論文99-36S-12-2

DNA 코딩방법을 이용한 셀룰라 오토마타 신경망의 진화

(An Evolution of Cellular Automata Neural Systems using DNA Coding Method)

李 東 昱 *, 沈 貴 寶 **

(Dong-Wook Lee and Kwee-Bo Sim)

요 약

셀룰라 오토마타 신경망(CANS)은 생물학적 발생과 진화에 기반한 신경망 모델이다. CANS에서 각 뉴런은 상호간에 국소적인 연결을 갖고 있으며 카오스 뉴런 모델의 동작 방정식에 따라 펄스의 형태로 동작한다. 신경망은 초기 패턴을 셀룰라 오토마타(CA) 규칙에 따라 발생시켜 얻어진다. 기존의 연구에서는 유용한 기능을 얻기 위하여 초기패턴을 진화시켰다. 그러나 이 방법은 신경망의 표현공간을 모두 나타낼 수 없다. 따라서 본 논문에서는 신경망의 표현공간이 작아지는 문제점을 개선하기 위한 CA의 발생규칙을 진화시키는 방법을 제안한다. DNA 코딩은 코딩의 중복과 여분을 효과적으로 사용하며 규칙의 표현에 매우 적합하다. 본 논문에서는 CA 규칙의 일반적인 표현방법을 제시하고 DNA 코드를 CA 규칙으로 해석하는 방법을 제안한다. 제안된 방법은 자율이동로봇의 제어기에 사용하여 주행 문제에 적용함으로써 그 유효성을 확인하였다.

Abstract

Cellular Automata Neural Systems(CANS) are neural networks based on biological development and evolution. Each neuron of CANS has local connection and acts as a form of pulse according to the dynamics of the chaotic neuron. CANS are generated from initial cells according to the CA rule. In the previous study, to obtain the useful ability of CANS, we make the pattern of initial cells evolve. However, it is impossible to represent all solution space, so we propose an evolving method of CA rule to overcome this defect in this paper. DNA coding has the redundancy and overlapping of gene and is apt for the representation of the rule. In this paper, we show the general expression of CA rule and propose translating method from DNA code to CA rule. The effectiveness of the proposed scheme was verified by applying it to the navigation problem of autonomous mobile robot.

Ⅰ. 서 론

최근, 인공생명(artificial life), 신경망, 로보틱스와 관련된 국제적인 학회에서 인공두뇌(artificial brain)를 개발하기 위한 초보적인 논문이 많이 발표되고 있다[1~4]. 이러한 연구들 중에는 생물학적 두뇌 모델을 연구하여 이를 공학적으로 구현하기 위한 방법이 많다. 특히 기존의 신경망의 파라미터나 구조를 진화하던 단계에서 더 나아가 신경망의 합성규칙을 진화시키는 방법이 연구되고 있다. 이것은 생물체의 발생과 진화 메커니즘에서부터 힌트를 얻은 것으로서, 신경망의 크기가 커지더라도 진화의 복잡성이 증가하지 않는 장점이 있다. 따라서, 궁극적으로 인공 뇌와 같이 고도로 복잡한 시스템을 구축하는데 유용한 방법론이 될 수 있다.

* 正會員, 中央大學校 制御計測學科
(Dept. of Control and Instrumentation Engineering, Chung-Ang University)
** 正會員, 中央大學校 電子電氣工學部
(School of Electrical and Electronics Engineering, Chung-Ang University)
接受日字:1998年12月8日, 수정완료일:1999年11月16日

일반적으로 유전자 알고리즘(genetic algorithms)에서는 유전자의 배열이 바로 표현형의 구조를 나타낸다. 이 경우 표현형이 매우 복잡한 구조를 가진다면 유전형의 길이도 매우 길어지는 단점을 가진다. 따라서 본 논문에서는 유전형과 표현형의 변환 과정에 발생모델을 이용함으로써 적은 길이의 유전자형을 가지고 복잡한 표현형을 나타낼 수 있는 방법을 이용한다. 생물학적 발생을 모델링 한 도구로서 L-시스템, 셀룰라 오토마타(이하 CA) 및 세포의 분열을 모델링 한 방법 등이 있다. Boers[1], Gruau[2] 등은 L-시스템 기반의 신경망의 설계방법을 제안하였고, Garis[3] 등은 CA 기반의 인공 뇌를 개발하기 위한 연구를 수행하고 있다. 본 저자들은 CA와 진화 알고리즘(evolutionary algorithms)을 기반으로 한 새로운 형태의 신경망 구성방법을 제안하였다[5]. 우리는 이것을 진화적 셀룰라 오토마타 신경망(Evolutionary Cellular Automata Neural Systems)이라 하고 간단히 ECANS로 명명하기로 한다. 이전의 연구에서는 신경망을 진화시키기 위하여 초기셀의 배열을 염색체로 표현하였다. 그러나 이 방식에 의해서는 신경망의 모든 표현공간을 나타낼 수 없기 때문에 복잡한 문제에 대하여 성능이 떨어진다. 본 논문에서는 이러한 단점을 개선하기 위하여 CA의 발생규칙을 코드화하였다. 이것은 실제로 생물체가 발생하는 과정에서 각각의 세포가 유전정보(발생규칙) 및 주변 상태에 따라 발달하는 것과 유사하다. 본 논문에서는 DNA 코딩방법[6]을 이용하여 CA의 규칙을 표현하는 방법을 제안한다. 기존에는 CA의 규칙을 진화시키기 위하여 모든 상태가 고려되는 테이블 표현방식을 이용하였다[7]. 그러나 이 방법은 상태가 늘어남에 따라 염색체의 길이가 길어질 뿐만 아니라 유사한 상태에 대한 다음 상태의 결정에 대한 연관성이 없다. 따라서 모든 가능한 상태에 대하여 개별적인 다음 상태의 결정값을 갖으며 규칙의 일반성이 결여된다. 본 논문에서 제안하는 셀룰라 오토마타 규칙의 DNA 코딩방법은 이러한 문제점을 해결한다.

CANS를 진화적으로 설계하기 위하여 우리는 기본적인 신경망의 동작방식을 바꿀 필요가 있었다. 따라서 연결강도를 없애고 그 대신 뉴런의 종류와 배열이 신경망의 역할에 중요한 역할이 되도록 하였다. 신경망은 초기셀의 배열 및 규칙에 따라 다양한 형태를 갖게 되는데 이것은 CA의 발생에 의해 결정된다. CA에서 각각의 셀은 신경망의 뉴런에, 셀의 상태는 뉴런의 종류에 대응된다. 이때 셀은 주변 셀과의 연결관계에 따라 여러 가지 모양을 갖는다. 이러한 특징을 얻기 위하여 각 뉴런은 기존의 시그모이드 뉴런에 비하여 복잡한 특성을 갖는 카오스 뉴런[8]을 사용하였다. 카오스 뉴런은 실제 생물체의 뉴런에 가까운 모델이다. 카오스 뉴런 모델에서 출력값을 0과 1을 사용함으로써 네트워크의 신호는 펄스의 형태로 전달되고 이 펄스의 밀도에 따라 출력의 세기를 나타낼 수 있다.

2절에서는 본 저자가 제안한 ECANS에 대하여 설명하고 3절에서는 ECANS의 발생 규칙의 진화를 위한 DNA 코딩방법을 제안한다. 4절에서는 본 신경망을 이용한 시뮬레이션 결과를 소개한다.

Ⅱ. 진화적 셀룰라 오토마타 신경망(ECANS)

ECANS[5]는 발생과 진화에 기반한 새로운 형태의 신경망이다. 본 논문에서 사용하는 ECANS의 특징을 정리해보면 다음과 같다.

가. 신경망의 구성요소

(1) 뉴런의 특성방정식으로 카오스 뉴런 모델을 사용한다.

(2) 각 뉴런은 pulse(on/off)에 의해 동작한다.

나. 신경망의 구조

(3) 각 뉴런은 이웃하는 뉴런과만 연결을 갖는다.

(4) 뉴런간의 연결방식이 신경망의 특징을 결정한다.

다. 신경망의 생성방법

(5) 신경망의 발생은 CA의 규칙을 따른다.(발생규칙은 DNA 코딩방법으로 표현됨)

(6) 진화 방식에 의하여 기능을 획득한다.

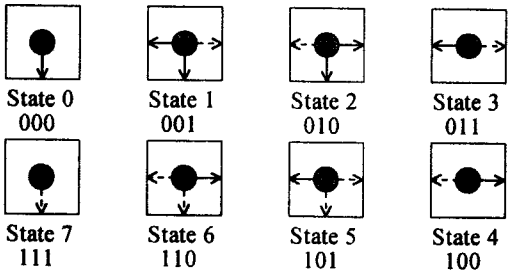

그림 1. 연결방식에 따른 셀의 상태(실선 : 흥분성 연결, 점선 : 억제성 연결)

Fig. 1. State of cell which is determined by it's connection type(solid line : excitatory synapse, dashed line : inhibitory synapse).

셀의 상태는 억제성 연결과 흥분성 연결을 갖는 셀의 상태가 비트로 표현했을 때 서로 보수 관계가 되도록 그림 1과 같이 설정하였다.

ECANS의 뉴런의 동작 방정식은 다음과 같은 카오스 뉴런모델[8]을 사용한다.

$$y(t+1) = u(x(t+1)) \tag{1}$$

$$x(t+1) = I(t) - \alpha \sum_{d=0} k^d y(t-d) - \theta \tag{2}$$

단, $y(t)$는 시간 t에서의 출력, $x(t)$는 내부 상태, $I(t)$는 입력, $u()$는 단위 계단 함수, k는 0에서 1사이의 값을 가지는 감쇠계수로서 뉴런의 카오스 특성을 결정한다. 상수 α는 양의 파라미터이며 θ는 뉴런의 문턱 값이다.

위의 방정식에 의하여 한 뉴런의 출력은 입력에 비례하는 펄스밀도를 갖는 신호를 출력한다(뉴런이 발화할 때 펄스 생성). 이때 뉴런의 카오스 특성 때문에 펄스의 배열은 비주기적이며 랜덤한 성질을 갖는다. 이것은 네트워크가 정상적으로 동작하도록 하는 역할을 한다. 그 이유는 만약 이웃하는 뉴런에 같은 신호가 들어왔을 경우 두 뉴런이 같은 주기의 펄스를 내보낸다면 위상차에 의해서 같은 경우라도 신호가 증폭되거나 소멸하는 등 다른 결과가 나타날 수 있다. 디지털 신경망에서도 이러한 문제를 해결하기 위하여 확률적 펄스 발생기를 이용한다. 그러나 카오스 뉴런은 자신의 과거 정보를 이용하고 펄스의 발생이 비주기적이기 때문에 신호의 증폭이나 소멸 등의 현상은 일어나지 않으며 별도의 확률적 펄스 발생기를 필요로 하지 않는다. ECANS에서 한 뉴런은 오직 이웃하는 뉴런과 연결을 갖으며 이때 연결강도는 -1, 0, 1중 하나를 갖는다. 즉 연결은 단지 흥분성인가 억제성 연결인가만 구분된다. 그림 2는 펄스형태로 동작하는 생물학적 뉴런의 모델을 나타낸다. 뉴런의 최종 출력은 펄스의 밀도에 의해 계산된다.

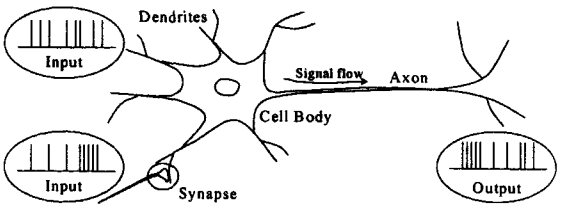

그림 2. 펄스에 의해 동작하는 생물학적 뉴런 모델
Fig. 2. Biological neuron model which acts as pulse.

그림 3은 CANS의 발생에 의해 신경망을 구성하는 방법을 나타낸다. 그림 1과 같은 셀들이 CA 규칙에 의하여 발생을 반복하여 신경망을 구성한다.

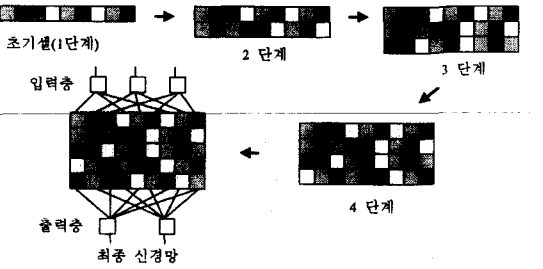

그림 3. CANS의 발생과정
Fig. 3. Developmental process of CANS.

Ⅲ. DNA 코딩에 의한 셀룰라 오토마타 신경망의 진화

1. 생물학적 DNA

모든 생물체는 각자 고유의 DNA를 가지고 있다. DNA는 개체의 특성을 발현시키는 유전코드로서, A

표 1. RNA(DNA) 코돈과 생성하는 아미노산
Table 1. RNA(DNA) codon and it's amino acid.

	U		C		A		G		
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
	UUC		UCC		UAC		UGC		C
	UUA	Leu	UCA		UAA	정지	UGA	정지	A
	UUG		UCG		UAG		UGG	Trp	G
C	CUU		CCU	Pro	CAU	His	CGU	Arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Gln	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	Lys	AGA	Arg	A
	AUG	Met	ACG		AAG		AGG		G
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glu	GGA		A
	GUG		GCG		GAG		GGG		G

아미노산 약어: 알라닌-Ala, 아르기닌-Arg, 아스파라긴-Asn, 아스파르트산-Asp, 시스테인-Cys, 글루탐산-Glu, 글루타민-Gln, 글리신-Gly, 히스티딘-His, 이소루신-Ile, 루신-Leu, 리신-Lys, 메티오닌-Met, 페닐알라닌-Phe, 프롤린-Pro, 세린-Ser, 트레오닌-Thr, 트립토판-Trp, 티로신-Tyr, 발린-Val.

* DNA에서는 U대신 T를 사용한다.

(아데닌) T(티민, RNA에서는 U:우라실) G(구아닌) C(시토신)의 4개의 염기배열로 이루어져 있다. 또한 염기 3개의 배열이 한 의미단위를 이루어 해석된다. 이 의미단위를 생물학적인 용어로 코돈(codon)이라 한다. 코돈의 가지수는 4^3 = 64개이며 이것이 코드화하는 아미노산은 20가지이다. 코돈의 64가지 패턴에 대하여 생성하는 아미노산이 20가지인 이유는 다른 코돈이 같은 아미노산을 만들기도 하기 때문이다. 이것은 표 1에 나타나 있다[9].

DNA는 RNA로 전사되어 리보솜에서 단백질로 번역된다. 즉 아미노산을 암호화하는 DNA의 배열에 따라 아미노산의 합성순서를 결정하여 여러 종류의 단백질을 만들어낸다. RNA의 단백질로의 번역은 AUG에서 시작해서 UGA(UAA,UAG)에서 번역이 끝난다.

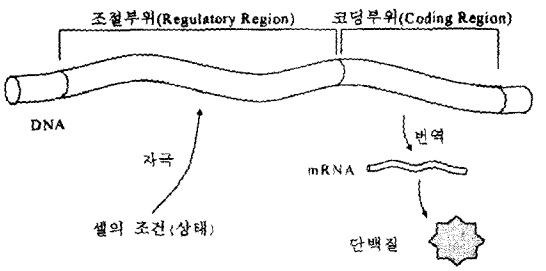

그림 4. 유전자 코드의 일반적인 구조
Fig. 4. General structure of genetic code.

생물학적 유전자의 기본적인 구조는 그림 4와 같다. 즉, 단백질을 직접 코드부위(coding region)와 그 코딩 부위의 발현을 조절하는 조절부위(regulatory region)로 구성되어 있다. 코딩부위는 조절부위의 명령에 의해 세포내의 조건이 조절부위를 자극할 때 단백질로 번역된다. 이것은 발생모델의 규칙의 표현방식과 유사하다. 즉, 조절부위는 규칙의 조건부 또는 세포의 현재 주변 상태, 코드화 부위는 결론부 또는 세포의 다음 상태에 대응된다.

2. DNA 코딩방법의 특징

DNA의 동작을 모방한 코딩방법에 대한 연구는 Yoshikawa[6] 등에 의하여 기본적인 방법이 연구되었다. 염색체는 기본적으로 4가지 염기의 배열로 이루어져 있고 아미노산을 번역하는 것과 마찬가지로 코돈 단위로 번역한다. 또한 그림 6과 같이 유전자의 번역 시작점이 일정하지 않기 때문에 중복 번역을 허용하며 때에 따라서는 번역되지 않는 부분도 존재한다. 표 1에서 알 수 있듯이 하나의 아미노산을 생성하는 코돈이 여러 개이기 때문에 염색체의 중복을 효율적으로 이용할 수 있다. 교차점도 임의로 주어지기 때문에 염색체의 길이도 가변적이다. Yoshikawa의 방법은 번역 테이블을 만듦으로서 규칙의 표현에 적합하게 구성되어 있다. 이상의 특징을 아래와 같은 4가지로 정리할 수 있다.

◆ DNA 코딩방법의 특징
(1) 자식의 융통성 있는 표현이 가능하다.
(2) 코딩에 여분과 중복이 있다.
(3) 염색체의 길이가 가변적이다.
(4) 교차점이 강제적으로 주어지지 않는다.

Wu[10] 등은 GA에서의 가변위치 표현법(floating representation)에 대한 스키마 분석을 통하여 그 유효성을 증명하였다. 가변위치 표현법이란 DNA 코딩에서와 같이 시작 기호(start tag)를 가지고 있어서 시작 기호가 발견되는 곳에서부터 유전자의 번역을 시작하는 방법이다. Wu는 비록 비트 스트링을 가지고 실험을 하였으나 이 방법도 DNA 표현 방법의 (2), (3), (4)의 특징을 모두 가지고 있다. 유용한 스키마를 찾는 능력에 대한 일반 GA방법(고정위치 표현법)과의 비교에서 그는 다음과 같은 결론을 얻어냈다. (a) 염색체의 길이가 짧을 경우에는 일반적인 고정위치 표현방법(fixed representation)이 우수했으나 염색체의 길이가 길수록 가변위치 표현법의 성능이 더 좋아진다. (b) 가변위치 표현법에서 개체군의 다양성이 더욱 높다. 따라서 매우 복잡한 문제에 대하여 적은 수의 개체군을 가지고 개체의 다양성을 최대로 유지하며 탐색을 수행한다.

3. CA 규칙의 표현

CA의 규칙을 코드화하고 번역하기 위해서는 표 1의 아미노산의 코드표와 같은 번역(decoding)표가 필요하다. 우선 CA의 규칙을 표현하기 위한 방법을 결정하고 이에 따라 필요한 표를 작성해 나간다.

1차원 CA에서 셀의 이웃반경을 r이라 했을 때 규칙(ϕ)을 나타내는 일반적인 표현은 (3)식과 같다.

$$\sigma_i^{new} = \phi(\sigma_{i-r}, \cdots, \sigma_{i+r}) \tag{3}$$

이웃반경(r)이 1인 경우를 생각할 때 j 단계의 i 번째 셀의 상태와 그 이웃 상태에 따라 $j+1$ 단계의

i 셀의 상태는 결국 σ_{i-1}^{t}, σ_{i}^{t}, σ_{i+1}^{t} → σ_{i}^{t+1}의 함수이다. 또한 함수 ϕ가 n 개의 부분규칙 함수로 이루어져 있다고 가정하면 (3)식의 ϕ는 (4)식과 같이 표현할 수 있다.

$$\phi = \begin{cases} \phi_1, & \text{조건1} \\ \phi_2, & \text{조건2} \\ \cdots & \\ \phi_n, & \text{조건}n \end{cases} \tag{4}$$

단, n은 최대 (총 상태)$^{2r+1}$의 값을 가질 수 있다.

(4)식에서 조건1 ~ 조건n은 CA에서 현재의 이웃에 의하여 결정되는 셀의 현재 상태이다. 본 논문에서는 이 조건부를 어떤 이웃 셀이 다음 상태에 영향을 주는가에 따라서 (5)~(7)식의 3가지 형태(type)로 나누었다.

$$\text{Type 1: If } \sigma_n = s_1 \tag{5}$$

$$\text{Type 2: If } \sigma_l = s_1 \text{ and } \sigma_m = s_2 \tag{6}$$

$$\text{Type 3: If } \sigma_{i-1} = s_1 \text{ and } \sigma_i = s_2 \text{ and } \sigma_{i+1} = s_3 \tag{7}$$

단, s_1, s_2, s_3는 각각 셀의 상태값 중 하나를 나타내며 $l, m, n \in \{i-1, i, i+1\}$, $l \neq m$ 이다.

(5)~(7)식은 함수에 따라 각각 한 개, 두 개, 또는 세 개의 이웃상태를 나타내는 조건부이다.

여기에서 (7)식을 조건으로 사용한 규칙은 현재 이웃하는 셀의 상태를 모두 고려하는 상태로 매우 세부적이며 특정한 규칙이 된다. 이것을 type 3 규칙으로 나타낸다. 또한 (5)식과 (6)식의 조건에 의해 결정된 규칙은 여러 가지의 상태를 포함하는 포괄적인 규칙으로 각각 type 1 및 type 2 규칙이 된다. 이때 다음상태를 결정하는 부분규칙(결론부) ϕ_k는 다음에 정의하는 논리 연산자(operator)를 사용하여 구성하였다.

표 2. ϕ_k에서 사용되는 비트 논리 연산자
Table 2. Bit logic operators used in ϕ_k.

No.	Operator	# of operand	Method
0	3XOR	3	σi-1^σi^σi+1
1	AND	2	σl & σm
2	OR	2	σl | σm
3	ADD	2	(σl + σm) % q
4	DIFFERENCE	2	|σl - σm|
5	XOR	2	σl ^ σm
6	NOT	1	~σn
7	State	1	sl

4. DNA 코딩방법에 의한 CA 규칙의 번역

DNA가 번역되는 기본단위는 3개 단위의 염기인 코돈이다. DNA 코드는 시작 코돈이 발견된 시점부터 표 2~4를 참조하여 그림 5의 흐름도에 따라 번역한다. 조건부에서 규칙의 상태를 고려하며, 결론부에서 피연산자의 순서를 결정하기 위한 셀의 순서를 나타내는 Order(표 4)를 도입하였고 A/B(And/Break)를 이용해 번역의 흐름을 제어하였다. 즉 A/B의 값에 따라 조건부에서 몇 개의 상태를 고려할 것인가가 결정된다. Type 3 규칙이 나타나기 위해서는 A/B에서 모두 A(and)가 나와야 하며 type 1의 규칙은 첫 번째 A/B에서 B(break)가 나와야 한다. 결론부의 연산(operation)은 기본적으로 두 가지 셀의 상태를 가지고 수행하나 NOT은 하나의 상태, 3XOR은 세 가지 상태를 이용한다. 또한 연산자(operator)에서 state가 선택된 경우 다음 코돈을 해석하여 바로 다음 상태를 결정한다. 이와 같이 CA의 규칙을 표현함으로써 기존의 테이블 표현방식에 비해 인접한 셀 간에는 공통된 성질을 갖게 할 수 있으며 적은 수의 규칙을 가지고 많은 상태를 나타낼 수 있다.

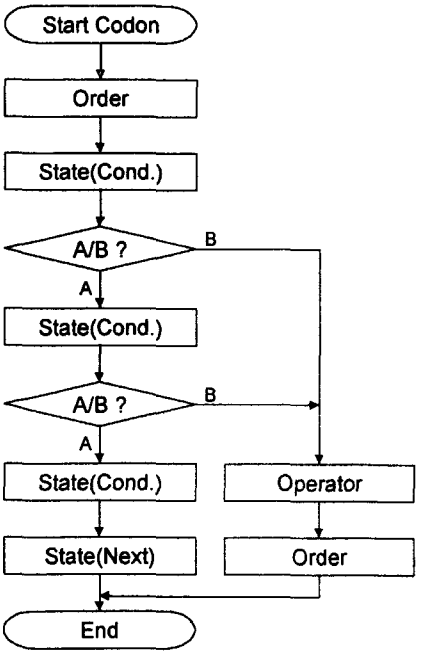

그림 5. DNA 코드의 번역 순서(각 블록은 코돈을 나타냄)
Fig. 5. Flowchart of translating DNA code(each block represents the codon).

표 3은 DNA 코돈을 해석하기 위하여 각 코돈에 대하여 order, state, A/B, operation을 대응시킨 표이다. 또한 표 4는 규칙에서 상태(state)를 고려할 때 순서(order)를 나타내는 표이다. 코돈의 배열 순서에 따라 그림 4와 표 2~4을 참조하면 셀룰라 오토마타의 부분규칙이 생성된다. 표 3을 설계하는 규칙은 특별히 없으나 생물학적인 연구[11]에 따르면 하나의 코돈이 돌연변이를 일으켰을 때 다른 어떠한 종류의 상태로 바뀔 수 있도록(changeability) 설계하는 것이 좋다.

표 3. 아미노산의 해석 표
Table 3. Translation table of amino acid.

Amino Acid	Phe	Leu	Val	Ser	Pro	Thr	Ala	Tyr	His
Order	0	2	1	3	4	1	5	1	3
State	0	0	1	3	7	2	7	4	5
And/Break	A	A	A	B	A	A	A	A	A
Operator	0	0	1	3	7	2	7	4	5
Amino Acid	Gln	Asn	Lys	Asp	Glu	Cys	Arg	Gly	
Order	1	5	4	4	2	5	0	4	
State	5	2	5	1	6	3	4	6	
And/Break	A	A	A	A	A	B	B	B	
Operator	5	2	5	1	6	3	4	6	

* Ile, Met, Trp 및 정지 코돈은 시작 코돈으로 사용하기 위해 사용하지 않음

표 4. 첨자의 순서
Table 4. Order of index.

No.	첨자의 순서	No.	첨자의 순서
0	i-1 → i → i+1	3	i → i+1 → i-1
1	i-1 → i+1 → i	4	i+1 → i-1 → i
2	i → i-1 → i+1	5	i+1 → i → i-1

생물체의 세포에서 만들어지는 단백질은 일반적으로 최소 55개에서 많게는 수백 개의 아미노산으로 구성되어 있다. 따라서 한 개의 시작 코돈만 가지고도 충분히 유전적인 암호를 코드화 할 수 있다. 그러나 하나의 셀룰라 오토마타 부분규칙은 6~8개의 코돈으로 구성되므로 1/64의 비율로 나타나는 하나의 시작 코돈을 가지고 진화를 하는 것은 염색체를 상당히 비효율적으로 사용하는 것이 된다. 따라서 본 실험에서는 4개의 코돈을 시작 코돈으로 사용하였다.

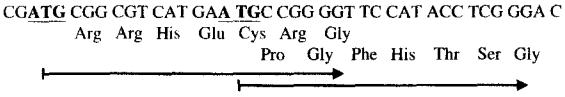

그림 6. DNA 염색체의 번역
Fig. 6. Translation of DNA chromosome.

그림 6은 DNA 염색체의 예이다. 여기에는 규칙으로 해석할 수 있는 시작 코돈이 두 개 발견된다. 이 중 첫 번째 규칙을 해석하면 다음과 같다. 우선 시작 코돈인 ATG에서 번역이 시작되고 차례로 3개씩(코돈) 다음과 같이 번역된다.

① ATG : start codon
② Arg : order(0: i-1 → i → i+1)
③ Arg : state(4)
④ His : A/B(And)
⑤ Glu : state(6)
⑥ Cys : A/B(Break)
⑦ Arg : operator(DIFFERENCE)
⑧ Gly : order(4: i+1 → i-1 → i)

이것을 규칙으로 표현하면 『If $\sigma_{i-1}^{j}=4$ and $\sigma_{i}^{j}=6$ then $\sigma_{i}^{j+1}=|\sigma_{i+1}^{j}-\sigma_{i-1}^{j}|$ 』 과 같이 된다. ②번의 order는 조건부 상태의 순서를 결정하는 것이고 ⑧번의 order는 결론부 상태의 순서를 결정하는 것이다. 이 규칙은 두개의 이웃 상태를 고려하고 왼쪽(i-1)과 오른쪽(i+1) 셀의 상태의 차이 값으로 다음의 상태를 구하는 규칙으로서 전건부에서 2개의 상태를 고려하므로 type 2 규칙에 해당된다. 또한 두 번째로 나타나는 규칙을 번역해 보면 아래와 같다.

① ATG : start codon
② Pro : order(4: i+1 → i-1 → i)
③ Gly : state(6)
④ Phe : A/B(And)
⑤ His : state(5)
⑥ Thr : A/B(And)
⑦ Ser : state(3)
⑧ Gly : next state(6)

이것을 규칙으로 나타내면 『If $\sigma_{i-1}^{j}=5$ and $\sigma_{i}^{j}=3$ and $\sigma_{i+1}^{j}=6$ then $\sigma_{i}^{j+1}=6$』 가 된다. 이것은 전건부

에서 3개의 상태를 모두 고려하므로 type 3 규칙이다.

5. 발생모델 설계를 위한 Bit 코딩과 DNA 코딩

진화적 기법을 통하여 유용한 CA의 규칙을 얻어내는 시스템을 진화 CA(Evolving CA)라고 한다. 이 진화 CA는 패턴분류문제, 복잡성의 진화실험, 신경망 발생 등에 이용된다. 이때 CA의 규칙 진화를 위하여 주로 사용되는 방식은 비트 열(bit string) 코딩방법이다. 하나의 셀이 처할 수 있는 상태를 테이블로 만들고 각각에 대한 셀의 다음 상태를 비트 열로 표현하는 방법이다.

1차원 CA에서 이웃(neighborhood)을 r, 상태(state)를 k라 할 때 가능한 이웃의 조합은 k^{2r+1}이다. 따라서 규칙을 표현하기 위해서는 k^{2r+1}의 크기를 갖는 테이블이 필요하다. 예를 들어 $k = 5$, $r = 2$ 일 때 규칙 테이블의 크기는 $5^{2\cdot 2+1}$, 즉 3125가 된다. 이 방법은 k와 r이 증가함에 따라 테이블의 크기가 지수적으로 증가하기 때문에 큰 k와 r에 대하여 규칙 테이블 자체를 진화시키기는 불가능하다. 따라서 이와 같은 CA의 규칙을 진화시키기 위해서는 다른 방법이 필요하다.

한편, CA의 규칙테이블이 아무리 크다하더라도 거기에 있는 모든 규칙을 다 사용하는 것은 아니다. 즉, 사용하는 규칙도 있고 사용하지 않는 규칙도 있다. 따라서 사용하는 규칙만 표현할 수 있으면 염색체의 크기를 줄일 수 있다. 또한 일반적으로 CA의 규칙은 테이블 형태가 아닌 함수의 형태로 표현할 수 있다. 규칙을 함수의 형태로 표현함으로서 표현해야할 양을 줄일 수 있고 상태간의 일관성을 유지할 수 있다. 본 논문에서는 제안한 DNA 코딩을 이용함으로서 이 두 가지 문제를 해결하였다.

그림 7은 CA의 규칙을 비트 열(bit string) 및 DNA 코드로 표현했을 때 상태 수에 따라 필요한 염색체의 길이를 나타낸 그래프이다. 이때 이웃반경(r)은 1이다. 주어진 CA의 규칙을 비트 열로 표현할 경우 필요한 비트 수 l_{bit}은 (8)식과 같이 구할 수 있다. 한편, DNA 코드의 길이는 진화에 따라서 가변이고 중복과 여분을 포함하기 때문에 특정한 숫자로 나타내기는 어렵지만 일반적으로 주어진 문제가 복잡할수록 규칙이 더 많이 필요해지기 때문에 염색체의 길이가 길어질 것이다. 그러나 문제가 고정되고 단지 상태 수만 늘어나면 DNA 코드의 길이에는 별 영향을 주지 않을 것이다.

$$l_{bit} = [\log_2 k] \times k^{2r+1} \tag{8}$$

단 []는 가우스 기호를 나타낸다.

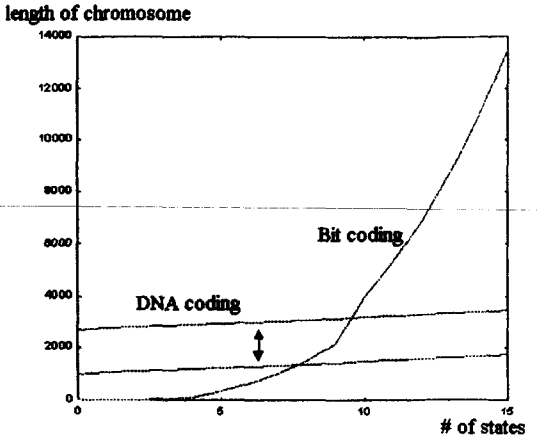

그림 7. 상태 수에 따른 염색체의 길이
Fig. 7. The length of chromosome vs number of states

비트 열 코딩방법의 염색체 길이는 상태가 늘어남에 따라 지수적으로 늘어난다. 반면 DNA 코딩의 염색체의 길이는 주어진 문제의 복잡도에 의존하기 때문에 상태 수가 증가함에 따라 여분의 규칙이 코딩되기 때문에 약간 증가한다. 그림에서와 같이 일반적으로 상태의 수가 클수록 비트 열 코드의 길이는 진화를 시키기에는 너무 커져 버리는 것을 알 수 있다. 반면 DNA 코딩은 상태 수보다는 문제의 복잡도에 따라 필요한 규칙 수에 의존한다.

Ⅳ. 시뮬레이션 결과

진화적 셀룰라 오토마타 신경망(ECANS)의 유효성을 입증하기 위하여 자율이동로봇의 미로주행 문제에 적용하였다. 시뮬레이션을 위하여 사용한 미로(maze)는 그림 10과 같다. 이 미로는 좌회전, 우회전, 좌로 180도 회전, 우로 180도 회전 등의 상황이 고려되어 설계되었다. 로봇의 제어기는 전방, 좌전방, 우전방, 좌측, 우측의 벽까지의 거리를 입력으로 받으며 두 바퀴의 힘을 출력한다.

셀룰라 오토마타 신경망의 진화를 위한 파라미터는 아래와 같이 설정하였다. DNA 코드의 중복과 여분을 고려하여 시작 코돈의 개수를 4개로 하였고 초기 염색

체의 길이는 1024로 하였다. 교차(crossover) 연산 수행시 두 염색체의 교차점이 각각 정해지므로 자손의 염색체의 길이는 변하게 된다. 따라서 적당한 초기길이를 설정하여도 알맞은 길이의 염색체로 진화해 나간다.

DNA 코딩방법은 고정된 위치가 특별한 의미를 가지지 않고 교차점이 일정하지 않기 때문에 염색체의 다양성이 매우 높다. 따라서 적합도 비례 선택법은 선택 압력이 잘 작용하지 않고 오히려 랜덤 탐색에 가까워진다. 결국 선택압력을 높이는 방법이 필요한데 본 논문에서는 Breeder GA(사육자 GA)를 사용함으로서 선택압력을 유지하였다. Breeder GA는 적합도가 우수한 개체(breeder)에게만 재생산의 기회를 줌으로서 선택압력을 높이는 방법이다. Breeder GA에서는 μ 개의 사육자(breeder) 개체를 가지고 유전연산을 거쳐 λ 개의 자손개체를 생성한다. 일반적으로 $\lambda > \mu$ 이다. 다음세대의 사육자(breeder)는 이전세대의 μ 개의 사육자와 λ 개의 자손 중에서 다시 μ 개를 선택한다. 이 선택방법은 진화전략(ES)의 $(\mu+\lambda)$-ES와 유사하다. ES와 비교해보면 μ 개의 부모개체를 순위 선택법에 의해 뽑는다는 것은 같지만 ES에서는 일단 부모로 선택되면 자손을 생성할 기회가 똑같이 주어지지만 Breeder GA에서는 적합도에 비례하게 주어진다는 점이 다르다. 본 시뮬레이션에서 μ=150, λ=350으로 설정하였다.

돌연변이의 탐색능력(exploration)과 이용(exploitation)능력의 균형을 유지하기 위하여 두 가지 돌연변이 확률을 도입하였다. 즉, 개체에 따라서 0.01과 0.05의 돌연변이율을 적용하였다. 이것을 Doi[12]의 비대칭 돌연변이(asymmetric mutation)라 한다. 한편, 교차율은 0.8로 하였다.

로봇의 제어기로서 신경망의 크기는 그림 9와 같이 5×10의 크기를 가지며 초기 10개의 셀로부터 나머지 셀을 CA 규칙에 따라 발생시킨다. 미로주행을 위한 로봇의 적합도는 (9)식과 같다.

$$fit = \frac{dist_R}{dist_M} \times w_1 + \left(1 - \frac{col_R}{col_{\max}}\right) \times w_2 \qquad (9)$$

단, w_1 및 w_2는 각각 거리와 충돌에 대한 중요성을 나타내는 가중치로서 0.8, 0.2로 설정하였다. $dist_R$은 로봇이 미로 중앙을 따라 움직인 거리, $dist_M$ 은 미로의 중앙을 따라 처음부터 끝까지의 총 거리, col_R은 충돌 횟수, $col_{\max}$(=100) 충돌회수의 최대 설정값이다. 이때 $col_{\max} < col_R$ 인 경우 $col_R = col_{\max}$로 하였다.

그림 8은 세대에 따른 최고 개체와 사육자 및 평균의 적합도 변화 그래프이고, 그림 9는 127세대 진화 후 얻어진 최고 개체의 신경망의 구조를 나타낸다. 이때 염색체의 길이는 1270이였다. 그림 10은 최고 개체의 미로 주행모습이다.

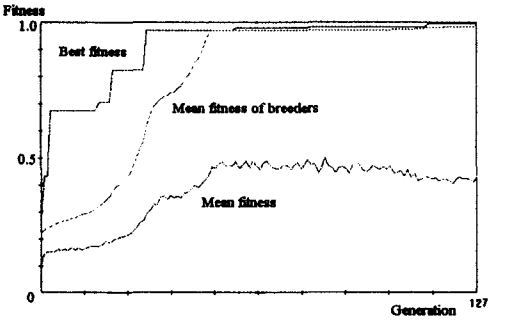

그림 8. 적합도 변화 그래프
Fig. 8. Graph of fitness change.

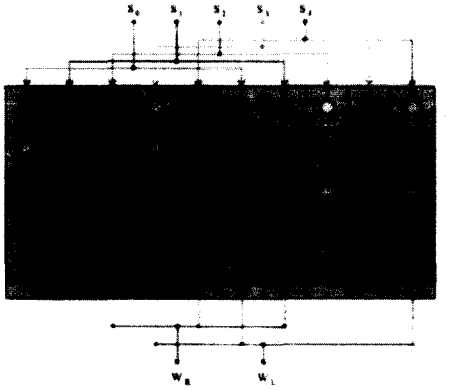

그림 9. 획득한 진화적 셀룰라 오토마타 신경망
Fig. 9. Obtained evolutionary cellular automata neural systems.

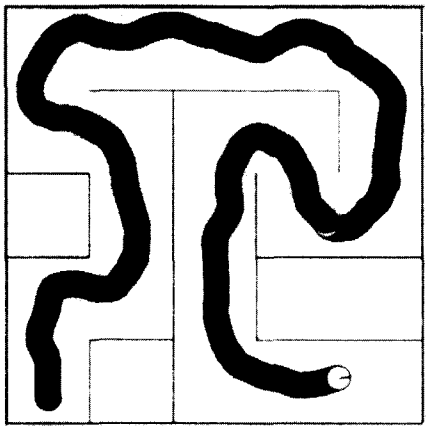

그림 10. 최고 개체의 미로 주행 결과
Fig. 10. Trace of best individual.

위의 시뮬레이션을 5회 수행했을 때 얻어진 결과는 표 5와 같다. 매 수행 시 충돌 없이 목적지에 도달하는 로봇의 제어기를 구할 수 있었다.

표 5. 시행회수 별 최고 적합도 개체의 획득 세대

Table 5. Trial number vs obtaining generation of best individual.

회수	1	2	3	4	5	평균
세대	127	44	153	104	125	110.6

V. 결 론

본 논문에서는 진화적 셀룰라 오토마타 신경망(ECANS)의 구성방법 및 효과적인 진화방법을 제안하였다. 뉴런을 하나의 셀로 여기고 이것을 CA의 규칙에 따라 발생하여 신경망을 구성하였다. 신경망의 작동을 연결강도가 아닌 연결방식으로 바꾸기 위하여 카오스 뉴런을 도입하였고 신호의 세기는 펄스 밀도로서 표현되도록 하였다. 본 신경망은 CA의 규칙에 따라서 다양한 형태를 갖게 된다. 본 논문에서는 문제에 적합한 신경망을 발생시키는 규칙을 찾기 위하여 규칙을 염색체로 코드화 하여 진화를 시켰다. 일반적으로 규칙을 직접 코드화 하면 염색체의 길이가 매우 길어져 간단한 문제를 진화시키는 데에도 시간이 오래 걸리게 된다. 본 논문에서는 이러한 문제점을 개선하기 위하여 효율적인 규칙의 진화를 위한 DNA 코딩방법을 제안하였다. 규칙을 일반적인 규칙(type 1, type 2)과 세부적인 규칙(type 3)으로 나누어 적은 수의 규칙으로 다양한 표현이 가능하도록 하였다. 제안된 방법에 의하여 주어진 목적을 수행할 수 있는 신경망 제어기가 구성되는 가를 간단한 시뮬레이션을 통하여 알아보았다.

인공생명은 기본적인 구성요소 및 법칙을 가지고 생물학적 현상을 합성하는 상향식 접근방식을 취한다. 반면 기존의 생물학은 생물의 특성을 분석하여 그 원리를 밝혀 내는 하향식 접근방법을 이용한다. 본 논문에서 제안한 신경망은 인공생명의 상향식 접근방법에 의하여 설계되었다. 즉 신경망의 구성요소의 특징을 설정하고 이것을 발생 및 진화를 통하여 이 구성요소의 상호작용에 의한 전체적인 유용한 특성이 나타나도록 하였다. 앞으로는 이와 같이 설계된 신경망을 분석하여(하향식 방법) 더 발전된 구조를 설계하는데 이용할 예정이다. 본 신경망은 재구성 가능한 반도체 소자(FPGA)로 구성하기에 매우 적합한 구조를 가지고 있으며 현재 제안한 신경망을 하드웨어로 구현하는 문제도 연구하고 있다.

참 고 문 헌

[1] E.J.W. Boers, H. Kuiper, B.L.M. Happel, and S. Kuyper, "Designing Modular Artificial Neural Networks," *Proceedings of Computer Science in the Netherlands*, pp. 87-96, 1993.

[2] F. Gruau, D. Whitley, "Adding Learning to the Cellular Development of Neural Networks: Evolution and the Baldwin Effect," Evolutionary Computation, vol. 1-3, pp. 213-233, 1993

[3] Hugo de Garis, "CAM-BRAIN: The Genetic Programming of an Artificial Brain Which Grows/Evolves at Electronic Speeds in a Cellular Automata Machine," *Proceedings of The First International Conference on Evolutionary Computation*, vol. 1, pp. 337-339b, 1994.

[4] M. Sugisaka, "Design of an Artificial Brain for Robots," *Proceedings of The Third International Symposium on Artificial Life and Robotics*, pp. (I-2)-(I-11), 1998.

[5] 이동욱, 심귀보, "셀룰라 오토마타 기법을 이용한 신경망의 자동설계에 관한 연구," *전자공학회 논문지*, 제35권 S편, 제11호, pp. 88-95, 1998.

[6] T. Yoshikawa, T. Furuhashi, Y. Uchikawa, "Effects of New Mechanism of Development from Artificial DNA and Discovery of Fuzzy Control Rules," *Proceedings of IIZUKA '96*, pp. 498-501, 1996.

[7] M. Sipper, "Non-Uniform Cellular Automata: Evaluation in Rule Space and Formation of Complex Structures," *Artificial Life IV*, The MIT Press, pp. 394-399, 1994.

[8] 合原一幸 編著, *ニュラルシステムにおけるカオス*, 東京電氣大學出版局, 1993.(정호선, 여진경 공역, *뇌와 카오스*, Ohm사, 1994.)

[9] R.A. Wallace, G.P. Sanders, R.J. Ferl, *BIOLOGY : The Science of Life 3rd eds.*, HarperCollins Publishers Inc., 1991.(이광웅 외 공역, *생물학*, 을유문화사, 1993.)

[10] A.S. Wu, R.K. Lindsay, "A Comparison of the Fixed and Floating Building Block Representation in Genetic Algorithm," *Evolutionary Computation*, vol. 4, no. 2, pp. 169-193, 1996.

[11] Tetsuya Maeshiro, "Prediction of Deviant Genetic codes-Why They Evolve-," *Proceedings of The Fourth International Symposium on Artificial Life and Robotics*, vol. 1, pp. 258-261, 1999.

[12] H. Doi, K. Wada, M. Furusawa, "Asymmetric Mutations Due to Semiconservative DNA Replication: Double-stranded DNA Type Genetic Algorithms," *Artificial Life IV*, pp. 359-364, 1994.

저 자 소 개

李 東 昱(正會員) 第 36 卷 S 編 第 6 號 參照

1998년~현재 중앙대학교 제어계측학과 박사과정. 주관심분야 : 인공생명, 인공두뇌, 인공면역계, 자율분산시스템, 가상현실 등

沈 貴 寶(正會員) 第 36 卷 S 編 第 6 號 參照

1991년~현재 중앙대학교 전자전기공학부 교수. 주관심분야 : 인공생명, 진화연산, 지능로봇시스템, 뉴로-퍼지 및 소프트 컴퓨팅, 자율분산시스템, 로봇 비전, 진화하는 하드웨어, 인공면역계 등