http://dx.doi.org/10.7236/JIIBC.2013.13.6.187
JIIBC 2013-6-24

혼성 다중에이전트 학습 전략

Hybrid Multi-agent Learning Strategy

김병천*, 이창훈**

Byung-Chun Kim, Chang-Hoon Lee

요 약 다중 에이전트 시스템에서 학습을 통해 여러 에이전트들의 행동을 어떻게 조절할 것인가는 매우 중요한 문제이다. 가장 중요한 문제는 여러 에이전트가 서로 효율적인 협동을 통해 목표를 성취하는 것과 다른 에이전트들과 충돌을 방지하는 것이다.
본 논문에서는 혼성 학습 전략을 제안하였다. 제안된 방법은 다중에이전트를 효율적으로 제어하기 위해 에이전트들 사이의 공간적 관계를 이용하였다. 실험을 통해 제안된 방법은 에이전트들과 충돌을 피하면서 에이전트들의 목표에 빠르게 수렴함을 알 수 있었다.

Abstract In multi-agent systems, How to coordinate the behaviors of the agents through learning is a very important problem. The most important problems in the multi-agent system are to accomplish a goal through the efficient coordination of several agents and to prevent collision with other agents.
In this paper, we propose a novel approach by using hybrid learning strategy. It is used hybrid learning strategy to control the multi-agent system efficiently by using the spatial relationship among the agents. Through experiments, we can see approximate faster the goal then other strategies and avoids collision among the agents.

Key Words : Multi-agent, Pursuit Problem, Reinforcement Learning, Q-learning, Control Strategy

Ⅰ. 서 론

최근 실세계의 복잡한 문제를 해결하기 위해 다중 에이전트(multi-agent) 시스템에 대한 연구가 활발히 진해되고 있다[1]. 일반적으로 다중 에이전트에 대한 연구는 에이전트들의 행동을 적당하게 조절하여 목표에 빠르게 도달하기 위한 행동 전략을 개발하는데 있다[2].

다중에이전트 시스템은 각 에이전트와 환경이 시간에 따라 변하는 동적 환경(dynamic environment)이기 때문에 다중에이전트의 행동을 조절하기 위해 강화 학습(reinforcement learning)이 널리 이용되고 있다[3]. 강화 학습은 동적 환경과 시행-착오(trial-and-error)를 통해 상호 작용하면서 학습을 수행하기 때문에 동적 환경에서 학습을 수행하기 위해 널리 이용되고 있다[4]. 그러나 강화 학습은 다른 에이전트들에 대한 (상태, 행동) 쌍을 고려하지 않고 학습을 수행하며, 강화 값(reinforcement value)이 0 또는 1이므로 에이전트가 선택한 (상태, 행동) 쌍이 얼마나 좋은가를 나타낼 수 없고, 에이전트들 간의 충돌 현상이 많이 발생하기 때문에 다중 에이전트 환경에 적합하지 않다[5].

*정회원, 국립한경대학교 컴퓨터웹정보공학과
**정회원, 국립한경대학교 컴퓨터웹정보공학과
접수일자 : 2013년 11월 19일, 수정완료 : 2013년 12월 7일
게재확정일자 : 2013년 12월 13일

Received: 19 November, 2013 / Revised: 7 December, 2013
Accepted: 13 December, 2013
*Corresponding Author: bckim@hknu.ac.kr
Dept. of Computer&Web Information Engineering, HanKyung National University, Korea

본 논문에서는 다중 에이전트 시스템에서 각 에이전트들의 공동 목표에 빠르게 수렴하고, 에이전트들 간의 충돌현상을 피하기 위한 혼성 학습 전략을 제안하였다. 제안된 방법은 에이전트들 간의 공간적 관계를 학습 전략으로 이용하였다.

본 논문에서 제안한 방법을 Stephen과 Merx가 제안한 먹이 추적 문제(prey pursuit problem)[6]에 적용한 결과 기존의 지역제어 전략 또는 분산제어 전략을 이용한 제어전략보다 에이전트들이 서로 충돌하지 않고 공동의 목표 즉, 먹이를 매우 빠르게 포획할 수 있음을 알 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서는 강화 학습과 먹이 추적 문제를 설명하였고, 3장에서는 본 논문에서 제안한 혼성 학습 전략에 대하여 성명하였다. 4장에서는 제안된 학습 전략을 먹이 추적 문제에 적용하여 다른 학습 방법과 비교 분석하였고 5장에서는 결론 및 향후 연구 방향을 제시하였다.

II. 관련 연구

1. 강화 학습

강화 학습은 그림 1과 같이 학습을 수행하는 에이전트와 외부 환경이 서로 시행-착오를 통해 상호 작용하면서 학습을 수행한다. 즉, 에이전트는 학습을 수행하는 동안 주어진 환경에서 선택할 수 있는 행동(a_t)을 수행하고, 외부 환경으로부터 수행한 행동에 대한 평가로서 강화값(r_t)을 받아 강화된다[7].

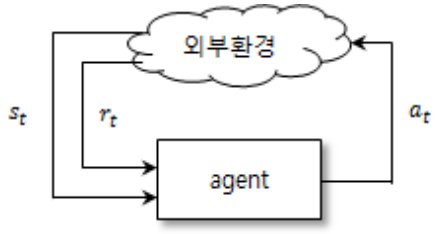

그림 1. 강화 학습 모델
Fig. 1. Reinforcement learning Model

일반적으로 강화 학습을 위해 $Q-$학습이 널리 이용되고 있다[8]. $Q-$학습은 통계적 동적프로그래밍에 근거한 학습 방법으로서 학습을 수행하는 에이전트가 현재 상태(S_t)에서 행동(a_t)을 수행하였을 때 외부 환경으로부터 받는 강화 값(reinforcement value)을 (상태-행동) 쌍에 대한 $Q-$함수 $Q(s_t, a_t)$에 할당한다. 그리고 나서 다음 상태 (s_{t+1})의 (상태-행동) 쌍에 대한 $Q-$함수 $Q(s_{t+1}, a_{t+1})$가 최대가 되는 행동(a_{t+1})을 선택하여 식1과 같이 갱신하면서 학습한다.

$$Q(s_t, a_t) = (1-\alpha) \cdot Q(s_t, a_t) + \alpha \cdot \delta_t \qquad (1)$$

식1에서 α는 학습율(learning rate)이고, δ_t는 현재 상태에서 선택한 (상태, 행동) 쌍에 대한 오류로서 식2와 같이 계산된다.

$$\delta_t = r_{t+1} + \gamma \max_{a \in A(s_t)} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \qquad (2)$$

식2에서 $\gamma(0 < \gamma < 1)$ 은 할인율(discount rate)이고, $A(s_t)$는 에이전트가 현재 상태 s_t에서 선택 가능한 행동들의 집합이다. 현재 상태에서 에이전트가 어떤 행동을 선택할 것인가는 식3과 같은 볼쯔만 확률 분포가 널리 이용된다.

$$p(\bar{a}|s) = \frac{e^{\frac{Q(s_t, \bar{a})}{T}}}{\sum_{a \in A(s_t)} e^{\frac{Q(s_t, a)}{T}}} \qquad (3)$$

식3에서 T는 행동 선택의 임의성(randomness) 정도를 제어하는 변수이다.

2. 먹이 추적 문제

먹이 추적 문제는 분산 인공지능 분야에서 잘 알려진 실험 환경으로 M. Benda에 의해 소개되었으며[9], 실세계의 개념을 구체화 해나가는 모델로 이용되고 있다. 특히, 서로 협력하는 다중에이전트 시스템의 상호 작용을 설명하기 위해 널리 이용되고 있다. 먹이 추적 문제는 그림1과 같이 30×30 격자 환경에서 4개의 추적 에이전트와 1개의 먹이 에이전트(p)들로 구성되어있다.

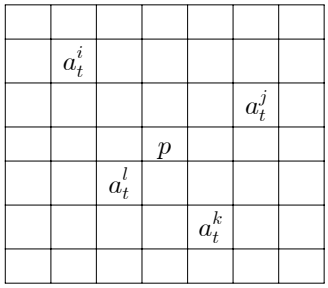

그림 2. 먹이 추적 문제
Fig. 2. Pursuit problem

현재 상태에서 각 에이전트들(a_t^i)은 이동 가능한 방향 $\{N, S, E, W, s\}$에서 제어 전략에 따라 특정 방향으로 이동하며, 먹이 에이전트는 $\{N, S, E, W\}$ 방향 중 임의의 한 방향으로 이동한다. 여기에서 s(stay)는 이동하지 않고 제자리에 있음을 의미한다. 먹이 추적 문제에서 각 에이전트들의 공동 목표는 그림 3과 같이 먹이 에이전트가 움직일 수 없도록 포획하는 것이다.

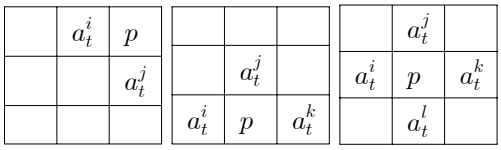

그림 3. 포획 상태
Fig. 3. state of capture

일반적으로 먹이 추적문제에 대한 성능 평가는 얼마나 빠르게 먹이를 포획하였는가? 그리고 에이전트들 간의 충돌이 얼마나 적게 일어났는가를 기준으로 하고 있다. 에이전트들 간의 충돌은 그림 4와 같이 먹이 p를 포획하기 위해 에이전트 (a_t^i)와 (a_t^j)가 S 위치로 이동할 경우 발생한다. 이와 같은 충돌 현상은 에이전트들 간의 공동 목표에 빠르게 수렴할 수 없다.

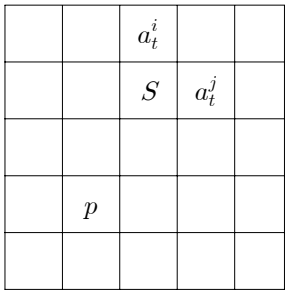

그림 4. 충돌 상태
Fig. 4. Collision state

먹이 추적 문제에서 여러 에이전트들이 먹이를 효율적으로 포획하기 위해 Stephen과 Max가 제안한 지역 제어 전략과 분산 제어 전략이 널리 이용되고 있다.

지역 제어 전략은 비 대화형 제어 전략으로서 각 에이전트들은 먹이를 포획하기 위해 포획 위치를 먼저 설정하고 나서 이동한다. 그림 5와 같이 에이전트 (a_t^i)가 포획 위치를 선점하면 다른 에이전트들은 먹이의 위치를 인지하고 포획하기 위해 이동한다. 지역 제어전략은 반드시 하나 이상의 에이전트가 먹이를 포획하기 위한 위치를 선점해야만 각 에이전트들 간의 협동이 가능하다. 그러므로 지역 제어전략은 먹이 포획을 위해 많은 이동이 필요하며 에이전트들의 포획 위치 중복으로 인한 충돌 현상이 많이 발생하는 문제점이 있다.

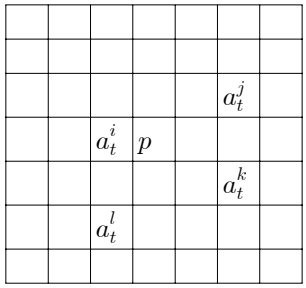

그림 5. 지역 제어 전략
Fig. 5. Local control strategy

분산 제어 전략은 그림 6과 같이 각 에이전트들은 먹이 p를 포획하기 위한 포획 위치 N, S, E, W에 이르는 거리 정보를 표1과 같이 생성한다. 그리고나서 각 에이전트들은 먹이(p)와 가장 멀리 있는 에이전트(a_t^i)에게 가장 가까운 포획 위치N 위치를 할당하고, 할당된 위치와 가장 가까운 거리를 갖는 상태로 이동하여 먹이를 포획하는 전략이다.

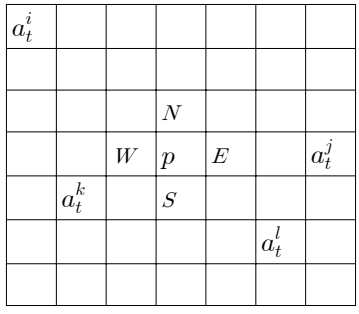

그림 6. 분산 제어 전략
Fig. 6. Distributed control strategy

표 1. 에이전트와 먹이의 거리
Table 1. Distance agent and prey

	N	S	E	W	p
a_t^i	**3.60**	5.00	5.00	3.60	**4.24**
a_t^j	3.16	3.16	**2.00**	4.00	3.00
a_t^k	2.82	2.00	3.16	**1.41**	2.23
a_t^l	4.47	**2.82**	3.16	4.24	3.60

분산 제어 전략은 지역 제어 전략보다 충돌 현상을 매우 감소 시켰으며, 에이전트들 간의 협동을 잘 나타낸 전략이다. 그러나 그림7과 같이 먹이(p)와 거리가 가장 먼 에이전트 a_t^l이 먼저 포획 위치로 E를 선정하고 나서 이동한다. 그리고 나서 에이전트 a_t^i와 a_t^j가 각각 포획 위치 N과 W로 이동하기 위해 화살표 방향으로 이동하였을 때 충돌이 발생한다.

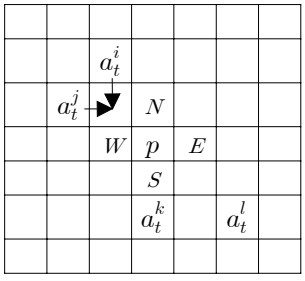

그림 6. 분산 제어 전략의 충돌
Fig. 6. Collision of Distributed control strategy

Ⅲ. 혼성 학습 전략

본 논문에서는 먹이 추적 문제에서 다중 에이전트들의 충돌 현상을 방지하기 위한 혼성 학습 전략을 제안한다. 제안된 방법에서 각 에이전트들은 에이전트들 간의 공간적 관계를 이용하여 학습을 수행한다.

학습을 수행하는 각 에이전트들은 그림7과 같이 자신의 영역(5×5) 영역 안에 다른 에이전트들이 없는 경우 분산 제어 전략을 이용하여 먹이를 추적한다.

그림7에서 에이전트 (a_t^i)는 먹이를 추적하기 위해 이동 가능한 위치 $\{N, S, E, W\}$에서 먹이 p 까지 거리를 구한 다음 최단거리인 S 위치로 이동한다.

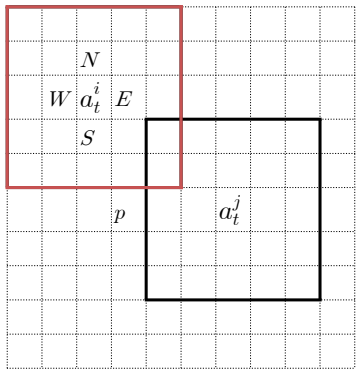

그림 7. 에이전트 영역
Fig. 7. Agent area

만일 에이전트 a_t^i의 영역에 그림8과 같이 다른 에이전트 a_t^j가 발견된다면 각 에이전트는 현재 상태(s_t)에서 선택 가능한 모든 (상태-행동) 쌍에 대한 평가 값을 식4와 같이 갱신한다.

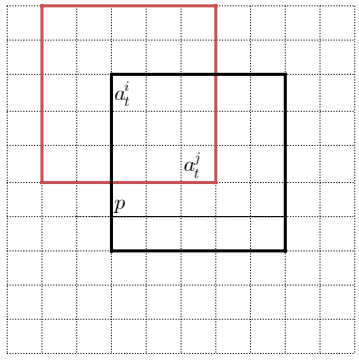

그림 8. 에이전트 a_t^i 의 중첩 영역
Fig. 8. Overlap-region of agent a_t^i

$$For\ \ all$$
$$Q(s_{t,}a_t)=(1-\alpha)Q(s_{t,}a_t)+ \quad (4)$$
$$\alpha[r_{t+1}+\gamma\{\max Q(s_{t+1,}a_{t+1})-Q(s_{t,}a_t)\}]$$

식4에서 α는 학습율, γ는 할인율(discount factor) 그리고 r은 강화 값이며 식5와 같이 계산된다.

$$r_{t+1}=\left[DR(s_t^i p)+SR(a_{t,}^j a_{t,}^i a_{t,}^k)\right]/2 \quad (5)$$

식5에서 $DR(s_{t,}^i p)$ 는 에이전트 (a_t^i)와 먹이 p와의 거리 관계를 의미하며, 식6과 와 같이 계산된다.

$$DR(a^i_{t+1}, p) = 1/\sqrt{(s^i_{(t+1,x)} - p_x)^2 + (s^i_{(t+1,y)} - p_y)^2} \quad (6)$$

식5에서 SR은 에이전트와 인접한 다른 에이전트들 사이의 공간적 관계를 의미하며, 에이전트들 간의 충돌을 방지하는 효과가 있으며 식7와 같이 계산된다.

$$SR(a^j_t, a^i_t, a^k_t) = 1 - (\max(\theta^j_t, \theta^k_t)/2) \quad (7)$$

식7에서 a^i_t와 인접 에이전트 a^j_t와 a^k_t는 에이전트 a^i_t와 먹이 p 그리고 다른 에이전트들과의 각 θ_i, θ_j 그리고 θ_k를 그림9와 같이 구한다.

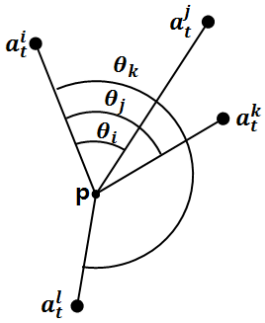

그림 9. 에이전트들 간의 각
Fig. 9. Angle of among the agents

그리고 나서 각이 최대인 에이전트와 최소인 에이전트를 식8과 같이 인접 에이전트로 선정한다.

$$a^j_t = \min(\angle a^i_t, p, a^j_t), \quad a^k_t = \max(\angle a^i_t, p, a^k_t) \quad (8)$$

두 인접 에이전트 a^j_t와 a^k_t 가 선정되면 식8의 θ^j_t 와 θ^k_t 는 식9와 같이 정의된다.

$$\theta^j_t = (\angle a^i_t, p, a^j_t), \quad \theta^k_t = 2\pi - (\angle a^i_t, p, a^k_t) \quad (9)$$

에이전트 a^i_t가 선택할 수 있는 모든 (상태, 행동) 쌍에 대한 강화 값이 계산되면 식10과 같은 상태전이 함수 h에 의해 행동을 선택하여 다음 상태로 이동하며, 혼성 다중에이전트 학습 전략의 알고리즘은 그림 10과 같다.

$$h = \max(Q(s_t, a)), \quad a \in A(s_t) \quad (10)$$

```
HybridMultagent()
{
   Initialize Q(s_t, a_t);
   if(not overlap region)  {
     Move shortest path;
   }
   else {
     For al agents {
       Observe current state s_t;
       For all actions {
         Selection an action a_t at s_t;
           // 다음 상태와 강화 값을 구한다.
         Observe s_t+1, r_t+1;
         Update_Qvalue(s_t, a_t)
         Move agent until capture prey;
       }
     }
   }
}
```

그림 10. 혼성 학습 전략 알고리즘
Fig. 10. Hybrid learning strategy Algorithm

Ⅳ. 실험 및 결과

본 논문에서 제안한 혼성 다중에이전트 학습 전략에 대한 학습 평가를 위해 30×30 격자 환경을 이용하였다. 동일한 학습 환경을 위해 각 에이전트들은 각각 $(0,0)$, $(0,29)$, $(20,0)$, $(29,29)$에서 제어 전략에 따라 (N, S, E, W, s) 방향으로 이동하며 먹이는 $(14,14)$에서 임의의 방향으로 이동한다.

일반적으로 먹이 추적 문제의 성능 평가 기준은 먹이 포획의 성공률, 각 에이전트의 상태전이 회수 그리고 에이전트들 간의 충돌 회수를 기준으로 한다. 본 논문에서 제안한 방법과 Stephan, Max가 제안한 지역제어 전략 그리고 분산 제어 전략을 각 100회 실험한 결과는 표2와 같다.

표 2. 실험 결과
Table 2. Experiment result

	포획 회수	탈출 회수	충돌회수	
			최소	최대
지역제어전략	92	8	49	138
분산제어전략	95	5	5	98
혼성제어전략	100	0	2	26

표2에서 나타난 것처럼 각 제어 전략을 100회 반복하는 동안 지역 제어 전략은 92회, 분산 제어 전략은 95회 포획에 성공하였다. 에이전트들 간의 충돌 회수는 지역 제어전략은 49회에서 138회로 많은 충돌이 발생하였고, 분산 제어 전략은 최소 5회 최대 98회의 충돌 회수가 발생하였다.

본 논문에서 제안한 혼성 제어전략은 100회 모두 포획에 성공하였고, 충돌회수는 최소 2회에서 최대 26회 발생하였다. 그러므로 본 논문에서 제안한 제어 전략이 지역, 분산 제어 전략보다 더 효율적임을 알 수 있었다. 이는 에이전트와의 거리 관계만을 고려한 제어 전략보다 공간적 관계를 고려하는 것이 더 효율적임을 알 수 있다. 먹이를 포획하는 동안 에이전트의 상태 전이 수는 표3과 같다.

표 3. 상태 전이 수
Table 3. Number of state transition

상태전이 / 제어전략	상태 전이 회수		
	최소	최대	평균
지역 제어 전략	161	2443	978
분산 제어 전략	129	2317	656
제안된 방법	98	452	261

표3에서 나타난 것처럼 먹이를 포획하는 동안 지역 제어 전략은 상태 전이가 평균 978회, 분산 제어전략은 평균 656회 발생하였으나 본 논문에서 제안한 혼성 제어 전략은 평균 261회 발생하였다. 먹이를 100회 포획하는 동안 매 10회 포획할 때 마다 평균 상태 전이 회수는 그림 11과 같다.

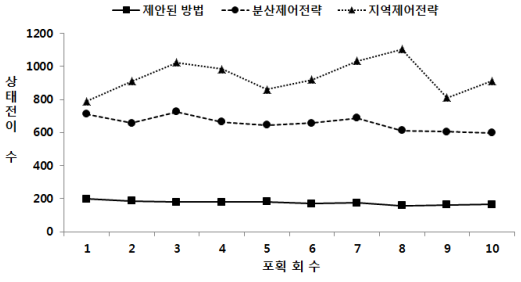

그림 11. 평균 상태 전이 수
Fig. 11. Average number of state transitions

그림 11에 나타난 것처럼 지역 제어 전략과 분산 제어 전략은 먹이의 이동 위치에 따라 상태 전이 수가 많은 변화를 일으켰으나 본 논문에서 제안한 방법은 먹이의 위치와 상관없이 매우 빠르고 안정적으로 먹이를 포획하는 것을 알 수 있었다.

V. 결 론

먹이 추적 문제는 다중 에이전트의 성능 평가와 다양한 실세계의 개념을 구체화하기 위해 널리 이용되고 있다. 본 연구를 통해 다중 에이전트를 제어하기 위해 에이전트와 목표만을 고려하는 것 보다 에이전트와 에이전트 사이의 공간적 관계를 고려하는 것이 더 효율적임을 알 수 있었다.

본 논문에서 제안한 방법은 목표와 다른 에이전트들의 상태 정보를 알고 있는 대화형 제어 전략이므로 목표와 다른 에이전트들의 상태 정보 없이 에이전트의 임무를 수행할 수 있는 비 대화형 에이전트의 제어전략에 관한 연구가 필요하다.

References

[1] Peter Stone, Manuela Veloso, Multi-agent system : “A Survey from a Machine Learning”, Technical Report CMU-CS-97-193, The University of Carnegies Mellon, December, 4, 1997.

[2] Sandip Sen, Mahendra Seckaran, “Multi-agent coordination with learning classifier system”, In Proceeding of the AAAI 99 Worshop on Negotiation, pp.44-49, 1999.

[3] Claus, C. and C. Boutilier, 1998. The dynamics of reinforcement learning in coorperative multiagent systems. Proceeding of AAAI-98, pp. 746-752.

[4] R.S. Sutton and A.G. Barto, Reinforcement Learning : An Introduction, MIT Press, 1998.

[5] C. Guestrin, M.Lagoudakis and R. Parr. “Coordinated reinforcement learning”, In Proc. of th 19th International Conference on machine Learning, 2002.

[6] L.M. Sephens 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] A. G. Barto and D.A. White and D.A. Sofge, “Reinforcement Learning with less data and less real time”, Machine Learning, 13, pp.103-130, 1993.

[8] C.J.C.H. Watkins, “Technical note : Q-learning”, Machine Learning, 8, pp. 279-292, 1989.

[9] M.Benda, V. Jagannathan and R.Dodhiawala, “On optimal cooperation of knowledge source an empirical investigation”, Technical Reoprt BCS-G2010-28, Boeing Advanced Technology Center, Boeing Computing Services, Seattle, Washington, July, 1986.

※ 본 연구는 2013년도 한경대학교 교비 파견연구비의 지원에 의한 것임

저자 소개

김 병 천(정회원)

- 1999년 명지대학교 컴퓨터공학과(박사)
- 경력 1991년~현재 : 국립한경대학교 컴퓨터웹정보공학과 교수

<주관심분야 : Machine Learning, Multiagent System>

이 창 훈(정회원)

- 1998년 중앙대학교 컴퓨터공학과(박사)
- 경력 : 2002년~현재 : 국립한경대학교 컴퓨터웹정보공학과 교수

<주관심분야 : 정형화 방법, 컴포넌트>