 게이트와 하나의 OR 게이트로 로직을 생성한다. 입력에 대한 로직을 만들기 위하여 입력은 NOT 게이트가 포함된 항 5개를 포함하여 모두 10개가 입력되며 초기에는 접점이 무작위 적으로 생성되어 무작위 적인 로직이 만들어진다. N1~N5에서 먹이가 있는 곳은 값이 1로 그렇지 않은 곳은 0으로 입력이 들어온다.

보수입력을 포함한 10개의 입력은 행동결정 로직을 만들기 위한 8개의 AND 게이트(A1~A8)로 연결될 수 있다. 어떤 입력이 연결되느냐에 따라서 로직이 결정된다. 예를 들어 A1 AND 게이트의 10개의 입력 접점에서 1, 3, 6, 10 접점이 연결되었다면 이는 N1N2N3'N5' 의 AND 항을 만든다. A1~A4의 네 개의 AND 출력은 O1으로 들어가 첫 번째 출력을 만들며 A5~A8의 네 개의 AND 출력은 O2로 들어가 두 번째 출력을 만든다. 그러므로 로직은 AND 항의 접점만으로 변화가 되며 이 출력에 의존해 행동을 결정하게 된다. 표 1은 출력 O1과 O2에 따른 개체의 행동을 보여준다. 00인 경우 개체는 정지해있으며, 01인 경우 오른쪽으로 회전하고, 10인 경우 왼쪽으로 회전하며, 11인 경우 앞으로 직진한다.

3.3 진화 과정

개체는 N1~N5의 입력에 대한 자신의 로직 출력 O1, O2에 따라서 결정된 행동을 한 시간단위 당 한 번씩 실행한다. 행동에는 네 개의 행동이 있기 때문에 각 개체는 입력상황에 따라서 정지해있거나, 왼쪽으로 회전하거나, 오른쪽으로 회전하거나, 직진한다. 개체가 이동한 곳에 먹이가 있는 경우 먹이를 먹은 것으로 간주한다. 그러므로 일정기간 동안 먹이에 대하여 올바른 행동을 하지 못하는 개체의 경우 굶어죽게 되며 먹이를 잘 먹는 개체의 경우 성장하여 자식을 번식하게 된다. 결국 먹이를 잘 먹는 개체의 좋은 유전자가 자식에게 유전되어 점점 더 먹이를 잘 찾아가는 개체로 진화하게 된다. 자식을 번식할 때는 부모 개체의 로직 접점에 일정 비율로 돌연변이를 발생시켜 보다 더 좋은 로직을 갖는 개체로 진화하게 된다. 본 논문에서 제안한 CC는 개체의 특성과 환경적 특성에 따라서 개체가 잘 살아남는 방법으로 진화하기 때문에 유전자 알고리즘에서처럼 각 개체의 적합도를 일일이 평가할 필요가 없다[6,7]. 다만 진화가 적절히 일어나는지를 확인하기 위하여 개체의 적합도를 평가할 수 있다. 이를 위하여 본 논문에서는 개체가 얼마나 먹이를 잘 먹는가를 적합도(fitness)로 나타내었다. 적합도는 모든 입력(2^5=32가지)상황에 대하여 먹이를 잘 먹는 경우에 1점 그렇지 않은 경우 0점을 주어서 계산하였다. 그러므로 최고점은 32점이 되며 32점의 적합도를 갖는 개체는 최적으로 진화된 상태라고 볼 수 있다.

4. 시뮬레이션 및 결과

인공개체-먹이 응용문제를 이용하여 본 논문에서 제안한 CC를 시뮬레이션 하였다. 프로그램은 PC상에서 Matlab으로 작성하였다. 시뮬레이션을 위해서 설정한 CC의 파라미터는 표 2와 같다. CC의 셀 공간은 35x35로 설정하였다. 개체의 초기 개수는 셀 공간의 한쪽 크기 35를 2로 나눈 후 반 내림한 17을 사용하였다. 특별한 이유가 있는 것이 아니고 개체의 움직임을 적절히 관찰하기 위하여 셀 공간 크기와 비례하여 설정한 것이다. 개체의 최소 개수는 초기 개체수와 동일하도록 설정하였고, 개체의 최대 개수는 개체의

표 2. CC의 파라미터
Table 2. Parameters of CC

Map size setting of CC	35 x 35
Initial no. of individuals and prey	17 / 170
Max. no. of individuals and prey	43 / 170
Min. no. of individuals and prey	17 / 170
Time unit: generation ratio	35 : 1
Maximum generation (Simulation termination conditions)	200
Mutation probability(per logic connection)	0.05

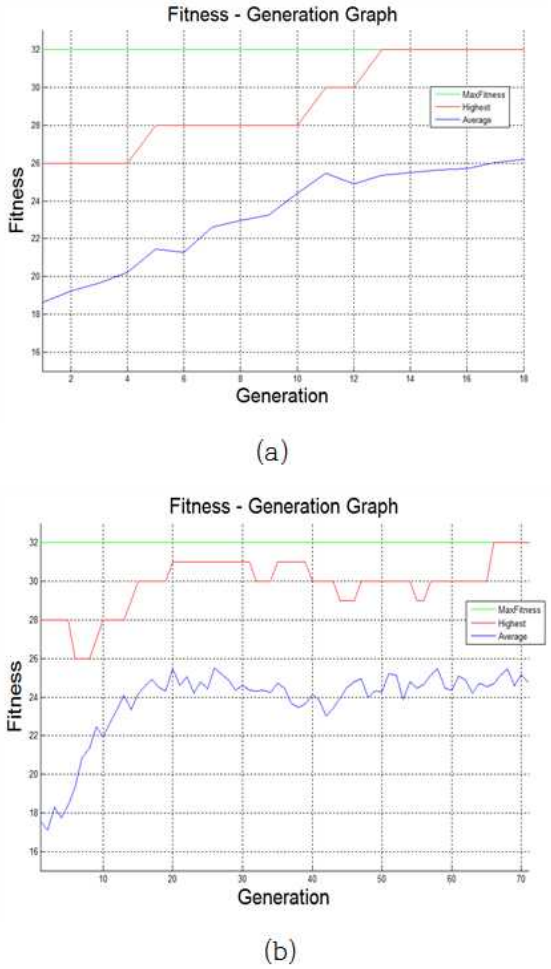

그림 4. Fitness-Generation 그래프
Fig. 4. Fitness-Generation graph

초기 개수의 2.5배에 반올림한 값을 사용하였다. 개체의 최대 개수를 한정한 이유는 개체수가 너무 많아지면 먹이가 부족해져서 적절한 진화 상황을 만들기 어렵기 때문이다. 먹이의 개수는 개체의 초기 개수의 10배에 해당하는 값으로 설정하였다. 개체진화에서 적당한 먹이의 분포는 중요하다. 먹이가 부족한 경우 굶어죽는 개체가 너무 많이 발생하며 먹이가 너무 많은 경우 경쟁력이 없는 개체도 살아남아 진화에 도움이 되지 않는다. 그러나 개체 개수에 비하여 어느 정도의 먹이가 있는 것이 적당한지는 알 수 없다. 그래서 본 논문에서는 경험적인 값으로 개체 초기 개수에 10배를 사용하였다. 최대 세대는 200으로 다른 값들과 연관성이 없는 고정 값이다. 최고 적합도 32를 갖는 개체를 찾지 못한 경우 200세대에서 시뮬레이션이 종료한다. 그러나 실험결과 대부분의 경우 200세대 이전에 최고 적합도를 갖는 개체가 생성되었다. CC는 시간단위로 시뮬레이션이 진행되며 35 시간단위가 경과된 경우 한 세대가 진행된 것으로 간주한다. CC 상에서 개체의 대부분의 동작(번식이나 혹은 죽음 등)은 세대를 기준으로 일어난다. 자식 개체를 생성할 때 돌연변이 확률은 로직 프레임 접점 당 0.05의 확률을 사용하였다.

시뮬레이션이 시작되면 CC의 셀 공간 내에 개체와 먹이가 임의의 위치에 생성된다. 다만, 먹이와 개체가 동일한 셀에 생성되지는 않는다. 각 개체의 초기 로직은 무작위로 생성되므로 초기에는 대부분의 개체가 다른 로직을 갖는다. 초기에 개체 로직은 임의로 만들어져서 대부분 적합한 행동을 하지 못한다. 그 경우 먹이를 먹지 못해 죽게 된다. 다음 3가지 조건은 개체가 죽게 되는 조건이다.

(1) 한 세대 내에서 먹이를 먹지 못한 경우
(2) 5세대 이상 살아온 경우
(3) 적합도가 낮은 경우

한 세대동안 먹이를 먹지 못한 개체는 굶어서 다음 세대가 시작하기 전, 즉 해당 세대의 35번째 시간단위에 죽게 된다. 적합도가 낮아서 죽는 경우는 한 세대가 끝날 때 개체의 개수가 최대 개체 개수를 초과할 경우에 적합도가 낮은 개체 순으로 제거하여 최대 개체 개수를 맞추어준다. 개체가 사는 조건은 다음과 같다.

(1) 한 세대 내에서 먹이를 하나 이상 먹은 경우
(2) 적합도가 높은 경우
(3) 부모 개체에 의해 생성된 자식 개체의 경우

개체의 번식은 매 세대가 끝나고 시작할 때 살아있는 개체를 대상으로 이뤄진다. 세대가 진행되면서 먹이를 잘 먹는 개체들은 살아남아 자식개체를 생성하므로 점점 더 먹이를 잘 먹는 개체들로 진화된다.

최적의 개체로 진화된 경우 시뮬레이션의 목적을 달성한 것이기 때문에 시뮬레이션을 종료하면 된다. 만약 최적의 진화를 모르는 경우에는 이 방법으로 종료를 하기 어렵기 때문에 보통 최대 세대수로 시뮬레이션을 종료한다. 본 연구에서는 최대 적합도 32점을 알기 때문에 최대 적합도를 갖는 개체가 발생한 경우 시뮬레이션을 종료한다.

개체가 잘 진화되는지 여러 번의 실험을 통하여 확인하였다. 확인 결과 세대가 지나가면서 점점 더 높은 적합도를 갖는 개체로 진화되는 것을 볼 수 있었다. 초기에 생성된 개체의 로직들은 대략 평균 26점의 적합도를 가졌고, 이 개체들이 32점의 적합도를 갖는데 걸린 평균 세대수는 대략 36세대이다. 그림 4(a)그래프는 하나의 시뮬레이션에서의 최고적합도와 평균 적합도를 보여준다. 그림에서 보듯이 평균적합도가 세대가 지날수록 안정적으로 상승하고 있고 13세대에서 최대 적합도 32를 갖는 개체로 진화하였다. 그림 4(b) 그래프에서는 상대적으로 진화가 더딘 결과를 보여준다. 또한 최고 적합도를 유지하지 못하는 것을 보여준다. 이것은 이전 세대에서 최고 적합도를 갖는 개체가 다음 세대에서 자식개체를 생성하고 죽은 경우, 자식 개체에 발생한 돌연변이가 개체의 적합도를 낮추어 최고 적합도를 유지하지 못하는 경우이다. 돌연변이는 더 좋은 개체를 찾기 위해 필요하지만 반드시 좋은 방향으로의 돌연변이만 발생하는 것은 아니다. 특히 진화가 많이 일어난 상황에서는 추가적

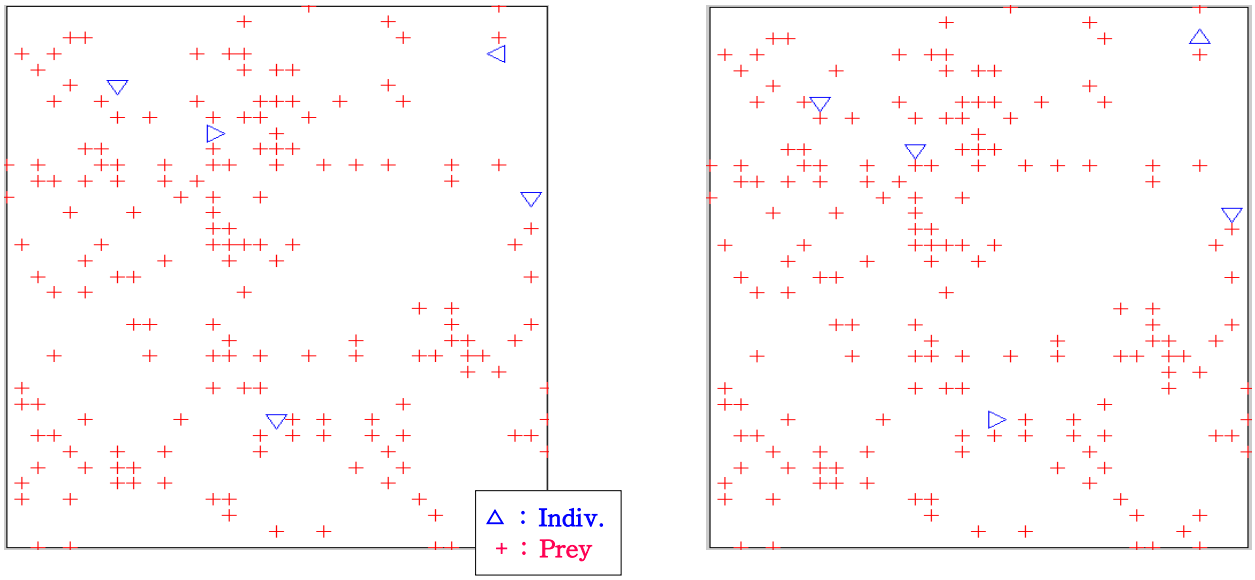

그림 5. 최고 적합도 32를 갖는 개체의 행동 (왼쪽: t, 오른쪽 : t+1)
Fig. 5. Behaviors of individuals with the highest fitness of 32 (left: t, right : t+1)

인 돌연변이가 좋은 쪽으로 일어나기 어렵기 때문에 주로 나쁜 쪽으로 발생한다.

그림 5는 최대 적합도를 갖는 개체의 행동을 살펴보기 위해 최대 적합도를 갖는 개체가 발생한 후에 해당 개체의 로직을 5개의 개체로 테스트해본 결과화면이다. 푸른색의 삼각형은 개체를 나타내며 붉은색의 십자가 모형은 먹이를 나타낸다. 개체의 삼각 모형 중 동, 서, 남, 북을 향하고 있는 방향이 현재 개체의 이동 방향이다. 그림 5의 왼쪽은 시간 t에서, 오른쪽은 시간 t+1에서의 실행화면이다. 3개의 개체가 먹이를 먹기 위해 방향을 튼 모습을 볼 수 있다. 맨 오른쪽 위의 개체의 경우 지난 간 자리에 먹이가 새로 생성된 것을 보여준다. 수십 세대 동안 살펴본 결과 먹이를 잘 찾아서 먹는 것을 살펴볼 수 있었다. 두 번의 실험에서 최대 적합도를 갖는 개체의 로직을 각각 살펴보니 다음과 같았다.

첫 번째 실험에서 최대 적합도를 갖는 개체 1

- O1 = N5′ + N2N3N4′
- O2 = N4′ + N5

두 번째 실험에서 최대 적합도를 갖는 개체 2

- O1 = N5′
- O2 = N1′N3′N4′ + N2 + N5

최대 적합도는 행동으로 결정하는 것이기에 최대 적합도를 보이는 입/출력 진리표가 다수 개 존재할 수 있다. 왜냐하면 동일한 상황에서 다수 개의 적합한 행동이 있기 때문이다. 예를 들면 N1~N5에 모두 먹이가 있는 경우 N1~N5 어느 곳으로 가던지 적합한 행동이 된다. 또한 하나의 진리표를 구현하는 것도 다수의 로직이 가능하다. 그러므로 최대 적합도를 보이는 개체의 로직은 매우 많이 존재할 수 있다.

그림 6은 최대 적합도를 갖는 개체 1, 2의 행동 결정표를 보여준다. N1~N5는 입력이고 O1과 O2는 입력에 대한 두 OR게이트의 출력이다. 출력을 통하여 표 1처럼 각 개체의 행동이 결정된다. 두 개체는 몇 개의 경우에 서로 다른 행동을 결정하는 것을 보여준다. 이는 두 개체가 서로 다르게 진화해 왔지만 둘 다 환경에 적합한 형태로 진화했음을 보여준다. 이런 결과로 보았을 때 본 논문에서 제안한 CC가 셀 공간상에서 진화적 모델링 및 시뮬레이션 프레임워크로 유용함을 볼 수 있었다.

Input					Sample 1			Sample 2		
N1	N2	N3	N4	N5	O1	O2	Action	O1	O2	Action
0	0	0	0	0	1	1	straight	1	1	straight
1	0	0	0	0	1	1	straight	1	0	turn left
0	1	0	0	0	1	1	straight	1	1	straight
1	1	0	0	0	1	1	straight	1	1	straight
0	0	1	0	0	1	1	straight	1	1	straight
1	0	1	0	0	1	1	straight	1	0	turn left
0	1	1	0	0	1	1	straight	1	1	straight
1	1	1	0	0	1	1	straight	1	1	straight
0	0	0	1	0	1	0	turn left	1	0	turn left
1	0	0	1	0	1	0	turn left	1	0	turn left
0	1	0	1	0	1	0	turn left	1	1	straight
1	1	0	1	0	1	0	turn left	1	1	straight
0	0	1	1	0	1	0	turn left	1	0	turn left
1	0	1	1	0	1	0	turn left	1	0	turn left
0	1	1	1	0	1	0	turn left	1	1	straight
1	1	1	1	0	1	0	turn left	1	1	straight
0	0	0	0	1	0	1	turn right	0	1	turn right
1	0	0	0	1	0	1	turn right	0	1	turn right
0	1	0	0	1	0	1	turn right	0	1	turn right
1	1	0	0	1	0	1	turn right	0	1	turn right
0	0	1	0	1	0	1	turn right	0	1	turn right
1	0	1	0	1	0	1	turn right	0	1	turn right
0	1	1	0	1	1	1	straight	0	1	turn right
1	1	1	0	1	1	1	straight	0	1	turn right
0	0	0	1	1	0	1	turn right	0	1	turn right
1	0	0	1	1	0	1	turn right	0	1	turn right
0	1	0	1	1	0	1	turn right	0	1	turn right
1	1	0	1	1	0	1	turn right	0	1	turn right
0	0	1	1	1	0	1	turn right	0	1	turn right
1	0	1	1	1	0	1	turn right	0	1	turn right
0	1	1	1	1	0	1	turn right	0	1	turn right
1	1	1	1	1	0	1	turn right	0	1	turn right

그림 6. 관찰 개체의 행동결정표
Fig. 6. Behavior table of some individuals

5. 결론

본 논문에서는 셀 공간상에서 개체의 진화적 특성을 모델링 및 시뮬레이션 할 수 있는 프레임워크를 제안하였다. 이러한 연구는 기존의 세포 자동자와 유사한 공간 개념을 갖으나 셀 공간에 상태가 존재하는 것과 다르게 개체가 존재해서 진화해간다는 새로운 특징이 있다. 이것을 통하여 개체와 환경 수준의 진화적 특성을 모사할 수 있다. 본 논문에서는 개체-먹이 응용을 통하여 본 논문에서 제안한 CC의 유용성을 살펴보았다. 시뮬레이션결과 초기 무작위로 생성된 개체내부의 행동결정 로직이 세대가 지나면서 빠르게 진화해서 수십 세대 만에 먹이를 잘 찾아가는 로직으로 진화됨을 볼 수 있었다. 본 연구에서는 단순히 현재의 관찰 상황만을 이용하여 행동을 결정하는 행동결정 로직 프레임으로 시뮬레이션 하였으나 향후 이전 결정 행동에도 영향을 받는 로직 프레임으로 변경하여 시뮬레이션해 볼 생각이다. 이렇게 할 경우 메모리 효과에 의하여 보다 창발적인 행동을 보일 수 있을 것으로 기대하고 있다. 또한 관찰영역을 확대해 본다던지 다른 개체의 존재도 인식하는 것으로 확장할 경우 보다 흥미로운 실험결과가 나올 것으로 기대하고 있다.

References

[1] Paul Davies, Mun-Ju Go translator, "*The fifth miracle: origin of life*," Bookshill, 2000.

[2] Gerhard Staguhn, Hye-Kyung Jang translator, "*Find the blueprint of life-genetics and evolution of life by miracle and coincidence facets*," HaeNaMoo, 2004.

[3] Charles Darwin, "*On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life,*" John Murray, 1859.

[4] Von Neumann, J. and A. W. Burks, "*Theory of self-reproducing automata*," Urbana, University of Illinois Press, 1966.

[5] Christopher G. Langton, "Self-reproduction in cellular automata," *Physica 10D*, pp. 135-144, 1984.

[6] D. Goldberg, "*Genetic Algorithms in Search, Optimization and Machine Learning*," Addison-Wesley, 1989.

[7] M. Srinivas and L. M. Patnaik, "Genetic Algorithms: A Survey," *IEEE Computer Magazine*, pp. 17-26, June 1994.

[8] L. Sekanina, "Evolutionary Design of Digital Circuits: Where Are Current Limits?," *Adaptive Hardware and Systems, First NASA/ESA Conference* on, pp.171-178, June, 2006.

[9] J. F. Miller, D. Job, and V. K. Vassilev, "Principles in the evolutionary design of digital circuits-Part I," *Genetic Programming and Evolvable Machines*, vol. 1, no. 1, pp. 8-35, 2000.

[10] J. Torresen, "A scalable approach to evolvable hardware," *Genetic Programming and Evolvable Machines*, vol. 3, no. 3, pp. 259-282, September, 2002.

[11] J. Torresen, "A divide-and-conquer approach to evolvable hardware," *in Proc. of the Second International Conference on Evolvable Systems: From Biology to Hardware*, vol. 1478, pp. 57-65. 1998.

[12] J. Torresen, "Evolving multiplier circuits by training set and training vector partitioning" *in Proc. 5th International Conference on Evolvable Systems*, Trondheim, Norway, Mar. 17-20, 2003, vol. 2606, Lecture Notes in Computer Science, pp. 228-237.

[13] V. Vassilev and J. F. Miller, "Scalability problems of digital circuit evolution evolvability and efficient designs," *in Proc. 2nd NASA/DoD Workshop on Evolvable Hardware*. Los Alamitos, CA: IEEE Computer Society, pp.55-64. 2001.

[14] J. F. Miller and P. Thomson, "Cartesian genetic programming," *in Proc. 3rd Eur. Conf. Genetic Programming (EuroGP 2000)*, Edinburgh, 2000, vol. 1802, Lecture Notes in Computer Science, pp. 121?132.

[15] Kwee-Bo Sim, Dong-Wook Lee, "Behavior Learning and Evolution of Individual Robot for Cooperative Behavior of Swarm Robot System," *Journal of The Korean Institute of Intelligent System*, vol. 16, no. 2, pp. 131-137, Mar 2006.

[16] Min-Kyung Kim, Kwang-Eun Ko, Kwee-Bo Sim, "Behavior Learning and Evolution of Swarm Robot based on Harmony Search Algorithm," *Journal of The Korean Institute of Intelligent System*, vol. 20, no. 3, pp. 441-446, Jun 2010.

저 자 소 개

정보선(Bo-Sun Jung)

2012년: 가천대학교 전자공학과 공학사
2013년 ~ 현재: 코아리버 연구원
2014년~현재 : 한성대학교 일반대학원 정보통신공학과 재학

관심분야 : 지능시스템
Phone : +82-10-9479-7738
E-mail : tommy15433@gmail.com

정성훈(Sung Hoon Jung)

1988년: 한양대학교 전자공학과 공학사
1991년: KAIST 전기 및 전자공학과 공학석사
1995년: KAIST 전기 및 전자공학과 공학박사
1996년~현재 : 한성대학교 정보통신공학과 교수

관심분야 : 지능시스템, 시스템생물학, 융합과학
Phone : +82-2-760-4344
E-mail : shjung@hansung.ac.kr