2000년 한국정보처리학회 추계 학술발표논문집 제 7 권 제 2 호

강화 학습을 이용한 다중 에이전트 조정 전략

김수현*, 김병천**, 윤병주*
*명지대학교 컴퓨터공학과
**한경대학교 컴퓨터공학과
e-mail : {kimsh, yoonbj}@mju.ac.kr
e-mail : bckim@hnu.hankyong.ac.kr

Multi-agent Coordination Strategy Using Reinforcement Learning

Suhyun Kim*, Byungcheon Kim**, Byungjoo Yoon*
*Dept. of Computer Engineering, Myongji University
**Dept. of Computer Engineering, Hankyong University

요 약

본 논문에서는 다중 에이전트(multi-agent) 환경에서 에이전트들의 행동을 효율적으로 조정(coordination)하기 위해 강화 학습(reinforcement learning)을 이용하였다. 제안된 방법은 각 에이전트가 목표(goal)와의 거리 관계(distance relationship)와 인접 에이전트들과의 공간 관계(spatial relationship)를 이용하였다. 그러므로 각 에이전트는 다른 에이전트와 충돌(collision) 현상이 발생하지 않으면서, 최적의 다음 상태를 선택할 수 있다. 또한, 상태 공간으로부터 입력되는 강화 값이 0 과 1 사이의 값을 갖기 때문에 각 에이전트가 선택한 (상태, 행동) 쌍이 얼마나 좋은가를 나타낼 수 있다. 제안된 방법을 먹이 포획 문제(prey pursuit problem)에 적용한 결과 지역 제어(local control)나, 분산 제어(distributed control) 전략을 이용한 방법보다 여러 에이전트들의 행동을 효율적으로 조정할 수 있었으며, 매우 빠르게 먹이를 포획할 수 있음을 알 수 있었다.

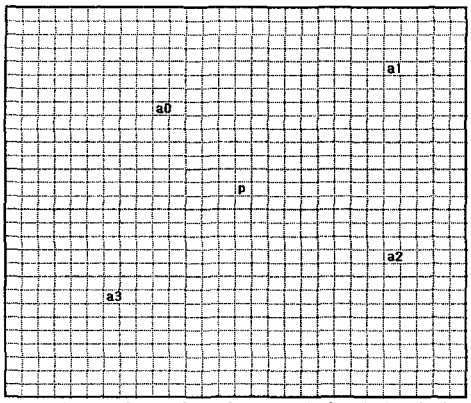
1. 서론

최근 실세계의 복잡한 문제를 해결하기 위해 다중 에이전트 시스템에 대한 연구가 활발히 진행되고 있다[1]. 일반적으로 다중 에이전트에 대한 연구는 여러 개의 에이전트들의 행동을 조정하기 위한 행동 전략(behavioral strategy)을 개발하는데 있다[2]. 본 논문에서는 다중 에이전트 시스템은 각 에이전트와 환경이 시간에 따라 변하는 동적 환경(dynamic environments)이기 때문에 에이전트들의 행동 전략을 개발하기 위해 강화 학습을 이용한 에이전트들의 행동을 조정하기 위한 전략을 제안하였다. 강화 학습은 동적 환경과 시행-착오(trial-error)를 통해 상호 작용하면서 학습을 수행하기 때문에 동적 환경에서 학습을 수행하기 위해 널리 이용되고 있다[3]. 그러나 강화 학습은 다른 에이전트들에 대한 (상태, 행동)쌍을 고려하지 않고 학습을 수행하며, 강화 값(reinforcement value)이 0 또는 1 이므로 에이전트가 선택한 (상태, 행동)쌍이 얼마나 좋은지를 나타낼 수 없기 때문에 다중 에이전트 환경에서 효율적으로 학습할 수 없다[4].

본 논문에서 제안한 방법은 각 에이전트가 목표와의 거리 관계와 인접 에이전트들과의 공간 관계를 이용하였다. 그러므로 각 에이전트는 다른 에이전트와 충돌 현상이 발생하지 않으면서, 최적의 다음 상태를 선택할 수 있으며, 상태 공간으로부터 입력되는 강화 값이 0 과 1 사이의 값을 갖기 때문에 각 에이전트가 선택한 (상태, 행동) 쌍이 얼마나 좋은가를 나타낼 수 있다. 본 논문에서 제안된 방법을 Stephen 과 Merx 가 제안한 먹이 포획 문제(prey pursuit problem)[5]에 적용한 결과 지역 제어(local control) 전략[6]나 분산 제어(distributed control) 전략[7]을 이용한 방법보다 여러 에이전트들의 행동을 효율적으로 조정할 수 있었으며, 매우 빠르게 먹이를 포획할 수 있음을 알 수 있었다.

2. 먹이 추적 문제

Stephens 과 Merx 가 제안한 먹이 추적 문제는 (그림 1)과 같이 30*30 의 Grid 환경에서 1 개의 먹이와 4 개의 에이전트들로 구성되어 있다.



(그림 1) 다중 에이전트를 위한 실험 환경

먹이는 중앙에서 시작하여 임의의 방향{북, 남 , 동, 서, ***s***}으로 이동할 수 있으며 ***s*** (stay)는 이동하지 않고 제자리에 머물러 있는 것을 말한다. 에이전트는 임의의 위치에서 시작하여 {북, 남, 동, 서}의 방향으로 수직 또는 수평으로 이동할 수 있다.

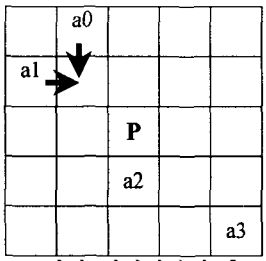
먹이 추적 문제에서 에이전트들의 목표는 먹이가 움직일 수 없도록 (그림 2)와 같이 4 개의 에이전트가 상호 협력하여 먹이를 포위하는 것이다.

		a0		
	a3	**P**	a1	
		a2		

(그림 2) 에이전트의 목표

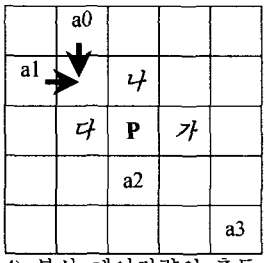
일반적으로 각 에이전트가 먹이를 효율적으로 포획하기 위해 널리 이용되고 있는 전략은 지역 제어 전략과 분산 제어 전략이 있다. 지역 제어 전략은 비-대화형 전략으로서 다른 에이전트의 위치를 알 수 없으며, 먹이를 발견하였을 때 먹이의 위치를 다른 에이전트에게 알려주어 먹이와 가장 가까운 위치로 상태 전이를 수행하여 먹이를 포획하는 전략이다. 분산 제어 전략은 대화형 전략으로서 각 에이전트들은 자신의 위치와 먹이의 위치를 다른 에이전트들에게 전달하여 먹이와 가장 먼 곳에 있는 에이전트가 먹이와 가장 가까운 위치로 상태 전이를 수행하여 먹이를 포획하는 전략이다. 지역 제어 전략과 분산 제어 전략의 문제점은 에이전트들이 먹이를 추적하는 과정에서 충돌이 발생하는 것이다[8].

지역 제어 전략의 경우 다른 에이전트의 상태를 고려하지 않으므로 (그림 3)과 같이 에이전트 a0 와 a1 가 화살표 방향으로 이동할 경우 충돌 현상이 발생한다.



(그림 3) 지역 제어전략의 충돌 현상

지역 제어 전략에서 발생하는 충돌 현상을 방지하기 위해 제안된 분산 제어 전략도 (그림 4)와 같이 충돌 현상이 발생한다.



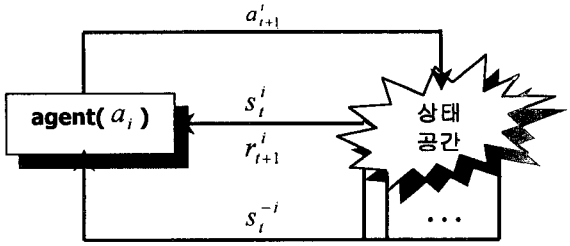
(그림 4) 분산 제어전략의 충돌 현상

(그림 4)에서 먹이와 가장 먼 거리에 있는 에이전트 a3 가 먼저 ***가*** 지역을 목표로 이동한다. 그리고 나서 a0 와 a1 은 ***나*** 와 ***다*** 지역이 같은 거리에 있기 때문에 임의의 한 곳을 목표로 정한다. 이 과정에서 a0 는 ***다***, a1 은 ***나*** 를 목표로 하여 화살표 방향으로 이동할 경우 충돌이 발생한다.

이처럼 충돌 문제는 다중 에이전트 시스템에서 중요한 문제로 제기되고 있으며, 본 논문에서는 먹이와의 거리 관계와 인접한 에이전트들과의 공간적 관계를 이용하여 먹이 포획과정에서 충돌 문제를 해결하였다.

3. 다중 에이전트 조정 전략

다중 에이전트 환경에서 각 에이전트들은 <S, A, *h*, R>로 정의되며, S 는 상태들의 유한 집합, A 는 에이전트가 선택할 수 있는 행동들의 유한집합, *h* 는 상태전이 함수, $R(r^0, r^1, \cdots, r^{n-1})$ 은 각 에이전트가 외부환경으로부터 받는 강화 값을 의미한다. 다중 에이전트 환경에서 에이전트들의 행동을 효율적으로 조정하기 위해 각 에이전트와 상태 공간과의 관계는 (그림 5)과 같다.



(그림 5) 에이전트와 상태 공간과의 관계

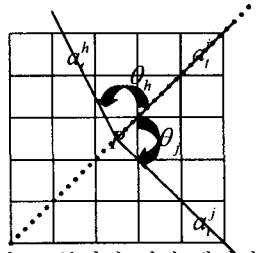
(그림 5)에서 에이전트 a_t^i 는 상태 공간으로부터 에이전트 상태(s_t^i), 강화 값(r_{t+1}^i) 그리고 에이전트와 인접한 다른 에이전트들의 상태(s_t^{-i})들을 입력 받아, 최적의 행동을 선택한다. 에이전트가 특정 행동을 선택하였을 때 외부 환경으로부터 받는 강화 값은 식(1)과 같이 계산된다.

$$r_t^i = \frac{DR\ (a_t^i, p) + SR\ (a_t^h, a_t^i, a_t^j)}{2} \qquad \text{식(1)}$$

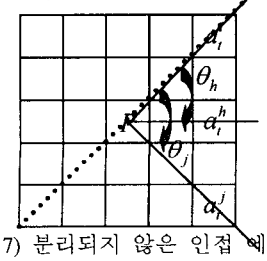
식(1)에서 $DR(a_t^i, p)$ 은 에이전트와 먹이와의 거리 관계를 의미하며 식(2)와 같이 계산된다.

$$DR\ (a_t^i, p) = \frac{1}{\sqrt{(s_{t,x}^i - p_x)^2 + (s_{t,y}^i - p_y)^2}} \qquad \text{식(2)}$$

식(1)에서 $SR(a_t^h, a_t^i, a_t^j)$ 은 에이전트와 먹이 그리고 인접 에이전트들과의 공간적 관계를 의미하며, 인접 에이전트가 (그림 6)과 같이 에이전트와 먹이에 이르는 직선을 기준으로 서로 분리된 경우와 (그림 7)과 같이 인접 에이전트가 분리되지 않은 경우로 나누어진다.



(그림 6) 분리된 인접 에이전트



(그림 7) 분리되지 않은 인접 에이전트

먹이 P 의 위치를 중심으로 에이전트 a_t^i 와 인접 에이전트 a_t^h 가 이루는 각을 θ_h, 에이전트 a_t^i 와 인접 에이전트 a_t^j 가 이루는 각을 θ_j 라 하자. 분리된 인접 에이전트인 경우의 공간관계 $SR(a_t^h, a_t^i, a_t^j)$ 은 식(3)과 같이 계산하며, 먹이와 에이전트 a_t^i 를 잇는 직선으로 분리되지 않은 경우 인접 에이전트의 공간관계 $SR(a_t^h, a_t^i, a_t^j)$ 은 식(4)와 같이 계산된다.

$$SR\ (a_t^h, a_t^i, a_t^j) = \frac{\max(\ \theta_h, \theta_j\ )}{2\pi} \qquad \text{식(3)}$$

$$SR(a_t^h, a_t^i, a_t^j) = \frac{2\pi - \max(\theta_h, \theta_j)}{2\pi} \qquad \text{식(4)}$$

본 논문에서 사용한 식(1)과 같은 강화 값은 0 과 1 사이의 값을 가지며, 에이전트가 상태 전이를 위해 선택한 (상태, 행동) 쌍이 얼마나 좋은지를 나타낼 수 있다.

에이전트는 외부 환경으로부터 입력되는 강화 값에 따라 현재 상태에서 식(5)과 같은 상태 전이 함수에 의해 강화 값이 최대인 행동을 선택하여 다음 상태로 이동한다.

$$h = \max(r_{t+1}^i, a_t^i) \qquad \text{식(3)}$$

먹이 추적 문제에서, 각 에이전트들의 행동을 효율적으로 조정하기 위해 본 논문에서 제안한 방법의 수행 절차는 (그림 8)과 같다.

```
PursuitProblem()
{
  Initialize Q(s,a)   for all s, a;
  Repeat {
    PreyMove ()       //임의의 방향으로 움직인다.
    For all  s_t^i, a, (a ∈ A(s_t^i))
        Calculate  r_{t+1}^i ;
    AgentMove();      //가장 큰 r_{t+1}^i 값을 갖는 방향
  }until (Prey can't move)
}
```

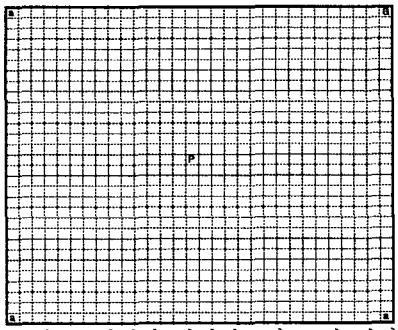
(그림 8) 제안된 방법의 수행 절차

4. 평가 및 분석

일반적으로 다중 에이전트 환경에서, 에이전트들의 성능을 평가하기 위해 Stephen 과 Merx 가 제안한 (그림 1)과 같은 환경을 이용하고 있으며, 성능 평가의 기준은 먹이 포획의 성공 율(success ratio)과 얼마나 빨리 먹이를 포획할 수 있는가(number of state transitions)를 기준으로 하고 있다.

본 논문에서 제안한 강화 학습을 이용한 행동 조

정 전략과 지역 제어 전략, 분산 제어 전략들을 비교 분석하기 위해 (그림 9)과 같이 에이전트와 먹이의 상태를 고정 시켰다. 즉, 먹이는 (14, 14) 에서 임의의 방향으로 움직이도록 하였고, 에이전트들은 각각 (0, 0), (0, 29), (29, 0), (29, 29)에서 수평 또는 수직 방향으로 움직이도록 하였다. 각 방법을 30 번씩 수행한 후 평균 값으로 비교 하였다.



(그림 9) 먹이와 에이전트의 초기 위치

논문에서 제안된 강화 학습을 이용한 행동 전략과 지역 제어전략 그리고 분산 제어 전략의 상태 전이 수와 먹이를 포획할 성공 율은 [표 1]과 같다.

[표 1] 성능 분석

기준 / 전략	성공 율	상태 전이 수
지역 제어	57	228.301
분산 제어	96	947.891
제안된 방법	**100**	**214.537**

[표 1]에서 나타난 것처럼 지역 제어 전략은 먹이 포획의 성공률이 57%이고, 분산 제어 전략은 96%이다. 그러나 제안된 방법은 30 회 실행하여 30 회 모두 먹이를 포획하였다. 또한 먹이를 포획하기 위해 지역 제어 전략은 228.301 회, 전역 제어 전략은 947.891 회의 상태 전이가 발생하였지만 제안된 방법은 214.537 회의 상태 전이후 먹이를 포획할 수 있었다.

5. 결론 및 향후 연구과제

실험을 통해 제안된 방법은 먹이 포획을 위해 제안된 지역 제어 전략이나 분산 제어 전략을 이용한 방법보다 매우 빠르게 먹이를 포획할 수 있었으며, 먹이를 포획할 성공 율도 더 좋음을 알 수 있었다. 또한 제안된 방법은 인접 에이전트들과의 충돌 현상 없이 먹이를 포획할 수 있었다. 그러므로 다중 에이전트 환경에서는 에이전트들의 행동을 효율적으로 조정하기 위해서 목표와 에이전트 사이의 거리만을 고려하는 것 보다 목표와의 거리 그리고 인접 에이전트와의 공간적 관계를 함께 고려하여 행동하는 것이 더 효율적임을 알 수 있었다.

그러나 제안된 방법은 목표와 다른 에이전트들의 위치를 알고 있는 대화형(communication) 다중 에이전트이므로, 다른 에이전트들의 위치나 정보를 알지 못한 상태에서 목표를 효율적으로 찾을 수 있는 비대화형(non-communication) 다중 에이전트 시스템의 개발이 필요하다. 또한 제안된 방법은 먹이 추적 과정에서 먹이의 경로를 예측할 수 없다. 따라서, 먹이가 어떻게 움직일 것인가를 예측할 수 있는 방법에 대한 연구가 필요하다.

참 고 문 헌

[1] Peter Stone, Manuela Veloso, "Multiagent System : A Survey from a Machine Learning", Technical Report CMU-CS-97-193, The University of Carnegie Mellon, December, 4, 1997.

[2] Sandip Sen, Mahendra Sekaran, "Multiagent coordination with learning classifier systems", In Proceeding of the AAAI 99 Workshop on Negotiation, pp. 44-49, 1999.

[3] Michael L. Littman, "Markov games as a framework for multi-agent reinforcement learning", pp.157-163, SanMateo, CA., 1994.

[4] Caroline Claus, Craig Boutilier, "The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems", Proceedings of the 8th European Workshop on MAAMAW'97, 1997.

[5] L. M. Stephens, M. B. Merx, "The effect of agent control strategy on the performance of a DAI pursuit problem", In Proceeding of the 1990 Distributed AI Workshop, October, 1990.

[6] Edwin de Jong, "Multi-Agent Coordination by Communication of Evaluation", In Proceeding of the 8th Europeon Workshop on MAAMAW '97, 1997.

[7] Thomas Haynes, Sandip Sen, "Evolving behavioral strategies in predators and prey", Adaptation and Learning in Multiagent System, pp.113-126, Springer Verlag, Berlin, 1996.

[8] Thomas Haynes, Kit Lau and Sandip Sen, "Learning Cases to Compliment Rules for Conflict Resolution in Multiagent Systems" Co-evolution, and Learning in Multiagent Systems, Stanford, CA., March, 1996.