

방향 벡터를 이용한 다중에이전트 휴리스틱

김 현*, 이승관, 정태충
경희대학교 컴퓨터공학과
e-mail:kimhyun@iislab.khu.ac.kr

The multi agent control heuristic using direction vector

Hyun Kim*, SeungGwan Lee, TaeChoong Chung
*Dept of Computer Engineering, Kyunghee University

요 약

먹이추적문제(preys pursuit problem)는 가상 격자로 이루어진 공간 내에 다중의 에이전트를 이용하여 먹이를 포획하는 것이다. 에이전트들은 먹이를 포획하기 위해 30x30으로 이루어진 격자공간 (gride)안에서 기존 제한된 지역 제어, 분산 제어, 강화학습을 이용한 분산 제어 전략들을 적용하여 먹이를 포획하는 전략을 구현하였다. 제한된 격자 공간은 현실세계를 표현하기에는 너무도 역부족이어서 본 논문에서는 제한된 격자공간이 아닌 현실 세계와 흡사한 무한 공간 환경을 표현하고자 하였다. 표현된 환경의 모델은 순환구조(circular)형 격자 공간이라는 새로운 실험 공간이며, 새로운 공간에 맞는 전략은 에이전트와 먹이와의 추적 관계를 방향 벡터를 고려한 모델로 구현하였다. 기존 실험과는 차별화된 환경에서 에이전트들은 휴리스틱을 통한 학습을 할 수 있다는 가정과 먹이의 효율적 포획, 충돌문제 해결이라는 결과를 얻었다.

1. 서 론

다양하고 복잡한 환경에서 사용자의 요구와 문제 해결을 위해 에이전트가 개발 되어졌으며, 그 결과 단일 에이전트로 해결하지 못하는 복잡한 문제에 대한 해결 방안을 위해 동일 조건의 다수 에이전트의 협동을 통한 다중 에이전트 시스템(multi-agent system)이 제안되었다[1]. 효율성 있는 다중 에이전트 시스템 개발의 대표적인 실험환경인 먹이추적 문제는 격자 공간(grid)내에 4 개의 독립적 에이전트가 하나의 먹이를 포획하는 실험으로서 복잡한 현실 세계를 표현하기 위한 가장 간단한 모델이다[2].

기존 먹이추적 문제의 실험 환경은 격자 공간(grid) 상에 4개의 에이전트와 1개의 먹이가 존재하였고, 에이전트들이 협동을 통해 먹이를 포획하여 에피소드를 끝내는 것이다[3]. 여기서 먹이는 에이전트가 도달하고자 하는 목표이며, 각 에이전트는 최소의 비용과 시간을 통해 효율성 있게 목표 획득을 하는 것이 다중에이전트 연구의 목적과 동일하다 할 수 있다.

그러나 기존 실험 환경은 30x30의 격자 구조로 이루어져 일반성이 많이 떨어지며, 먹이는 에이전트를 피해 효과적으로 도망치지 못하고 단순히 불규칙적으로 움직이는 약점이 있다.

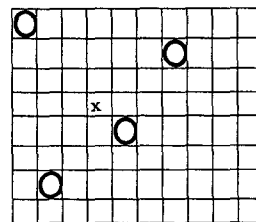
본 논문에서는 이러한 문제에 대해서 순환구조(circular)형 격자 공간이라는 새로운 실험 환경으로 바꾸어 실험을 하였다. 그리고 에이전트와 에이전트,

먹이와 에이전트 사이의 관계를 방향 벡터로 표현하고, 방향성을 고려한 휴리스틱을 통한 목표 획득을 하고자 하였다.

2. 관련 연구

2.1 먹이 추적 문제

먹이 추적 문제는 복잡한 현실세계에서의 에이전트의 활용을 실험하는 모델로 사용되어지고 있다. 먹이 추적 문제는 (그림 1)과 같이 격자(grid)환경에서 4개의 에이전트가 1개의 먹이를 전략에 따라 4방향에서 포획하는 실험이다. 에이전트와 먹이의 이동 조건(속도, 방향)은 동일하며 단, 먹이가 우선 움직임을 가진다. 이 실험을 통해 다중에이전트의 최적화 작업을 구현하고자 하였지만 에이전트간의 충돌과 불규칙한 포획의 성공률로 인해 다양한 전략들이 소개되었다.



(그림1) 먹이 추적 문제

2.2 비대화형 에이전트와 대화형 에이전트

효율적인 포획을 위해 Stephen과 Merx은 지역 제어 전략과 분산 제어 전략을 이용한 제어방법을 제안했다[4,5].

지역제어전략은 비대화형 에이전트들로 구성되며 각 에이전트는 목표인 먹이를 포획하기 위해 포획 위치(capture position)인 지역 목표를 먼저 설정한 후 이동을 한다. 초기에 에이전트들은 위치 및 먹이에 대한 정보가 전혀 없으며 한 개의 에이전트가 먹이 포획 위치를 설정시에만 에이전트들 간의 협동이 가능하다. 하지만 이 전략은 수렴 단계까지의 많은 상태 전이 발생과 포획위치 중복 결정에 따른 충돌 현상이 발생하는 문제점이 있다.

분산 제어 전략은 대화형 에이전트로 위의 지역 제어 전략과는 달리 초기에 각 에이전트들은 먹이를 중심으로 거리 정보를 생성한다. 거리의 생성 후 각 에이전트는 먹이와 가장 멀리 있는 에이전트에게 가장 가까운 포획 위치를 선정하도록 할당하며, 나머지 에이전트들도 먹이와 가장 가까운 위치를 포획 위치로 설정한다. 분산 제어 전략은 지역 제어 전략의 포획 위치의 중복으로 인한 충돌 현상을 극복하기 위해 제안된 전략으로 충돌 현상을 감소시켰으며 먹이를 중심으로 에이전트간의 협동을 잘 나타낸 전략이다. 하지만 먹이를 중심으로 위치는 다르고 거리가 같은 에이전트가 존재할 경우 충돌 문제가 발생한다. 다중에이전트 시스템에서 충돌 문제는 중요한 문제로 제기되기 때문에 충돌 해결을 위한 새로운 전략이 필요하게 되었다[6].

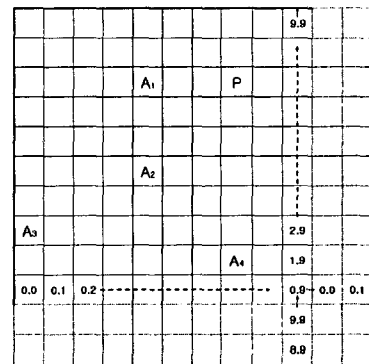
3. 순환구조(circular)형 격자공간에서의 에이전트 제어전략

3.1 현실성을 고려한 순환구조(circular)형 격자공간

본 논문에서 제안한 실험 환경의 변화는 기존 먹이 추적 문제의 실험환경이 공간적 제한과 주어진 환경을 100% 쓰지 못하는 불완전 포획이라는 점이 있었기 때문이다. 먹이포획을 위해 에이전트들은 주어진 공간 내에서 전략을 구사하지만 기존 실험환경으로는 현실성이 상당히 결여되어있는 것 같았다[7]. 이런 이유로 본 논문에서는 기존 실험환경을 변화시켜 무한공간과 유사한 순환구조(circular)형 격자 공간을 제안하였다. 또한 기존 단순화된 먹이가 아닌 에이전트를 회피하여 탈출할 수 있는 학습력과 지능을 주어 현실세계에 적용시킬 수 있는 실험을 하였다. 기존의 환경에서는 30*30의 격자 공간상에 4개의 에이전트가 먹이를 포획하여 에피소드가 종료되는 방식이었다. 하지만 본 논문에서는 제한된 공간의 실험환경은 현실성이 떨어지며 상태 행동에 대한 평가 시간이 길어진다는 것에 착안하여 보다 현실적이고 일반화된 환경을 제시하며 사각형의 그리드 구조를 순환구조(circular)형으로 만들어 무한 공간 상태와 비슷한 환경을 제시하였다.

본 논문에서 제안된 순환구조(circular)형 격자공간이란 무한공간 상태를 표현하기 위해 (그림 2)와 같이 구형한 연속된 격자 공간이다. 이것은 기존 실험 모델에서 먹이는 항상 가운데 있고 에이전트가 포위

할 수 있는 위치에서 시작하고 있다는 점과 먹이는 효과적으로 도망치지 않고 단순히 불규칙적으로 움직인다는 문제에 대하여 일반성을 고려하여 무한 공간의 비슷한 형태의 환경인 순환구조(circular)형 격자 공간이라는 것을 제안하고 그에 따른 전략을 세웠다. 에이전트가 목표를 획득하는 것은 무한 공간 상태라 할 수 있다. 무한 공간에서 먹이포획 방법을 한다면 에이전트의 속도가 먹이와 같을 때 에이전트들이 한쪽 방향으로 모여 있고 먹이가 한쪽으로만 도망간다면 영원히 포획하지 못한다. 이런 무한 공간의 문제점을 고려하고 다중에이전트의 실용성 있는 실험을 위해 기존 유한 그리드 구조를 무한 순환구조(circular)로 만들었다. 즉 (0,0)~(0,10)의 격자 공간에서 (0,10)의 오른쪽이 (0,11)이되어야 하지만 순환(round)시켜서 (0,0)이 되게 하는 것이다. 예를 들어 (3,3)의 상하좌우는 (2,3), (4,3), (3,2), (3,4)이지만 (0,0)의 상하좌우는 (29,0), (1,0), (0,29), (0,1)이 되는 것이다. 이럴 경우 에이전트들과 먹이의 속도가 같을 때에도 다중 에이전트이므로 먹이포획이 가능할 수 있다. 본 실험환경은 무한거리 모델인 메비우스 띠를 모델로 하여 순환구조(circular)형 격자 공간이라 명하였다. 본 환경으로 실험시 먹이와 에이전트의 위치는 무작위(random)가 될 수 있으며, 오직 완전포획만이 존재하게 되는 것이다. 또한 기존 실험의 에이전트의 학습과 달리 에이전트 뿐만 아니라 먹이에게도 에이전트들의 접근시 도망할 수 있는 능력을 부여했다.



(그림 2) 순환구조형 격자 공간

3.2 먹이의 에이전트에 대한 회피 능력 부여를 통한 문제의 일반화

기존 먹이추적문제에서 먹이의 역할은 에이전트의 목표대상이었다. 에이전트에겐 추적과 포획, 협동의 능력 주어졌지만 먹이는 무작위로 이동만을 반복하였다. 본 논문에서는 현실성을 고려해 먹이에게도 에이전트를 파악할 수 있는 능력(시야)을 주었다. 에이전트의 행동-상태에 따라 시야를 주어 에이전트와의 거리를 계산하여 회피할 수 있는 능력을 주었다.

3.3 방향벡터를 이용한 먹이 포획 전략 과 탈출 전략

기존 포획 전략들의 문제는 에이전트들이 한곳에 모여 있을 때의 상황이다. 무한 공간 상태에서 먹이는 에이전트의 반대 방향으로 이동시 분명 포획이 불가능하다. 순환구조(circular)형 격자 공간 역시 무한공간과 비슷한 꼴이 없는 순환(circular)하는 환경이기에 기본적인 전략으로는 포획하기가 힘들다. 실험 횟수의 반복에 따라 먹이를 포획하더라도 비효율적인 것을 알 수 있다. 이런 점에서 본 논문에서는 먹이와 에이전트간의 거리관계를 방향벡터로 표현하는 새로운 전략법을 구현하였다.

각 에이전트들은 먹이를 기점으로 서로의 상태 정보 값을 확인하고 먹이를 중심으로 이동한다. 먹이는 분명 에이전트를 피하기 위해 에이전트와의 공간관계를 계산하고 에이전트들이 없는 상태 공간 쪽으로 이동을 한다. 초기 설정에서 에이전트와 먹이의 이동속도가 같고 먹이가 에이전트보다 한 단계 먼저 이동할 때 포획이 상당히 어렵다는 것을 알 수 있을 것이다. 에이전트들은 먹이 포획을 위해 효율적으로 쫓아 가야하는 것을 판단하는 함수를 만들고, 먹이는 근접상태의 공간을 파악할 수 있는 능력을 주어 도망할 수 있는 정책이 필요하다. 이 함수들을 만들기 위해 방향벡터를 도입하였다. 인접한 에이전트와 먹이와의 거리와 다른 에이전트들의 상관관계를 반영하는 벡터를 만든다.

에이전트(A_i)가 먹이(P)쪽으로 이동하는 방향벡터는 먹이와 에이전트와의 거리를 반영한 것이다. 각 에이전트와 먹이간의 방향 벡터 \vec{VP} 의 크기는 먹이와 각 에이전트간의 거리에 반비례한다. 먹이가 가까울수록 먹이 방향으로 가려고 하는 벡터가 커져나오는 함수의 벡터가 커진다.

$$\vec{VP} = \vec{A_iP}$$

$$\rightarrow |\vec{A_iP}| = \frac{1}{(Px - A_{ix})^2 + (Py - A_{iy})^2} \quad (식1)$$

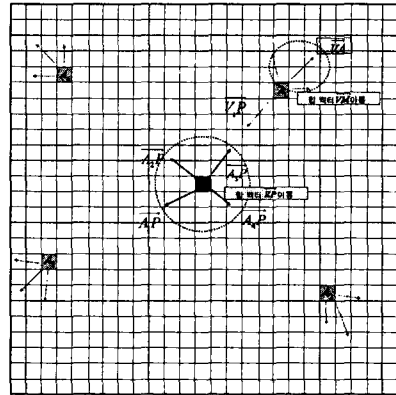
기존의 먹이추적문제 실험에서 제기되었던 문제는 에이전트간의 충돌 문제였다. 이것을 해결하고자 에이전트와 다른 에이전트들간의 상관관계를 고려한 벡터를 만들었다. 이 방식은 다른 에이전트들과 가까울수록 반대방향으로 값이 커지는 벡터 \vec{VA} 를 만드는 것이다. 값은 에이전트사이의 모든 방향 벡터의 합으로 구해진다.

$$\vec{VA} = \sum_{i \in J(A)} \vec{R}_i$$

$$\rightarrow |\vec{R}_i| = \left(\frac{1}{(R_{ix} - Ax)^2 + (R_{iy} - Ay)^2} \right) \quad (식2)$$

J(A)는 에이전트(A)이외의 이웃 에이전트의 집합, R은 에이전트(A)와 이웃 에이전트, \vec{R}_i 은 에이전트(A)와 에이전트(R)의 방향 벡터이다. 이것은 에이전트들 간의 거리가 가까울수록 반대방향으로 커진다.

에이전트와 다른 에이전트와의 벡터 값은 두 에이전트 거리의 제곱에 반비례하는 값을 갖게 하면 된다. 이 값은 에이전트 사이를 멀어지게 하는 효과가 있다. 이렇게 하면 에이전트는 다른 에이전트들과 다른 방향으로 가려고 하는 성격이 생기므로 에이전트간의 충돌문제를 해결하며 공동작업을 효과적으로



(그림 3) 방향 벡터를 이용한 전략

할 수 있다.

에이전트의 이동 방향은 \vec{VP} 와 \vec{VA} 를 합하는 방향으로 가면 되는 것이다. 그 합을 \vec{VM} 이라고 하자.

$$\vec{VM} = \vec{VP} + \vec{VA} \quad (식3)$$

그런데 \vec{VM} 을 구할 때 단순히 합하지 않고 $\alpha * \vec{VP} + \beta * \vec{VA}$ 해야 한다. α, β 값에 따라 어느 쪽을 중요시하는지를 알려주는 것이다. (먹이 쪽으로 가는 것이 방향의 다양성보다는 중요하므로 일반적으로 $\alpha > \beta$ 이다)

$$\vec{VM} = \alpha * \vec{VP} + \beta * \vec{VA} \quad (식4)$$

여기서 \vec{VM} 은 벡터이므로 벡터의 방향은 에이전트의 이동 방향을, 벡터의 크기는 현재 좌표의 평가 값이 된다. 따라서 해당 벡터 방향과 가장 가까운 셀(좌표)로 이동하는 것을 시도할 수 있다. 이동 후 보들과 현재 위치 각각에 대해서 \vec{VM} 을 구해서 제일 큰 \vec{VM} 의 자리로 이동하면 된다. 여기서 고려할 점은 먹이나 에이전트나 꼭 움직임이 없어도 된다는 것이다. 제자리에 있는 것도 한 가지 선택사항이다. 같 방향이 현재 위치보다 나쁘면 움직일 필요가 없는 것이다. 이 문제는 \vec{VM} 값을 이용하면 해결된다. 충돌이 일어나면 그 다음 \vec{VM} 값을 갖는 위치로 움직이면 된다. 먹이는 각 에이전트에서 먹이 쪽 방향으로 거리 제곱에 반비례하는 벡터를 만들어 합한 방향으로 도망가면 된다. 다음은 먹이의 탈출 벡터이다.

$$\vec{EP} = \vec{A_1P} + \vec{A_2P} + \vec{A_3P} + \vec{A_4P}$$

$$= \sum_i \vec{A_iP} \quad (식5)$$

$$\rightarrow |\vec{A_iP}| = \frac{1}{(Px - A_{ix})^2 + (Py - A_{iy})^2}$$

다음은 방향벡터를 이용한 먹이포획문제 알고리즘이다.

1. 먹이와 에이전트는 제안한 순환구조(circular)형 격자 공간에 무작위로 놓여진다.
2. 에이전트와 먹이는 서로의 현재 상태를 감지하고 방향벡터를 계산한다.
3. 먹이는 방향벡터 \vec{EP} 에 따라 이동한다.
4. 에이전트는 다음 상태의 위치와 에이전트들, 먹이의 방향벡터를 고려하여 다음상태를 결정하여 이동시킨다.

$$\vec{VM} = \alpha \cdot \vec{VP} + \beta \cdot \vec{VA}$$
5. 환경은 새로운 상태에 놓여지며 먹이를 포획함으로써 상태가 끝나거나 먹이 추적을 계속한다.
6. 2-5 까지의 과정이 계속 반복된다.

4. 실험 및 분석

먹이 추적 문제에서 성능 평가 기준은 먹이 포획의 성공률(success rate), 각 에이전트의 상태 전이 수(number of transitions)를 기준으로 한다. 하지만 새롭게 제시한 순환구조(circular)형 격자 공간을 통해 다중 에이전트의 목표에 최대한 근접하기 위한 실험으로 먹이의 100% 포획과 에이전트의 충돌 회피를 동시에 하는 전략을 평가의 기준으로 하고자 한다. 본 실험에서는 에이전트와 먹이사이의 공간벡터를 이용한 포획전략과 에이전트간의 충돌방지를 고려하였다. 또한 현실세계에서의 다중에이전트 문제 해결에 대해 근접하게 접근해보고자 순환구조(circular)형 격자 공간이라는 새로운 실험환경을 제시하였고 먹이에 시야를 주어 에이전트를 피해 회피할 수 있는 능력을 부여 하였다. 본 실험 환경은 현실성을 고려하여 불완전 포획이라는 요소를 없애고 초기 먹이와 에이전트의 위치는 무작위로 주어지며 먹이의 완전포획을 상황의 종료로 하였다. 다음의 실험결과와 같이 먹이의 포획과 에이전트간의 충돌 방지, 먹이와 에이전트의 α, β 에 따른 방향벡터를 고려한 알고리즘의 효율성을 입증 할 수 있었다. 또한 α 의 값이 5이상일때 좋은 상태전이와 100% 포획률을 보였다.(무작위 배열, 최대 상태전이 100회 기준)

먹이 가중치(α)	에이전트 가중치(β)	초기 위치	교정		무작위	
			포획 확률	상태 전이	포획 확률	상태 전이
1	1	2%	34회	3%	28회	
1	2	0%	0회	0%	0회	
1	3	0%	0회	0%	0회	
2	1	45%	42회	24%	52회	
3	1	74%	37회	86%	41회	
4	1	93%	31회	97%	35회	
5	1	100%	26회	100%	30회	

5. 결론 및 향후 연구방향

다중 에이전트에 관한 연구를 위해 다양한 실험속에서 먹이 추적 문제는 많이 사용되어지고 있다. 그만큼 현실 세계와 가깝게 표현되었기 때문이다. 하지만 제한된 격자 공간 내에서 실험은 현실 세계와는 많이 동떨어져 있다. 현실 세계에서는 무한공

간의 환경이라 할 수 있다. 다양한 현실 세계를 대신 할 수 있는 실험 환경을 통한 연구는 반드시 필요하다.

본 논문에서 제안한 순환구조(circular)형 격자 공간은 기존의 제한된 격자 공간이 아닌 에이전트와 먹이의 순환구조(circular)를 통한 무한공간 개념의 현실세계와 유사하게 표현하고자 하였다. 또한 새로운 실험 환경에 맞는 방향벡터를 이용한 휴리스틱을 통해 에이전트간의 충돌 방지와 효율적인 먹이 포획을 입증할 수 있었으며, 현실에서 응용할 수 있는 다중에이전트의 목표 획득에 대한 또 하나의 전략을 구현할 수 있었다.

본 논문에서 제안한 순환구조(circular)형 격자 공간에서 방향벡터를 이용한 포획 전략은 무작위 배열 상태에서도 효율성을 검증할 수 있었지만 특정상황(초기 배열이 한쪽으로 몰리는 현상)에서는 순환구조(circular)형 격자 공간에서는 먹이의 포획률이 낮아 이것을 해결할 수 있는 추가적인 연구가 필요하다. 또한 새로운 환경인 순환구조(circular)형 격자 공간에서 적용될 수 있는 독립된 에이전트간의 먹이 포획 전략(비대화형 에이전트) 연구가 절실히 필요하다.

참고문헌

- [1] Peter Stone, Manuela Veloso, Multiagent coordination with learning classifier systems, In Proceeding of the AAAI 99 Workshop on Negotiation, pp. 44-49, 1999.
- [2] Peter Stone and Manuela Veloso, "Multiagent System : A Survey from a Machine Learning," Technical Report CMU-CS-97-193, The University of Carnegie Mellon, December-1997
- [3] Thomas Haynes and Sandip Sen. "Evolving behavioral strategies in predators and prey" , In Gerhard Weiband Sandip Sen, editors, Adaptation and Learning in Multiagent Systems, Springer Verlag, Berlin, 1996
- [4] Ran Levy and Jeffrey S.Rosenschein, "A game theoretic approach to the pursuit problem", In Working Papers of the 11th International Workshop on Distributed Artificial Intelligence, February 1992.
- [5] S. Cammarata, D. McArthur, And &, Steeb, "Strategies of Cooperation in Distributed Problem Solving", Proceedings of Eighth International Joint Conference on Artificial Intelligence, Karlsruhe West Germany, August 1993.
- [6] L.M.Stephens and 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.
- [7] 이형일, 김병천 "강화학습을 이용한 다중에이전트 전략" 정보처리학회논문제 B 제 10-B권제 3호. 2003.6