# 감독 지식을 융합하는 강화 학습 기법을 사용하는 셀룰러 네트워크에서 동적 채널 할당 기법

# (A Dynamic Channel Assignment Method in Cellular Networks Using Reinforcement Learning Method that Combines Supervised Knowledge)

김 성 완 † 장 형 수 ††

(Sung Wan Kim) (Hyeong Soo Chang)

**요 약** 최근에 제안된 강화 학습 기법인 "potential-based" reinforcement learning(RL) 기법은 다수 학습들과 expert advice들을 감독 지식으로 강화 학습 알고리즘에 융합하는 것을 가능하게 했고 그 효용성은 최적 정책으로의 이론적 수렴성 보장으로 증명되었다. 본 논문에서는 potential-based RL 기법을 셀룰러 네트워크에서의 채널 할당 문제에 적용한다. Potential-based RL 기반의 동적 채널 할당 기법이 기존의 fixed channel assignment, Maxavail, Q-learning-based dynamic channel assignment 채널 할당 기법들보다 효율적으로 채널을 할당한다. 또한, potential-based RL 기법이 기존의 강화 학습 알고리즘인 Q-learning, SARSA(0)에 비하여 최적 정책에 더 빠르게 수렴함을 실험적으로 보인다.

**키워드** : 강화 학습, SARSA(0), 셀룰러 네트워크, 채널 할당 기법

***Abstract*** The recently proposed "potential-based" reinforcement learning (RL) method made it possible to combine multiple learnings and expert advices as supervised knowledge within an RL framework. The effectiveness of the approach has been established by a theoretical convergence guarantee to an optimal policy. In this paper, the potential-based RL method is applied to a dynamic channel assignment (DCA) problem in a cellular networks. It is empirically shown that the potential-based RL assigns channels more efficiently than fixed channel assignment, Maxavail, and Q-learning-based DCA, and it converges to an optimal policy more rapidly than other RL algorithms, SARSA(0) and PRQ-learning.

**Key words** : Reinforcement Learning, Cellular networks, Channel Assignment Methods

## 1. 서 론

강화 학습(Reinforcement Learning, 이하 RL)[1] 알고리즘은 Markov Decision Process(MDP)[1,2]로 형식화된 순차적 의사결정 문제(sequential decision problem)[2]를 해결한다. RL 알고리즘은 유한한 시간 내에 최적 정책에 수렴하지만, 그 수렴 속도가 느리기 때문에 실제 문제에 RL을 적용하는 데는 어려움이 있다. 최근까지 "감독" 지식(supervised knowledge)을 RL의 과정에 융합하여 수렴 속도를 향상시키려는 연구가 진행되어 있다[3,4].

특히 Chang[4] 은 Ng et al.[5]의 "potential-based reinforcement function"을 이용하여 감독 지식을 융합한 학습 기법(이하 potential-based RL 기법)을 제시하였다. 이는 다수 학습(multiple learning)들과 expert advice들을 감독 지식으로 RL 알고리즘에 융합하여 그 수렴 속도를 향상시키고자 하였고, 이전의 관련 연구들과는 다르게 그 이론적인 최적 정책으로의 수렴성을 확립하였다.

Potential-based RL 기법이 기존의 RL 알고리즘에 비해 수렴 속도에 대한 성능 향상을 보이는지에 대한 실험적인 결과는 제시된 바가 없었다. 본 논문에서는 potential-based RL 기법을 셀룰러 네트워크에서의 채널 할당 문제[6]에 적용한다. Potential-based RL 기반의 채널 할당 기법은 Maxavail, Q-learning-based dynamic channel assignment(DCA)[6]등 기존의 채널 할당 기법들보다 효율적으로 채널을 할당한다. 또한, potential-based RL 기법이 SARSA(0)[1,7]와 Q-learning 등 기존의 강화 학습 알고리즘들에 비해 최적 정책에 더 빠르게 수렴한다는 것을 실험적으로 보인다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 RL 알고리즘에 대한 간략한 소개, 3장에서는 potential RL 기법과 이론적인 수렴성에 대한 설명, 4장에서는 셀룰러

· 이 논문은 2007년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R01-2007-000-10511-0)
· 이 논문은 제34회 추계학술대회에서 '감독 지식을 융합하는 강화 학습 기법을 사용하는 셀룰러 네트워크에서의 동적 채널 할당 기법'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 서강대학교 컴퓨터공학과
inaina21@sogang.ac.kr
†† 정 회 원 : 서강대학교 컴퓨터공학과 교수
hschang@sogang.ac.kr

논문접수 : 2007년 12월 4일
심사완료 : 2008년 4월 5일



정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제5호(2008.7)

네트워크 문제로의 적용을 통한 potential-based RL 기법과 다른 RL 알고리즘들과의 성능 비교 결과 분석, 5장에서는 본 논문의 결론을 제시한다.

## 2. 연구 배경

강화 학습에서 에이전트는 MDP 환경과 상호작용한다. MDP 모델 $M=(X,A,P,R)$이 있다고 하자. $X$는 상태들의 집합, $A$는 행동들의 집합이며 $A(x)$를 상태 $x$에서 선택 가능한 행동의 집합으로 정의한다. $P$는 집합 $\{(x,a)|x\in X,\ a\in A(x)\}$를 집합 $X$에 대하여 정의되는 확률분포들의 집합으로 mapping하는 상태전이함수이다. 상태 $x$에서 어떤 행동 $a\in A(x)$를 선택하여 상태 $y$로 전이할 수 있는 확률을 $P(y|x,a)$라 하자. $R$은 $X\times A\times X$를 실수집합 $\mathbb{R}$로 mapping하는 보상함수이고, 상태 $x$에서 어떤 행동 $a\in A(x)$를 선택하여 상태 $y$로 이동하였을 때의 보상(reward)을 $R(x,a,y)$라 하자.

정책 $\pi$는 $\pi: X\to A$, $\pi(x)\in A(x)$로 정의되며, $\Pi$를 모든 정책들의 집합이라고 하자. 정책 $\pi$에 따라 초기 상태 $x$에서 매 상태마다 행동을 선택했을 때 얻어지는 보상값의 평균을 $V^{\pi}(x)$라 하고 다음과 같이 정의한다 :

$$V^{\pi}(x)=E\left[\sum_{t=0}^{\infty}\gamma^{t}R(X_t,\pi(X_t),X_{t+1})\,|\,X_0=x\right],\ x\in X. \quad (1)$$

$X_t$는 시간 $t$의 상태를 나타내는 random variable이며, $\gamma\in(0,1)$은 수렴을 위한 고정된 discount factor이다. $x\in X$에 대해 $V^{*}(x)=\max_{\pi\in\Pi}V^{\pi}(x)$라고 할 때 Bellman's optimality principle에 의하여 각 $x\in X$에 대해 다음과 같이 주어진다[1]:

$$V^{*}(x)=\max_{a\in A(x)}\left\{\sum_{y\in X}P(y|x,a)\big(R(x,a,y)+\gamma V^{*}(y)\big)\right\}. \quad (2)$$

위 식을 만족하는 $V^{*}(x)$는 오직 하나의 값을 가지며 최적 정책(optimal policy) $\pi^{*}$는 각 $x\in X$에 대해 다음과 같이 주어진다:.

$$\pi^{*}(x)\in\arg\max_{a\in A(x)}\left\{\sum_{y\in X}P(y|x,a)\big(R(x,a,y)+\gamma V^{*}(y)\big)\right\}. \quad (3)$$

함수 $Q^{*}$를 $X\times A$에 대해서 다음과 같이 정의하면,

$$Q^{*}(x,a)=\sum_{y\in X}P(y|x,a)\big(R(x,a,y)+\gamma V^{*}(y)\big),$$
$$x\in X,\ a\in A(x) \quad (4)$$

이 때 $Q^{*}$는 다음의 식을 만족시키고,

$$Q^{*}(x,a)=\sum_{y\in X}P(y|x,a)\big(R(x,a,y)+\gamma\max_{a'\in A(y)}Q^{*}(y,a')\big) \quad (5)$$

최적 정책 $\pi^{*}$는 각 $x\in X$에 대해 다음과 같이 주어진다:

$$\pi^{*}(x)\in\arg\max_{a\in A(x)}Q^{*}(x,a). \quad (6)$$

MDP 모델의 parameter들이 모두 주어져 있을 경우 Value Iteration, Policy Iteration[2] 등의 알고리즘을 사용하여 최적 정책 $\pi^{*}$를 구할 수 있다. 하지만 상태전이 함수 $P$와 보상함수 $R$을 알지 못할 경우, 즉 hidden MDP 모델의 경우 위의 방법들을 사용할 수 없다.

RL 알고리즘은 $Q^{*}$함수의 근사 값을 "학습"함으로써 hidden MDP 모델에서도 최적 정책 $\pi^{*}$를 구할 수 있으며, 그 종류에는 Q-learning, SARSA(0)등이 있다[1,2].

Q-learning은 이산 시간 $t\geq 0$마다 에이전트가 상태 $x_t$에서 학습 전략 $\phi_t^{w}$ 에 따라 행동 $a_t\in A(x_t)$를 선택한다. $\phi_t^{w}$는 시간 $t$의 exploration-exploitation rule(EE rule)[1]이며, $w$는 현재 사용하는 EE rule의 종류를 의미한다. $a_t$를 선택하여 얻어지는 보상 $R(x_t,a_t,x_{t+1})$과 다음 상태 $x_{t+1}$에서의 maximum $Q$ 값을 이용하여 $Q^{*}(x_t,a_t)$의 추정치를 다음에 의해 업데이트(update)한다 :

$$Q_{t+1}(x_t,a_t)\leftarrow Q_t(x_t,a_t)+\beta[R(x_t,a_t,x_{t+1})+\gamma\max_{a\in A(x_{t+1})}Q_t(x_{t+1},a)-Q_t(x_t,a_t)]. \quad (7)$$

$(x,a)\neq(x_t,a_t)$인 모든 $(x,a)$쌍에 대해 $Q_{t+1}(x,a)\leftarrow Q_t(x,a)$로 업데이트한다. $\gamma\in(0,1)$는 discount factor, $\beta\geq 0$는 에이전트의 학습을 $Q$ 함수 값의 업데이트 식에 반영하는 정도를 나타내는 계수이다.

SARSA(0)는 Q-learning과 마찬가지로 에이전트가 상태 $x_t$에서 $\phi_t^{w}$에 따라 행동 $a_t$를 선택함으로써 얻어지는 보상 $R(x_t,a_t,x_{t+1})$과 다음 상태 $x_{t+1}$을 관찰한다. 하지만 다음 상태 $x_{t+1}$에서 EE-rule에 따라 행동 $a_{t+1}$을 선택하되, 실제로 실행하지는 않는다. 이렇게 얻은 $R(x_t,a_t,x_{t+1})$과 $Q_t(x_{t+1},a_{t+1})$ 값을 이용하여 $Q^{*}(x_t,a_t)$ 추정 값을 다음과 같이 업데이트한다 :

$$Q_{t+1}(x_t,a_t)\leftarrow Q_t(x_t,a_t)+\beta[R(x_t,a_t,x_{t+1})+\gamma Q_t(x_{t+1},a_{t+1})-Q_t(x_t,a_t)]. \quad (8)$$

$(x,a)\neq(x_t,a_t)$인 모든 $(x,a)$쌍에 대해 $Q_{t+1}(x,a)\leftarrow Q_t(x,a)$로 업데이트한다. 몇 가지 일반적인 수렴 조건을 만족할 경우 Q-learning과 SARSA(0)는 $t\to\infty$일 때 $Q_t\to Q^{*}$로 수렴한다[1,2]. 본 논문에서의 EE rule $w$로는 $\epsilon_t$-greedy strategy를 사용한다[1,5]. $\epsilon_t$-greedy strategy은 시간 스텝 $t$에서 $1-\epsilon_t$의 확률로 $a_t\in\arg\max_{a\in A(x_t)}Q_t(x_t,a)$인 행동이 선택되며, $\epsilon_t$의 확률로 임의의 행동이 균일하게 선택된다. $\epsilon_t$의 값은 1부터 시작하여 시간이 지날수록 그 값이 감소하도록 설정한다.

## 3. 감독 지식과 강화학습의 융합

MDP $M=(X,A,P,R)$이 주어져 있을 때 $X$에 대한 potential function $\Phi: X\rightarrow\mathbb{R}$에 의하여 $M'=(X,A,P,R')$로 다음과 같이 변환되었다고 하자.

$$R'(x,a,y)=R(x,a,y)+\gamma\Phi(y)-\Phi(x) \tag{9}$$

위의 식에서 정의된 $X\times X$에 대하여 정의된 함수 $F(x,y)=\gamma\Phi(y)-\Phi(x)$를 potential-based reinforcement function이라 한다. Ng *et al.* [5]은 함수 $F$에 의해 변형된 $M'$의 최적 정책이 $M$의 최적 정책과 동일함을 보였다. (9)의 식을 SARSA(0)에 융합하면 $Q$값의 업데이트 공식은 다음과 같이 변형되어질 수 있다[4].

$$Q_{t+1}(x_t, a_t)\leftarrow Q_t(x_t, a_t)+\alpha_t(x_t, a_t)[R(x_t,a_t,x_{t+1}) +\gamma\Phi(x_{t+1})-\Phi(x_t)+\gamma Q_t(x_{t+1}, a_{t+1})-Q_t(x_t, a_t)]. \tag{10}$$

$(x, a)\neq(x_t, a_t)$인 모든 $(x, a)$쌍에 대해서는 $Q_{t+1}(x, a)\leftarrow Q_t(x, a)$로 업데이트한다. 위의 식을 사용하는 변형된 SARSA(0) 역시 몇 가지 일반적인 수렴 조건을 만족할 경우 $t\rightarrow\infty$일 때, $Q_t\rightarrow Q^*$로 수렴한다[4,5]. 이를 이용하여 Chang[4]은 다수 학습들이 각각의 RL 알고리즘들을 통해 학습한 $Q^*$의 추정 값들과 expert들에 의해서 제공되는 확률 분포를 반영하는 $\Phi$를 사용하는 potential-based RL 기법을 제안하였다.

### 3.1 다수 학습(Multiple Learning)

(10)의 SARSA(0) 업데이트 식을 사용하는 기본에이전트와, Q-learning, SARSA($\lambda$)[7] 등을 사용하는 서브에이전트들이 있다고 하자. $m$개의 서브에이전트들이 있고, $t_i$를 서브에이전트 $i$의 시간이라고 할 때 $\Phi$는 다음과 같이 정의된다.

$$\Phi(x_t;t_1,\ldots,t_m) = \sum_{a\in A(x_t)}\left(\frac{1}{m}\sum_{i=1}^{m}Q^i_{t_i}(x_t, a_t)\times\theta(x_t, a;t_1,\ldots,t_m)\right). \tag{11}$$

$Q^i_{t_i}$-함수는 서브에이전트 $i$가 자신의 RL 알고리즘을 사용하여 학습하는 $Q^i$ 함수의 시간 $t_i$에서의 추정 값을 말한다. $\theta(x_t, a;t_1,\ldots,t_m)$는 다음과 같이 주어진다.

$$\theta(x_t, a;t_1,\ldots,t_m) = \frac{\sum_{i=1}^{m}I\left(a\in\arg\max_{b\in A(x_t)}Q^i_{t_i}(x_t, b)\right)}{\sum_{a'\in A(x_t)}\sum_{i=1}^{m}I\left(a'\in\arg\max_{b\in A(x_t)}Q^i_{t_i}(x_t, b)\right)}. \tag{12}$$

함수 $\Phi$는 $m$개의 서브에이전트들이 MDP $M$에 대하여 학습한 각각의 $Q^*$ 추정 값의 가중 평균(weighted average)들의 $a\in A(x)$, $x\in X$에 대한 합이며, 이는 MDP $M$에 대한 $V^*$의 추정 값이기도 하다.

### 3.2 Expert Advice

$m$개의 서브에이전트 이외에도 $k=1,\ldots,l$ 개의 expert들이 있다고 하고, 각각의 expert들은 각 상태(state)에서 행동(action)들의 집합 $A$에 대한 확률 분포의 형태로 기본 에이전트에 expert advice들을 제시한다고 하자. Expert는 CBR(Cased Based Reasoning) [8], model-based/model free RL[1,2] 등이 될 수 있으며, 이 때 함수 $\theta$는 다음과 같이 확장된다.

$$\theta(x_t, a;t_1, \ldots, t_m, s_1,\ldots,s_l) = \theta(x_t, a;t_1, \ldots, t_m)\times\frac{\sum_{i=1}^{l}\rho^k_{s_k}(x_t, a)}{\sum_{a'\in A(x_t)}\sum_{i=1}^{l}\rho^k_{s_k}(x_t, a')}. \tag{13}$$

$\rho^k_{s_k}(x_t,a)$는 전문가 $s_k$가 상태 $x_t$에서 행동 $a$를 택할 확률이다. (13)의 $\theta$를 사용한 $\Phi$는 $l$개의 expert들의 정보를 강화 학습 과정에 반영하게 된다.

## 4. Channel Allocation in Cellular Networks

### 4.1 문제 정의

셀마다 균일한 트래픽을 갖는 49개의 셀들로 이루어진 셀룰러 망이 있다고 하자(그림 1). 각 셀의 call 발생률은 평균 arrival rate $\lambda$의 Poisson 분포를 따르며, call이 발생한 셀에 채널이 할당되면 exponential 분포를 갖는 call holding time $\tau_{holding}$ 뒤에 channel은 release된다. 셀 $i$에 call arrival이 발생하였다고 하자. 채널 간 간섭 거리를 $D$라고 할 때 그림 1에서 색이 칠해진 부분은 셀 $i$와 간섭 거리 $D$이내에 놓인 셀들을 의미한다. Cell $i$에 채널 $k$를 할당하면, $i$로부터 간섭 거리 이내의 셀들은 채널 $k$를 사용할 수 없다. 이 조건을 만족하면서 cell $i$와 최단거리에 놓인 cell들을 $i$의 compact cell이라 한다[6]. Q-learning-based dynamic channel allocation (DCA)[6]에서는 에이전트로 하여금 같은 채널을 사용하는 compact cell의 수가 증가하도록

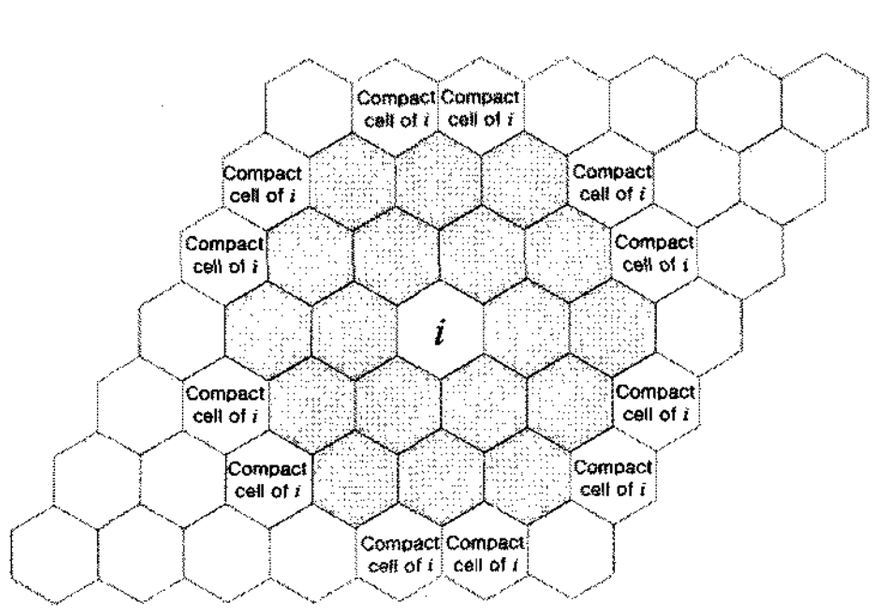


그림 1 Cellular Networks

학습함으로써 채널 할당 효율을 높였다. DCA를 MDP $M=(X,A,P,C)$에 대해서 정의하면 아래와 같으며 이는 [6]의 문제 정의를 따른 것이다. Reward 함수 $R$ 대신 cost 함수 $C$를 사용한 이유는 이 문제가 시간에 따른 cost들의 합에 대한 기댓값을 최소화시키는 문제이기 때문이다.

$N$개의 셀들과 $M$개의 채널들이 있다고 하자. 상태 $x \in X$는 다음과 같이 정의된다 :

$$x=(i,avail(i)). \tag{14}$$

여기서 $i \in \{1,2,...,N\}$는 call이 발생한 셀의 번호를 나타내며, $avail(i) \in \{1,2,...,M\}$은 셀 $i$에 할당 가능한 채널의 수를 나타낸다. $avail(i)$의 값은 셀 $i$에서 사용하고 있는 채널들과 셀 $i$와 간섭 거리 $D$ 이내에 있는 셀들의 집합인 $I(i)$에 의해 결정된다[6]. 상태 $x \in X$에서 $a \in A(x)$인 행동 $a$는 다음과 같이 정의된다 :

$a=k$ , $k \in \{1,2,...,M\}$, $k$는 $i$에 할당 가능한 채널.

$C(x,a,y)$를 상태 $x$에서 행동 $a \in A(x)$를 선택하여 상태 $y$로 transition하면서 발생하는 cost라고 하자. $x, y \in X$일 때 이는 다음과 같이 정의된다 :

$$C(x,a,y)=n_1(a)r_1+n_2(a)r_2+n_3(a)r_3 . \tag{15}$$

여기서 $x=(i,avail(i))$, $a=k$일 때 $n_1(a)$는 채널 $k$를 사용하는 셀 $i$의 compact 셀들 중 같은 채널 $k$를 사용하는 셀들의 수이며, $n_2(a)$는 셀 $i$와 2-tier 거리에 있으면서 compact 셀이 아닌 셀들 중 채널 $k$를 사용하는 셀들의 수를 뜻한다. $n_3(a)$는 그 외의 셀들 중 채널 $k$를 사용하는 셀들의 수이다. 이 식에 의하면, 같은 채널을 사용하는 compact 셀들의 수가 많을수록, 낮은 cost를 받게 된다($r_1 < r_2 < r_3$).

### 4.2 실험 환경 및 실험 결과

7×7 개의 육각형의 cell들로 구성된, 70개의 채널을 가진 셀룰러 네트워크가 있다고 하자($N=49$, $M=70$). 본 실험에서 사용되는 parameter들은 다음과 같이 정의된다. Call arrival 확률은 평균 arrival rate $\lambda$의 Poisson 분포를 따르며, $\lambda$는 20~250 calls/h의 값을 갖는다. $\tau_{holding}$은 평균 call-duration $1/\mu$의 exponential 분포를 따르며, 여기서는 모든 call에 대해 $1/\mu=180$초로 정의한다. Cell $i$의 traffic load $\omega_i$는 다음과 같이 정의된다.

$$\omega_i=\frac{\lambda}{\mu} . \tag{16}$$

(15)의 cost function $C$에서 $r_1$, $r_2$, $r_3$의 값은 각각 -5, -1, 1을 사용하였다. 즉 같은 채널을 사용하는 compact cell의 수는 cost 값과 반비례한다.

성능 척도로 사용되는 New call blocking probability는 다음과 같이 정의된다 :

$$P=\frac{number\ of\ blocked\ calls}{number\ of\ new\ call\ arrivals}. \tag{17}$$

다른 성능 척도로는 RL 알고리즘들끼리의 성능 비교를 위해서 다음과 같은 sample average total reward $E^{\pi}$가 사용된다.

$$E^{\pi}=\frac{1}{u}\sum_{v=1}^{u}\left[\sum_{t=0}^{k}\gamma^t R\left(x_t^v,\pi(x_t^v),x_{t+1}^v\right)\right],\ x_0^v=x,\ x\in X \tag{18}$$

$u$은 실험 회수, $k$는 전체 실험 시간, $\pi$는 각 강화 학습 기법의 정책, $\gamma$는 discount factor를 뜻하며, $x_t^v$는 $v$번째 실험에서 시간 스텝 $t$에서의 상태를 의미한다. 이 실험에서는 $u=5$, $\gamma=0.99995$를 사용하였다.

위 실험 환경에서 fixed channel allocation(FCA), Maxavail[9], Q-learning-based DCA, SARSA(0)-based DCA, potential-based RL을 적용한 DCA의 blocking probability를 측정하여 각각의 성능을 비교하였다. FCA는 각 셀마다 10개의 고정된 채널만을 사용하며, Maxavail은 cell $i$에 call이 발생했을 때, $avail(i)$개의 할당 가능한 채널들 중에서 모든 셀의 할당 가능한 채널의 수의 합, 즉 전체 셀의 개수가 49개일 경우 $\sum_{i=1}^{49} avail(i)$ 를 최대로 만드는 채널을 선택하여 할당한다. Potential-based RL 기법은 SARSA(0)을 사용하는 기본 에이전트와 Q-learning을 사용하는 1개의 서브에이전트, Maxavail을 사용하는 1개의 expert로 이루어져 있다. Potential-based RL 기법과 SARSA(0), Q-learning의 EE-rule로는 $\epsilon_t$-greedy strategy를 사용하였으며, $\epsilon_t$의 값은 다음과 같이 설정한다 :

$$\epsilon_t=\frac{q}{n_t(x_t)},\ q\in(0,1),\ x_t\in X. \tag{19}$$

여기서 $n_t(x_t)$는 $t$ 시간 스텝까지 에이전트가 상태 $x_t$에 방문한 횟수를 말한다.

이 실험에서는 simulation 시간을 사용하며, 1초의 simulation 시간마다 call arrival rate $\lambda$에 따른 확률로 각 셀에 call이 발생한다. 성능 측정은 다음과 같이 이루어졌다. FCA와 Maxavail은 5 simulation 시간동안 각각의 채널 할당 기법을 수행한 뒤 blocking probability를 측정하였고, Q-learning-based DCA와 SARSA(0)-based DCA, 그리고 potential-based RL을 적용한 DCA는 15 simulation 시간동안 각각의 정책을 학습한 뒤, 해당 정책을 사용하여 5 simulation 시간동안 채널 할당을 수행한 뒤 blocking probability를 측정하였다. 그림 2의 그래프에서 $x$축은 traffic load $w_i$를 나타내며, $w_i$의 값이 5에서 10으로 늘어남에 따라 call arrival rate $\lambda$는 100에서 200으로 늘어난다. 그림 2와 표 1의 결과에서 확인할 수 있듯이, potential-based

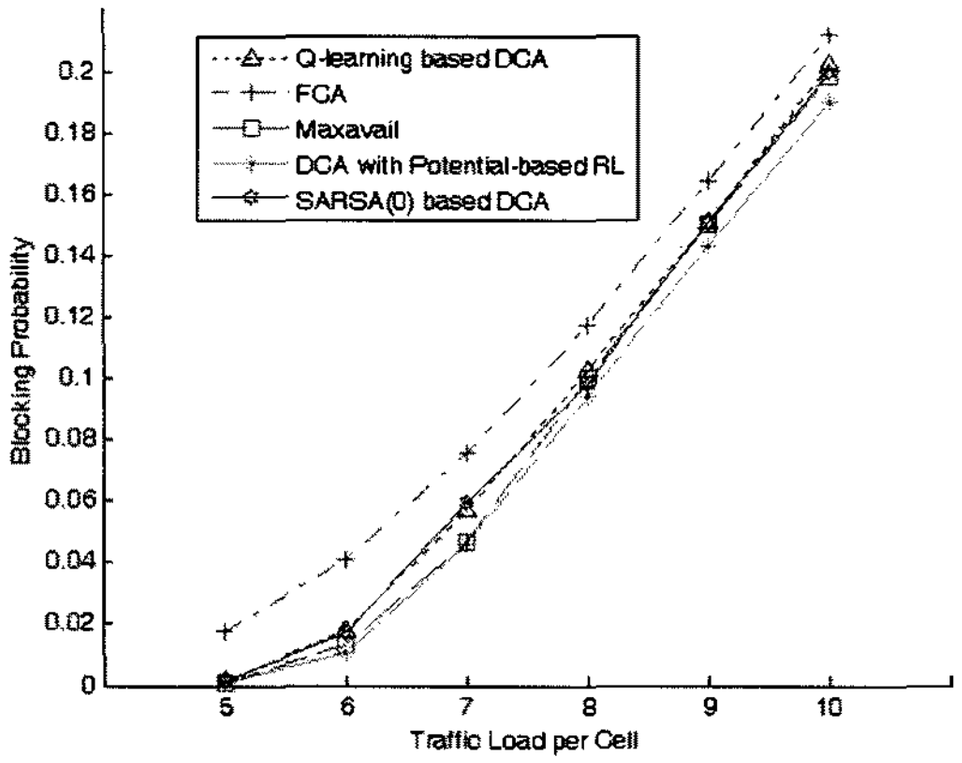


그림 2 셀 당 traffic load에 따른 blocking probability 비교

표 1 셀 당 traffic load에 따른 blocking probability 비교

| Traffic Load per Cell | Q-learning | FCA | Maxavail | Potential-Based RL | SARSA(0) |
|---|---|---|---|---|---|
| 5 | 0.139% | 1.750% | 0.086% | 0.102% | 0.192% |
| 6 | 1.766% | 4.101% | 1.344% | 1.090% | 1.735% |
| 7 | 5.708% | 7.570% | 4.673% | 4.590% | 5.935% |
| 8 | 10.235% | 11.724% | 10.004% | 9.433% | 9.830% |
| 9 | 15.076% | 16.409% | 15.014% | 14.303% | 15.086% |
| 10 | 20.216% | 21.196% | 19.824% | 18.983% | 19.974% |

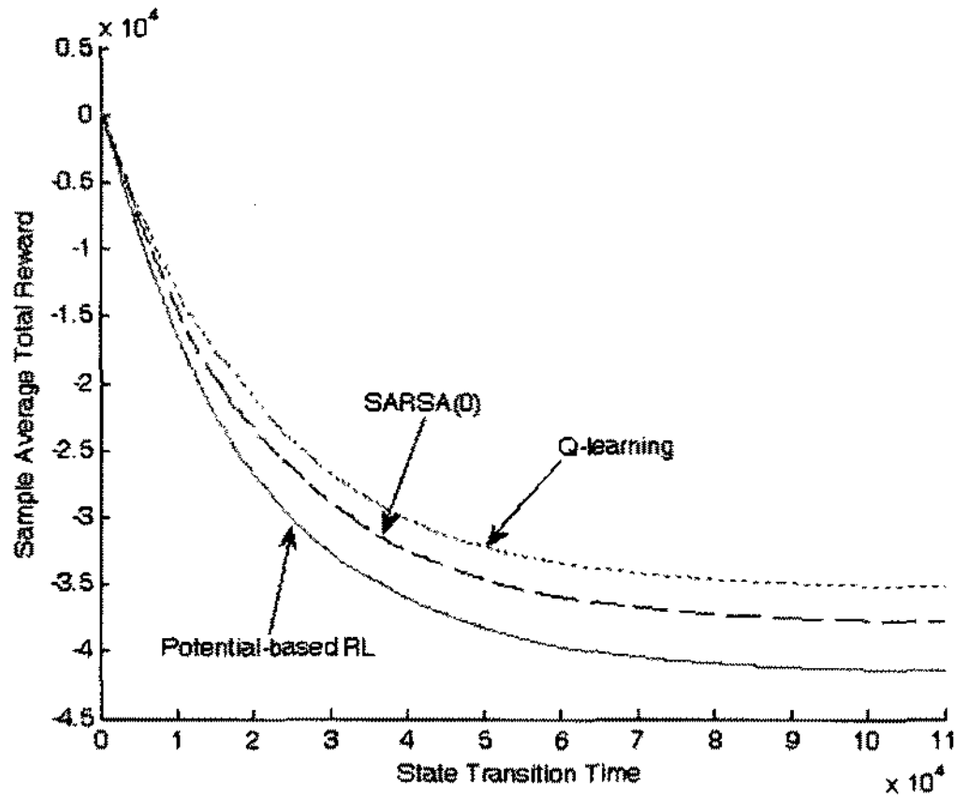


그림 3 Optimal Policy로의 수렴 과정 비교

RL을 사용한 DCA의 경우 다른 4개의 채널 할당 기법보다 낮은 blocking probability를 보임을 알 수 있다.

그림 3은 Potential-based RL 기법을 사용한 DCA와 Q-learning-based DCA, SARSA(0)-based DCA의 수렴 과정을 비교한 그래프이다. $11\times10^4$ 번의 상태 전이를 거친 결과 potential-based RL 기법이 Q-learning에 비해서 낮은 expected total reward값을 보이며, 이는 potential-based RL 기법이 최적 정책에 Q-learning과 SARSA(0)보다 더 빠르게 수렴한다는 것을 뜻한다.

## 5. 결 론

지금까지 감독 지식을 융합하는 강화 학습 기법인 potential-based RL 기법에 대하여 알아보고, 이를 셀룰러 네트워크에서의 채널 할당 문제에 적용하였다. Potential-based RL 기법을 사용하는 동적 채널 할당 기법의 성능을 기존의 채널 할당 기법들과 비교한 결과, Maxvail을 expert로, Q-learning을 다수 학습으로 사용한 potential-based RL 기반의 DCA가 가장 효율적으로 채널을 할당하였다. 또한, RL 알고리즘들 간의 expected total reward들의 비교를 통해 potential-based RL 기법이 기존의 강화 학습 기법의 성능을 향상시켰음을 확인하였다..

## 참 고 문 헌

[1] R. Sutton and A. Barto, *Reinforcement Learning*. MIT Press, 2000.

[2] M. L. Littman. Algorithms for sequential decision making. Unpublished Ph.D. Thesis, Brown University, Providence, R.I. 1996.

[3] M. N. ahmadabadi and M. Asadpour, "Expertness based cooperative Q-learning," *IEEE Trans. on Systems, Man, and Cybernetics*, part B, Vol.32, No.1, pp. 66-76, 2002.

[4] H. S. Chang, "Reinforcement Learning with Supervision by Combining Multiple Learnings and Expert Advices," in Proc. of the 2006 American Control Conference, pp. 4159-4164, 2006.

[5] A. Y. Ng, D. Harada, and S. Russel. "Policy invariance under reward transformations: theory and application to reward shaping," in Proc. of the 16th Int. Conf. on Machine Learning, pp. 278-287, 1999.

[6] Junhong Nie; Haykin, S., "A dynamic channel assignment policy through Q-learning," IEEE Trans. on Neural Networks, Vol.10, No.6, pp. 1443-1455, 1999.

[7] Singh, S., Jaakkola, T., Littman, M. L., Szepesv'ari, C, "Convergence results for single-step on-policy reinforcement-learning algorithms," Journal of Machine Learning, Vol.38, No.3, pp. 287-308, 2000.

[8] T. Mitchell, Machine Learning, McGraw Hill, 1989.

[9] Tekinay, S.; Jabbari, B., "Handover and channel assignment in mobile cellular networks," Communications Magazine, IEEE, Vol.29, No.11, pp. 42-46, 1991.