상태 공간 압축을 이용한 강화학습

김 병 천† · 윤 병 주††

요 약

강화학습(reinforcement learning)은 동적 환경에서 시도와 오류(trial-and-error)에 의해 상호 작용하면서 학습을 수행한다. 그러므로 동적 환경에서 Q-학습(learning)이나 TD(Temporal Difference)-학습과 같은 강화학습 방법들은 전통적인 통계적 학습 방법보다 더 빠르게 학습을 할 수 있다. 그러나 제안된 대부분의 강화학습 알고리즘들은 학습을 수행하는 에이전트(agent)가 목표 상태에 도달하였을 때만 강화 값이(reinforcement value) 주어지기 때문에 최적 해(optimal solution)에 매우 늦게 수렴한다.

본 논문에서는 미로 환경(maze environment)에서 최단 경로를 빠르게 찾을 수 있는 강화학습 방법(COMREL : COMpressed REinforcement Learning)을 제안하였다. COMREL은 주어진 미로 환경을 압축하고 나서 압축된 미로 환경에서 최단 경로가 될 수 있는 후보 상태들을 선택한다. 그리고 최단 경로를 찾기 위해 후보 상태들에 대해서만 학습을 수행한다. COMREL을 기존의 Q-학습과 Priortized Sweeping 알고리즘과 비교한 결과 학습 시간이 매우 단축됨을 알 수 있었다.

Reinforcement Learning Using State Space Compression

Byung-Cheon Kim† · Byung-Joo Yoon††

ABSTRACT

Reinforcement learning performs learning through interacting with trial-and-error in dynamic environment. Therefore, in dynamic environment, reinforcement learning method like Q-learning and TD(Temporal Difference)-learning are faster in learning than the conventional stochastic learning method. However, because many of the proposed reinforcement learning algorithms are given the reinforcement value only when the learning agent has reached its goal state, most of the reinforcement algorithms converge to the optimal solution too slowly.

In this paper, we present COMREL(COMpressed REinforcemetn Learning) algorithm for finding the shortest path fast in a maze environment. COMREL compress the given maze environment, select the candidate states that can guide the shortest path in compressed maze environment, and learn only the candidate states to find the shortest path. After comparing COMREL algorithm with the already existing Q-learning and Priortized Sweeping algorithm, we could see that the learning time shortened very much.

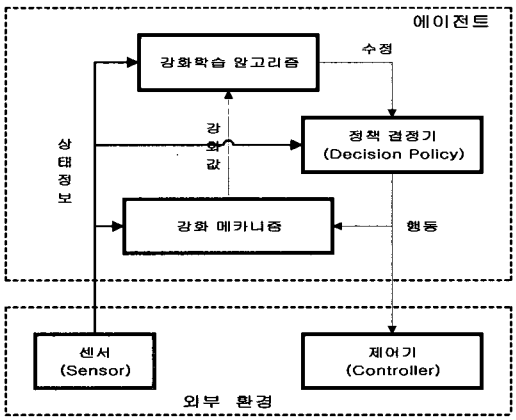
1. 서 론

학습(learning)이란 과거의 경험을 이용하여 현재의 문제를 해결하기 위한 지식이나 기술(skill)을 획득하고 개선하는 것을 의미한다. 이와 같은 학습을 통해 주변 환경의 변화에 유연하게 대처해 나갈 수 있는 능력이 생기게 되며, 어떻게 훈련(training)하여 학습할 것인가에 따라 교사학습(supervised learning), 비교사학습(unsupervised learning) 그리고 강화학습(reinforcement learning)으로 분류된다[1]. 일반적으로 교사학습은 입력과 출력이 제시되어야 하고 결정적 환경(deterministic environment)에서 학습을 수행할 수 있다. 그러나 실제 상황에서는 교사 학습에서 요구하는

† 정 회 원 : 한경대학교 컴퓨터공학과 교수
†† 종신회원 : 명지대학교 컴퓨터공학과 교수
논문접수 : 1998년 8월 28일, 심사완료 : 1998년 12월 7일

정확한 출력을 알 수 없는 경우가 매우 많기 때문에 최근 강화학습으로 알려진 새로운 학습 방법에 대한 연구와 관심이 증가하고 있다.

Minsky[2]에 의해 제시된 강화학습은 동적 프로그래밍과 교사학습을 혼합한 학습 방법으로서 입력과 출력이 제시되지 않으며, 비 결정적인 환경(non-deterministic environment)에서 학습을 수행할 수 있다. 강화학습은 학습을 수행하는 에이전트(agent)를 강화(reinforce)시키기 위해 주어지는 스칼라(scalar) 형의 강화 값(reinforcement value)에 의해 학습을 수행한다. 일반적으로 강화학습은 (그림 1)과 같이 학습을 수행하는 에이전트와 외부 환경으로 구성되어 있다[3]. 에이전트는 센서(sensor)를 통해 외부 환경에 대한 상태 정보를 획득한다. 학습 단계에서 에이전트는 현재 상태(s_t)를 관찰하고 정책 결정기(decision policy)에 따라 현재 상태에서 수행할 수 있는 최적 행동(optimal action)을 결정한다. 최적 행동이 결정되면 제어기(controller)에 의해 그 행동을 수행하며, 강화 메커니즘(mechanism)은 강화 값을 발생시킨다. 강화 값은 현재 상태에서 수행된 행동에 대한 보상(reward)이라 할 수 있다. 강화 메커니즘에 의해 발생된 강화 값은 강화학습 알고리즘에 따라 에이전트의 정책 결정 과정을 수정한다. 이와 같은 학습의 목적은 상태와 행동을 사상시키는 최적의 정책(optimal policy)을 구성하는 것이다.



(그림 1) 강화학습 구조
(Fig. 1) Reinforcement learning structure

장애물을 포함하고 있는 미로(maze) 환경에서 최단 경로를 찾기 위해 많은 강화학습 알고리즘들이 제안되었다[4,5,6,7]. 그러나 제안된 알고리즘들은 학습을 수행하는 에이전트가 목표 상태에 도달하였을 때만 강화 값이 주어지고 최단 경로와 관계가 없는 상태들을 많이 방문하기 때문에 학습 시간이 오래 걸리는 단점이 있다[8].

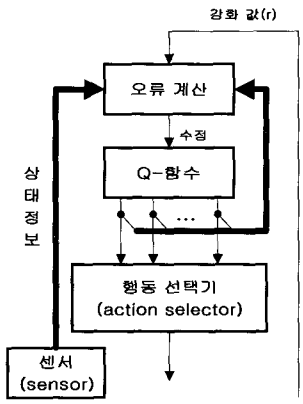
본 논문에서는 미로 환경에서 최단 경로를 보다 빠르고 효율적으로 찾을 수 있는 알고리즘을 제안하였고 이를 COMREL(COMpressed REinforcement Learning)이라 하였다. COMREL은 주어진 환경을 압축하고 나서 압축된 환경에서 중간 강화 값(intermediate reinforcement value : r')을 이용하여 최단 경로가 될 수 있는 후보 상태(candidate state)들을 선택한다. 그리고 나서 선택된 후보 상태들에 대해 지연 강화 값(delayed reinforcement value : r)을 이용하여 학습을 수행한다. 중간 강화 값이란 학습 과정에서 에이전트에게 주어지는 강화 값을 의미하고 지연 강화 값이란 목표 상태를 발견하였을 때 에이전트에게 주어지는 강화 값을 의미한다. COMREL 알고리즘을 복잡한 장애물을 포함하고 있는 미로 환경에 적용한 결과 기존의 Q-학습[9]과 Prioritized Sweeping[10] 보다 빠르고 효율적으로 최단 경로를 찾을 수 있음을 알 수 있었다.

본 논문의 전개는 다음과 같다. 2장에서는 MDP (Markov Decision Problem) 환경에서 학습하기 위해 널리 이용되고 있는 Q-learning 알고리즘에 대하여 소개하고, 3장에서는 본 논문에서 제안한 COMREL 알고리즘에 대하여 설명하였다. 4장에서는 COMREL과 Q-학습 및 Prioritized Sweeping 알고리즘을 미로 환경에 각각 적용하여 성능을 분석하였고, 5장에서는 결론 및 향후 연구 방향을 제안하였다.

2. Q-learning

유한 상태(finite state)의 MDP 환경[11]에서 최단 경로를 찾기 위한 강화학습 방법은 Watkins가 제안한 Q-학습이 널리 이용되고 있다. Q-학습은 통계적 동적 프로그래밍(stochastic dynamic programming)에 근거한 강화학습으로서 (그림 2)와 같이 구성되어 있다[3].

(그림 2)에서 오류 계산은 식(1)과 같이 TD-오류(temporal difference error)를 계산한다. 식(1)에서 r_t는 강화 값이고 γ는 할인율(discount rate)이다.



(그림 2) Q-학습 구조
(Fig. 2) Q-learning structure

$$\Delta = r_t + \gamma \cdot (\max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q(s_t, a_t)) \quad \text{식(1)}$$

TD-오류를 계산하고 나서 식(2)를 이용하여 Q-함수(Q-function) 값을 계산하며, α는 학습율(learning rate)이다.

$$Q(s_t, a_t) = (1-\alpha) \cdot Q(s_t, a_t) + \alpha \cdot \Delta \quad \text{식(2)}$$

행동 선택기(action selector)는 Q-함수에 의해 계산된 Q-값들 중에서 가장 적당한 행동을 선택한다. 가장 적당한 행동을 선택하는 방법은 식(3)과 같은 볼쯔만(Boltzmann) 확률 분포에 따라 행동을 선택하는 방법이 널리 사용되고 있다.

$$p(\bar{a} \mid x) = \frac{e^{\frac{Q(s_t, \bar{a})}{T}}}{\sum_{a \in A(s_t)} e^{\frac{Q(s_t, a)}{T}}} \quad \text{식(3)}$$

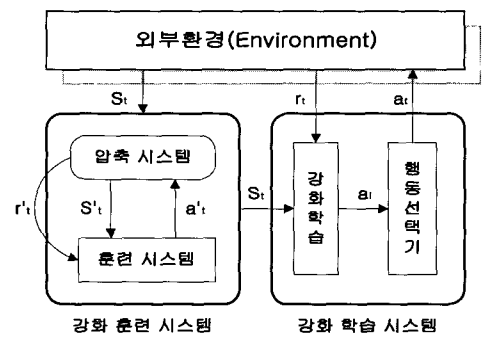
식(3)에서 T는 행동 선택의 임의성(randomness) 정도를 제어하는 온도(temperature) 변수이다. Q-함수를 이용한 학습 알고리즘은 (그림 3)과 같다.

REPEAT {
1. Observe current state s_t ;
2. Select an action a_t *for state* s_t ;
3. Perform action a_t ;
3.1 Observe new state s_{t+1},
reinforcement r_t ;
4. $\Delta = r_t + \gamma \cdot (\max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q(s_t, a_t))$;
5. $Q(s_t, a_t) = (1-\alpha) \cdot Q(s_t, a_t) + \alpha \cdot \Delta$;
} UNTIL(the required number of cycles)

(그림 3) Q-학습 알고리즘
(Fig. 3) Q-learning algorithm

3. COMREL 알고리즘

본 논문에서 제안한 COMREL 알고리즘은 Q-학습의 학습 속도를 개선하기 위해 (그림 4)와 같이 중간 강화 값(r')을 이용하여 최단 경로가 될 수 있는 후보 상태들을 선택하는 강화 훈련 시스템(training system)과 선택된 후보 상태들에 대해 지연 강화 값(r)을 이용하여 최단 경로를 찾기 위해 학습을 수행하는 강화 학습 시스템(learning system)으로 구성되어 있다.



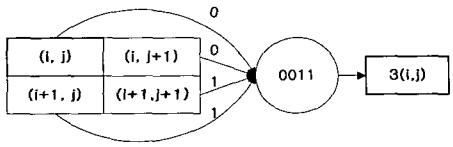
(그림 4) COMREL의 구조
(Fig. 4) COMREL's structure

3.1 강화 훈련 시스템

강화 훈련 시스템은 주어진 환경($S_t(n \times n)$)을 압축하여 새로운 환경($S'_t(n/2 \times n/2)$)으로 구성하는 압축 시스템(compression system)과 압축된 환경에서 중간 강화 값(r')을 이용하여 최단 경로가 될 수 있는 후보 상태들을 선택하는 훈련 시스템(training system)으로 구성되어 있다. 압축된 환경을 이용하는 이유는 탐색 공간이 1/4로 축소되어 최단 경로를 위한 후보 상태를 빠르게 찾을 수 있기 때문이다.

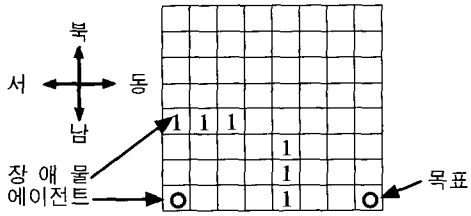
압축 시스템은 주어진 환경을 왼쪽에서 오른쪽(left to right)으로 위에서 아래(top to bottom)로 이동하면서 전체 상태 공간을 1/4로 압축된 환경으로 재구성하

는 작업을 수행한다. 즉, (그림 5)와 같이 현재의 위치 (i, j)를 포함하고 있는 (2×2) 상태들을 시계방향(clockwise)으로 장애물의 유무에 따라 0에서 15사이의 값을 갖는 (1×1) 상태로 압축한다.

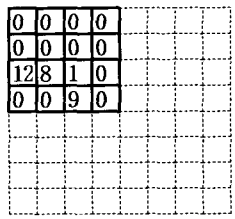


(그림 5) 압축 과정
(Fig. 5) Compression process

예를 들어 S. P. Singh[12]가 제안한 (그림 6)과 같이 장애물을 포함하고 있는 미로 환경에서 장애물은 1, 장애물이 아니면 0으로 표현되었을 때 압축시스템에 의한 압축 결과는 (그림 7)과 같다.



(그림 6) 미로 환경
(Fig. 6) Maze environment



(그림 7) 압축된 환경
(Fig. 7) Compressed environment

훈련 시스템은 압축시스템에 의해 (그림 7)과 같이 압축된 환경에서 선택 가능한 다음 상태들에 대해 중간 강화 값을 이용하여 시작 상태에서 목표 상태까지 최단 경로가 될 수 있는 후보 상태들을 선택한다. 예를 들어 현재 상태 값이 12(⊟)인 경우 훈련 시스템이 북쪽 방향에 대해 선택할 수 있는 다음 상태는 없고, 동쪽 방향은 다음 상태 값이 0, 2, 4, 6, 8, 10, 12, 또는 14인 상태들 중 한 상태를 선택할 수 있고, 남쪽 방향은 다음 상태 값이 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 또는 11인 상태들 중 한 상태를 선택할 수 있다. 그리고 서쪽 방향은 다음 상태 값이 0, 1, 2, 3, 4, 5, 8, 9, 12, 또는 13인 상태들 중 한 상태를 선택할 수 있다.

압축된 상태 공간(S_t')의 현재 상태(s_t')에서 다음 상태(s_{t+1}')는 식(4)와 같이 $Q'(s_t', a_t')$에 의해 결정된다.

$$Q'(s_t', a_t') = \max r_t' \qquad \text{식(4)}$$
$$r_t' = -R \cdot sign(d_{t+1} - d_t)$$

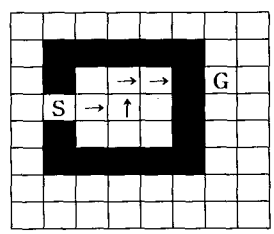
식(4)에서 r'는 중간 강화 값이고, R은 0과 1사이의 값 ($0 < R < 1$)을 갖는다. $sign()$ 값은 식(5)와 같다.

$$sign(z) = \begin{cases} 1, & \text{if } z > 0 \\ 0, & \text{if } z = 0 \\ -1, & \text{if } z < 0 \end{cases} \qquad \text{식(5)}$$

식(4)의 d_t는 현재 상태($s_t(x_t, y_t)$)에서 목표 상태 ($g(\bar{x}_t, \bar{y}_t)$)까지의 거리이며 식(6)과 같이 계산된다.

$$d_t = |x_t - \bar{x}_t| + |y_t - \bar{y}_t| \qquad \text{식(6)}$$

훈련 시스템이 현재 상태에서 선택 가능한 다음 상태들 중에서 선택한 다음 상태가 목표 상태에 가까워지면 중간 강화 값은 양의 값을 가지며, 목표 상태로부터 멀어지면 음의 값을 갖는다. 이와 같은 방법은 장애물이 없는 환경에서는 매우 효율적으로 목표 상태를 찾을 수 있지만 (그림 8)과 같이 오목한(concave) 형태의 장애물이 있는 환경에서는 학습을 수행하면 장애물의 안쪽으로 수렴해 나간다. 이를 오도된 강화(misleading reinforcement)라 한다.



(그림 8) 오목한 형태의 환경
(Fig. 8) Concave environment

훈련 시스템은 중간 강화 값에 의해 오도하는 것을 방지하기 위해 (그림 9)와 같은 알고리즘을 적용하였다.

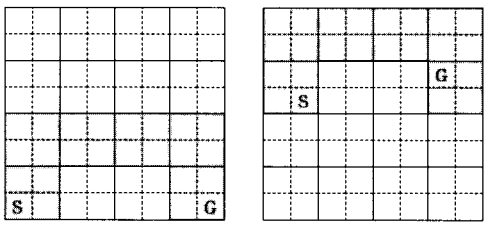
이 알고리즘은 압축된 환경에서 훈련 시스템이 선택한 다음 상태가 이미 방문한 상태인 경우 후보 상태로 선택하지 않고 이전 상태로 되돌아가는 역할을 한다.

```
check_new_state(I, s'_{t+1})
/* s'_{t+1} : new state, I : STACK */
{
  if ( s'_{t+1} = not visited state) {
     Q'(s_t', a_t') = max r_t' ;

     PUSH (I, s'_{t+1});
  }
  else {
     Q'(s_t', a_t') = Large Negative Value;
     POP (I, s'_{t+1});
  }
}
```

(그림 9) 오도된 강화를 방지하기 위한 알고리즘
(Fig. 9) Algorithm for prevent misleading reinforcement

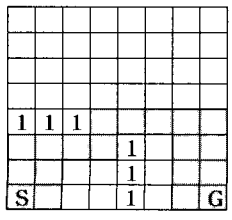
강화훈련 시스템을 (그림 6)과 (그림 8)에 적용한 결과 (그림 10) (a)와 (그림 10) (b)와 같이 최단 경로가 될 수 있는 후보 상태들이 선택되었다.



(a) (b)

(그림 10) 후보 상태들
(Fig. 10) Candidate states

강화 훈련 시스템은 (그림 10)과 같은 후보 상태들을 (그림 11)과 같이 주어진 미로 환경에 사상시켜 강화학습 시스템에 입력한다.



(그림 11) 사상된 상태 공간
(Fig. 11) Mapped state space

3.2 강화학습 시스템

강화학습 시스템은 최단 경로를 찾기 위해 강화훈련 시스템에 의해 선택된 후보 상태들에 대해서 지연 강화 값을 이용하여 학습을 수행한다. 강화학습 시스템은 후보 상태들과 사상된 상태(s_t)들에 대해서 식(7)을 이용하여 Q-함수 값을 계산한다.

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \cdot \max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q(s_t, a_t)) \quad \text{식(7)}$$

식(7)에서 α는 학습율(learning rate), r_t은 지연 강화 값 그리고 γ는 discounted factor이다. 지연 강화 값은 다음 상태가 목표 상태인 경우 1($r_t = 1$), 다음 상태가 목표 상태가 아닌 경우 -1($r_t = -1$) 값을 갖는다. 강화학습 시스템이 후보 상태들에 대해서만 학습을 수행하는 이유는 시작 위치에서 목표 위치까지 가능한 불필요한 상태들을 방문하지 않고 최단 경로에 빠르게 수렴하기 위함이다. 현재 상태에서 선택 가능한 행동들의 집합 $A(s_t)$에 속해 있는 특정 행동(a)을 선택하기 위한 방법은 식(8)과 같은 Gibbs 확률 분포(probability distribution)에 의해 선택하였다.

$$P(a \mid x) = \frac{e^{\beta Q_t(s_t, a)}}{\sum_{a' \in A(s_t)} e^{\beta Q_t(s_t, a')}} \quad \text{식(8)}$$

식(8)에서 β는 임의성(randomness) 정도를 제어하는 상수이고 COMREL 알고리즘은 (그림 12)와 같다.

```
1. Initialize stack I, Q'(s', a'), Q(s, a);
2. Compress of Maze Environment;
3. In Compressed Environment( S_t');
   REPEAT {
     Get current state( s_t') and Choose best action( a_t');
     Get the new next state( s'_{t+1} ) and
         intermediate reinforcement value( r'_t);
     Q'(s_t', a_t') = max r'_t ;

     check_new_state(I, s'_{t+1});
   } UNTIL( s'_{t+1} == goal state)
4. In Maze Environment( S_t)
   REPEAT {
     Get start state( s_t), Choose best action( a_t);
     Get the new next state( s_{t+1}) and
           delayed reinforcement value( r_t);
```

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \cdot \max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q(s_t, a_t))$$

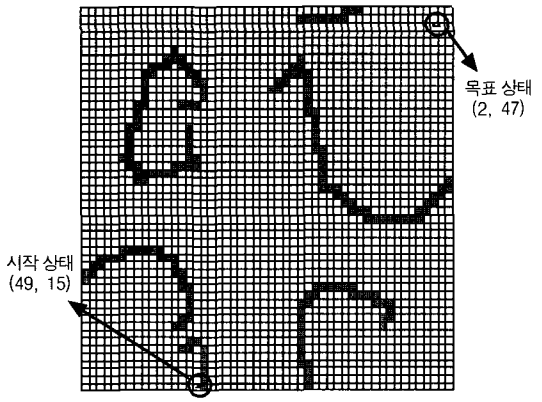
} ***UNTIL***(the required number of cycles)

(그림 12) COMREL 알고리즘
(Fig. 12) COMREL algorithm

4. 실험 및 평가

장애물을 포함하고 있는 미로 환경에서 강화학습 알고리즘의 성능 평가 기준은 일반적으로 에피소드(episode)에 의한 방법을 사용하고 있다[13]. 에피소드란 학습을 수행하는 에이전트와 외부 환경과의 상호작용이 자연적으로 끝나는 것을 의미한다. 미로 환경에서 에피소드란 시작 상태에서 목표 상태를 찾을 때까지의 상태 전이를 의미하며 학습 알고리즘의 평가 기준은 최단 경로를 찾을 때까지 몇 번의 에피소드가 발생하였는가? 그리고 각 에피소드 당 몇 번의 상태전이가 발생하였는가를 의미한다.

본 논문에서 제안한 COMREL의 성능 평가를 위해 (1) 몇 번의 에피소드가 발생하여 최단 경로를 찾았는가? (2) 각 에피소드 당 상태 전이 수는 몇 번인가? (3) 최단 경로를 찾기 위해 한 번 이상 방문한 상태들의 수 몇 개인가? 등을 기준으로 하고 있으며 실험을 위한 환경은 64개의 상태들로 구성된 그림(6)과 2500개의 상태들로 구성된 (그림 13)과 같은 미로 환경에 각각 적용하였다.



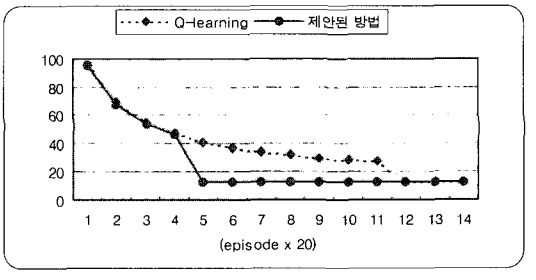
(그림 13) 미로 환경
(Fig. 13) Maze environment

64개의 상태 공간과 2500개의 상태 공간에서 Q-학습과 COMREL의 학습 결과는 <표 1>과 같다. 최단 경로를 찾기 위해 Q-학습은 64개의 상태 공간에서는 약 300,000번의 상태 전이를 하였고 57개의 상태들을 한 번 이상 방문하였다. 그리고 2500개의 상태 공간에서는 약 3,800,000번의 상태 전이를 하였고 2339개의 상태들을 한 번 이상 방문하였다. 그러나 COMREL은 64개의 상태공간에서 약 3,800번의 상태 전이를 하였고 24개의 상태들을 1번 이상 방문하였다. 그리고 2500개의 상태 공간에서 약 6,700번의 상태전이를 하였고 144개의 상태들만 한 번 이상 방문하였다.

〈표 1〉 학습 결과
〈Table 1〉 Learning result

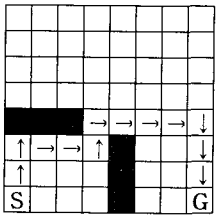
학습 방법 / 상태 공간	Q-learning		COMREL	
	상태 전이 수	방문 상태 수	상태 전이 수	방문 상태 수
64개	300,000	57	3,800	24
2500개	3,800,000	2,339	6,700	144

64개의 상태들로 구성된 미로 환경에서 시작 상태(7, 0)에서 목표 상태(7, 7)까지 최단 경로를 찾기 위해 수행한 각 에피소드 당 상태 전이 수를 Q-학습과 비교한 결과는 (그림 14)와 같다.



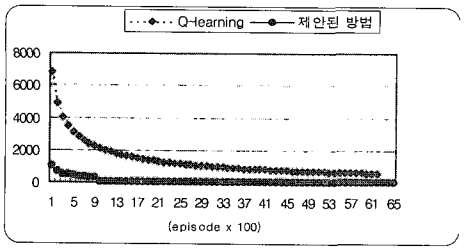
(그림 14) 64개의 상태 공간에 대한 학습 결과
(Fig. 14) Learning result of 64 state space

(그림 14)에서 y축은 에피소드가 20회 발생할 때마다 평균 상태 전이 수를 의미한다. Q-학습은 최단 경로를 찾기 위해 약 240(12×20)의 에피소드가 발생하였다. 즉 시작 상태에서 목표 상태를 약 240번 찾은 후 최단 경로를 찾을 수 있었다. COMREL 알고리즘은 약 100(5×20)번의 에피소드가 발생한 후 (그림 15)와 같은 최단 경로를 찾을 수 있었다.



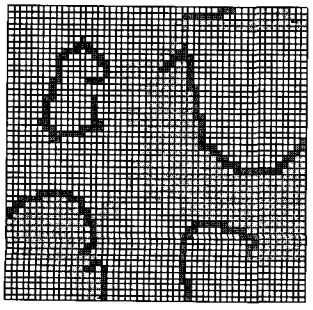
(그림 15) 학습 결과
(Fig. 15) Learning result

2500개의 상태들로 구성된 환경에서 최단 경로를 찾기 위해 수행한 각 에피소드 당 상태 전이 수를 Q-학습과 비교한 결과는 (그림 16)과 같다. (그림 16)에서 y축은 에피소드가 100번 발생할 때마다 평균 상태 전이 수를 의미한다. Q-학습은 6300(63×100)번의 에피소드가 발생한 후 최단 경로를 찾을 수 있었고, COMREL은 1000(10×100)번의 에피소드 후에 최단 경로를 찾을 수 있었다.



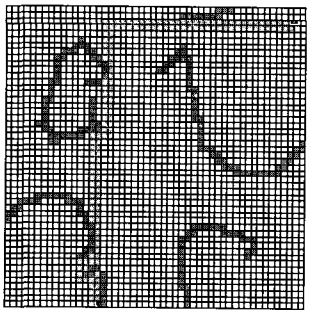
(그림 16) 2500개의 상태 공간에 대한 학습 결과
(Fig. 16) Learning result of 2500 state space

Andrew W. Moore가 제안한 Prioritized Sweeping 알고리즘은 2500개의 상태들로 이루어진 환경에서 시작 상태에서 목표 상태까지 최단 경로를 찾기 위해 1번 이상 방문한 상태들은 (그림 17)과 같이 약 35%의 상태들을 방문하였다.

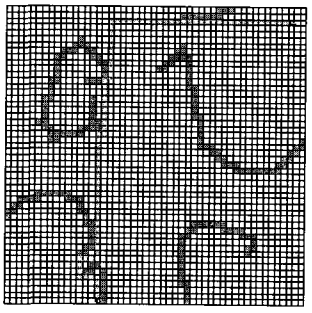


(그림 17) Prioritized Sweeping 알고리즘의 방문 상태들
(Fig. 17) Visited states of Prioritized Sweeping algorithm

그러나 COMREL 알고리즘은 강화훈련 시스템이 최단 경로 찾기 위해 1번 이상 방문한 상태들은 (그림 18)과 같이 2500개의 상태들 중 약 6%의 상태들만 방문하여 후보 상태들을 선택하였다. 선택된 후보 상태들에 대해 강화학습 시스템은 최단 경로를 찾기 위해 약 6700번의 상태 전이 후 (그림 19)와 같은 최단 경로를 찾았다.



(그림 18) COMREL 알고리즘의 방문 상태들
(Fig. 18) Visited states of COMREL algorithm



(그림 19) COMREL의 학습 결과
(Fig. 19) COMRES's learning result

5. 결 론

본 논문에서 제안한 COMREL 알고리즘을 64개의 상태 공간과 2500개의 상태 공간을 가진 환경에서 학습한 결과 Q-learning과 Prioritized Sweeping 알고리즘보다 매우 빠르게 학습한다는 것을 알았다. 이는 압축된 환경에서 최단 경로에 대한 후보 상태들을 선택하였고 선택된 후보 상태들에 대해서만 학습을 수행하였기 때문에 전체 학습시간을 단축할 수 있었다. 그러므로 학습 과정에서 문제 영역에 관한 지식(domain knowledge)을 활용하는 것이 바람직하다는 것을 알 수 있었다.

그러나 본 논문에서 제안한 CORMEL 알고리즘 뿐만 아니라 거의 모든 강화학습 알고리즘이 학습을 언제까

지 수행하여야 최적 해에 수렴하는가에 대해 명확하게 기술하지 않고 있다. 그러므로 학습을 언제까지 수행하여야 하는가에 대한 연구가 필요하다. 그리고 제안된 COMREL 알고리즘은 하나의 목표 상태를 찾는 문제에 적용하였으나 여러 개의 목표 상태가 합성(composite)되어 있는 환경에서 잘 학습할 수 있는 알고리즘에 관한 연구가 필요하며, 실세계의 다양한 환경에 잘 적응할 수 있는 강화학습에 관한 연구도 필요하다.

참 고 문 헌

[1] L. P. Kaelbling, M. L. Littman and A. W. Moore, "Reinforcement Learning : A Survey," Journal of AI Research 4, pp.237-285, 1996.

[2] M. L. Minsky, "Theory of Neural-Analog Reinforcement Systems and its Applications to the Brain-Model Problem," PhD thesis, Princeton University, 1954.

[3] P. Cichosz, "Reinforcement Learning Algorithms Based on the Method of Temporal Difference," MS thesis, University of Warsaw, Computer Science, 1996.

[4] S. Sehad and C. Touzet, "Reinforcement Learning and Neural Reinforcement Learning," ESANN, 1994,

[5] M. Wiering and J. Schmidhuber, "HQ-Learning," Adaptive Behavior 6, 1997.

[6] A. W. Moore, "The parti-game algorithm for variable resolution reinforcement learning in multidimensional space," Advances in Neural Information Processing Systems, pp.711-718, 1994.

[7] L. P. Kaelbling, "Learning to Achieve Goals", IJCAI, 1993

[8] P. Dayan, "The convergence of TD(λ) for general λ," Machine Learning, 8(3), pp.341-362, 1992.

[9] C. J. C. H. Watkins and P. Dayan, "Technical note : Q-learning," Machine Learning, 8 : pp.279-292, 1992.

[10] A. W. Moore and C. G. Atkeson, "Priroitized Sweeping : Reinforcement Learning with Less Data and Less Real Time," Machine Learning, 13, 1993.

[11] M. L. Puterman, "Markov Decision Processes - Discrete Stochastic Dynamic Programming," John Wiley & Sons, New York, 1994.

[12] S. P. Singh, "Transfer of learning across compositions of sequential tasks," Machine learning : Proceedings of the Eighth International Workshop, pp.348-352, San Mateo, CA, 1991.

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

김 병 천

e-mail : bckim@anu.ansung.ac.kr

1988년 한남대학교 전자계산학과(학사)

1990년 숭실대학교 대학원 전자계산학과(석사)

1997년 명지대학교 대학원 컴퓨터공학과 박사과정 수료

1991년~1993년 안성농업전문대학교 전자계산학과 전임강사

1993년~현재 한경대학교 컴퓨터공학과 조교수

관심분야 : Machine Learning, Artificial Neural Network, Knowledge-Based Systems

윤 병 주

e-mail : yoonbj@wh.myongji.ac.kr

1975년 경북대학교 수학과(학사)

1982년 한국과학기술원 전산학과(석사)

1994년 Florida State University 전산학과(박사)

1982년~현재 명지대학교 컴퓨터공학과 교수

관심분야 : Machine Learning, Knowledge-Based System, Hybrid Intelligent Systems