한국지능정보시스템학회논문지 제12권 제3호 2006년 9월(pp.67~82)

규칙 구성자와 연결 구성자를 이용한 혼합형 행동 진화 모델

박사준
대구한의대학교 모바일콘텐츠학부
(*phdjoon@dhu.ac.kr*)

가상 로봇의 행동 진화를 위해서 규칙 구성자와 연결 구성자를 구성하여 분류 규칙과 진화 신경망을 형성하는 혼합형 행동 진화 모델(Hybrid Behavior Evolution Model)을 제안한다. 본 모델에서는 행동 지식을 두 수준에서 표현하였다. 상위 수준에서는 규칙 구성자와 연결 구성자를 구성하여 표현력을 향상시켰다. 하위 수준에서는 행동 지식을 비트 스트링 형태의 염색체로 표현하여, 이들 염색체를 대상으로 유전자 연산을 적용하여 학습을 수행시켰다. 적합도가 최적인 염색체를 추출하여 가상 로봇을 구성하였다. 구성된 가상 로봇은 주변 상황을 인식하여 입력 정보와 규칙 정보를 이용하여 패턴을 분류하였고, 그 결과를 신경망에서 처리하여 행동하였다. 제안된 모델을 평가하기 위해서 HBES(Hybrid Behavior Evolution System)를 개발하여 가상 로봇의 먹이 수집 문제에 적용하였다. 제안한 시스템을 실험한 결과, 동일한 조건의 진화 신경망보다 학습 시간이 적게 소요되었다. 그리고, 규칙이 적합도 향상에 주는 영향을 평가하기 위해서, 학습이 완료된 염색체들에 대해서 규칙을 적용한 것과, 그렇지 않은 것을 각각 수행하여 적합도를 측정하였다. 그 결과, 규칙을 적용하지 않으면 적합도가 저하되는 것을 확인하였다. 제안된 모델은 가상 로봇의 행동 진화에 있어서 기존의 진화 신경망 방식 보다 학습 성능이 우수하고 규칙적인 행동을 수행하는 것을 확인하였다.

논문접수일 : 2006년 6월 게재확정일 : 2006년 9월 교신저자 : 박사준

1. 서론

가상 로봇들을 위한 성공적인 행동 학습 시스템을 개발하기 위해서는 행동 지식을 획득하고 수행하는데 있어서 비용이 적어야 한다. 여러 상황들에 대한 행동 지식들을 일반적인 프로그래밍 방법과 전통적인 인공지능 접근방법이 있다. 일반적인 프로그래밍 방법은 표현 방법에 있어서 효율성도 떨어지고 오류 발생의 가능성이 매우 높다. 전통적인 인공지능 접근방법은 비순서적인 규칙들과 사실들을 이용하여 행동 지식들을 표현한다. 그러나, 복잡한 추론 과정으로 인하여실시간 응답성이 좋지 않기 때문에 실용성이 떨어진다. 그리고, 기계학습 접근방법으로, 유전자 프로그래밍 방식은 구조적으로 표현된 프로그램을 자동으로 생성한다. 그러나, 학습 과정 중에 인간이 개입하여 응용 분야에 적합한 종단 노드와 함수 노드를 정의해야 한다. 이렇게 개발자에 의해서 직접 코딩되어야 하는 것은 시스템의 개발 및 정확성에 많은 부하를 초래할 수 있으며, 문제 영역에 대한 깊은 이해가

필요하다. 성공적인 행동 학습 시스템을 개발하기 위해서는 최소한의 노력으로 학습이 이루어질 수 있는 방법이 제공되어야 한다.

본 논문에서는 문제 영역에 대한 사전 지식을 제공하지 않는 형태의 혼합형 행동 진화 모델을 제안한다. 혼합형 행동 진화 모델은 분류 규칙과 진화 신경망을 이용하여 행동 지식들을 표현하고, 유전자 알고리즘에 의해서 학습을 수행한다. 분류 규칙과 진화 신경망이 염색체 상의 이진화된 비트 스트링의 형태로 표현되고, 이를 유전자 알고리즘을 이용하여 학습시킨다. 비트 스트링 형태로 분류 규칙을 표현하기 위해서는 규칙 구성자를 사용하며, 진화 신경망을 표현하기 위해서는 연결 구성자를 사용한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 검토하고 3장에서는 모델을 제시한다. 그리고 4장에서는 모델을 적용하여 시스템을 구현과 실험에 대해서 기술한다. 마지막 5장에서 결론을 언급한다.

2. 관련 연구

자율적인 이동 에이전트 또는 로봇 등에 있어서 주어진 상황에서 적절한 행위(action)들을 결정하기 위한 방법으로서 계획자(planner) 방식과 반응형(reactive) 방식이 있다. 계획자 방식은 하향식(top-down) 방법으로 감지된 정보들을 내부적인 월드 모델(world model)에 이용하여 가장 적절한 행위의 순서들을 결정하는 방식이다. 그러나 환경이 불확실하고 변화하는 경우에는 빈번하게 재계획을(replanning) 수행해야 하기 때문에 복잡한 시스템에서는 너무 많은 응답 시간의 지체를 초래할 수 있다. 따라서 거의 이용되지 않는다.

이에 비하여 반응형 방식은 미리 정해진 조건-행위(condition-action)들로부터 지체 없이 행위를 수행하는 최소한의 내부 상태만을 갖는 방식이다. 이 방식은 월드(world)에 대한 모델이 전혀 없고 자극에 대해서 적절한 응답만이 있을 뿐이다. 단점으로는 정보를 동적으로 저장할 수 있는 능력이 없기 때문에 수행시의 융통성이 부족하다.

행동 기반 시스템(BBS : Behavior Based System)은 계획자 방식과 반응형 방식의 중간에 위치하는 것으로서 리액티브 컴포넌트와 다양한 형태의 상태 표현을 보유한다. 행동 기반 시스템의 특징들을 살펴보면 다음과 같다(Mataric, 1997; Brooks, 1991a).

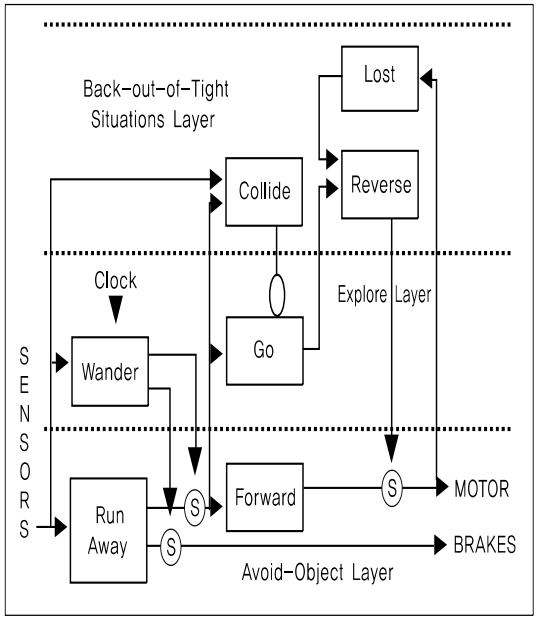
1) 하나 또는 그 이상의 추론 엔진(reasoning engine)에서 작동하는 중앙 집중화된 표현(representation)을 채용하지 않는다.
2) 전형적으로 다양한 형태의 분산된 표현들과 그 위에서 동작하는 분산된 계산에 의존한다.
3) 표현은 정적이지 않고 조작할 수 있는 구조이며 능동적이며 절차적이다.
4) 창발적(emergent)이다. 즉, 시스템 어디에서나 가시적으로 명시하지 않았어도 나타나는 특징이다.

행동 기반 시스템은 다수의 가상 로봇들을 제어하는 문제에 있어서, 특히 다수가 학습을 하는 경우에 매우 효과적인 방법이다. BBS는 반응적인 시스템이 실시간 응답성, 확장성, 강건성(robustness)에도 불구하고 과거 또는 미래에 대한 상태를 고려하지 않아서 발생하는 문제점을 해결하기 위해서 발전해왔다.

2.1 포섭 구조

대표적인 행동 기반 시스템으로는 Brooks에 의하여 1986년 소개된 포섭 구조이다(Brooks, 1986;

Brooks, 1987). 각각의 행동이 층(layer)로 표현되며 여러 층이 모여서 과제(task)를 수행하도록 되어 있다. 층마다 별도의 목표를 갖고서 자신의 역할을 병렬적으로 수행하도록 되어 있으며, 각 계층은 AFSM(Augmented Finite State Machine)이라는 모듈들로 구성되는데, 입력에 따라서 행위(action)를 출력하는 가장 단순한 함수로서의 기능을 수행한다. 하위층에 속하는 AFSM은 상위층으로부터 억제 또는 차단될 수 있다. 즉, 상위층이 하위층을 총괄하는 구조라고 볼 수 있다. 하위층은 상위층의 존재를 알 수 없으며 자신의 동작만을 계속 수행할 뿐이다. 도식적으로 구조를 살펴보면 [그림 1]과 같다.



[그림 1] Brooks의 포섭 구조

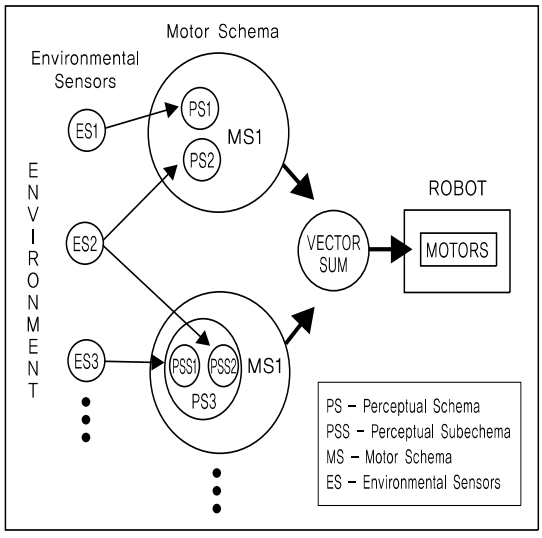
Brooks의 포섭 구조는 3개의 계층으로 구성되어 있으며 각각의 계층은 장애물 회피층(Avoid-Object Layer), 탐색층(Explore Layer), 협소 공간에서의 탈출층(Back-out-of-Tight Situation Layer)로 구성되어 있다. 3계층은 병렬적으로 수행된다. 가장 하위 계층인 장애물 회피층은 센서 입력을 감지하여 장애물이 있으면 멈추거나 앞으로 가는 역할을 수행하며, 탐색층은 되도록이면 넓은 곳을 다니도록 하는 기능을 최상위 계층인 탈출층은 하위의 층의 능력으로는 빠져 나올 수 없는 협소 공간에 빠졌을 경우에 뒤돌아 나오는 기능을 제공한다. 그림에서 표시된 사각형이 바로 AFSM 모듈들이며 AFSM들은 센서로부터 또는 다른 AFSM으로부터 입력을 받아서 모터를 구동하거나 다른 AFSM에게 자신을 출력을 전달한다. 이러한 경우에 상위층은 하위층을 억제하거나 차단하는 것이 가능하며, 자신이 생성한 출력으로 대체하는 것도 가능하다. 예를 들면, 떠돌기(Wander)는 도망(RunAway)를 가로채서 다른 방향으로 이동을 하도록 하여 다른 곳으로 돌아다니도록 할 수 있다.

이 시스템은 실시간에서 수행시에 매우 뛰어난 동작과 성능을 보여주었다. 그러나 만약 여러 다른 영역에 적응시키고자 할 때, 가장 큰 문제점으로는 바로 AFSM에 대해서 적응이나 학습 등이 자동으로 되는 것이 아니라는 점이다. Brooks는 행동 언어(Behavior Language)를 사용하여 사용자가 규칙들로 표현하면 AFSM 형태로 컴파일되는 구조를 제안하고 있다. 그러나, 이러한 학습 방법은 사람이 개입되어 규칙들을 표현하고 수정하는 작업이 필요하게 되어 스스로 자동 학습 할 수 있는 방법이 필요하다. 이 시스템의 장점으로는 각각의 계층이 독립적으로 병렬 수행될 수 있다는 점이며 단점으로는 상위 계층이 하위 계층의 행동에 간여하여 모듈성이 떨어진다는 점이다.

2.2 모터 스키마

1989년에 Arkin은 모터 스키마(Motor Schema)

행동 기반 로봇을 소개하였다(Arkin, 1989). 모터 스키마 방식은 모든 모듈들이 스키마로 구성되며, 스키마는 인지 및 행위에 대한 지식과 계산 과정으로 구성된다. 단위행동은 모터 스키마와 퍼셉션 스키마(perception schema)로 구성된다. 퍼셉션 스키마는 모터 스키마에 포함되어 로봇이 처한 상황에 대한 정보를 제공하는 역할을 한다. 모터 스키마는 이러한 퍼셉션 스키마를 이용하여 행동의 형식을 벡터로 표현한다. 로봇의 행동 제어기는 이와 같이 모터 스키마들이 다수가 모여서 구성된다. 여러 모터 스키마들이 각자의 벡터를 출력하면 이러한 벡터들이 취합되어 최종적으로 움직일 방향과 속도를 얻게 된다. [그림 2]를 통해서 이러한 과정들을 참조 바란다.



[그림 2] 모터 스키마 시스템의 구조도

이 방법의 단점은 국소해에 민감하여, 같은 행동을 반복하는 루프(loop)에 빠지는 경우가 있다는 점이다. 장점으로는 병렬성이 우수하고 수행시의 적응이 가능한 구조이며, 모듈성이 있다.

2.3 유전자 프로그래밍 기반 시스템

Brooks는 행동 학습 방법으로 유전자 프로그래밍에 의한 방법을 제안하였다(Brooks, 1992). Koza는 이러한 아이디어에 기반하여 유전자 프로그래밍 방법으로 로봇의 행동 학습법을 적용하였다(Koza, 1992). 적용 영역으로는 컴퓨터 속의 가상의 공간에서 개미로봇들이 먹이를 찾아서 모으는 과정을 시뮬레이션 하였다. 25 × 25 격자 공간에서 활동하는 20마리의 개미들이 먹이를 찾아서 한 곳으로 모으는 것으로, 동일한 형질의 개미들이 함께 활동하는 방식이다. 프로그램을 자동으로 생성하기 위한 함수 집합 F와 종단 집합 T는 다음과 같다.

F = {IF-FOOD-HERE, IF-CARRYING-FOOD, IF-FOOD-ADJECENT, PROGN2}
T = {(MOVE-RANDOM), (PICK-UP), (MOVE), (DROP-FOOD)}

유전자 프로그래밍 방법으로 위의 함수집합을 이용하여 맨 처음에는 무작위로 행동을 제어하기 위한 프로그램들의 집합체를 구성한다. 그리고 각각의 프로그램들에 대해서 환경에서 주어진 시간만큼 수행한 후 적합도를 구하게 된다. 이와 같은 방식으로 500개의 프로그램들에 대해서 30세대 만에 먹이를 한 곳으로 모으는 프로그램을 진화시켰다. Koza의 유전자 프로그래밍 방식은 단순한 행위들로부터 복잡한 행동이 창발될 수 있음을 보여주었다.

이 후의 유전자 프로그래밍 관련 연구들을 살펴보면, 1996년 Bennet는 독립적으로 행동하는 로봇들의 집단에 대해서 특정 행동들을 진화하도록 하였으며, (Zhao, 2000)에서는 실시간 계획자를 생성하기 위해서 유전자 프로그래밍 방법을 채용하여 상호간 통신 기능이 없는 로봇들을 대상으로 협력

을 위한 기본 지식을 보유하지 않고 상황에 적응하도록 하였다. (Kubca, 2002)에서는 분산적으로 제어하는 모듈화된 로봇들의 프리미티브들을 생성하기 위한 분야에 적용하고 있다. 즉, 수작업에 의한 코드 작성과 유전자 프로그래밍을 결합하여 프리미티브들을 정제하는 기법을 제안하고 있다.

이와 같이 많은 연구들에서 행동 지식을 표현하기 위해서 유전자 프로그래밍 방법을 사용하고 있다. 그러나 근본적으로 유전자 프로그래밍은 인간이 개입하여 함수 집합과 종단 집합을 정의해야 한다는 것이 단점이다. 또한 순서적으로 수행하는 프로그램의 형태이기 때문에 병렬적인 수행이 어렵고 프로그램 크기가 커지게 되면 점차 응답성이 저하된다.

3. 혼합형 행동 진화 모델

이번 장에서는 본 논문에서 제안한 혼합형 행동 진화 모델을 기술한다. 먼저, 본 모델의 유전자 연산의 대상이 되는 염색체에 대해서 기술한다. 또한, 행동 지식의 핵심적인 역할을 하는 분류 규칙과 진화 신경망에 대한 표현, 이와 관련된 알고리즘에 대해서 설명한다. 그리고, 가상 로봇의 행동 수행과 모델을 학습시키는 방법에 대해서 기술한다.

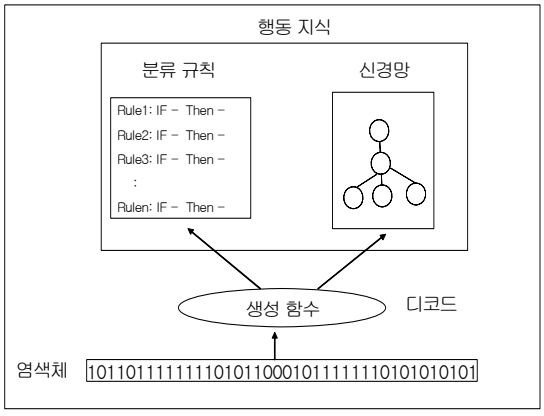
3.1 염색체 표현

행동 진화 모델의 표현하기 위해 가장 기본 요소인 염색체를 {0, 1}로 이루어진 고정 길이의 비트 스트링 형태로 표현하며, 염색체에는 행동 지식을 포함시킨다. 행동 지식을 표현하기 위해서 염색체는 다음과 같이 세 부분으로 구성된다.

- **메타 데이터**
  분류 규칙과 신경망을 구성하기 위해 염색체 자신이 포함하고 있는 정보들에 대해 표현한다.
- **규칙 구성자**
  규칙을 표현하기 위해 연속적으로 규칙들을 나열한다.
- **연결 구성자**
  신경망의 구성을 위해 유닛간의 연결에 관한 정보를 표현한다.

이들 염색체들에 대해서 유전자 연산인 선택, 교배, 돌연변이를 적용한 후 적합도 함수를 사용하여 가장 적합도가 높은 염색체들이 계속해서 생성되어 가상 로봇의 행동이 진화될 수 있도록 행동 지식으로 결정한다.

염색체에 기반한 가상 로봇의 행동을 수행하기 위해서는 수행 가능한 형태의 행동 지식으로 전환되어야 하는데 이러한 과정은 생성 함수에 의해서 수행된다. 생성 함수는 염색체의 비트 스트링 정보로부터 분류 규칙과 진화 신경망을 생성하기 위해서 메타데이터, 규칙 구성자, 연결 구성자로 나누기 위한 파싱, 각각의 부분을 해석하기 위한 해석, 해석된 결과로부터 규칙과 진화 신경망을 구성하는 구성 알고리즘으로 구성된다.

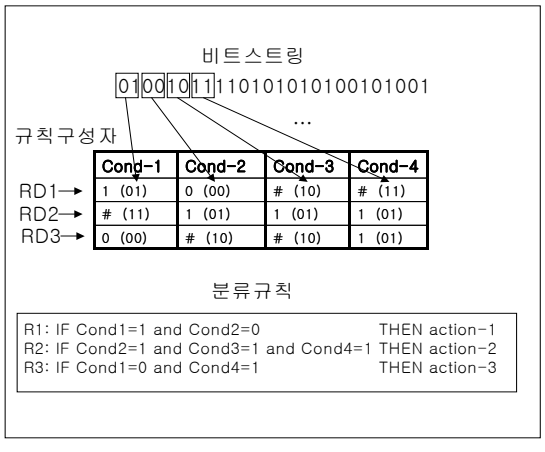


[그림 3] 염색체로부터 행동 지식을 생성하는 과정

3.2 규칙 구성자로부터 분류 규칙 생성

분류 규칙은 조건부와 결론부로 구성되는 형태의 규칙이다. 조건부는 {0, 1, #}로 이루어진 스트링으로 표현되며 여기에서 #은 'don't care'이다. 결론부는 {0, 1}로 표현된 스트링 형태로 표현된다. 입력값은 현재의 상태 s(t)⊆이고, 시간 t에서 매칭되는 규칙의 결론부 행위 a(t)∈가 수행된다. 입력값의 가능한 모든 상태는 $S\in, 1, \#\}^{L}$이며, L은 고정 길이의 입력 스트링을 나타낸다. 분류 규칙 "IF C THEN a"의 조건부 $C\in, 1, \#\}^{L}$은 입력값과 매칭이 가능하다. 결론부의 a∈는 매칭이 성공하면 수행되는 행위를 지시한다.

[그림 4]는 입력된 규칙 구성자의 비트 스트링으로 부터 분류 규칙를 형성하는 과정을 보여주고 있다. 이 예에서는 3개가 형성되는 과정이다.



[그림 4] 입력된 규칙 구성자의 비트 스트링으로 부터 분류 규칙 생성

염색체의 규칙 구성자 비트 스트링에서 분류 규칙의 수만큼 해당하는 비트 스트링을 나누어 배열에 분류 규칙들이 생성되어 저장된다. 본 논문에서 제시한 분류 규칙 생성 알고리즘은 [그림 5]와 같다.

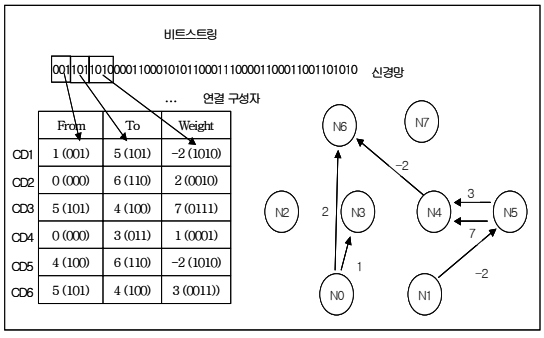
```
RuleCreate(RDbitString, R, Rbase[])
      RDbitString : 규칙 구성자 비트스트링
      R : RDbitString에 포함된 규칙의 총 수
      Rbase[] : 분리된 규칙을 저장할 배열

- Rbase 배열을 초기화한다.
- 규칙의 수만큼 다음을 반복한다.
  1. RDbitString에서 규칙의 서브스트링를 추출하고 추출
     된 서브스트링에서 조건 부분을 추출한다.
  2. 정해진 생성 규칙에 따라 규칙을 정의한다.
- 규칙의 총 수만큼 규칙을 반환한다.
```

[그림 5] 규칙 구성자로부터 분류 규칙을 생성하는 알고리즘

3.3 연결 구성자에 의한 진화 신경망 구성

염색체 상에 표현된 연결 구성자의 비트 스트링으로부터 신경망을 구성하는 방법이다. 연결 구성자는 From, To, Weight 세 개의 부분으로 구성되어 있으며, 신경망의 구성과 유닛과 유닛간의 연결강도를 표현한다. 연결이 시작될 유닛을 나타내기 위해 From을 사용하고, 연결의 대상이 되는 유닛을 나타내는데 To를 사용한다. 그리고 이 연결의 강도를 표현하기 위해 Weight를 사용한다. [그림 6]은 From과 To의 비트 길이를 3으로 설정하고, Weight 비트 길이는 4(부호 비트 포함)로 설정하여 비트 스트링으로부터 신경망 구조를 생성한 과정을 나타낸 것이다.



[그림 6] 연결 구성자로부터 신경망 생성

연결 구성자는 두 유닛간에 순환적 또는 중복 연결되더라도 유효하다. 또한, 출력 유닛에서 입력 유닛으로 연결된다 하더라도 유효하지만, 동작에 있어서는 아무런 영향을 미치지 못한다. 이러한 연결 구성자를 이용하여 신경망을 염색체에 표현하게 되면 두 가지 효과를 얻게 된다. 돌연변이 또는 교배를 통해서 특정 연결의 가중치와 연결 위치를 모두 변경할 수 있다. 연결 구성자로 표현한 신경망 구성의 커다란 장점은, 제약이 없고 상속이 가능한 신경망의 패턴을 제공한다는 점이다. 또한 연결 구성자는 유전 정보 상의 위치와는 상관없이 같은 효과를 나타낸다.

[그림 7]는 염색체의 연결 구성자 비트 스트링으로부터 신경망을 생성하는 알고리즘이다. C는 연결 구성자의 개수이고, 이 개수만큼 CDbitString을 나누어 디코딩하여 신경망의 유닛간 연결과 강도를 표현하는 2차원 배열을 생성한다. 2차원 배열에는 유닛간 연결의 가중치 값이 저장되며, 0인 경우는 유닛간 연결이 없는 것으로 해석한다.

NeuralNetworkCreate(CDbitString, C, NN[][])
  CDbitString : 연결 구성자의 비트스트링
  C : CDbitString에 포함된 연결구성자의 총 수
NN[][] : 신경망을 위한 배열

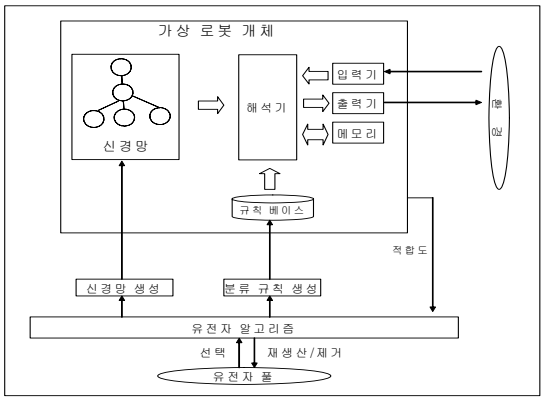
  - NN 2차원 배열을 초기화한다.
  - 연결 구성자의 수만큼 다음을 반복한다.
1. CDbitString에서 n번째 연결구성자의 서브스트링을 추출한다.
2. 추출된 n번째 서브스트링에서 from 값을 추출한다.
3. 추출된 n번째 서브스트링에서 to 값을 추출한다.
4. 추출된 n번째 서브스트링에서 weight 값을 추출한다.
5. from, to, weight를 이용하여 NN[][] 2차원 배열을 구성한다.
  - 2차원 배열 NN을 반환한다.

[그림 7] 연결 구성자로부터 신경망을 생성하는 알고리즘

3.4 혼합형 행동 진화 모델의 구조

혼합형 행동 진화 모델은 가상 로봇의 행동 지식을 결정하기 위해 분류 규칙을 표현하기 위한 규칙 구성자와 진화 신경망을 표현하기 위한 연결 구성자를 이용하여 행동 지식들을 표현한다. 그리고 이들 염색체 정보를 이용하여 행동 학습 수행을 걸쳐 최적의 유전자를 구할 수 있도록 환경에 반응하는 행동을 수행하여 최적의 유전자를 찾아서 가상 로봇을 구성한다. [그림 8]은 본 논문에서 제시하는 혼합형 행동 진화 모델의 전체 구조이다.

유전자 풀에는 규칙 구성자와 연결 구성자를 이용하여 행동 지식을 표현하여 다량의 염색체를 저장하고 염색체에 대해서 유전자 알고리즘을 적용하여 최적의 유전자를 추출한다. 최적의 유전자를 추출하기 위해서 염색체의 규칙 구성자를 해석하여 분류 규칙을 생성하고 연결 구성자를 이용하여 신경망을 생성한다. 생성된 분류 규칙과 신경망 구조를 가상 로봇 개체에 적용하여 환경에 대해서 학습한다. 적절한 적합도 함수를 이용하여 행동 학습을 수행하여 최적의 유전자가 결정될 때까지 유전자 알고리즘을 반복, 수행하여 최적으로 여겨지는 유전자를 추출하여 가상 로봇 개체에 적용한다.



[그림 8] 혼합형 행동 진화 모델의 전체 구조

3.5 행동 학습 절차

본 모델에서 분류 규칙과 신경망을 이용하여 행동 학습을 수행하기 위한 절차는 다음과 같다.

1단계 : 행동 지식을 표현하는 염색체를 비트스트링으로 구성한다.

염색체는 분류 규칙을 표현하기 위한 규칙 구성자, 진화 신경망을 표현하기 위한 연결 구성자, 몇 개의 분류 규칙과 진화 신경망을 구성하는 노드와 연결의 수와 같은 메타 정보를 표현하는 메타 데이터 부분으로 구성된다.

2단계 : 임의의 염색체들을 {0,1}로 랜덤하게 생성하여 개체군(population)을 구성하고 유전자 풀에 저장하여 관리한다.

3단계 : 유전자 풀에 저장된 개체군을 대상으로 하여 유전자 알고리즘을 적용하여 학습을 수행한다.

개체군에서 어떤 염색체들을 선택하여, 이들로부터 분류 규칙과 신경망을 생성하여 테스트 환경에서 행동을 수행한다. 테스트 환경에서 이미 정해진 적합도 함수에 따라서 적합도를 구한다. 적합도에 따라서 유전자 연산을수행하여 우수한 것은 재생산되도록 하며 그렇지 않은 것은 도태되도록 한다. 이러한 학습의 과정을 계속적으로 반복, 수행하면 유전자 풀에는 점차적으로 우수한 형질의 염색체들이 늘어나게 된다.

4단계 : 3단계에서 학습된 염색체 중 가장 우수한 행동 전략을 가진 염색체를 추출하여 문제 영역에 적용하여 행동하도록 수행한다.

3.6 행동 수행 절차

가상 로봇들의 집단이 주어진 환경에서 지엽적인 감지 정보를 입력받아서 행동을 수행한다. 집단에 소속된 가상 로봇 개체 r∈R는 환경으로부터 지엽적인 환경 정보를 감지기를 통해 받아드린다. 그리고 감지된 정보와 규칙베이스에 있는 모든 규칙들을 매칭 검사하여 매칭된 해당 규칙들을 활성화한다. 그런 후에, 감지기의 정보와 분류 규칙의 수행 결과를 신경망 입력 유닛들로 전달하고, 신경망의 처리된 결과는 출력 유닛을 통하여 출력기로 전달되어 적절한 행동이 수행된다.

3.7 학습

가상 로봇의 행동 학습의 경우에 있어서는 전체 개체에 대해서 의미 있는 적합도를 구하는 것은 어려운 문제이다. 본 논문에서는 [그림 9]에서 제시하는 행동 지식에 대한 학습 알고리즘을 이용한다. 먼저 개체군(population)을 생성하기 위해, 모든 염색체들에 대해서 {0,1}로 랜덤하게 초기값을 할당한다. 그러므로, 초기에는 랜덤한 방법으로 분류 규칙들과 신경망에 대한 정보가 발생된다. 이렇게 형성된 염색체로부터 만들어진 표현형인 개체들은 환경에서 운이 좋은 경우를 제외하고는 원시적인 수준의 행동밖에는 하지 않는다. 다음 단계로는 승자 두 개와 패자 두 개를 선정하기 위한 토너먼트를 수행한다. 개별적인 염색체 i의 적합도 F_i는 주어진 환경에서 동일 염색체를 갖는 로봇 개체들이 시간 스텝별로 얻은 보상 점수들의 합이 된다.

$$F_i = \sum_{j=1}^{n} \sum_{t=1}^{m} r_{ijt}$$

여기에서 n은 집단을 구성하는 로봇의 수이고, m은 수행한 시간 스텝이고, r_{ijt}는 i번 염색체로부터 생성된 j번 로봇의 시간 t에 수행한 보상 값이다. 이러한 식에 의하여 적합도 계산 결과를 이용하여 승자와 패자를 결정한다. 그리고 승자의 두 염색체로 교배 연산을 수행하여 자식 염색체를 만

든다. 자식 염색체는 다시 돌연변이 과정을 거친 후 개체군에서 패자 염색체를 대체한다.

```
BehaviorLearning(P)
P : 개체군
- 랜덤하게 개체군을 생성한다.
- 정해진 세대만큼 다음을 반복한다.
  1. 2D 선택방법에 의해 부개체군을 생성한다.
  2. 선택한 부개체군에 대해 적합도를 구한다.
  3. 적합도에 따라서 승자와 패자를 결정한다.
  4. 승자의 두 염색체에 대해 교배 연산을 수행하여
     자식 염색체를 만든다.
  5. 자식 염색체에 대해 돌연변이 연산을 적용하여
     패자 염색체를 대체한다.
```

[그림 9] 행동 지식 학습 알고리즘

이러한 과정이 계속적으로 반복되면서 진화 학습이 진행되며, 결국 유전자 풀에는 우수한 적합도를 보유한 염색체만이 살아남게 된다. 학습 과정이 끝난 후에는 우수 염색체를 채취하여 해당하는 환경 속에서 목표한 작업을 시킬 수 있다.

4. 구현 및 실험

3장에서 행동 진화 모델을 제안하였다. 이를 가상 로봇의 먹이 수집 문제에 적용하기 위해서 HBES (Hybrid Behavior Evolution System)를 설계하고 구현하였다. 그리고, 시뮬레이션을 통해서 학습 능력과 특성을 분석 및 비교 평가한다.

4.1 HBES의 설계 및 구현

4.1.1 가상 로봇과 환경

가상 로봇은 몸체와 매우 단순한 센서들 그리고 모터들로 구성된다. 센서들은 거리 1만큼에 대해서만 인접한 전방향, 좌측방향, 우측방향을 인식할 수 있다. 또한 앞으로 움직이거나 방향을 전환하기 위해서 몸체 좌측과 우측에 모터를 갖고 있다. 그리고 몸체의 전방향에는 암(arm)이 있어서 물체를 집어 올리거나 내려놓을 수 있다.

환경은 격자들로 이루어져 있다. 환경의 중앙에 Nest로 표시된 둥지가 있고, 가상 로봇들이 둥지에 먹이를 모은다. 격자들에는 'Food'로 표시된 먹이와 'Block'으로 표시된 장애물과 로봇들이 위치할 수 있다. 제약 사항으로 같은 격자에 동시에 두 가지가 존재할 수 없다. 예를 들면, 어떤 로봇의 앞에 먹이가 있다면 앞으로 이동하는 것은 불가능하다. 또한 환경 속에 있는 장애물은 옮겨 질 수 없다. 그러나 먹이는 로봇에 의해서 옮겨질 수 있다. 만약 어떤 로봇이 먹이를 집어올리면 환경에는 자동으로 새로운 먹이가 랜덤한 위치에 자동으로 생기도록 하였다. 그리고 로봇이 먹이를 둥지에 놓는 경우는 해당 위치에 먹이를 표시하지는 않고 다만 보상 점수를 준다. 로봇은 둥지가 아닌 곳에서 내려놓는 것은 기본적으로 불가능하도록 하였다. 내려놓는 동작 속에는 현재 위치가 둥지가 아니라면 무효화 된다. <표 1>은 로봇들이 취할 수 있는 모든 행동 프리미티브들을 나타낸다.

<표 1> 로봇의 행동 프리미티브

행동 프리미티브	의 미
Go Forward	현재의 방향으로 한 칸 이동
Turn Left	현 위치에서 왼쪽으로 90도 회전
Turn Right	현 위치에서 오른쪽으로 90도 회전
Lift Up	전방의 먹이를 집어 올리기
Drop Down	갖고 있는 먹이를 내려놓기
Pheromone 1	Pheromone 1을 현 위치에 뿌리기
Pheromone 2	Pheromone 2을 현 위치에 뿌리기

4.1.2 규칙 베이스

본 시스템에서는 총 10개로 구축된 규칙베이스를 사용한다. <표 2>에서는 사용된 규칙베이스를 나타내고 있다. 이 규칙베이스에는 총 10개의 규칙들이 존재한다. 각각의 규칙들은 12개의 문자로 구성되며, 하나의 문자는 조건을 나타낸다. 12개의 이진 상태로 입력되는 데이터는 어떤 조건을 기술하는 정보이다.

<표 2> 구축된 규칙베이스의 예

#	규칙	심볼화된 규칙 표현
1	#0#0##0##110	IF Front != Food & Left !=Robot & Left !=Block & Carrying = True & Dir=True & Nest != True
2	10#1#10##0#1	IF Left==Food & Front != Food & Left = Robot & Right = Robot & Left !=Block & Carrying = False & Nest=True
3	######0##1#0	IF Left !=Block & Carrying=True &
4	#1#0#101##0	IF Front=Food &Left != Robot & Right= Robot && Right = Block & Nest!=True
5	##00##0##0##	IF Right!=Food & Left != Robot & Left !=Block & Carrying = False

예를 들어 규칙 1번을 보면, 로봇 개체가 먹이를 나르는 상태에서 둥지로의 방향과 현재의 위치가 둥지가 아닐 때의 상태를 나타내고 있다. 이러한 조건은 먹이를 보유한 상태이면서 현재의 방향이 둥지로 향하는 방향이면 '앞으로 이동'의 결과를 초래한다. 따라서 매우 중요한 지식을 보유하고 있다고 볼 수 있다. 신경망에서 규칙에 대한 정보가 행동과 연관되어 적합도를 향상시키며 결과에 영향을 주고 있다.

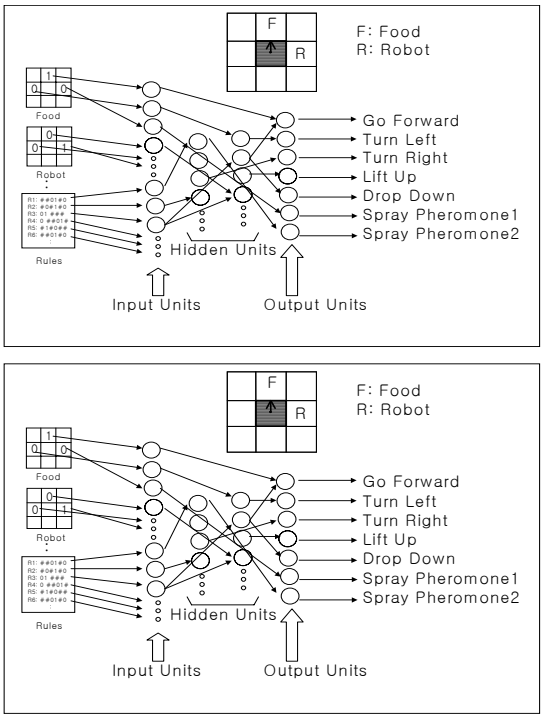
4.1.3 신경망의 구성

다음의 <표 3>에는 신경망의 입력과 출력 노드를 정의한다. 입력 유닛으로는 3방향에 있는 음식, 로봇, 블록, 둥지에 대한 존재 여부와 페르몬과 규칙 수행 결과로 이루어진다. 출력 유닛은 <표 1>에서 정의한 로봇의 행동 프리미티브를 이용하여 정의한다.

<표 3> 신경망의 구성

입력유닛	<총 44개> 3방향의 Food 존재 3개 3방향의 Robot 존재 3개 3방향의 Block 존재 3개 Nest 여부 1개 Direction 상태 1개 3방향+현 위치의 페르몬-1 존재 4개 3방향+현 위치의 페르몬-2 존재 4개 규칙 수행 결과 10개
출력유닛	<총 7개> 앞으로 가기 1개 방향 전환 2개(Left Turn, Right Turn) Food 집어 올리기 또는 내리기 2개 페르몬 뿌리기 2개

[그림 10]에서는 <표 3>에서 정의한 신경망의 입력과 출력에 대해서 연결 구성자를 이용하여 형성된 가상 로봇의 신경망 구조를 나타내고 있다. 형성된 신경망 구조에 의해서 들어온 입력 정보를 바탕으로 7가지 출력 중 하나를 선택하는 행동 결정 과정도 나타내고 있다. [그림 10]의 상단의 격자 모양은 로봇의 위치와 인접 격자의 상태를 나타낸다. 진한 음영은 가상 로봇 자신의 위치이고 앞쪽 방향에는 F로 표시된 먹이가 있고, 우측 방향에는 다른 로봇이 있는 것을 나타낸다. 이러한 주변 상황을 센서를 통해서 신경망의 입력 유닛으로 입력되고, 규칙에도 매칭되어 매칭된 결과값이 신경망으로 함께 유입된다. 그리고 신경망은 입력된 신호들을 처리하여 하나의 출력 유닛을 선택하여 적절한 행동을 활성화시킨다. 이와 같이 가상 로봇은 주변 상황을 인식하고 적절한 행동을 결정하여 수행한다.



[그림 10] 가상 로봇의 신경망 구성과 상황 인식 및 행동 결정 과정

4.1.4 적합도

본 시스템에서는 적합도를 단순하고 정적인 방식을 채용한다. 특정 염색체에 대한 적합도는 그것으로부터 발생된 로봇들이 주어진 시간 동안 활동하면서 얻은 보상 값의 합으로 한다.

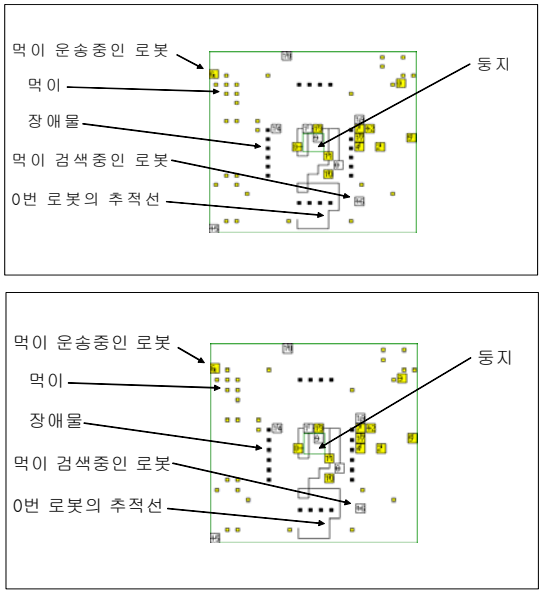
로봇 개체가 보상 점수를 받는 방식을 살펴보면, 움직인 경우에 전면에 먹이가 존재 한다면 1점을 받는다. 그리고 먹이를 들어올린 경우에도 1점을 받는다. 만약 먹이를 성공적으로 둥지에 놓은 경우에는 1000점을 받는다. 최종적인 목표를 달성할 때에만 점수를 주는 방법도 있으나 초기 단계에는 거의 먹이를 모으지 못한다. 이러한 경우에는 작은 수준의 행동이라도 유리한 결과를 초래할 수 있는 경우는 작은 점수를 주는 전략을 채용하였다. 염색체 P의 적합도 수식은 다음과 같다.

$$\text{Fitness(P)} = \sum_{i=1}^{n} Ri$$

여기에서 n은 로봇의 수, Ri는 P 염색체로부터 발생된 로봇 개체 i가 생명시간 동안 받은 보상 점수를 지시한다.

4.1.5 구현

본 시스템은 윈도우즈 운영체제(Windows XP) 상에서 Borland C++ Builder를 이용하여 구현하였다. [그림 11]은 한 로봇들이 노란색의 먹이를 나르는 과정을 쉽게 관찰할 수 있도록 시각화한 시뮬레이션의 한 예를 보여주고 있다. 가상 로봇에 대해서 추적선을 그려 어떤 형태의 이동 경로로 움직이는지 쉽게 관찰 할 수 있다.



[그림 11] 시뮬레이션의 한 예

<표 4>는 주요 파라메터이다. 교차율은 수행을 통해서 선정된 두 개의 부모 염색체들을 교차하여 자식들을 만들 때 교차의 확률을 나타낸다. 돌연변이율은 교차에 의해 만들어진 자식들에 대해서 수행되며, 특정 비트가 돌연변이를 일으켜서 다른 값으로 바뀔 확률이다. 개체군은 유전자 풀에 들어있는 염색체들의 수를 나타낸다. 그리고 로봇 집단의 크기는 특정 염색체의 적합도를 평가하기 위해서 환경에서 함께 활동하게 될 로봇들의 수이다. 환경의 크기는 격자들로 이루어진 칸과 열의 수이다. 환경의 먹이 유지량은 로봇들에 의해서 운송 중인 것과 수집된 것을 제외하고 환경 속에서 항상 유지되는 먹이의 수량을 나타낸다. 둥지의 크기는 중앙에 위치하고 있으며, 로봇들이 먹이를 내려놓아서 수집을 완료하게 되는 좌표 영역을 지시한다. 규칙베이스의 크기는 특정 염색체로부터 생성되는 분류 규칙의 수를 표현한다. 센서 입력 값의 길이는 규칙과 매칭의 대상이 되는 부분들로 구성된 값의 비트 스트링 길이를 표현한다. 활동 시간은 환경 속에서 활동하게 되는 개별 로봇들이 입력과 행동 싸이클의 총 회수를 지시한다.

<표 4> 시뮬레이션에 사용된 주요 파라미터들

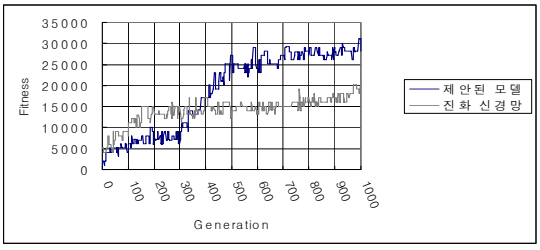
파라미터	값
교차율	90%
돌연변이율	0.05%
개체군 크기	100
로봇 집단의 크기	20
환경의 크기	20 X 20
환경의 먹이 유지량	30
둥지의 크기	2 X 2
규칙 베이스의 크기	10
센서 입력 값 길이	12
활동 시간	100

4.2. 실험 결과 분석

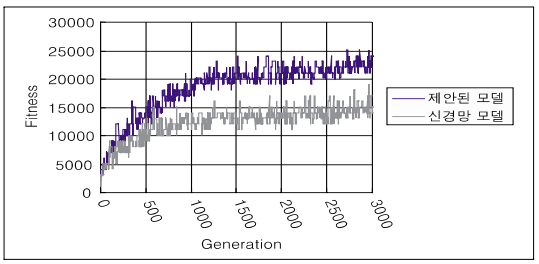
4.2.1 장애물 실험

이 실험은 장애물 환경의 복잡도를 늘려가면서 학습 능력을 평가하기 위한 것이다. 이 실험에서는 본 논문에서 제안된 모델과 동일한 조건하에서의 순수한 진화 신경망 방식을 비교 평가하였다. 장애물은 각각 벽이 없는 것과 2개 있는 것 그리고 4개 있는 것으로 환경의 복잡도를 다르게 하였다.

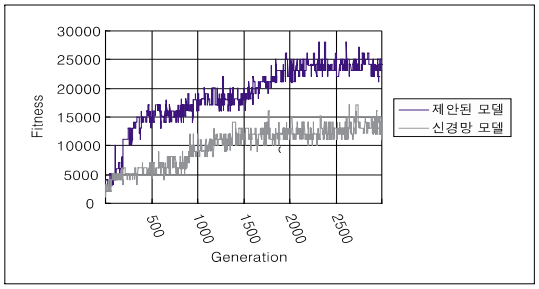
이러한 환경들에서 수행한 세대별(generation)와 최대 적합도(Fitness)의 결과는 [그림 12]~[그림 14]에 나타내었다. [그림 12]을 보면, 진화 신경망 방식 250세대 정도의 초기 단계에서 수렴하는 것을 보였고, 제안된 모델은 30 세대 이후부터 급격히 적합도가 높아져서 600세대 정도에서 수렴하기 시작했다. [그림 13], [그림 14]에서도 마찬가지로 초기 학습 단계에서 만들어진 격차는 좁혀지지 않았으며 제안된 모델이 보다 높은 적합도의 수준에서 수렴하였다.



[그림 12] 장애물이 없는 환경에서의 수행결과

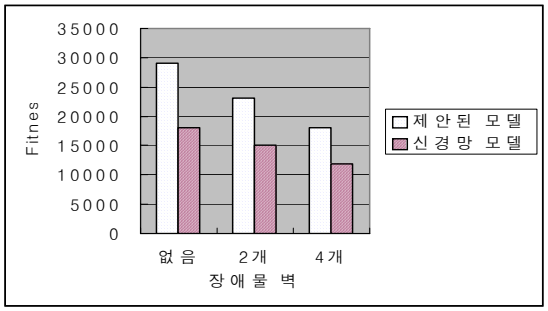


[그림 13] 장애물 벽이 2개 있는 환경에서의 수행결과



[그림 14] 장애물 벽이 4개 있는 환경에서의 수행결과

이러한 결과들을 하나의 차트에서 비교할 수 있도록 [그림 15]에서 1000세대에서의 적합도를 함께 보였다. 이들 실험을 통해서 제안된 모델은 학습의 속도와 학습의 질에 있어서 환경의 복잡도가 높아지는 경우에도 우수하다는 것을 실험적으로 입증하였다.

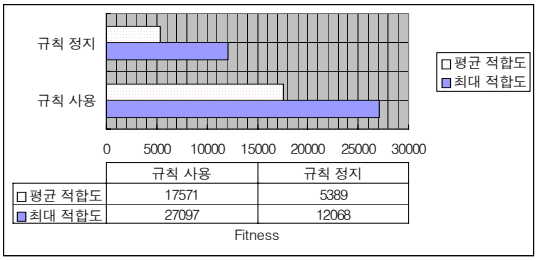


[그림 15] 환경 변화시의 적합도 변화비교

4.2.2 규칙과 적합도의 연관성 분석

본 모델에서 규칙이 적합도에 얼마나 기여하는가를 분석하였다. 먼저 10000세대 학습을 수행시킨 후에 전체 염색체들을 대상으로 하여 100 시간 스텝 만큼 활동시켰다. 그리고 각 염색체의 적합도에서 최대값과 평균값을 각각 구하였다. 같은 방법으로 이번에는 규칙 부분을 사용하지 않고 실험값을 추출하였다. [그림 16]에서는 이들 값들을 나타내고 있다. 비교 결과를 보면 규칙을 사용하지 않는 경우와 비교해서 최대 적합도는 2.2배, 평균 적합도에 있어서는 3.2배 높은 결과가 도출되었다.

이는 규칙이 적합도 형성에 기여함을 알 수 있다. 규칙 구성자에 의해 형성된 규칙이 계속적으로 동일한 결과를 지속적으로 유도해주며 유도된 결과가 적합도를 향상시킬 수 있는 적절한 결과임을 알 수 있다.



	규칙 사용	규칙 정지
평균 적합도	17571	5389
최대 적합도	27097	12068

[그림 16] 규칙 사용이 적합도에 미치는 영향비교

5. 결론

본 논문에서는 가상 로봇들을 위한 행동 진화 시스템을 위해서 규칙 구성자를 이용하여 분류규칙을 형성하고 연결 구성자를 이용하여 진화 신경망을 형성하는 혼합형 행동 진화 모델을 제안하였다. 제시된 모델에서는 행동 지식을 두 수준에서 표현한다. 먼저 상위수준에서는 분류규칙과 신경망을 함께 적용하여 표현력을 향상하였다. 가상 로봇이 주변 상황을 인식하면 규칙들을 통해서 패턴을 분류하고 그 결과를 신경망의 입력으로 적용하여 적절히 행동하게 하였다. 그리고 하위 수준에서는 행동 지식이 비트 스트링 형태로 염색체에 표현되도록 하였으며, 이들 염색체를 대상으로 유전자 연산을 적용하여 학습이 수행된다. 또한 특정 염색체의 적합도를

측정하기 위해서는 해당 염색체로부터 상위 수준의 행동 지식을 생성하고 수행해야 한다.

제안된 모델을 평가하기 위해서 가상 로봇들의 먹이 수집 문제에 적용하였다. 이를 위해서 HBES를 개발하여 시뮬레이션 하였다. HBES는 20 × 20 격자들로 이루어진 공간에서 20대의 가상 로봇들이 함께 활동하면서 먹이를 수집하는 환경을 지원하는 시스템이다. 첫 번째 실험으로는, 제안된 모델의 학습 능력을 확인하기 위해서 장애물이 없는 환경, 장애물 벽이 2개 있는 환경, 장애물 벽이 4개 있는 환경에서 각각 진화 학습을 수행하였다. 그 결과, 제안된 모델은 동일한 조건의 진화 신경망에 비하여 빠르게 학습하고 높은 적합도에서 수렴하였다. 다음으로는 규칙이 적합도 향상에 주는 영향을 평가하기 위해서, 학습이 완료된 염색체들에 대해서 규칙을 적용한 것과 그렇지 않은 것을 각각 수행하여 적합도를 측정하였다. 그 결과, 규칙을 적용하지 않게 되면 적합도가 급격히 저하되는 것을 확인하였다. 이러한 실험분석들을 통해서 제안된 모델은 행동 학습에 있어서 기존의 진화 신경망 방식에 비해서 우수하다는 것을 확인하였다

본 논문에서 제안된 행동 진화 모델은 비정형적이며 동적인 환경의 여러 응용 분야들에서 효과적으로 이용될 수 있을 것이다. 또한 시뮬레이션에서 사용된 가상 로봇은 구현하기 쉬운 구조로 설계되어 있기 때문에 쉽게 제작도 가능하다. 따라서 본 논문에서 제안된 행동 진화 모델은 실세계 로봇으로의 적용도 가능하다.

추후의 연구 과제로써는 본 모델에서는 동일 형질의 로봇들에 대해서 진화적으로 학습시켰다. 그러나 서로 구조와 역할이 다른 경우의 학습법에 대해서도 연구가 필요하다. 또한, 실세계에서 로봇들을 진화 학습시키기 위해서는 실제 단위 동작들을 수행하는데 많은 시간이 소요되어 매우 비효율적이다. 따라서 시뮬레이터에서 실제와 동일한 환경을 재현하여 빠르게 학습할 수 있는 방법에 대한 연구가 필요하다.

참고문헌

[1] Arkin, R. C., "Motor Schema-Based Mobile Robot Navigation", *The International Journal of Robotics Research*, Vol.8, No.4(1989).

[2] Brooks, R. A., "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, Vol.2, No.1 (1986), 14-23.

[3] Brooks, R. A., "A Hardware Retargetable Distributed Layered Architecture for Mobile Robot Contorl", *Proceedings of the IEEE International Conference on Robotics and Automation,* Raleigh, NC(1987), 106-110.

[4] Brooks, R. A., "Intelligence without Reason", Proceedings of 12th Int. Joint Conf. on Artificial Intelligence(1991), 569-595.

[5] Brooks, R. A., "Artificial Life and Real Robots", Proc. of the First Europian Conference on Artificial Life : Towards a Practice of Autonomous Systems, Morgan Kaufmann Publishers(1992), 3-10.

[6] Collins, R. J. and D. R. Jefferson, *AntFarm : Towards Simulated Evolution Artificial Life II*, Addison-Wesley, 1991.

[7] Collins, R. J., *Studies in Artificial Evolution, PhD Thesis, Philosophy in Computer Science*, University of California, Los Angeles, 1992.

[8] Koza, J. R., *Genetic Programming : On the Programming of Computers by Means of Natural Selection*, MIT Press. 1992.

[9] Kubica, J. and E. Rieffel, "Collaborating with

a Genetic Programming System to Generate Modular Robotic Code", A joint meeting of the eleventh International Conference on Genetic Algorithms (ICGA- 2002) and the seventh Annual Genetic Programming Conference (GP-2002)(2002), 804-811.

[10] Lee, M., "Evolution of Behaviors in Autonomous Robot Using Artificial Neural Network and Genetic Algorithm", *Journal of Information Science*, Vol.155, No.1 (2003), 43-60.

[11] Mataric, M. J., "Behavior-Based Control : Examples from Navigation, Learning, and Group Behavior", *Journal of Experimental and Theoretical Artificial Intelligence*, Special Issue on Software Architectures for Physical Agents, Vol.9, No.2(1997), 323-336.

[12] Pipe, A. G. and B. Carse, "Fuzzy Logic and the Pittsburgh Classifier System for Mobile Robot Control", In Proceedings of the 3rd Conference of the European Society for Fuzzy Logic & Technology (EUSFLAT)(2003), 120-125.

[13] Zhao, K. and J. Wang, "Multi-robot Cooperation and Competition with Genetic Programming", Proc. of Genetic Programming(EuroGP'2000)(2000), 804-811.

Abstract

Hybrid Behavior Evolution Model Using Rule and Link Descriptors

Sa Joon Park*

We propose the HBEM(Hybrid Behavior Evolution Model) composed of rule classification and evolutionary neural network using rule descriptor and link descriptor for evolutionary behavior of virtual robots. In our model, two levels of the knowledge of behaviors were represented. In the upper level, the representation was improved using rule and link descriptors together. And then in the lower level, behavior knowledge was represented in form of bit string and learned adapting their chromosomes by the genetic operators. A virtual robot was composed by the learned chromosome which had the best fitness. The composed virtual robot perceives the surrounding situations and they were classifying the pattern through rules and processing the result in neural network and behaving. To evaluate our proposed model, we developed HBES(Hybrid Behavior Evolution System) and adapted the problem of gathering food of the virtual robots. In the results of testing our system, the learning time was fewer than the evolution neural network of the condition which was same. And then, to evaluate the effect improving the fitness by the rules we respectively measured the fitness adapted or not about the chromosomes where the learning was completed. In the results of evaluating, if the rules were not adapted the fitness was lowered. It showed that our proposed model was better in the learning performance and more regular than the evolutionary neural network in the behavior evolution of the virtual robots.

Key words : Behavior Evolution Model, Rule Descriptor, Link Descriptor, Rule Representation

* School of Mobile Contents, Daegu Haany University