

논문 2005-42C1-3-3

신경망 지능 캐릭터의 게임 환경 변화에 대한 적응 방법

(Adaptation of Neural Network based Intelligent Characters to Change of Game Environments)

조 병 현*, 정 성 훈**, 성 영 락***, 오 하 령***

(Byeong-heon Cho, Sung-hoon Jung, Yeong-rak Sung, and Ha-ryoung Oh)

요 약

최근 컴퓨터 게임에서 지능 캐릭터는 게이머들의 흥미를 계속 유발시킬 수 있기 때문에 더욱 더 중요한 요소로 부각되고 있다. 지능 캐릭터를 구현하는 대표적인 방법으로 신경망을 사용하여 상대 캐릭터의 행동패턴과 게임 규칙을 학습하는 방법이 연구되었다. 그러나 게임의 규칙은 갑자기 변경될 수 있으며 온라인 게임과 같은 상황에서는 게이머에 따라서 행동 특성이 크게 다를 수 있다. 본 논문에서는 지능 캐릭터가 이러한 환경의 변화에 적응하는 방법으로서 개체 수준 적응 알고리즘과 개체군 수준 적응 알고리즘을 제안한다. 개체 수준 적응 알고리즘에서 각 지능 캐릭터는 자신의 게임 점수의 변화를 계속해서 관찰하면서, 최종적으로 획득한 점수들을 고려하여 환경의 변화를 판단하고, 만약 변화가 감지된 경우에는 다시 새로운 학습을 시작한다. 대용량 온라인 게임과 같이 다수의 사용자가 있는 게임에서는 지능 캐릭터들이 다양한 행동 패턴과 전략을 가지고 있는 여러 상대 캐릭터들과 대진한다. 개체군 수준 적응 알고리즘은 유전자 알고리즘을 이용하여 지능 캐릭터들의 출현을 조절하여 게임 월드내의 균형이 유지되도록 한다. 제안한 알고리즘의 성능을 평가하기 위하여 간단한 대전 액션 게임을 구현하고 그 환경 상에서 게임 규칙과 상대 캐릭터들의 행동 패턴을 변화시키면서 실험하였다. 실험 결과 지능 캐릭터는 제안한 기법을 이용하여 환경 변화에 적응할 수 있음을 보였다.

Abstract

Recently intelligent characters in computer games have been an important element more and more because they continually stimulate gamers' interests. As a typical method for implementing such intelligent characters, neural networks have been used for training action patterns of opponent's characters and game rules. However, some of the game rules can be abruptly changed and action properties of gamers in on-line game environments are quite different according to gamers. In this paper, we address how a neural network adapts to those environmental changes. Our adaptation solution includes two components: an individual adaptation mechanism and a group adaptation mechanism. With the individual adaptation algorithm, an intelligent character steadily checks its game score, assesses the environmental change with taking into consideration of the lastly earned scores, and initiates a new learning process when a change is detected. In multi-user games, including massively multiple on-line games, intelligent characters confront diverse opponents that have various action patterns and strategies depending on the gamers controlling the opponents. The group adaptation algorithm controls the birth of intelligent characters to conserve an equilibrium state of a game world by using a genetic algorithm. To show the performance of the proposed schemes, we implement a simple fighting action game and experiment on it with changing game rules and opponent characters' action patterns. The experimental results show that the proposed algorithms are able to make intelligent characters adapt themselves to the change.

Keywords : 게임, 인공지능, 신경망, 적응, 유전자 알고리즘

I. 서 론

지금까지의 많은 연구를 통하여 컴퓨터 게임 캐릭터에 지능을 부여하려는 노력이 있었으나 여러 가지 이유로 성공적으로 구현된 경우는 거의 없다^[1]. 특히 대전 액션 게임의 경우에는 관련된 연구조차도 그리 많지 않

* 학생회원, 국민대학교 전자공학과
(Department of Electrical Eng., Kookmin Univ.)

** 정회원, 한성대학교 정보통신공학과
(Department of Information and Communication Eng., Hansung Univ.)

*** 정회원, 국민대학교 전자정보통신공학부
(School of Electrical Eng., Kookmin Univ.)

접수일자: 2005년1월3일, 수정완료일: 2005년5월2일

다. 그러나 지능 캐릭터는 게이머들의 흥미를 계속적으로 유발시킬 수 있기 때문에 컴퓨터 게임 발전에 매우 중요한 요소로서 최근 들어 관련 연구가 집중하고 있다^[2]. 최근 들어 대전 액션 게임에서 지능 캐릭터를 구현하기 위해 신경망을 이용하여 상대 캐릭터의 행동 패턴과 게임 규칙을 학습하는 방법들이 연구되었다^[3,4,5]. 그 연구들의 결과에 따르면, 초기에는 게임의 규칙이나 상대방의 행동 특성을 전혀 모르던 지능 캐릭터도 여러 번의 대전을 통하여 게임의 규칙을 파악하고 상대방 캐릭터 상대방에게 적절한 대응을 할 수 있다. 그러나 위의 연구에서는 오직 하나의 상황에서 하나의 게임 캐릭터만을 상대로 지능 캐릭터가 학습하였기 때문에 게임의 규칙이 변경되거나 지능 캐릭터의 상대 캐릭터가 바뀌는 경우에는 지능 캐릭터가 이에 대응하지 못하는 문제가 있다. 이러한 이유로 최근 컴퓨터 게임 분야에서 중요한 이슈 중의 하나는 캐릭터가 외부 환경 변화에 능동적으로 적응하는 방법을 고안하는 것이다^[2,6].

본 논문에서는 학습 능력을 가진 지능 캐릭터를 환경 변화에 적응시키는 방법을 제안한다. 제안된 알고리즘은 개체 수준 적응 알고리즘과 개체군 수준 적응 알고리즘으로 구성된다. 개체 수준 적응 알고리즘은 하나의 지능 캐릭터가 독립적으로 환경 변화에 적응하는 알고리즘으로서, 지능 캐릭터가 이미 학습한 지식을 이용하여 행동을 결정했을 때 과거와는 현저히 다른 결과를 획득하면, 환경이 변화했다는 것으로 인식하여 다시 학습하는 방법이다. 개체군 수준 적응 알고리즘은 대용량 온라인 게임과 같이 지능 캐릭터가 군집을 이루고 환경에서 적응할 수 있도록 하는 방법으로, 유전자 알고리즘을 이용하여 변화된 환경에 적절히 대응하게 하는 방법이다. 여러 종류의 외부 환경의 변화가 가능하나 본 논문에서는 대표적인 것으로 상대 캐릭터의 행동 패턴의 변화와 게임 규칙의 변화를 사용하였다. 행동 패턴의 변화란 A라는 행동 패턴을 보이는 상대방에 맞게 학습된 지능 캐릭터가 B라는 행동 패턴을 보이는 상대방이 등장했을 경우에 이러한 상황 변화를 어떻게 인지하고 변화된 상황에 어떻게 적응할 것인지를 보는 것이다. 또한 게임 규칙의 변화란 예를 들어 게임에서의 점수 규칙이 변했다든가 하는 경우 이를 어떻게 인지하고 변화된 상황에 어떻게 학습을 할 것인지를 보는 것이다. 개체 수준 적응에서는 각 개체가 이러한 상황 변화를 인식하고 적응할 수 있는 알고리즘을 제시하며, 개체군 수준 적응에서는 변화된 상황에 잘 적응하는 개체가 살아남게 하는 유전자 알고리즘을 기반으로 세대별

로 적응하는 방법을 제안한다. 본 논문에서 제안한 적응 알고리즘들의 성능을 평가하기 위하여 실제 게임과 유사한 대전 액션 게임을 구현하여 실험하였다. 실험 결과, 제안한 두 가지 적응 알고리즘 모두 상대 캐릭터의 행동 패턴의 변화와 같은 환경 변화에 적절히 적응할 수 있음을 보였다.

본 논문의 구성은 다음과 같다. II절에서는 적응 알고리즘의 대상이 되는 신경망 지능 캐릭터와 기존 적응 기법을, III절에서는 개체 수준 적응 알고리즘을 설명한다. IV절에서는 개체군 수준 적응 알고리즘을, V절에서는 제안한 적응 알고리즘들을 검증하기 위해 구현한 대전 액션 게임과 실험 결과를 알아보고, 마지막으로 VI절에서 결론을 맺는다.

II. 신경망 지능 캐릭터와 적응 기법

지금까지 게임에 인공지능을 적용하기 위하여 다양한 방법들이 연구되었다^[1-6]. 본 논문의 저자들은 많은 방법 중 신경망을 이용하여 게임의 지능 캐릭터를 구현하는 방법을 이전 논문에서 제안하였고, 학습 능력이 있음을 보였다^[3,4,5]. 그러나 이러한 지능 캐릭터는 게임의 환경 변화나 게임 규칙의 변화에 대하여 아무런 적응 능력을 갖추지 못하였다. 따라서 본 논문에서는 환경 변화를 스스로 인지하고 이에 대처하는 지능 캐릭터를 구현하기 위한 개체 수준 및 개체군 수준 알고리즘을 제안한다. 제안한 적응 기법은 논문 [3-5]에서 구현한 지능 캐릭터에 적용하여 실험한다. 이 절에서는 본 논문에서 제안하는 적응 알고리즘의 대상이 되는 신경망 지능 캐릭터에 대해서 간단히 설명한다.

논문 [3-5]에서 구현한 신경망 지능 캐릭터는 대전 액션 게임에서 게임의 규칙을 학습하여 상대 캐릭터의 행동과 단계 등의 조건에 따라 자신에게 유리한 행동을 결정하는 지능 캐릭터이다. 지능 캐릭터를 구현하는 신경망의 구조는 그림 1과 같다^[5]. 신경망에 입력되는 값들로서 상대 캐릭터의 행동, 행동의 단계와 거리는 각각 이산적인 정수로 입력한다. 또한 상대 캐릭터의 행동 패턴에 따라 적절한 대응을 하기 위해 상대 캐릭터의 과거 행동도 정수로 입력받는다. 출력은 지능 캐릭터가 취할 수 있는 행동 개수만큼의 출력 노드가 있으며, 지능 캐릭터는 그 중에서 가장 큰 값을 출력하는 행동으로 자신의 행동을 결정한다.

예를 들어 거리 2에서 상대 캐릭터의 아래주먹공격이 단계 3을 수행 중이고 과거 행동은 <전진, 위주먹,

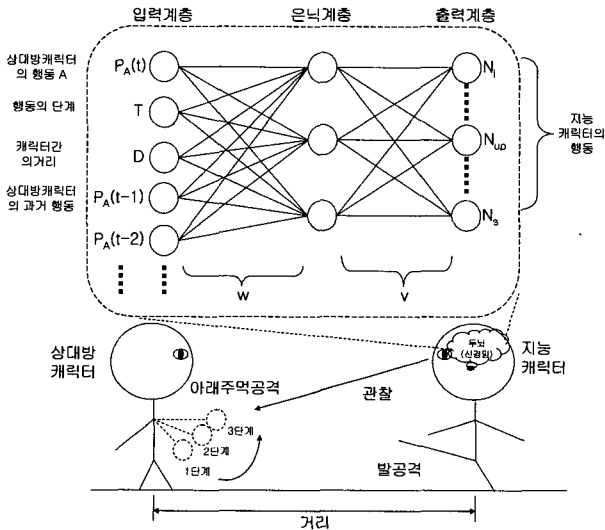


그림 1. 신경망의 구성
Fig. 1. The Structure of the neural network.

았기>였다면, 그 순간의 신경망의 입력은 $P_A(t)=7, T=3, D=2, P_A(t-1)=5, P_A(t-2)=8, P_A(t-3)=2$ 이 된다*. 결정된 입력값을 신경망에 적용하고 출력값을 계산한다. 신경망은 전방향 신경망 (feedforward neural networks)을 사용하며 일반적인 전방향 신경망과 동일하게 출력을 계산한다^[8,9].

신경망의 입출력 값이 결정되면 게임 규칙에 따라 계산된 두 캐릭터 간의 점수의 차를 강화값으로 사용하여 신경망을 강화 학습한다. 강화 학습은 오류 역전파 알고리즘을 이용하여 수행된다. 학습이 완료되면, 특정 상황에서 출력 노드들 중에서 가장 큰 값을 출력하는 노드의 행동이 바로 가장 지능 캐릭터가 많은 점수를 획득하는 행동과 일치하게 된다. 그러므로 지능 캐릭터는 신경망 출력에서 가장 큰 출력을 내는 노드의 행동으로 결정한다. 신경망 지능 캐릭터의 보다 세부적인 동작 방식 및 알고리즘은 논문 [3-5]를 참고하기 바란다.

위에서 설명한 지능 캐릭터는 변화된 환경에 적응하지 못하는 문제가 있다. 신경망을 이용하여 구현한 지능 캐릭터를 실제 게임에 투입하여 게임을 하는 사람의 상대 캐릭터로 사용한다고 가정하자. 실제 게임에 투입하기 전에 학습했다, 아니면 게임을 진행하면서 학습했다 이 지능 캐릭터는 현재 게임 규칙과 학습할 때 상대했던 캐릭터의 행동 패턴을 학습했을 것이다. 그런데 만약 상대하는 캐릭터의 행동 패턴이 변경되거나 게임 규칙이 변경된다면 지능 캐릭터가 이전에 학습한 행동

을 하더라도 좋지 않은 결과를 나타낼 것이다. 이러한 결과는 지능 캐릭터가 변화된 외부 환경에 적응하지 못했기 때문에 발생하며, 게임을 하는 사람에게 이 게임의 인공지능이 좋지 않다는 인상을 주게 된다. 이러한 문제를 해결하기 위해 게임에 적응 기법을 적용하기 위한 연구가 이루어졌다. 이러한 연구는 대부분 유전자 알고리즘으로 구현한 지능 캐릭터를 적용시키는데 초점을 맞추고 있다^[2,6,7]. 반면에 본 논문에서는 신경망을 이용한 지능 캐릭터를 적용시키는 방법을 제안한다.

III. 개체 수준 적응 알고리즘

본 논문에서는 외부 환경의 변화의 크기에 비례해서 지능 캐릭터가 획득하는 점수의 변화가 발생할 것이라는 가정을 기반으로 하여 개체 수준 적응 알고리즘을 제안한다. 개체 수준 적응 알고리즘을 간단히 설명하면 다음과 같다. 일정 기간 동안 지능 캐릭터가 학습한 행동으로 게임을 반복하면서 점수의 변화를 관찰한다. 그 점수 변화가 미리 정한 기준 값보다 크다면, 외부 환경이 변한 것으로 인식하여 다시 학습하는 과정을 반복한다. 다음은 개체 수준 적응 알고리즘을 기술한 것이다.

```

알고리즘 1: 개체 수준 적응 방법
learn 단계:
    while (n1 클럭 동안){
        r%의 확률로 지능 캐릭터의 결정 대신 무작위로 행동 선택
        행동의 결과로 지능 캐릭터 학습
    }
fight 단계:
    현재 점수차 = 0
    while (n2 클럭 동안){
        지능 캐릭터의 결정으로 행동
        현재 점수차 += 지능 캐릭터의 점수 - 상대 캐릭터의 점수
    }
    점수차의 변화 계산
    점수차의 변화 = |(현재 점수차-과거 점수차)/과거 점수차|
    과거 점수차 = 현재 점수차
    if (점수차의 변화 < ε){
        goto fight
    }
set 단계:
    점수차의 변화에 비례해서 n1과 r% 재조정
    n1 = A * diff_score
    r = diff_score * 100%
    goto learn
    
```

알고리즘 1에서 r은 학습할 때 지능 캐릭터의 행동을 무작위로 선택할 확률이고, n1은 학습시간을 결정할 때 사용하는 값이다. n2는 환경이 변했는지 알아보기 위해 대결하는 시간을 나타내고, ε은 점수 차의 허용 변화율이다. 본 논문에서는 각 값의 초기 값으로 r=100%,

* 여기에서 아래주먹은 7, 전진은 2, 위주먹은 8, 앓기는 5로 매핑된다고 가정한다.

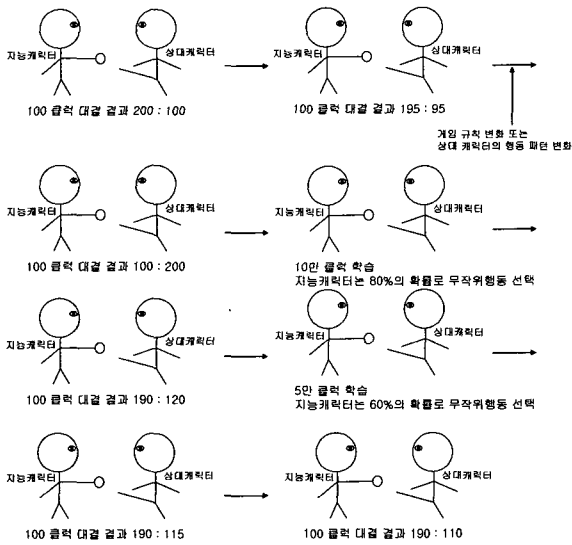


그림 2. 적응 알고리즘의 동작 방식
Fig. 2. Process of adaptation algorithm.

$n1=50$ 만, $n2=100$, $\epsilon=1\%$ 를 사용한다. 지능 캐릭터는 처음에는 게임에 대해서 어떠한 지식도 가지고 있지 않다. 따라서 지능 캐릭터는 무작위 행동으로 학습해야 한다(learn 단계). 어느 정도 학습 후 $n2$ 클릭 동안 신경망의 출력 즉 지능 캐릭터의 결정대로 행동해본다(fight 단계). 그 때 점수의 변화가 허용 변화율보다 작다면 환경이 변하지 않은 것으로 인식하고 다시 일정 시간 대결한다(fight 단계). 점수의 변화가 허용 변화율보다 크다면 환경이 변화한 것으로 인식하여 그 변화의 크기에 비례해서 학습 시간과 무작위 선택 확률을 조정하여(set 단계) 다시 학습한다(learn 단계). 본 논문에서는 $n1$ 을 조정하는 식에서 A 를 20만으로 설정했다. 지능 캐릭터는 이 과정을 반복하면서 환경 변화에 적응해간다. 그림 2는 개체 수준 알고리즘의 동작 방식을 예를 들어 설명한다.

먼저 지능 캐릭터가 상대 캐릭터와 100 클릭을 싸운 결과 점수가 200 대 100 (점수차 100)이었다(fight 단계). 그 후 100 클릭을 다시 싸웠더니 결과 점수가 195 대 95 (점수차 100)가 되었다(fight 단계). 이 때에는 점수차의 변화가 없으므로 게임 규칙이나 상대 캐릭터의 행동 변화가 없다고 생각할 수 있다. 따라서 적응할 필요 없이 다시 100 클릭 동안 싸운다(fight 단계). 그 결과 점수가 100 대 200 (점수차 -100)이라면, 점수차의 변화(200)가 크므로 도중에 게임 규칙이나 상대 캐릭터의 행동 변화가 있었다고 생각할 수 있다. 따라서 지능 캐릭터는 점수차의 변화 비율대로 확률과 학습 시간을 재조정한다(set 단계). 재조정된 확률만큼 신경망의 출

력대로 행동하지 않고 무작위로 행동하여, 결과로 나타나는 점수로 학습한다(learn 단계). 이 학습 과정은 지능 캐릭터가 과거에 학습한 행동이 나쁜 결과를 보이므로 지능 캐릭터에게 보다 나은 행동을 찾아보기 위한 과정이다. 예를 들어 그림에서는 점수차의 변화가 상당히 크므로 학습 시간은 10만 클릭, 무작위 확률은 80%로 크게 설정한다는 것을 보여준다*. 학습 후 다시 100 클릭 동안 싸운 후의 결과가 190 대 120 (점수차 70)이라면, 점수차가 -100에서 70으로 변한다(fight 단계). 이 때에도 점수차가 크게 변했지만 이전(200)보다는 변화의 크기가 작다. 따라서 이전 학습 과정보다는 학습 시간과 무작위 확률을 작게 하여(set 단계) 학습한다(learn 단계). 그 후 다시 100 클릭 동안 싸운 후의 점수가 190 대 115 라면(fight 단계), 이 때의 점수차는 75로 이전 점수차 70에서 75로 변한 것은 환경 변화에 어느 정도 적응했다고 판단할 수 있다. 따라서 학습하지 않고 100 클릭 동안 싸우는 과정을 반복한다.

이 절에서 설명한 개체 수준 적응 알고리즘은 하나의 지능 캐릭터가 독립적으로 외부 환경에 적응하는 알고리즘이다. 그러나 최근 들어 각광을 받고 있는 대용량 온라인 게임(MMOG: Massively Multi-player On-line Game)에는 지능 캐릭터로 구현할 수 있는 수많은 NPC(Non-Player Character)가 있다. 대용량 온라인 게임과 같이 지능 캐릭터가 많이 존재할 경우, 외부 환경이 변화했을 때 각각의 지능 캐릭터가 학습한 지식 서로 교환하면서, 진화해 나가면 변화된 환경에 쉽게 적응할 수 있을 것이다. IV절에서는 이에 대한 내용을 설명한다.

IV. 개체군 수준 적응 알고리즘

이 절에서는 대용량 온라인 게임과 같이 게임 내에 캐릭터가 다수가 있는 경우를 위한 개체군 수준 적응 알고리즘을 제안한다. 그림 3은 제안하는 알고리즘의 개략적인 동작 방식을 나타낸다.

* 실제 게임에서는 지능 캐릭터가 10만 클릭이라는 긴 시간을 무작위로 행동한다면, 게임을 하는 사람은 지능 캐릭터의 지능이 낮다고 생각할 것이다. 본 논문에서는 지능 캐릭터가 적응하는지 빠르게 알아보기 위해서 무작위로 행동하는 확률을 상당히 크게 설정했다. 그러나 실제 게임에 적용할 때는 무작위로 행동하는 확률을 줄이고, 대신 학습 시간을 크게 하면 비슷한 적응 효과를 보면서 게임을 하는 사람에게도 어느 정도 지능이 있는 것처럼 보일 수 있다.

그림 3에서 보여주듯이 게임과 캐릭터 풀을 분리하여 캐릭터가 게임에 가능한 한 독립적으로 구현될 수 있도록 한다. 개체군 수준 적응 알고리즘의 동작 방식은 다음과 같다. 두 캐릭터 풀에서 각각 하나의 캐릭터가 게임에 투입되어 게임을 한다. 게임의 결과가 지능 캐릭터의 적합도로 평가되고, 지능 캐릭터 풀에서는 각 지능 캐릭터의 적합도 정보를 이용하여 부모가 선택되고, 그 부모 캐릭터는 자식 캐릭터를 생성하는 과정을 반복하면서 진화해 나간다.

본 논문에서는 진화 방법으로서 “개체군” 수준에서 적용시키는데 적합한 알고리즘인 유전자 알고리즘을 사용한다. 그림 4는 유전자 알고리즘에서 세대가 진행되어 가면서 지능 캐릭터가 각각의 행동 패턴에 적응해가는 모습을 보여준다. 특정 행동 패턴 A, B, C를 각각

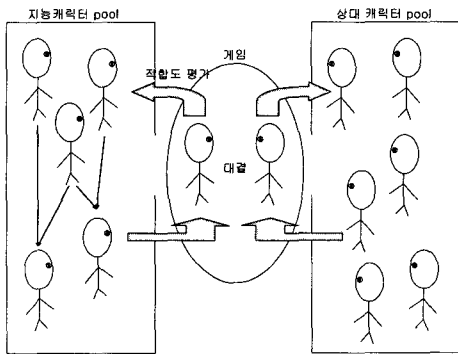


그림 3. 개체군 수준 적응 알고리즘의 개략적인 동작 방식
Fig. 3. Scheme of the adaptation algorithm for population.

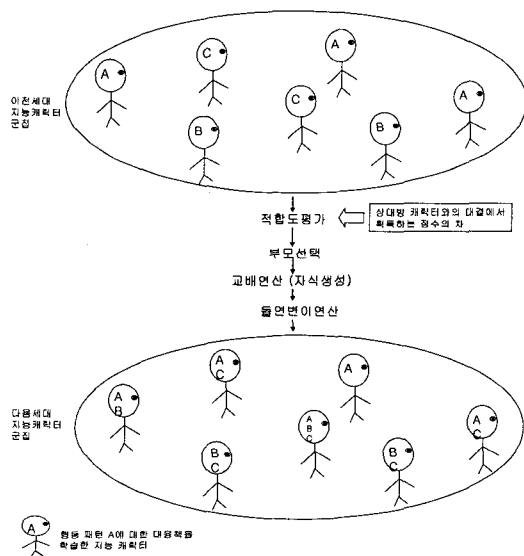


그림 4. 개체군 수준 적응 알고리즘
Fig. 4. Adaptation algorithm for population.

학습한 지능 캐릭터 군집이 있을 때, 먼저 군집에 있는 모든 지능 캐릭터에 대해 적합도를 평가한다. 이 적합도는 행동 패턴 A, B, C로 행동하는 상대 캐릭터와 차례대로 대결해서 획득한 점수의 차로 평가한다. 적합도가 계산되면 그 적합도에 비례해서 자식을 생성할 부모를 선택한 후, 그 부모가 교배 연산을 통해 자식을 생성한다. 그리고 국부 최적에 빠지지 않기 위해 각 지능 캐릭터에 대해 돌연변이 연산을 수행하면 다음 세대의 지능 캐릭터 군집이 생성된다. 세대를 반복할수록 지능 캐릭터 풀에는 모든 행동 패턴에 대해서 학습한 지능 캐릭터가 많아지게 된다. 이것을 하나의 지능 캐릭터의 관점에서 보면, 그 지능 캐릭터는 지금까지는 특정 행동 패턴에 대한 경험만을 해왔지만, 다른 행동 패턴을 가진 상대 캐릭터가 나타난 것, 즉 외부 환경의 변화에 대해 적응한 것으로 볼 수 있다. 개체군 수준 적응 알고리즘의 기본이 되는 유전자 알고리즘은 개체, 교배 연산자, 돌연변이 연산자, 적합도 함수 등으로 구성되며, 다음은 각각에 대한 구현 내용을 설명한다.

4.1 개체

유전자 알고리즘은 문제에 대한 해의 후보를 개체로 표현하며 일반적으로 이진 스트링으로 표현한다. 그러나 이진 스트링은 지능 캐릭터를 표현하는 데에는 부적합하므로 그림 5와 같이 신경망 링크의 가중치를 하나의 유전자로 표현하며, 개체는 지능 캐릭터의 모든 링크의 가중치를 1차원 배열 형태로 표현한다. 그림 5는 신경망의 링크가 18개일 경우, 개체가 모든 링크의 가중치를 저장하는 1차원 배열로 표현되는 것을 나타낸다.

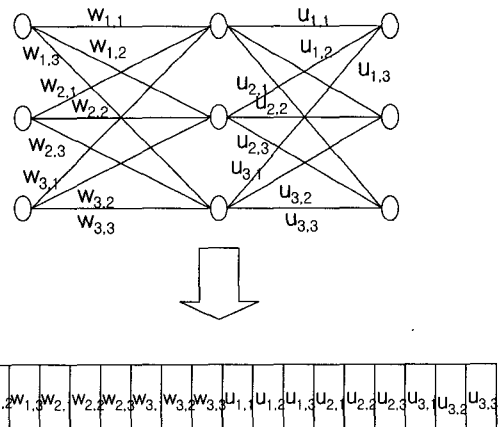


그림 5. 개체의 표현
Fig. 5. Representation of individual.

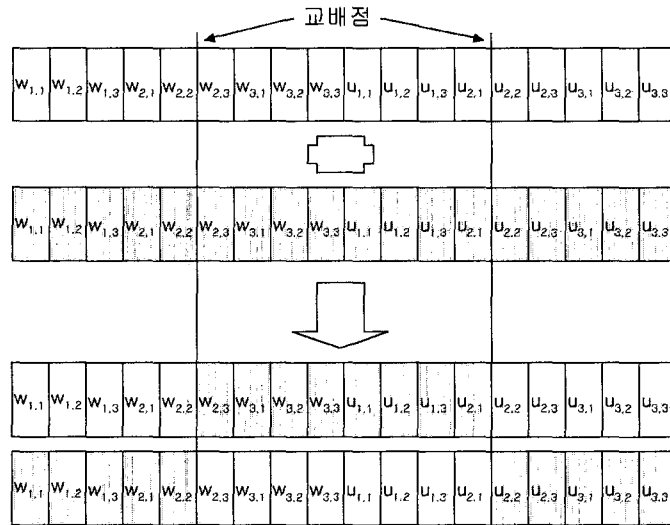


그림 6. 교배 연산자
Fig. 6. Crossover operator.

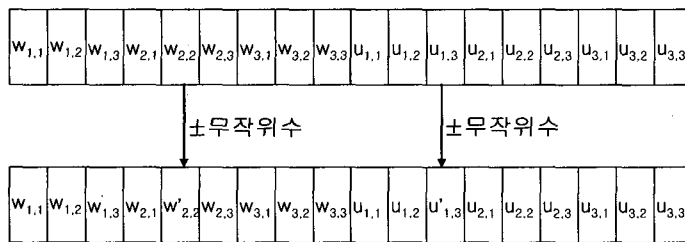


그림 7. 돌연변이 연산자
Fig. 7. Mutation operator.

4.2 교배 연산자

교배(crossover) 연산자는 선택된 개체들이 교배를 통해 서로의 유전 정보를 교환함으로써 새로운 개체를 생성하는 과정이다. 이때 개체의 어느 부분을 선택할 것인지는 무작위수에 의해 결정되도록 하였다. 본 논문에서는 개체의 형태가 1차원이므로 일반적으로 많이 사용되는 n-점 교배 연산을 사용하여 지능 캐릭터가 학습한 지식을 교환하도록 한다^[10]. 그림 6은 임의의 두 점을 선택한 후, 그 사이의 내용을 교환하여 새로운 두 개의 개체를 생성하는 2-점 교배 연산을 나타낸다.

4.3 돌연변이 연산자

돌연변이(mutation) 연산자는 하나의 개체 중에서 무작위로 선택된 문자열을 임의로 변경하는 것이다. 앞에서 설명한 교배 연산자에 의해서 개체들 사이의 유전 정보가 서로 교환되지만 모든 해의 공간을 탐색하기 위한 유전 정보가 현재 집단내의 유전자들 속에 들어 있지 않으면 교배 연산자를 아무리 적용시키더라도 더 이

상 탐색이 이루어질 수 없다. 이와 같이 부모 해에 존재하지 않는 속성을 도입하여 탐색 영역을 확대시킨다. 본 논문에서는 각 유전자에 대해 돌연변이 확률로, 일정 범위의 무작위 실수를 발생시켜 그 수를 더하거나 빼으로써, 지능 캐릭터가 새로운 지식을 탐색하도록 한다. 그림 7은 개체에서 돌연변이 확률로 선택된 두 유전자($w_{2,2}$, $u_{1,3}$)에서 무작위수를 더하거나 빼서, 새로운 유전자($w'_{2,2}$, $u'_{1,3}$)로 변경되는 것을 나타낸다.

4.4 적합도 함수

각 개체의 적합도는 다음 세대로 유전되는 비율에 영향을 미친다. 본 논문에서는 지능 캐릭터가 특정 행동 패턴대로 행동하는 상대 캐릭터와 일정 회수 대결 후, 두 캐릭터가 획득한 점수의 차를 적합도로 사용한다. 예를 들어 지능 캐릭터가 어떤 상대 캐릭터와 대결하여 50 대 10의 결과가 나왔다면, 그 지능 캐릭터의 적합도는 $40(=50-10)$ 으로 평가한다. 이와 같이 점수의 차를 적합도로 평가하면, 상대 캐릭터보다 점수를 많이 획득하

여 점수의 차가 큰 지능 캐릭터가 적합도가 높으므로, 다음 세대에 유전될 확률이 높아지고, 세대가 진행될 수록 상대 캐릭터에 대해 적절한 대응을 하는 지능 캐릭터가 많아진다.

V. 실험 결과

본 논문에서 제안한 지능 캐릭터의 적응 알고리즘을 검증하기 위하여 [5]에서 구현한 대전 액션 게임을 이용했다. 이 게임에서는 두 캐릭터가 제한된 1차원 게임 공간에서 이동하면서 공격과 방어를 하며, 그 결과에 따라 점수를 획득한다. 캐릭터의 모든 행동은 클릭에 동기가 맞춰져 있다. 각각의 행동을 하는데 소요되는 시간, 공격이 성공했을 때 획득하는 점수와 공격의 유효거리를 표 1에 나타냈다.

캐릭터가 할 수 있는 공격은 크게 주먹공격, 발공격 그리고 먼 거리에서 행할 수 있는 필살기의 세 종류로 구분되며, 주먹공격과 발공격은 각각 상대 캐릭터의 하반신을 공격하는 아래공격과 상반신을 공격하는 위공격으로 세분화된다. 실제 게임처럼 각각의 공격에 따라 공격의 효과를 얻을 수 있는 유효거리와 공격점수를 설

표 1. 각 행동의 소요시간/공격점수/유효거리
Table 1. Necessary time/Attack score/Effective range.

공격행동	소요시간(클릭)	공격점수	유효거리
정지	1	-	-
전진			
후진			
점프			
앉기			
막기			
아래주먹	2	1	0~2
위주먹	4	2	
아래발	6	3	2~3
위발	8	4	
필살기	10	5	3~5

표 2. 실험에 사용한 행동 패턴과 각 패턴에 대한 지능 캐릭터의 최적 행동

Table 2. Action patterns for experiments and the optimal actions for each action pattern.

번호	상대 캐릭터의 행동 패턴	지능 캐릭터의 최적 행동	캐릭터간의 초기거리	점수
A	전진, 위주먹, 후진, 아래발, 후진, 필살기, 전진, 위발	위주먹, 아래발, 필살기, 위발	3	14
B	전진, 위주먹, 후진, 아래발, 전진, 아래주먹, 후진, 위발	위주먹, 아래발, 아래주먹, 위발		10
C	후진, 필살기, 전진, 아래발, 후진, 필살기, 전진, 위발	필살기, 아래발, 필살기, 위발		17

정하였다. 아래주먹과 아래발공격은 상대방이 막거나 점프를 하면 점수를 얻지 못하고, 위주먹과 위발공격은 상대방이 막거나 앉으면 점수를 얻지 못한다. 필살기 공격은 상대방이 막으면 50%의 점수만을 획득한다.

5.1 개체 수준 적응 실험

[5]에서 구현한 지능 캐릭터에 개체 수준 적응 알고리즘을 추가하여 실험했고, 무작위 초기값 20개에 대해서 실험하여 평균 점수비를 조사했다. 실험에 사용한 행동 패턴을 표 2에 나타냈다.

그림 8에서 (1) 항목은 어떠한 행동 패턴에 대해서도 학습하지 않은 지능 캐릭터가 각각의 행동 패턴에 대해서 학습한 후의 평균 점수비이다. (2) 항목은 상대 캐릭터의 행동 패턴이 변경되었을 때 적응하지 않고 대결한 경우의 점수비이다. 그리고 (3) 항목은 어떤 행동 패턴에 대해서 학습 후 상대 캐릭터의 행동 패턴이 달라진 경우 적응 알고리즘이 수렴했을 때의 평균 점수비이다. 예를 들어 그림 8에서 학습하지 않은 지능 캐릭터가 행동 패턴 A에 대해서 학습한 결과 점수비는 39.7이고, 상대 캐릭터의 행동 패턴이 B로 변경되면 그 때 대결한 점수비는 평균 1.5가 된다. 지능 캐릭터가 변경된 행동 패턴 B에 적응한 후에는 점수비가 45.7로 증가한다. 실험 결과, 지능 캐릭터는 상대 캐릭터의 행동 패턴이 변경되면 점수비가 상당히 낮아지지만, 적응 후에는 점수비가 대부분 6이상으로 증가하여 행동 패턴 변화에 적응했음을 알 수 있다.

다음은 그림 8의 (1)과 (3)에 대해서 적응 알고리즘의 각 단계별 수행 내용에 대하여 설명한다. 먼저 게임에 대해서 어떠한 지식도 학습하지 않은 지능 캐릭터를 개



그림 8. 개체 수준 적응 알고리즘의 실험 결과

Fig. 8. Experimental results of adaptation algorithm for individual.

표 3. 행동 패턴 A에 대해 학습하는 과정
Table 3. Learning process for action pattern A.

순서	단계	학습시간 (n1)	대결시간 (n2)	무작위선택 확률(r)	지능캐릭 터점수	상대방캐 릭터점수
①	learn	500000	-	100%	-	-
②	fight & set	-	100	0%	18	0
③	learn	8500000	-	100%	-	-
④	fight & set	-	100	0%	47	0
⑤	learn	805555	-	100%	-	-
⑥	fight & set	-	100	0%	47	0

표 4. 행동 패턴 A에 대한 학습 결과 행동
Table 4. Actions in results of learning for action pattern A.

번호	상대 캐릭터의 행동 패턴	지능 캐릭터의 학습 행동
A	전진, 위주먹, 후진, 아래발, 후진, 필살기, 전진, 위발	위주먹, 아래발, 필살기, 위발

표 5. 행동 패턴 B에 대해 적응하는 과정
Table 5. Process of adaptation for action pattern B.

순서	단계	학습시간 (n1)	대결시간 (n2)	무작위선택 확률(r)	지능캐릭 터점수	상대방캐 릭터점수	점수차
①	fight & set	-	100	0%	25	20	5
②	learn	446808	-	89%	-	-	-
③	fight & set	-	100	0%	45	0	45
④	learn	4000000	-	100%	-	-	-
⑤	fight & set	-	100	0%	50	0	50
⑥	learn	55555	-	10%	-	-	-
⑦	fight & set	-	100	0%	50	0	50

체 수준 적응 알고리즘을 이용하여 행동 패턴 A에 대해 학습하는 과정을 표 3에 나타냈다.

①은 지능 캐릭터가 학습하지 않은 상태이기 때문에 우선 일정 시간 무작위 행동을 해서 학습한다. 그 후 ②에서 지능 캐릭터가 학습한대로 행동하여 점수를 계산한다. 지능 캐릭터는 이 과정에서 처음 대결한 것이기 때문에 old_score가 정해져 있지 않다. 따라서 본 논문에서는 초기 old_score를 1로 설정하여 점수의 변화(diff_score)를 계산한다. 점수의 변화가 상당히 크기 때문에 ③에서는 학습 시간도 상당히 길게, 무작위 선택 확률도 100%로 크게 설정되어 학습한다. 학습 후 ④에서 지능 캐릭터가 학습한대로 행동하여 점수를 계산한다. 이때에도 역시 점수의 변화가 18에서 47로 상당히 크기 때문에 ⑤에서 학습 시간과 무작위 선택 확률이 크게 설정되어 학습한다. 학습 후 ⑥에서 지능 캐릭터가 학습한대로 행동하면 점수의 변화가 없어서 이 때 적응 알고리즘은 학습하지 않고 계속 상대 캐릭터와 대결하게 된다. 마지막 학습 과정에서 지능 캐릭터는 표 4와 같은 행동 패턴 A에 대해서 최적 행동을 학습했다.

다음으로 행동 패턴 A에 대해 학습 후 상대 캐릭터의 행동 패턴이 B로 변경되는 경우 지능 캐릭터가 적응하는 과정을 표 5에 나타냈다.

표 6. 행동 패턴 B에 대한 학습 결과 행동
Table 6. Actions in results of learning for action pattern B.

번호	상대 캐릭터의 행동 패턴	지능 캐릭터의 학습 행동
B	전진, 위주먹, 후진, 아래발, 전진, 아래주먹, 후진, 위발	위주먹, 아래발, 아래주먹, 위발

표 7. 공격 점수의 변화
Table 7. Changes of score of attack.

공격행동	변경 전 공격점수	변경 후 공격점수
아래주먹	1	5
위주먹	2	4
아래발	3	3
위발	4	2
필살기	5	1



그림 9. 개체 수준 적응 알고리즘의 실험 결과
Fig. 9. Experimental results for adaptation algorithm for individual.

①에서 상대 캐릭터의 행동 패턴이 B로 변경된다면 지능 캐릭터는 그 패턴에 대해서 학습한 내용이 없기 때문에 이전 보다 점수를 많이 획득하지 못하고 점수의 변화가 발생한다. 따라서 ②에서 학습 시간과 무작위 선택 확률을 각각 446808 클릭과 89%로 설정하여 학습한다. 학습 후 ③에서 지능 캐릭터가 학습한대로 행동하여 점수를 계산한다. 이 때 계산되는 점수차의 변화도 상당히 크기 때문에 아직 환경 변화에 제대로 적응하지 못했다고 판단하여 ④에서 다시 학습한다. 이와 같은 과정을 반복하면 ⑦에서 점수의 변화가 발생하지 않게 되고, 적응 알고리즘은 더 이상 적용되지 않는다. 마지막 학습 과정에서 지능 캐릭터가 학습한 행동은 표 6과 같이 행동 패턴 B에 대하여 최적 행동을 학습했다.

지금까지는 상대 캐릭터의 행동 변화에 대한 적응 실험을 살펴보고, 다음은 게임 규칙의 변화에 대한 적응 여부를 알아보는 실험이다. 본 논문에서는 게임 규칙의 변화로서, 표 7과 같이 공격에 대한 점수를 변경하였다. 실험 방법은 앞의 실험과 비슷하게 무작위 초

기값 30개에 대해서 실험하여 평균 점수비를 조사하였다.

각각의 행동 패턴에 대해서 적응한 지능 캐릭터가 게임 규칙이 표 7과 같이 변경되었을 때의 실험 결과를 그림 9에 나타냈다. 그림 9에서 (1) 항목은 어떠한 행동 패턴에 대해서도 학습하지 않은 지능 캐릭터가 각각의 행동 패턴에 대해서 학습한 후의 평균 점수비이다. (2) 항목은 같은 행동 패턴에 대해서 게임 규칙이 변경되었을 때 적응하지 않고 대결한 경우의 점수비이다. 그리고 (3) 항목은 변경된 게임 규칙에 적응 한 후의 평균 점수비이다. 그림에서 알 수 있듯이, 적응을 함으로써 적응 전보다 높은 점수비를 획득함으로써, 변경된 게임 규칙에 적응했음을 알 수 있다.

5.2 개체군 수준 적응 실험

4절에서 제안한 개체군 수준 적응 알고리즘을 실험한 환경은 다음과 같다. 유전자 알고리즘의 개체수는 30, 교배 확률은 0.8, 돌연변이 확률은 0.1로 설정했고 최대 10000 세대까지 수행했다. 적합도는 상대 캐릭터와 일정 횟수 대결한 점수차로 평가한다. 초기 개체는 무작위 확률로 A, B, C 중 한 가지 행동 패턴에 대해서 최적 행동을 학습한 지능 캐릭터로 초기화된다. 확률적으로 초기 개체 중 33.3%는 행동 패턴 A에 대해 학습한 지능 캐릭터, 33.3%는 행동 패턴 B에 대해 학습한 지능 캐릭터, 나머지는 행동 패턴 C에 대해 학습한 지능 캐릭터로 초기화된다. 개체가 초기화되면 이 상황은

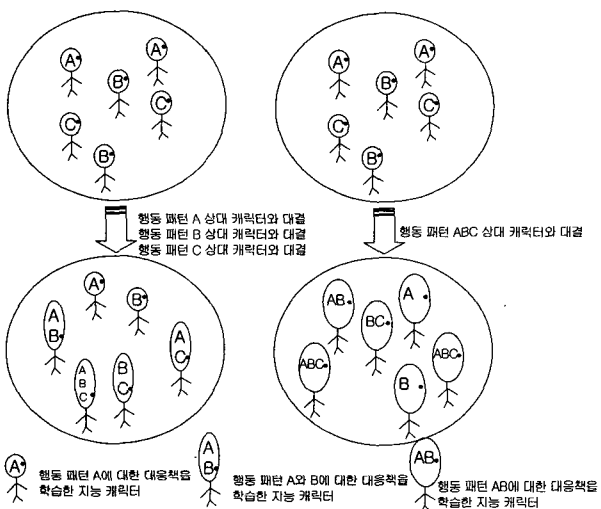


그림 10. 개별 행동 패턴 적응 실험과 연결 행동 패턴 적응 실험

Fig. 10. Experiments of adaptation for separate action pattern and connected action pattern.

각각의 지능 캐릭터가 하나의 행동 패턴에 대해서만 경험해본 것으로 생각할 수 있다. 그 후 그림 10에 나타난 개별 행동 패턴 적응과 연결 행동 패턴 적응에 대해서 실험한다.

개별 행동 패턴 적응 실험은 각각 행동 패턴 A, B, C로 행동하는 상대 캐릭터가 나타난 것이고, 연결 행동 패턴 적응 실험은 ABC가 연결된 행동 패턴을 갖는 지능 캐릭터가 나타난 것이다. 그림 10에 지능 캐릭터의 머리에 행동 패턴이 세로로 나타난 것과 가로로 나타난 것 두 가지가 있다. 행동 패턴이 세로로 나타난 것은 지능 캐릭터가 표시된 각각의 행동 패턴에 대해서 학습한 것을 의미하고, 가로로 나타난 것은 표시된 행동 패턴이 연결된 행동 패턴에 대해서 학습한 것을 의미한다. 예를 들어 행동 패턴 A는 전진과 주먹으로 이루어져 있고, 행동 패턴 B는 후진과 발차기로 이루어졌다고 가정하자. 가로로 AB가 나타난 지능 캐릭터는 두 행동 패턴이 연결된 <전진, 주먹, 후진, 발차기>라는 행동 패턴에 대해서 학습한 것을 의미한다. 이 행동 패턴은 지능 캐릭터의 입장에는 입력되는 과거 행동이 달라지기 때문에 새로운 행동 패턴이라고 할 수 있다. 본 실험에서 보이고자 하는 결과는 처음에는 하나의 행동 패턴에 대해서만 학습한 지능 캐릭터가 있는 상황에서, 개별 행동 패턴 적응 실험에서는 진화를 통해 이미 학습한 행동 패턴이 아니라 환경 변화로 나타난 다른 행동 패턴에 대해서도 학습한 지능 캐릭터가 나타난다는 것이다. 연결 행동 패턴 적응 실험에서는 진화 과정을 거쳐 지능 캐릭터가 기존에 학습한 행동 패턴과는 큰 차이가 있는 새로운 행동 패턴에 대해서도 적응할 수 있다는 것을 보이고자 한다.

개별 행동 패턴 적응 실험은 먼저 30개의 지능 캐릭터 중 10 개씩 하나의 행동 패턴에 적응시킨 후 행동

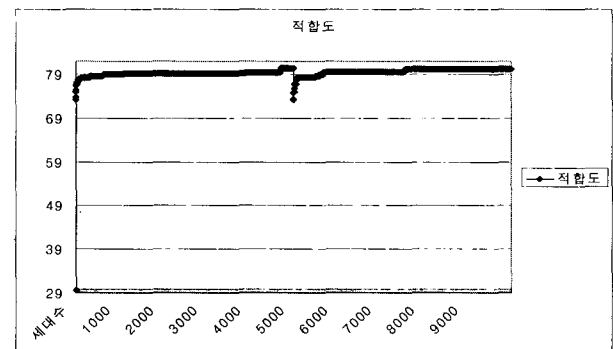


그림 11. 개별 행동 패턴 적응 실험

Fig. 11. Experiment of adaptation for separate action pattern.

패턴 A, B, C로 행동하는 상대 캐릭터가 나타난 경우에 대한 실험이다. 이 실험에서는 5점 교배 연산을 사용했다. 각 세대마다 가장 높은 적합도를 갖는 지능 캐릭터는 반드시 다음 세대에 유전되도록 했으며, 10개의 무작위 초기값에 대해 각 세대에서 가장 높은 적합도의 평균을 그림 11에 나타냈다.

그래프의 1세대에서 5000 세대까지는 10개 단위의 지능 캐릭터가 각각의 행동 패턴에 적응하는 과정이다. 최적 행동을 학습했을 때 지능 캐릭터는 세 행동 패턴과 1번씩 대결하면 41점의 점수차를 획득하지만, 그림 11의 결과에서는 5000 세대에서 지능 캐릭터는 평균적으로 40점의 점수차를 획득하는 행동으로 학습하였다. 그러나 10번의 실험 중 8번의 실험에서 최적 행동을 학습함을 볼 수 있었다. 5000 세대에서 행동 패턴 A, B, C로 행동하는 상대 캐릭터가 나타난 경우에도 초기 세대에 빠르게 적응하여 5000 세대 후에는 80까지 도달한다. 최적 행동을 학습했을 때 지능 캐릭터는 세 행동 패턴과 1번씩 대결하면 41점의 점수차를 획득하지만, 그림 11의 결과에서는 10000 세대에서 지능 캐릭터는 평균적으로 40점의 점수차를 획득하는 행동으로 학습하였다. 그러나 10번의 실험 중 8번의 실험에서 최적 행동을

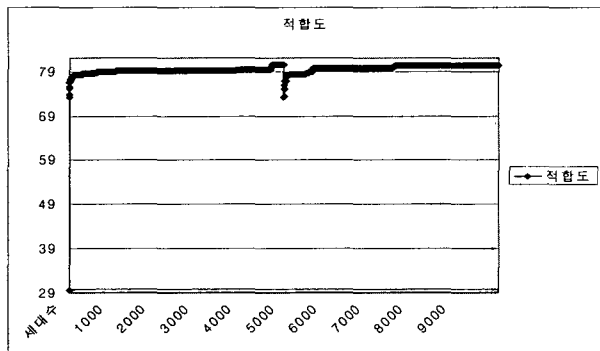


그림 12. 연결 행동 패턴 적응 실험
Fig. 12. Experiment of adaptation for connecting action pattern.

표 8. 실험에 사용한 행동 패턴과 각 패턴에 대한 지능 캐릭터의 최적 행동
Table 8. Action patterns for experiments and the optimal actions for each action pattern.

상대 캐릭터의 행동 패턴	지능 캐릭터의 최적 행동	캐릭터간의 초기거리	점수
전진, 위주먹, 후진, 아래발, 후진, 필살기, 전진, 위발, 전진, 위주먹, 후진, 아래발, 전진, 아래주먹, 후진, 위발, 후진, 필살기, 전진, 아래발, 후진, 필살기, 전진, 위발	위주먹, 아래발, 필살기, 위발, 위주먹, 아래발, 아래주먹, 위발, 필살기, 아래발, 필살기, 위발	3	41

을 학습함을 볼 수 있었다.

연결 행동 패턴 적응 실험에서도 5점 교배 연산을 사용하여 그림 12와 같은 결과를 보였다.

최적 행동을 학습했을 때 지능 캐릭터는 행동 패턴과 1번 대결하면 41점의 점수차를 획득한다(표 8 참조). 그림 12의 실험 결과에서 보듯이 5000 세대에서 환경이 변화하면 지능 캐릭터는 초기 세대에 빠르게 적응하여 5000 세대 후에는 80까지 증가하고, 10번 중 7번의 실험에서 최적 행동을 학습했다. 다음의 그림 13과 그림 14에서 1000 세대 이전은 30개의 지능 캐릭터 중 1/3씩 하나의 행동 패턴에 대해서 최적 행동을 학습했다는 가정 하에 그림 10의 실험을 수행한 결과이다. 1000 세대 이후는 표 7과 같이 게임 규칙이 변경되었을 때 개체군 수준 적응 알고리즘이 적응하는 과정을 보여준다.

그림 13과 14의 결과 모두 1000 세대에 게임 규칙이 변경되면 일시적으로 적합도가 감소하지만, 급격히 증가하여 평균적으로 120 가까이 증가함을 알 수 있다. 변경된 게임 규칙에서는 가까운 거리에서의 공격 점수가 높기 때문에 상대 캐릭터가 규칙적으로 행동할 경우 지능 캐릭터는 전진과 아래주먹을 반복하면 높은 점수를 획득할 수 있다. 위의 두 실험 모두 2000세대에서 지

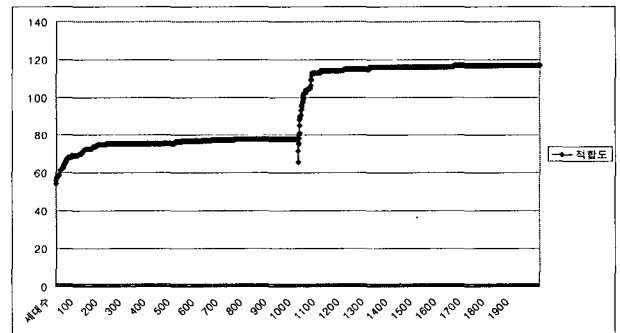


그림 13. 개별 행동 패턴 적응 실험-게임 규칙 변경 후
Fig. 13. Experiment of adaptation for separate action pattern - after the game rule is changed.



그림 14. 연결 행동 패턴 적응 실험-게임 규칙 변경 후
Fig. 14. Experiment of adaptation for connecting action pattern - after the game rule is changed.

능 캐릭터는 전진과 아래주먹을 반복하도록 적응하였음을 알 수 있었다.

5.1과 5.2의 실험 결과들로서 본 논문에서 제안한 개체 수준 적응 알고리즘과 개체군 수준 적응 알고리즘이 외부 환경 변화로 간주할 수 있는 상대 캐릭터의 행동 패턴의 변화와 게임 규칙의 변화에 대하여 적절히 적응할 수 있음을 보았다.

VI. 결 론

본 논문에서는 일반적인 대전 액션 게임에서 신경망을 이용하여 구현한 지능 캐릭터가 상대 캐릭터의 행동 양식의 변화와 같은 환경 변화에 적응하는 알고리즘으로서 개체 수준 적응 알고리즘과 개체군 수준 적응 알고리즘을 제안하였다. 개체 수준 적응 알고리즘은 하나의 지능 캐릭터가 독립적으로 환경 변화에 적응하는 알고리즘이다. 이 알고리즘에서는 지능 캐릭터가 과거에 학습한 지식을 이용하여 행동을 결정했을 때 과거와는 다른 결과를 획득하면, 환경이 변화했다는 것으로 인식하여 환경 변화에 비례한 시간만큼 학습한다. 개체군 수준 적응 알고리즘은 군집을 이루고 있는 지능 캐릭터가 진화하면서 환경 변화에 적응하는 알고리즘이다. 이 알고리즘에서는 유전자 알고리즘을 이용하여 교배와 돌연변이를 거치면서 변화된 환경에 적절히 대응하는 지능 캐릭터로 진화시킨다. 제안한 알고리즘의 성능을 평가하기 위하여 실제 게임과 유사한 게임 환경에서 외부 환경을 변화시키면서 실험하였다. 실험 결과, 제안한 두 가지 적응 기법 모두 상대 캐릭터의 행동 패턴의 변화와 게임 규칙의 변화와 같은 환경 변화에 적응할 수 있음을 보였다. 제안한 알고리즘은 대전 액션 게임은 물론 최근 각광을 받고 있는 온라인 게임의 캐릭터에도 쉽게 적용할 수 있다.

참 고 문 헌

- [1] 류광, 장원석, 인공지능 게임 프로그래밍, 정보문화사, 2003.
- [2] 이면섭, 조병헌, 정성훈, 성영락, 오하령, "대전 액션게임에서의 유전자 알고리즘을 이용한 지능 캐릭터," 전자공학회 논문지 제 41권 TE편 제4호, pp.119-128. 2004. 12.
- [3] 조병헌, 정성훈, 성영락, 오하령, "신경망을 이용한 지능형 게임 캐릭터의 구현," 한국정보처리학회 논문지, 제11-B권 제7호, pp.831-840, 2004. 12.
- [4] 조병헌, 정성훈, 성영락, 오하령, "대전 액션 게임을 위한 신경망 지능 캐릭터의 구현," 한국 퍼지 및 지능시스템학회 논문지 Vol. 14, No. 4, pp.383~389, 2004. 7.
- [5] 조병헌, 정성훈, 성영락, 오하령, "대전 게임에서 상대방 캐릭터의 행동 패턴을 학습하여 대응하는 신경망 지능 캐릭터," 전자공학회 논문지 제 41권 CI편 제6호, pp. 579-590, 2004. 11.
- [6] 안태홍, 강성관, 이상규, 김우정, 김흥기, "유전자 알고리즘을 사용한 지능적인 실시간 게임 캐릭터," 컴퓨터산업교육기술학회 논문지, VOL. 02, NO. 10, pp.1309~1316, 2001.10.
- [7] Laramee, Francois Dominic, AI Game Programming Wisdom, Charles River Media, 2002.
- [8] Chin-Teng Lin, C. S. George Lee, Neural Fuzzy Systems, Prentice Hall, 1996.
- [9] Richard. P. Lippman, An Introduction to Computing with Neural Nets, IEEE ASSP Magazine, pp. 4-22, April, 1987.
- [10] David E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning, Addison Wesley, 1989.

저 자 소 개



조 병 현(준회원)
 1997년 국민대학교 전자공학과 (공학사)
 1999년 국민대학교 전자공학과 (공학석사)
 2005년 국민대학교 전자공학과 (공학박사)

<주관심분야 : 유전자 알고리즘, 시뮬레이션, 신경망>



정 성 훈(정회원)
 1988년 한양대학교 전자공학과 (공학사)
 1991년 한국과학기술원 전기 및 전자공학과(공학석사)
 1995년 한국과학기술원 전기 및 전자공학과(공학박사)

1995년~1996년 한국과학기술원 전기 및 전자공학과(위촉연구원)
 1996년~1998년 한성대학교 정보전산학부 정보통신공학전공 전임강사
 1998년~2002년 한성대학교 정보전산학부 정보통신공학전공 조교수
 2002년~현재 한성대학교 정보통신공학과 부교수

<주관심분야: 지능시스템, 유전자알고리즘, 뉴로퍼지, 시스템생물학>



성 영 락(정회원)
 1989년 한양대학교 전자공학과 (공학사)
 1991년 한국과학기술원 전기 및 전자공학과(공학석사)
 1995년 한국과학기술원 전기 및 전자공학과(공학박사)

1995년~1996년 한국과학기술원 위촉연구원
 1996년~1998년 국민대학교 전자공학부 전임강사
 1998년~2002 국민대학교 전자공학부 조교수
 2002년~현재 국민대학교 전자정보통신공학부 부교수

<주관심분야 : 시뮬레이션, 고장감내, 내장형 시스템>



오 하 령(정회원)
 1983년 서울대학교 전기공학과 (공학사)
 1983년~1986년 삼성전자 종합연구소
 1988년 한국과학기술원 전기 전자과 컴퓨터공학전공 (공학석사)

1992년 한국과학기술원 전기전자과 컴퓨터공학전공(공학박사)
 1992년~1996년 국민대학교 전자공학부 조교수
 1996년~2001 국민대학교 전자공학부 부교수
 2001년~현재 국민대학교 전자정보통신공학부 교수

<주관심분야 : 병렬처리, 내장형 시스템, 고장감내>