

논문 2011-48CI-2-9

# 멀티에이전트 전략을 위한 방향벡터 함수 활용과 동적 환경에 적응하는 경로 추천시스템에 관한 연구

( The Application of Direction Vector Function for Multi Agents  
Strategy and The Route Recommendation System Research in A  
Dynamic Environment )

김 현\*, 정 태 충\*

( Hyun Kim and TaeChoong Chung )

## 요 약

본 논문에서는 운전자의 특성, 도로상황, 경로 추천을 담당하는 에이전트와 같은 동적환경정보(DEI:Dynamic Environment Information)를 반영하여 실시간으로 운전자에게 경로를 추천할 수 있는 시스템을 위해 멀티에이전트에 관한 연구를 수행하였다. DEI는  $n$ 개의 멀티 에이전트이며 운전자에게 최적화된 경로를 제공할 수 있는 경로추천시스템에 활용되는 환경변수이다. DEI가 반영되는 경로추천 시스템은 멀티 에이전트 연구의 새로운 연구 분야라 할 수 있겠다. 이를 위하여 멀티에이전트 연구의 대표적 실험 환경인 먹이추적문제<sup>[4]</sup>를 이용하여 새로운 해법을 찾고자 하였다. 본 논문에서는 기존의 먹이추적 실험은 현실성이 결여된 멀티에이전트 연구였기에 기존의 실험환경<sup>[5]</sup>과 달리 현실세계와 비슷한 실험환경을 제안 을 하며 새로운 전략인 Ant-Q 학습을 적용한 알고리즘과 기존의 방향벡터를 활용한 전략과의 비교를 통해 새로운 환경에서의 성능의 향상을 입증할 수 있었다.

## Abstract

In this paper, a research on multi-agent is carried out in order to develop a system that can provide drivers with real-time route recommendation by reflecting Dynamic Environment Information which acts as an agent in charge of Driver's trait, road condition and Route recommendation system. DEI is equivalent to number of  $n$  multi-agent and is an environment variable which is used in route recommendation system with optimal routes for drivers. Route recommendation system which reflects DEI can be considered as a new field of topic in multi-agent research. The representative research of Multi-agent, the Prey Pursuit Problem, was used to generate a fresh solution. In this thesis paper, you will be able to find the effort of indulging the lack of Prey Pursuit Problem, which ignored practicality. Compared to the experiment<sup>[5]</sup>, it was provided a real practical experiment applying the algorithm, the new Ant-Q method, plus a comparison between the strategies of the established direction vector was put into effect. Together with these methods, the increase of the efficiency was able to be proved.

**Keywords :** Multi-agent, Ant-Q learning, Dynamic Environment Information, Personalisation, Prey Pursuit Problem

## I. 서 론

\* 정회원, 경희대학교 컴퓨터공학과  
(Kyunghee University)

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (과제번호: 2010-0012609)

접수일자: 2010년8월16일, 수정완료일: 2011년3월4일

개인화 서비스(Personalization)는 사용자에게 최적화된 맞춤형 정보를 제공하는 것으로 유비쿼터스 분야에서 중요하게 다루어지고 있다. 이것의 핵심은 개인의

특성을 저장하고 있는 프로파일데이터(Profile data)로, 여러 방법들을 사용해 추출하고 있다. 사용자의 활동데이터(이동경로, 움직임, 조작)를 실시간으로 측정하고 이를 누적하는 방법, 성향이 비슷한 사람들로부터 이미 만들어진 데이터를 바탕으로 하는 통계적 추천하는 방법들이 대표적이다.<sup>[1]</sup>

본 논문에서는 개인의 특성을 고려한 네비게이션(Navigation) 서비스를 제안하며, 이것은 개인의 운전습성을 실시간으로 측정하고 이를 바탕으로 네비게이션 서비스를 할 수 있는 알고리즘이다. 차량운전을 하는 사람들은 각각의 운전습관들을 가지고 있다. 예를 들어 1차선만을 고집하는 사람과 우회전만을 고집하는 사람, 집중력이 떨어져 추천경로를 놓치는 사람 등이 그것이다. 운전자들은 대부분 네비게이션 서비스를 이용해 운전을 하는데 이것은 제조회사에서 제공하는 데이터를 바탕으로 설정된 에이전트가 입력된 경로를 분석하여 평균적인 최단거리를 출력하는 방식으로 저마다 운전습관이 다른 운전자들에게는 좋은 해법이 될 수 없다.<sup>[2]</sup>

에이전트가 제시하는 경로가 개인에게 최적화될 수 없는 것은 운전에 영향을 미치는 환경변수(개인의 운전습관, 도로사정)들을 전혀 고려하지 않았다는 것이다. 현실에서는 운전환경의 잦은 변화, 천차만별인 개인의 운전 습관 등이 최적화된 경로를 결정하는 중요한 환경변수가 된다. 대부분의 추천방식은 개인 사용자를 고려하기 보다는 데이터상의 최적화된 정보를 일방적으로 전달하는 역할밖에 없다는 것이다.<sup>[2, 16]</sup>

이런 이유로 본 논문에서는 다양한 환경변수들을 고려한 최적화된 경로를 추천할 수 있는 새로운 네비게이션 서비스 모델과 알고리즘을 제안하고자 한다.

통신 및 장비의 발달로 인해 양방향 데이터 송수신 서비스가 가능하고 다양한 데이터 수집 센서들이 개발됨에 따라 이를 이용하여 동적 정보를 측정하고 수집하여 본 논문에서 제시하는 알고리즘에 환경변수로 활용할 것이다. 또한 네비게이션에 설정된 기본 데이터, 양방향통신을 통해 수신되는 동적 환경 정보인 도로상황, 운전자의 특성, 추천을 담당하는 에이전트 등을 동적환경정보(DEI:Dynamic Environment Information)라 하고 이것은 각각  $n$ 개의 에이전트 ( $A_i$ ), DEI를 통해 최종 생성되어 개인에게 제공될 정보(최적화된 경로)를  $P$ 로 정의한다. DEI와  $P$ 의 관계는 수식(1)과 같이 표현할 수 있다.

$$P \leq \sum_{i=1}^n A_i \quad (1)$$

수식 (1)은 DEI인 에이전트  $A_i$ 들이 서로 유기적 협동관계를 통해  $P$ 를 획득하는 문제로 본 논문에서 M.Benda에 의해 처음 제안된 먹이추적문제(Prey pursuit problem)에 적용할 수 있었다<sup>[4]</sup>, 먹이추적문제는 멀티에이전트 연구를 위한 모델로 본 논문에서는 새로운 환경모델<sup>[5]</sup>을 기반으로 연구를 진행 하였다.

기존 먹이추적 문제의 실험 환경은 격자 공간(Grid) 상에 4개의 에이전트와 1개의 먹이가 존재하였고, 에이전트들은 협동을 통해 먹이를 포획함으로써 에피소드(Episode)를 끝내게 된다<sup>[4]</sup>. 먹이는 에이전트가 도달하고자 하는 목표이며, 각 에이전트는 유기적 관계속에 최소의 비용을 들여 효율적 방법으로 목표를 획득하는 것이 멀티에이전트 연구의 목적이라 할 수 있겠다.

그러나 기존 실험 환경은  $30 \times 30$  크기의 격자 구조로 이루어져 현실성이 결여되어 있으며, 에이전트의 먹이 포획만을 위주로 연구가 진행 되었다.

본 논문에서는 현실세계에 적용 가능한 멀티에이전트 연구를 위해 현실성을 고려한 순환구조(circular)형 격자 공간이라는 환경에서 제안한 알고리즘과 Ant-Q 학습을 이용하여 실험하였고 그 결과 에이전트들의 유기적 협력을 통해 DEI를 이용하여 개인에게 최적화된 경로를 제시할 수 있는 방법을 제안하였다.

본 논문의 구성은 다음과 같다. 본론에서는 논문과 연관된 관련 연구들에 대해 설명하고 실험을 위한 알고리즘을 설명한다. 결론에서는 알고리즘을 적용하여 기존의 멀티에이전트 전략들과 실험환경의 크기를 증가시키며 목표도달에 대한 평가 및 효율성을 그리고 향후 연구와 관련된 제안을 다루고 있다.

## II. 본 론

### 2. 관련 연구

#### 2.1 먹이 추적 문제

먹이 추적문제는 복잡한 현실세계에 적용할 수 있는 단일 혹은 멀티 에이전트의 목표획득을 실험하는 모델로 사용되어지고 있다<sup>[1~2]</sup>. 다양하고 복잡한 환경에서 사용자의 요구와 문제 해결을 위해 에이전트가 개발되어졌으며, 그 결과 단일 에이전트로 해결하지 못하는

$A_3$								
							$A_4$	
			$P$					
				$A_1$				
	$A_2$							

그림 1. 먹이 추적 문제

Fig. 1. Prey purist problem.

복잡한 문제에 대한 해결 방안을 위해 멀티 에이전트간의 협동을 통한 멀티 에이전트 시스템(multi-agent system)이 제안되었다<sup>[6~7]</sup>. 효율성 있는 멀티 에이전트의 대표적인 실험모델인 먹이추적문제는 현실세계를 표방하는 격자 공간(grid)내에 4개의 독립적 에이전트가 하나의 먹이를 포획하는 실험으로서 복잡한 현실 세계를 표현하기 위해 M.Benda에 의해 제안되었다<sup>[3]</sup>.

먹이 추적 문제는 (그림 1)과 같이  $N \times N$  크기의 환경에서 4개의 에이전트( $A_i$ )가 1개의 먹이( $P$ )를 4방향에서 포획하는 실험이다. 실험에서 에이전트와 먹이의 이동 조건(속도, 방향)은 동일하며 단, 먹이가 우선 이동을 하게 된다. 실험을 통해 멀티 에이전트들이 목표인 먹이를 포획하는 것에 대한 다양한 연구들이 진행되었지만 에이전트간의 충돌, 불안정포획(구석으로 물렸을 때 3방향, 2방향에서만 포획하는 문제, 불규칙한 포획시간(무작위 위치 배정으로 발생하는 문제들)등의 문제점으로 인해 이를 해결하고자 다양한 전략과 해법들이 소개되었다.<sup>[8~11]</sup>

## 2.2 Ant-Q(Ant colony Q-learning system)

Ant-Q는 본래 생물학에서 연구되어 졌던 개미집단의 습성 강화학습(Reinforcement Learning)을 적용한 알고리즘이다. 개미들은 생활의 기본이 되는 음식물을 찾아 탐험하고 발견된 음식물을 자신의 등지로 이동시키는데 페르몬 분비를 통해 길을 찾게 된다. 이때 페르몬은 시간이 지날수록 산도(Acidity)가 약해지는 단점이 있다. 때문에 많은 개미들, 혹은 초기에 이동한 개미가 먹이를 찾은 후 등지로 돌아올 때 새롭게 페르몬을 분비하며 갱신함으로, 산도가 높아지게 되고, 산도가 약한 다른 경로를 무시하고 높은 산도의 경로가 최적 이동경로로 정해지게 된다.<sup>[12~13]</sup> 이때 개미의 이동규칙은

식(2)와 같이 표현하며  $s$ 는 확률분포(Probability distribution)를 계산하는 식(3)에 의해 선택된 무작위 파라미터이다. 이것은 국부최적에 빠지기 쉬운 한계를 벗어나기 위해 적용된다.

$$s = \begin{cases} \arg \max_{u \in J_k(r)} [AQ(r, u)] \cdot [HE(r, u)]^\beta, & \text{if } q \leq q_0 \text{ (exploitation)} \\ S, & \text{otherwise (biased exploitation)} \end{cases} \quad (2)$$

$$P_k(r, s) = \begin{cases} \frac{[AQ(r, s)] \cdot [HE(r, s)]^\beta}{\sum_{u \in J_k(r)} [AQ(r, u)] \cdot [HE(r, u)]^\beta}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

식(2)(3)을 가지고 식(4)에 적용해 노드선택에 가장 중요한 환경변수인 페로몬을 갱신하게 된다.

$$AQ(r, s) \leftarrow (1 - \alpha) \cdot AQ(r, s) + \alpha \cdot (\Delta AQ(r, s) + \gamma \cdot \max_{z \in J_k(s)} AQ(s, z)) \quad (4)$$

$\alpha$  ( $0 < \alpha < 1$ )는 페로몬 자연 파라미터로 학습률(learning rate)이며,  $\gamma$ 는 할인율(discount rate)이다.  $\max_{z \in J_k(s)} AQ(s, z)$ 는 다음 상태에 대한 평가로 외부 환경으로부터 받는 강화 값을 최대화하는 것으로 전역 강화일 때는 0이다. 또한,  $\Delta AQ(r, s)$ 는 강화 값으로 지역 강화일 때는 항상 0이다.

## 2.3 현실을 고려한 순환구조(Circular)형 격자공간

기존 먹이추적 문제의 실험환경은  $N \times N$  크기로 공간적 제한과 이로 인해 발생하는 불완전 포획이라는 문제점이 발생했다. 또한 기존 실험환경은 현실성이 떨어진다.<sup>[5]</sup> 이런 이유로 본 논문에서는 현실세계를 반영하여 제한된 공간이 아닌 무한공간과 유사한 순환구조형 격자 공간을 사용하였다. 본 논문에서는 제한된 공간의 실험환경은 현실성이 떨어지며 상태 행동에 대한 평가 시간이 길어진다는 것에 착안하여 보다 현실적이고 일 반화된 환경을 제시하며 사각형의 그리드 구조를 순환 구조형으로 만들어진 무한 공간 상태와 비슷한 환경으로 연속된 격자 공간이다.<sup>[5]</sup>

기존 실험 모델에서 먹이는 에이전트들 가운데 배치되 쉽게 포위될 수 있는 위치에서 시작하고 있다는 불리함과 먹이는 어떠한 전략 없이 단순히 불규칙적으로 움직여 현실성이 떨어졌다.

순환구조형 격자 공간의 구조는 그림 2와 같다. 예를

(0,0)	(0,1)	(0,2)	$A_1$ ...	0,10	(0,0)
(1,0)					
(2,0)					
...					
(10,0)			$A_1$	10,10	
(0,0)				0,10	

그림 2. 순환구조(Circular)형 격자 공간  
Fig. 2. Circular gride space.

들어, (0,0)-(0,10)의 격자 공간에서 (0,10)다음은 오른쪽이 (0,11)이 되지만 이를 다시 (0,0)으로 연결시켜 순환이 되게 하는 방식이다. 예를 들어 (3,3)의 상하좌우는 (2,3), (4,3), (3,2), (3,4)이지만 (0,0)의 상하좌우는 (9,0), (1,0), (0,9), (0,1)이 되는 것이다.

이런 경우 에이전트들과 먹이의 속도가 같을 때에도 새로운 에이전트 전략으로 먹이포획이 가능할 수 있게 된다. 본 실험환경은 무한거리 모델인 메비우스를 모델로 하여 순환구조(circular)형 격자 공간이라 명하였다. 먹이와 에이전트의 위치가 무작위로(Random) 생성되어도 오직 완전포획만이 존재하게 되어 멀티에이전트는 완전한 목적을 달성하게 된다.

### 3.1 Ant-Q학습을 통한 DEI의 최적화

기존 먹이추적문제에서 먹이의 역할은 에이전트의 목표대상이었다. 에이전트에게 추적과 포획, 협동의 능력 주어졌지만 먹이는 무작위로 이동만을 반복하였다.

그러나 본 논문에서는 현실성을 고려해 먹이에게도 에이전트를 파악할 수 있는 능력과 학습을 통한 진화를 할 수 있도록 하였다. 에이전트의 (행동-상태)를 파악하고 거리를 계산하여 회피할 수 있는 능력을 준 것이다. 이것은 에이전트의 학습능력을 강화시키는 강화 값(Q-value)으로 식(5)을 통해 갱신한다..

$$\Delta AQ(r,s) = \begin{cases} \frac{W}{L_{A_i}}, & \text{if } (r,s) \in \text{tour done by the agent } A_i \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

여기서  $W$ 는 상수 값으로 여러 실험을 통해  $W=10$ 으로 고정한다<sup>[14]</sup>.

### 3.2 Ant-Q학습을 적용한 방향벡터알고리즘

새로운 실험 환경에 기존 연구들에서 제안된 전략들을 적용했을 때 다양한 문제가 발생하였다. 가장 큰 문제는 그림 3과 같이 에이전트들이 한곳에 모여 있을 때의 상황이다.

무한 공간 상태에서 그림 3과 같이 에이전트가 먹이와 대치상태로 생성될 시 기존의 전략들은 먹이가 계속 한쪽 방향으로 도망하여 포획이 불가능하였다. 그리고 실험 환경 크기가 커질수록 먹이포획의 횟수는 크게 감소하였다. 이런 이유에서 본 논문에서는 먹이와 에이전트, 에이전트와 에이전트의 상태 위치를 파악하고 이를 고려한 방향벡터 전략을 제안 하였다.<sup>[5]</sup> 하지만 기존의 방향벡터 전략은 상태공간의 크기가 증가함에 포획에 대한 시간 또한 증가함으로 그 효율성이 감소함을 보였다. 그래서 본 논문에서는 새롭게 에이전트와 먹이의 이동 위치를 더 나은 곳으로 결정짓는 값을 추가하였다. 또한 그림 3과 같이 기존의 4방향과 달리 현실성을 고려하여 대각선 방향으로 이동할 수 있도록 실험환경에 변화를 주었다.

각 에이전트들은 먹이를 기점으로 서로의 상태 정보 값을 확인하고 충돌을 고려하여 간격을 두고 먹이를 중심으로 이동한다. 먹이는 에이전트를 피하기 위해 에이전트들과의 공간관계를 고려하여 에이전트들이 없는 상태 공간 쪽으로 이동을 한다. 에이전트들은 먹이 포획을 위한 위치를 결정하는 판단하는 함수를 만든다. 에이전트는 기존의 연구들의 조건과 동일하게  $A_i(A_i \geq 4)$ 로 설정하였다. (최소 에이전트 값을 4로 해야만 완전하게 먹이를 포획할 수 있다.)

에이전트는 인접한 에이전트와 먹이와의 거리와 다른 에이전트들의 상관관계를 반영하는 벡터를 생성한다. 수식(6)은  $A_i$ 가  $P$ 쪽으로 이동을 위한 방향벡터로

		$A_4$				
$A_3$	↖	↑	↗			
	←	$A_1$	→	$P$		
	↙	↓	↘			
	$A_2$					

그림 3. 기존 전략들의 문제점

Fig. 3. The problem of old strategies.

먹이와 에이전트와의 거리를 반영한 것이다. 각 에이전트와 먹이간의 방향 벡터  $\overrightarrow{VP}$ 의 크기는 먹이와 각 에이전트간의 거리에 반비례한다. 이것은 에이전트가 먹이와 가까울수록 먹이 방향으로 가려고 하는 에이전트의 벡터 값이 커져 에이전트의 이동 함수를 만들게 된다.

$$\begin{cases} \overrightarrow{VP} = \sum_{i=1}^n \overrightarrow{A_iP} \\ |\overrightarrow{A_iP}| = ((Px - A_ix)^2 + (Py - A_iy)^2)^{-1} \cdot AQ(r, s) \end{cases} \quad (6)$$

기존의 연구들에서 가장 많이 제기되었던 문제는 에이전트간의 충돌 문제였다. 이것을 해결하고자 본 논문에서는 에이전트와 다른 에이전트들 간의 상관관계를 고려한 벡터 함수를 만들었다. 다른 에이전트들과 거리가 가까워지면 충돌을 피하기 위해 반대방향으로 값이 커지는 벡터  $\overrightarrow{VA}$ 를 생성하게 된다. 이것은 에이전트간의 일정한 거리를 유지하며 먹이에게로 접근할 수 있는 함수 값이며, 이 값은 에이전트사이의 모든 방향 벡터의 합으로 만들어 진다.

$$\begin{cases} \overrightarrow{VP} = \sum_{i \in J(A)} \overrightarrow{R_i} \\ |\overrightarrow{R_i}| = ((R_ix - A_ix)^2 + (R_iy - A_iy)^2)^{-1} \cdot AQ(r, s) \end{cases} \quad (7)$$

$J(A)$ 는 에이전트( $A$ )의 이웃 에이전트의 집합,  $R$ 은 에이전트( $A$ )와 이웃 에이전트,  $\overrightarrow{R}$ 은 에이전트( $A$ )와 에이전트( $R$ )의 방향 벡터이다. 이것은 에이전트들 간의 거리가 가까울수록 반대방향으로 커진다. 그리고 에이전트와 다른 에이전트와의 벡터 값은 두 에이전트 거리의 제곱에 반비례하는 값을 갖게 하면 되는데, 이것은 에이전트와 에이전트 사이를 멀어지게 하는 작용을 한다. 수식(7)을 통해 에이전트는 다른 에이전트들과 다른 방향으로 가려고 하는 함수가 만들어져 에이전트간의 충돌문제를 해결하며 다른 에이전트들과는 먹이를 포획하는데 있어 공동작전을 효과적으로 펼 수 있게 된다. 에이전트의 이동방향을 최종 결정짓는 함수는 수식(8)로 생성된다. 에이전트는  $\overrightarrow{VP}$ 와  $\overrightarrow{VA}$ 를 합친 쪽으로 이동하면 된다. 이렇게 만들어진 에이전트의 이동 함수를  $\overrightarrow{VM}$ 이라 한다.

$$\overrightarrow{VM} = \alpha(\overrightarrow{VP}) + \beta(\overrightarrow{VA}) \quad (8)$$

$\alpha, \beta$  값은 식5를 기반으로 생성된  $Q$  값으로 확률적으로 더 나은 위치를 선택할 수 있도록 해주는 값이고 먹이를 추적하는 방향을 결정하는 중요한 환경변수이다. (먹이 쪽으로 이동하는 것이 중요하므로 일반적으로  $\alpha > \beta$ 로 설정한다.  $\alpha$  획득 알고리즘은 다음과 같다.

$$\begin{aligned} \alpha &= \max(AQ(r_t, s)), s \in A(s_t) \\ \overrightarrow{VM} &= \alpha \cdot \overrightarrow{VP} + \beta \cdot \overrightarrow{VA} \end{aligned} \quad (9)$$

수식(9)를 통해 생성된  $\overrightarrow{VM}$ 을 가진 에이전트는 먹이와 가장 가까운 셀(좌표)로 이동하는 것을 시도할 수 있다. 이동 후보들과 현재 위치 각각에 대해서  $\overrightarrow{VM}$ 을 구해서 제일 큰  $\overrightarrow{VM}$ 의 위치로 이동하게 된다. 여기서 고려해야 할 사항은 먹이나 에이전트는 반드시 움직임이 없어도 된다. 제자리에 머무르는 것은 불필요한  $DEI$  생성을 차단하는 전략으로  $\overrightarrow{VM}$ 으로 생성 할 수 있다. 즉 생성된  $\overrightarrow{VM}$ 을 통해 에이전트가 이동할 방향이 현재 위치보다 나쁘다면 움직임 필요가 없는 전략인 것이다.

에이전트와 달리 먹이는 각 에이전트에서 먹이 쪽 방향으로 거리의 제곱에 반비례하는 벡터 함수를 생성하고 이것들을 합한 방향으로 이동을 하면 에이전트들을 따돌릴 수 있게 된다. 수식(10)은 먹이의 도망을 위한 벡터 함수이다. 단, 먹이의 도망을 결정하는  $\overrightarrow{EP}$ 는 에

```

α value from Ant-Q
{
  Initialize AQ(r, s) arbitrarily for all r, s;
  for {
    Move_P( ); // P는 도망친다.
    For all A_i {
      Observe current state r_t; // 에이전트가 선택할 수
      있는 모든 다음 상태에 대해
      For all actions {
        Selection an action r_t at state s_t;
        Observe s_{t+1}, γ_{t+1}; // 다음 상태와 강화
        값을 구한다.
      }
      Choose r_{t+1}; // 다음 상태를 선택
      Update_Q value (s_t, r_t) // 현재 상태의 Q값을 갱신
      Move_Agent();
    }
  } Until (Prey can't move)
}

```

그림 4. Ant-Q 학습 벡터 이동 알고리즘

Fig. 4. Ant-Q learning vector move algorithm.

이전트의 학습능력을 향상시키는 중요한 요소이기도 하다. 에이전트들이 먹이를 포획했을 때 학습 값을 생성하기 위한 일정의 포상 값을 만들게 되는데 이것이 강화 값<sup>[14]</sup>으로, 에이전트의 이동을 결정짓는  $\alpha$ 를 구성하는 중요한 변수가 되는 것이다. 본 논문에서는  $\overline{EP}$ 를 증가시키거나 감소시킴에 따라 에이전트의 이동에 변화가 있음을 알 수 있었다.

$$\begin{cases} \overline{EP} = \overline{A_1P} + \overline{A_2P} + \overline{A_3P} + \overline{A_4P} \\ = \sum_i \overline{A_iP} \\ |\overline{A_iP}| = ((Px - A_ix)^2 + (Py - A_iy)^2)^{-1} \end{cases} \quad (10)$$

### III. 실험

제안한 알고리즘의 성능 평가를 위한 실험은 에이전트의 상태 전이 수(number of transitions), 먹이 포획의 성공률(success rate), 에이전트와의 충돌횟수를 측정 하였다.

그리고 실험 공간의 크기를 변화시키고 기존의 연구들에서 제안된 대표적 알고리즘을 적용하여 비교 평가하였다.<sup>[5, 15]</sup>

상태공간이 커질수록 에이전트와 먹이의 이동속도가 같고 먹이가 에이전트보다 한 단계 먼저 이동하기 때문에 에이전트들의 먹이 포획이 상당히 더딤을 알 수 있었고 이것은 본 논문에 적용한 Ant-Q학습의 실험결과들과 비슷한 패턴을 보였다.<sup>[12~14]</sup>

본 실험 환경은 현실성을 고려하여 불완전 포획은 포획 실패로 정했으며 먹이와 에이전트의 위치는 무작위(Random)로 주어졌다.

표 1을 통해 상태공간이 커질수록 에이전트의 이동 횟수는 증가하지만 Q학습을 적용한 전략이 효율적임을

표 1.  $\alpha$ 값에 따른 먹이 포획 결과

Table 1. The result is capture prey to  $\alpha$  value.

( $\alpha$ )	( $\beta$ )	30×30		50×50		100×100	
		Capture probability	State transition	Capture probability	State transition	Capture probability	State transition
0.4	0.1	100%	494	75%	1296	50%	5320
0.7	0.1	100%	421	100%	1102	75%	5412
1.4	0.1	100%	373	100%	988	100%	4930

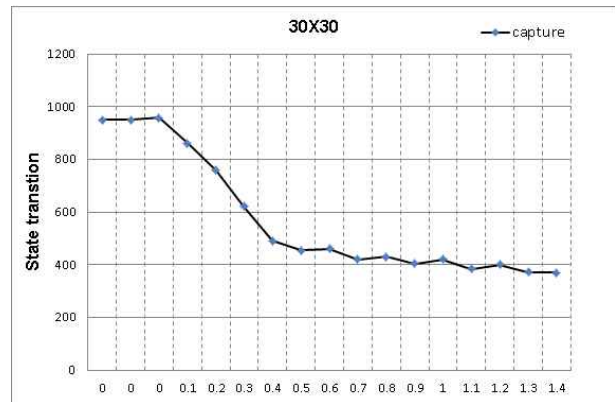


그림 5.  $\alpha$ 값 증가에 따른 상태전이의 변화

Fig. 5.  $\alpha$  values due to changes in state transitions.

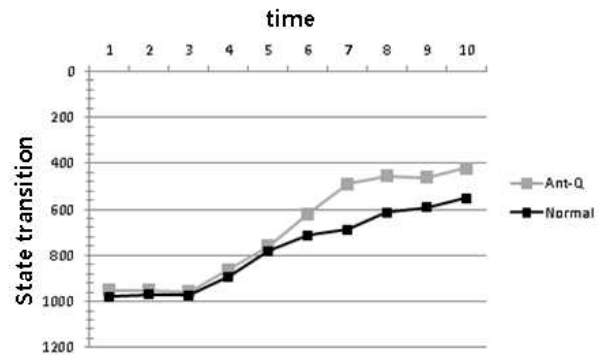


그림 6. Ant-Q 적용 알고리즘과 기존 알고리즘의 비교

Fig. 6. The compare Normal algorithm and Ant-Q algorithm.

알 수 있었다.

그리고 먹이의 완전포획을 기준으로 생성되어지는 Q값이 적용된  $\alpha$ 값을 점차 증가시킴으로 표 1, 그림 5와 같이 상태전이 횟수(state transition)가 점차 감소함을 보였다.

그림 6은 30×30 환경에서 기존의 연구에서 제안된 전략<sup>[5]</sup>과의 비교 결과이다. 기존의 논문에서 제안된 실험환경과 전략을 변형하였다. 초기 실험에서는 Ant-Q 학습 전략을 통해 강화학습을 이용한 전략<sup>[15]</sup>과 비슷한 결과를 보여주고 있지만 기존 강화학습전략은 새로운 실험환경에서 먹이 포획의 실패와 에이전트간의 충돌문제 등으로 인해 실험데이터가 충분히 만들어지지 않아 새로운 환경에 맞는 전략의 필요성을 보여주고 있었다. 그림 6의 기존 알고리즘과의 성능 분석에서는 새로 제안한 알고리즘의 실험 횟수(시간)가 증가할수록 기존의 전략<sup>[5]</sup>보다 먹이 포획을 위한 상태전이횟수가 적어짐을 알 수 있었다. 이것은 먹이포획의 횟수가 증가할 수록

학습능력이 향상되어 에이전트의 상태전이 횟수가 줄어드는 것이다.

#### IV. 결 론

본 논문에서는 기존의 네비게이션 시스템에 동적환경변수를 고려하여 사용자에게 최적화된 길안내 서비스를 제공할 수 있는 새로운 추천 시스템 연구를 위해 본 논문에서는 멀티 에이전트 연구의 대표적인 실험인 먹이추적문제 해결을 위한 연구를 진행하였다. 그 결과 우리는 Ant-Q 학습을 적용한 새로운 알고리즘을 제안하게 되었다. 기존의 먹이추적 연구의 실험환경은 많은 문제와 현실성이 결여되어 있었다. 먹이추적문제를 새로운 실험환경에 적용하고 4방향이 아닌 8방향으로의 이동을 정의한 이유는 현실 세계를 비슷하게 표현하고자 했기 때문이다. (현실 세계는 무한공간의 환경이며 모든 방향으로 이동 할 수 있다.) 그래서 현실성을 고려한 새로운 실험환경을 제안하며 Ant-Q를 이용한 새로운 포획 전략을 통해 새로운 해법을 제시할 수 있었다.

본 논문에서 제안한 전략과 환경을 통해 기존의 제한적이고 현실성이 미흡했던 멀티에이전트연구에 새로운 방법을 제시해 볼 수 있었다. 또한 제안한 알고리즘과 환경을 이용한다면 동적환경에서 사용자에게 최적화된 정보를 제공할 수 있는 시스템 개발에 충분히 응용할 수 있으리라 생각된다.

향후 연구에서는 기존의 먹이추적의 4개의 에이전트보다 더욱 다양한 동적환경변수를 감안하여 실시간으로 사용자에게 맞춤형 길안내 서비스를 제공할 수 있는 새로운 시스템을 구현할 수 있는 멀티에이전트의 해결 방법들이 지속적으로 연구되어야 할 것이다.

#### 참 고 문 헌

- [1] 김은희, 김문철 “협업 필터링을 통한 IPTV 프로그램 자동 추천” 대한전자공학회 2008년도 하계종합 학술대회, 제31권 제1호, 701-702쪽, 2008년
- [2] 이봉우, 최우경, 전홍태 “퍼지 로직을 이용한 지능형 네비게이션 시스템” 전자공학회논문지 제43권 제4호, 67-72쪽, 2006년 7월.
- [3] 한상우, 김종원 “다자간 협업 환경에서 에이전트 기반 서비스 합성” 전자공학회논문지, 제45권 CI편, 제5호, 2008년.
- [4] M. Benda, V. Jagannathan and R. Dodhiawala, “On optimal cooperation of knowledge source-an empirical investigation,” Technical Report BCS-G2010-28, Boeing Advanced Technology Center, Boeing Computing Services, Seattle, Washington, July, 1986.
- [5] 김현, 이승관, 정태충 “방향벡터를 이용한 다중 에이전트 휴리스틱” 제22회 한국정보처리학회 추계 학술발표대회 11권 2호, 2004.11.01.
- [6] Peter Stone, Manuela Veloso, Multiagent coordination with learning classifier systems, In Proceeding of the AAAI 99 Workshop on Negotiation, pp. 44-49, 1999.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] A. Colomi, M. Dorigo, and V. Maniezzo, An Investigation of Some Properties of An Ant Algorithm. Proceedings of the Parallel Problem Solving from Nature Conference (PPSN 92), 1992, R. Mäanner and B. Manderick (Eds.), Elsevier Publishing, pp. 509 - 520.
- [13] V. Maniezzo, A. Colomi, and M. Dorigo, The Ant System Applied To The Quadratic Assignment Problem, Tech. Rep. IRIDIA/94-28, 1994, Université Libre de Bruxelles, Belgium.
- [14] M. Dorigo, L.M. Gambardella “Ant-Q: a reinforcement learning approach to the traveling salesman problem,” Proceedings of ML-95, twelfth International Conference on Machine Learning, A. Prieditis and S. Russell (Eds.),

Morgan Kaufmann, pp.252-260, 1995.

- [15] 이형일, 김병천 “강화학습을 이용한 다중 에이전트 제어 전략”, 정보처리학회지 10-B권 3호 249p-256p 1598-2866
- [16] 박정훈, “멀티에이전트 시스템에서의 효율적인 작업수행방법”, 대한전자공학회, 전자공학회논문지, 제 21호, 1998년.

---

저 자 소 개

---



김 현(정회원)  
2005년 경희대학교 컴퓨터공학과 석사  
2011년 경희대학교 컴퓨터공학과 박사 졸업예정  
<주관심분야 : 멀티에이전트, 강화 학습, 스마트 TV>



정 태 충(정회원)  
1987년 KAIST 전산학과 공학박사  
1987년 KIST 시스템공학센터 선임 구원  
1988년~현재 경희대학교 컴퓨터 공학과 교수