# Dynamic Channel Assignment in Cellular Networks: A Reinforcement Learning Solution

Sidi-Mohammed Senouci, and Guy Pujolle

Laboratoire LIP6
Université de Paris VI
8, rue du Capitaine Scott
75015 Paris – France
Sidi-Mohammed.Senouci@lip6.fr
Guy.Pujolle@lip6.fr

*Abstract-* **The optimization of channel assignment in cellular networks is a very complex optimization problem and it becomes more difficult when the network handles different classes of traffic. The objective is that channel utility be maximized so as to maximize service in a stochastic caller environment. We address in this paper, the dynamic channel assignment (DCA) combined with call admission control (CAC) problem in a multimedia cellular network that handles several classes of traffic with different resource requirements. The problem is naturally formulated as a Semi-Markov Decision Process (SMDP) problem and we use an approach based on reinforcement learning (RL) [neuro-dynamic programming (NDP)] method to solving it. We show that the policy obtained using our Q-DCA algorithm provides a good solution and is able to earn significantly higher revenues than classical solutions. A broad set of experiments illustrates the robustness of our policy that improves the Quality of Service (QoS) and reduces call-blocking probabilities for handoff calls in spite of variations in the traffic conditions.**

## 1 Introduction

Technological advances and rapid development of handheld wireless terminals have facilitated the rapid growth of wireless communications and mobile computing. The tremendous growth of the wireless/mobile user population, coupled with the bandwidth requirements of multimedia applications, requires efficient reuse of the scarce radio spectrum allocated to wireless/mobile communications.

The total system bandwidth is divided into channels[1], with each channel centered around a frequency and the most important problem is to allocate these channels so as to maximize the service provided to a set of mobile callers. The assignment of this bandwidth fall into two categories: Fixed Channel Allocation (FCA), where each cell has a fixed number of channels, and dynamic channel allocation (DCA) where channels are dynamically assigned to cells. In FCA, the set of channels is partitioned according to some reuse pattern, and the partitions are permanently assigned to cells. When a call arrives in a cell, if any pre-assigned channel is unused; it is assigned, else the call is blocked. Such policies are very simple, however, they do not adapt to changing traffic conditions and user distribution. More efficient are DCA policies, where all channels are placed in a pool and are assigned to new calls as needed such that the carrier-to-interference ratio (CIR) criterion is satisfied. At the cost of higher complexity, DCA schemes provide flexibility and traffic adaptability.

In both FCA and DCA systems, when a mobile caller crosses from one cell to another, he needs to be allocated a new channel (one that does not violate the channel reuse[2] constraint) in the destination cell. This event (handoff) must be transparent to the user. If no such channel is available, the call must be dropped/disconnected from the system. One objective of a channel allocation policy is to minimize the number of calls that are dropped when they are handed off to a busy cell, since dropping existing calls is generally more undesirable than blocking new calls.

In [1] the authors provide an overview of different channel assignment algorithms and compare them in terms of performance, flexibility, and complexity. One of the best existing dynamic channel allocation strategies we found in the literature belongs to a class of algorithms called exhaustive searching DCA [2,3,4-7]. In these algorithms, a cost (reward) is associated with each available channel. When a new call arrives, the system searches exhaustively for the channel with minimum cost (maximum reward) and then that channel is assigned to the call. Some criteria including maximum availability, maximum interferers, and minimum damage have been used.

This paper proposes an alternative approach to solve dynamic channel assignment and call admission control problems in multimedia cellular networks. The optimal policy is obtained using a form of reinforcement learning (RL) algorithm known as Q-learning [8,9]. One of the most

[1] Channels could be frequencies, time slots or codes depending on the radio access technique used.

[2] A channel can be associated with many cells as long as the co-channel interference constraint is satisfied.

0-7803-7661-7/03/$17.00©2003 IEEE

significant and actively investigated RL algorithms is Q-learning. It has the nice property that it does not need a model of the environment, and the system is designed to learn an optimal policy by directly interacting with the environment. Our method learn a policy that outperforms the most commonly used policies in cellular systems. It is able to reduce the blocking probability for handoff calls and, also, able to generate higher revenues.

We consider a system with two classes of traffic. Our objective is to assign the best available channel to the customer so as to maximize the expected value of the rewards received over an infinite planning horizon. In such context (multi-class traffic framework) it is sometimes preferable to block a call of a less valuable class and to accept another call of a more valuable class. By making the assumptions of Poisson arrivals and a common exponential service time, this problem can be formulated as an SMDP (Semi-Markov Decision Process) and learning is a solution for this problem.

The remainder of this paper is organized as follows. After a brief description of the Q-Learning strategy and the formulation of the channel assignment problem as an SMDP in section 2, we detail the Q-learning implementation that solves this SMDP in section 3. Performance evaluation and numerical results are exposed in section 4. Finally, section 5 summarizes the main contributions of this work.

## 2 Problem definition

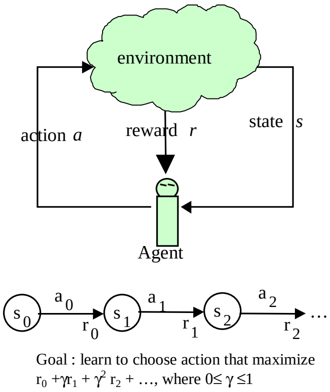
We propose an alternative approach to solving the call admission control and dynamic channel assignment problems. This approach is based on the judgment that DCA and CAC can be regarded as an SMDP, and learning is one of the effective ways to find a solution to this problem. A particular learning paradigm has been adopted, known as neuro-dynamic programming (NDP) [reinforcement learning (RL)]. In NDP, as shown in Fig. 1, an agent aims to learn an optimal control policy by repeatedly interacting with the controlled environment in such a way that its performance, evaluated by the sum of rewards (payoff) obtained from the environment, is maximized. There exist a variety of RL algorithms. A particular algorithm that appears to be suitable for these two tasks is called Q-learning. In what follows, we briefly describe this algorithm, and then present the details of how the CAC and DCA problems can be solved by means of Q-learning.

### 2.1 Q-learning Strategy

The agent, the environment it interacts with, and the task it has to achieve are the components that define the reinforcement-learning framework (cf. Fig. 1). The interaction between the agent and the environment is continuous. On one hand the agent's decision process selects actions according to the perceived situations of the environment, and on the other hand these situations evolve under the influence of the actions. Each time the agent performs an action, it receives a reward. A reward is a scalar value that tells the agent how well it is fulfilling the given task. To be formal let's denote *s* ($\in$ *S*, a finite set), a representation of the environment's state as it is perceived by the agent, *a* ($\in$ *A*, a finite set) the selected action, and *r* ($\in$ *R*, a finite set) the received reward. The agent's decision process is called policy and is a mapping from states to actions ($\pi$: $S \to A$). The interaction between the agent and the environment is continuous and a learning agent modifies its policy according to its experience and to its goal which is to maximize the cumulated rewards over time $V^{\pi}(s_t)$ defined as follows:

$$V^{\pi}(s_t) = \sum_{i=0}^{\infty} \gamma^i r_i \tag{1}$$

where $0 \le \gamma \le 1$ is a discount factor.



Goal : learn to choose action that maximize
$r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$, where $0 \le \gamma \le 1$

Fig. 1. The Agent-environment interaction.

A function *Q(s,a)* is used to memorize the expected reward for the action *a* and state *s*. The *action-value* function *Q* can be represented either by a look-up table or a function approximator (neural network, regression tree, etc.). On each step of interaction, and in the case of a look-up table representation, the action-value function is updated with equation (2):

$$Q_{t+1}(s,a) = \begin{cases} Q_t(s,a) + \alpha_t \Delta Q_t(s,a), & if\ s = s_t and\ a = a_t \\ Q_t(s,a), & otherwise \end{cases} \tag{2}$$

Where

$$\Delta Q_t(s,a) = \left\{ r_t + \gamma \max_b \left[ Q_t(s'_t, b) \right] \right\} - Q_t(s,a) \tag{3}$$

and $\alpha_t$ is the learning rate.

It has been shown in [9] that if the *Q*-value of each admissible *(s,a)* pair is visited infinitely often, and if the learning rate is decreased to zero in a suitable way, then as $t \to \infty$, $Q_t(s,a)$ converges to $Q^*(s,a)$ with probability 1. The optimal policy $\pi^*(s)$ is the one with the maximum Q-value: $\pi^*(s) = \arg\max_{a \in A(s)} Q^*(s,a)$.

We consider two classes of traffic, but the ideas in this paper can be extended easily to several classes of traffic as well. This cellular system can be considered as a discrete event system. The major events that may occur in a cell include new and handoff calls arrivals and call departures for the two traffic classes. These events are modeled as stochastic variables with appropriate probability distributions. In particular, new call arrivals in a cell obey a Poisson distribution. Call holding time is assumed to be exponentially distributed (cf. Table 2).

### 2.2 Learning Dynamic Channel Assignment and Call Admission Control

This work is an extension of a previous work of *Nie and al.* [10]. We consider in this paper not only channel assignment task but also the call admission control problem in a cellular network. We consider a DCA system handling not only one class of traffic as in [10] but two classes of traffic ($C_1$ and $C_2$) with *N* cells and *M* available channels kept in a common pool. Any channel can be temporarily allocated to any cell, provided that the constraint on the reuse distance is fulfilled. We develop, in this section, the dynamic programming formulation suitable for this problem.

Calls arrive and leave over time and the network can choose to accept or reject connection requests. If the call is accepted, the system assigns to it one of the available channels. The goal of the network operator is to find a DCA policy that maximizes the long-term revenue/utility and reduces handoff blocking probabilities (Contrary to [10] who does not give any priority to handoff calls neither in its simulation nor in the NDP formulation).

The experimental parameters are shown in Table 2. We identify the system states *s*, the actions *a* and the associated rewards *r* as follows:

1) **States:** We define the state $s=(i,A(i),x,e)$ as:
   - $i\in\{1,\dots,N\}$ is the cell index specifying there is an event *e* occurring in cell *i*.
   - $A(i)\in\{1, 2,\dots,M\}$ is the number of available channels in cell *i*, which depends on the channel usage conditions in this cell and in its interfering cells[3] $I(i)$.
     To obtain $A(i)$, we define an *M*-dimensional availability vector $u_q$ for cell *q*, $q= 1,2,\dots,N$ as:
     $$u_{qk} = \begin{cases} 0, & \textit{if channel } k \textit{ available for use in cell } q \\ n, & \textit{otherwise} \end{cases}$$
     where n>0.
     By using $u_{ik}$, $A(i)$ can be obtained from
     $$A(i) = \sum_{k=1}^{M} \bar{u}_{ik}$$
     where $\bar{u}_{ik} = \begin{cases} 1, & \textit{if } u_{ik} = 0 \\ 0, & \textit{otherwise} \end{cases}$
   - $x=(x_1, x_2)$ where $x_1$ and $x_2$ are the number of calls of each class of traffic ($C_1$ and $C_2$ respectively) in cell *i*.
   - $e=\{1,2,3,4\}$ where

$$e = \begin{cases} 1, & \textit{arrival of a new call of class } C_1 \textit{ in cell } i \\ 2, & \textit{arrival of a new call of class } C_2 \textit{ in cell } i \\ 3, & \textit{arrival of a handoff call of class } C_1 \textit{ in cell } i \\ 4, & \textit{arrival of a handoff call of class } C_2 \textit{ in cell } i \end{cases} \qquad (4)$$

We do not take into account the states associated with a call departure. The reason for this simplification is that call departure is not a decision point for the admission controller, and therefore no action needs to be taken.

2) **Actions:** We combine the notions of call admission control and channel allocation. Thus, applying an action is to reject the current call request call in cell *i*, or to assign a channel from the $A(i)$ available channels to it. So, the possible actions are defined as $A=\{0,1,2,\dots,M\}$ where
   $$a_i = \begin{cases} k, & \textit{accept the call and assign channel } k \textit{ to it} \\ 0, & \textit{reject the call} \end{cases}$$
   where $k = 1, 2, \dots, M$ and $u_{ik} = 0$.

3) **Rewards:** The reward $r(s, a)$ represents the cost of choosing the action *a* in the state *s*.

$$r(s,a) = \begin{cases} (n_1(a)r_1 + n_2(a)r_2 + n_3(a)r_3)\times\eta_i, & \textit{if } a = 1,\dots,M \textit{ and } e = e_i \\ 0, & \textit{if } a = 0 \end{cases} \qquad (5)$$

When there is a call arrival in cell *i*, the reward parameter will be equal to zero when the action is to reject the call ($a$=0) and it represents the cost of choosing channel *a* to serve this call when it is accepted ($a\neq0$). There are many possibilities to define *r*. Here, we consider the type of the call and as in [10] the usage conditions in cochannel[4] cells associated with cell *i*. In the above equation, $n_1(k)$ is the number of compact cells in reference to cell *i* in which channel *k* is being used. Compact cells are the cells with minimum averaging distance between cochannel cells [7]. In the case of a regular hexagonal layout shown in Fig. 6, compact cells are located on the third tier with three cells apart; $n_2(k)$ is the number of cochannel cells which are located on the third tier, but not compact cells in which channel *k* is being used; $n_3(k)$ is the number of other cochannel cells currently using channel k; and $r_1$, $r_2$, *and* $r_3$ are constant associated with the above-mentioned conditions related to $n_1(k)$, $n_2(k)$, and $n_3(k)$ respectively. To obtain $n_1(k)$, $n_2(k)$, and $n_3(k)$ at time *t*, we define the an *M*-dimensional channel status vector for each cell *q*, $q= 1,2,\dots,N$ as:

$$s_{qk} = \begin{cases} 1, & \textit{if channel } k \textit{ is in use in cell } q \\ 0, & \textit{otherwise} \end{cases}$$

The parameter $\eta_i$ represents a constant associated with the type of the current call (new call or handoff

[3] The set of neighborhood cells that lie at a distance less than a reuse distance *D*.

[4] Cells using the same channel without causing interference.

call), and it is defined in Table 1. To prioritize handoff calls, larger reward values have been chosen for handoff calls. We also suppose that $C_1$ calls are more important than $C_2$ calls.

| $\eta_1$ | $\eta_2$ | $\eta_3$ | $\eta_4$ |
|---|---|---|---|
| 5 | 1 | 50 | 10 |

*Table 1.* The reward parameter $\eta_i$.

In summary, we choose the state descriptor to be $s = (i, A(i), (x_1, x_2), e)$, where *A(i)* is the number of available channels in cell *i*; $x_k$ is the number of calls of class $C_k$ in progress, and $e \in \{1,2,3,4\}$ stands for a new or handoff call arrival. When an event occurs, the agent has to choose a feasible action for that event. The action set is $A(s)=\{0=\text{reject}\}\cup\{1,\ldots,M\}$ upon a call arrival. Call terminations are not decision points, so no action needs to be taken. The agent has to determine a policy for accepting and choosing the most appropriate channel for calls given *s*, which maximizes the long-run average revenue, over an infinite horizon. The system constitutes an SMDP with a finite state space $S = \{(i, A(i), x, e)\}$ and a finite action space $A=\{0,1,\ldots,M\}$.

## 3 Algorithm Implementation

After the specification of the SMDP (states, actions, and rewards) associated with the channel assignment problem, let us describe the online implementation of the Q-learning algorithm for solving it. Here, an important issue arises as to how to store the values of the *Q*-function.

### 3.1 Q-values representation

A number of powerful convergence proofs have been given showing that Q-learning is guaranteed to converge with probability 1, in cases where the state space is small enough so that look-up table representation can be used. Furthermore, the major difficulty with SMDP problems is the curse of dimensionality (the exponential state space explosion with the problem dimension). Clearly, when the number of state-action pairs becomes large, look-up table representation will be infeasible, and a compact representation where *Q* is represented as a function of a smaller set of parameters using a function approximator in necessary (state aggregation [11,12], neural networks [13], regression trees [14]). In a previous work [15] we used a neural network but in this paper we choose state aggregation approximation architecture defined below.

### 3.2 Implementation

We note that the only interesting states in which decisions need to be made are those associated with call arrivals. So, we avoid the updates of Q-values at departure states. This will reduce the amount of computation and storage of Q-values significantly.

The flowchart of *Fig.*2 summarizes the procedures involved in the Q-DCA[5] algorithm. When there is a call arrival (new or handoff call), the algorithm first determines if accepting this call will violate QoS. If this case, the call is rejected; else the action is chosen according to

$$a = \arg\max_{a\in A(s)} Q^*(s,a) \qquad (6)$$

Where $A(s)=\{0=\text{reject}, 1,2,\ldots,M\}$.

In particular, (6) implies the following procedures. When a call arrives, the Q-value of rejecting the currently concerned call attempt and the Q-value of accepting and choosing channel *a* to serve this call are determined from the lookup table. If rejection has the higher value, the call is dropped. Otherwise, if acceptance has the higher value, the call is accepted and channel *a* is assigned to it.

To learn the optimal Q-values *Q*(s,a)*, the value function is updated at each transition from state *s* to *s'* under action *a* using (2). The parameters in cost evaluation of (5) were $r_1$ = +5, $r_2$ = +1, and $r_3$ = -1. Such a setting would result in a situation in which the channels being used in the compact cells (associated with $r_1$) have maximum Q-values and thus become the most favorable candidates to be chosen.

We consider a mobile communication system consisting of *N* = 36 hexagonal cells with *M* = 70 channels available in a pool. With the reuse distance $D = \sqrt{21}R$ (*R* is the cell radius), it turns out that if a channel is allocated to cell *i*, it cannot be reused in two tiers of adjacent cells with *i* because of unacceptable cochannel interference levels. Thus, there are at most 18 interfering cells for a specified reference cell. The discount factor is chosen to be $\gamma = 0.5$. Training runs typically used a fixed learning rate $\alpha$ = 0.1, which seemed to give results even though convergence theorems require decreasing $\alpha$ with time.

### 3.3 Exploration

Basically, the convergence theorem of Q-learning requires that all state-action pairs (*s*,*a*) are tried infinitely. To overcome the slow convergence, during the training period, when there are more than one feasible action, the control action is chosen, not according to (6), but according to a *Boltzman* distribution [16]. The idea is to start with high exploration and decrease it to nothing as time goes on, so that after a while we are only exploring (*s*,*a*)'s that have worked out at least moderately well before.

[5] Q-DCA-the online implementation of the Q-learning algorithm for solving the channel assignment and CAC problems in a DCA system.

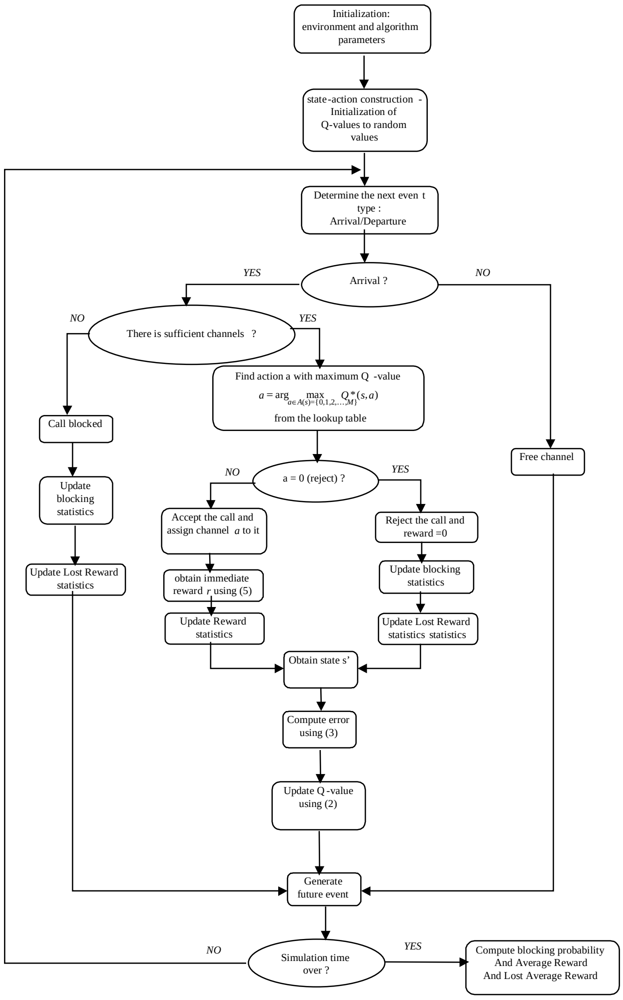

Initialization: environment and algorithm parameters
state-action construction - Initialization of Q-values to random values
Determine the next even t type : Arrival/Departure
Arrival ?
YES
NO
There is sufficient channels ?
NO
YES
Find action a with maximum Q -value
$a = \arg \max_{a \in A(s)=\{0,1,2,\ldots,M\}} Q^*(s,a)$
from the lookup table
Call blocked
Update blocking statistics
Update Lost Reward statistics
a = 0 (reject) ?
NO
YES
Accept the call and assign channel a to it
obtain immediate reward r using (5)
Update Reward statistics
Reject the call and reward =0
Update blocking statistics
Update Lost Reward statistics statistics
Free channel
Obtain state s'
Compute error using (3)
Update Q -value using (2)
Generate future event
Simulation time over ?
NO
YES
Compute blocking probability And Average Reward And Lost Average Reward

*Fig. 2.* Q-DCA algorithm.

## 4 Experimental results

In order to evaluate the performance of our solution, we apply a test data set to compare it with the greedy-DCA[6] and the DCA algorithm presented in [10] that we call *DCA-Nie*. The performance of the algorithms has been evaluated on the basis of the total rewards of the accepted calls (*Total rewards*), the total rewards of the rejected calls (*Total Lost Rewards*), and by measuring the handoff blocking probability.

The parameters used in the training period are given in *Table* 2.

| | Source Type | |
|---|---|---|
| | $C_1$ | $C_2$ |
| Call duration ($1/\mu_C$) | 180 s | 180 s |
| Sojourn time in a cell ($1/\mu_H$) | 60 s | 60 s |
| Call arrival rate | $\lambda_1 = 120\ calls/hour$ | $\lambda_2 = \lambda_1/2 = 60\ calls/hour$ |

*Table 2.* Experimental Parameters

A set of simulations was carried out, including the cases of uniform distribution, non-uniform distribution, time-varying traffic load, and equipment failure. The experimental results are shown in *Fig.* 3 through *Fig.* 13. The results show that the reinforcement learning is a good solution for channel assignment and call admission control problems. The proposed algorithm, Q-DCA, is considerably powerful compared to the greedy policy. In all cases the lost reward due to rejection of customers and blocking probability of handoff calls are significantly reduced. The total rewards due to acceptance of customers are also significantly increased.

### 4.1 Uniform Distribution

Our first set of experiments involved a constant traffic load for the different classes of traffic among all 36 cells. In this case we used the policy learned in the training period but with eight different traffic load conditions among all 36 cells as shown in *Table* 3.

| | Sources Type | |
|---|---|---|
| | $C_1$ | $C_2$ |
| Traffic Load (calls/hour) | 60 | 30 |
| | 80 | 40 |
| | 100 | 50 |
| | 120 | 60 |
| | 140 | 70 |
| | 160 | 80 |
| | 180 | 90 |
| | 200 | 100 |

*Table 3.* Experimental Parameters

From the results shown in Fig. 3, we see that the handoff blocking probability decreases significantly using Q-learning compared to the greedy policy for all the traffic loads considered in *Table* 3 and especially when the traffic load is heavy. The handoff blocking probability metric is given by

$$P_{HO} = \frac{number\ of\ \ handoff\ \ calls\ blocked\ in\ the\ system}{number\ of\ \ hndoffs\ \ in\ the\ system} \tag{7}$$

*Fig.* 4(a) shows the total rewards of using learning computed over one simulated hour with the eight different traffic loads of Table 3. We can see that the total rewards due to the acceptance of new or handoff calls of the two classes of traffic ($C_1$ or $C_2$) in the cellular network using Q-DCA is more important compared to those of the greedy-DCA policy. Fig. 4(b) shows that the total loss rewards due to rejection of new calls or the failure of handoff calls were reduced significantly using Q-learning for all the traffic loads and especially when the traffic load is heavy. Q-DCA outperforms the traditional schemes (greedy-DCA) and it performs as well as the DCA-Nie does.

[6] Greedy-DAC: Policy that randomly selects a channel to serve a call without any interference measurements. The channels are selected based on a uniform distribution and hence each of the *M* channels has an equal probability of being selected.

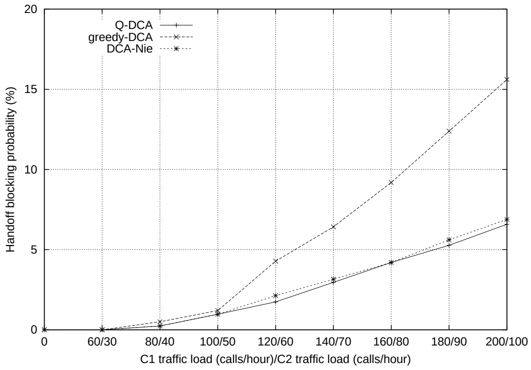
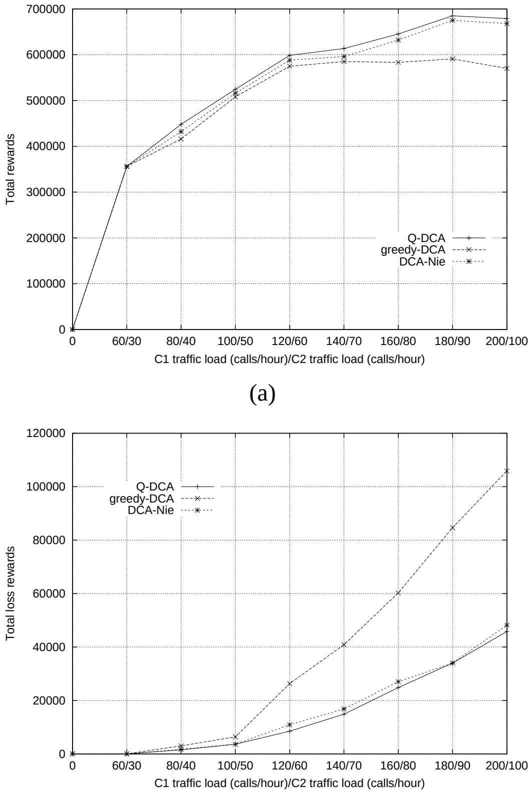


*Fig.* 3. Handoff blocking probability.



(a)

(b)

*Fig.* 4 (a) Total rewards/1 hour (b) Total Loss rewards/1 hour.

We also compare the handoff blocking probability of $C_1$ traffic vs. $C_2$ traffic using Q-DCA (Fig. 5). This metric is given, for each class of traffic $C_i$, by

$$P_{HO(C_i)} = \frac{number\ of\ handoff\ calls\ of\ type\ C_i\ blocked\ in\ the\ system}{number\ of\ handoffs\ of\ type\ C_i\ in\ the\ system} \qquad (8)$$

Since $C_1$ calls have priority than $C_2$ calls[7], we notice that the handoff blocking probability of $C_1$( $P_{HO(C1)}$ ) is less than the handoff blocking probability of $C_2$ ( $P_{HO(C2)}$ ).

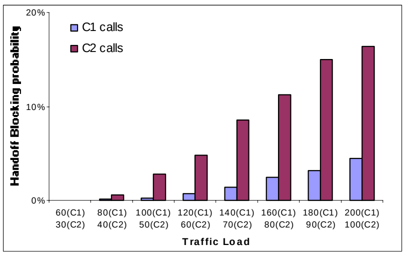


*Fig. 5.* Handoff blocking probability of $C_1$ vs. $C_2$ using Q-DAC.

This illustrates clearly that Q-DCA has the potential to significantly improve the performance of the system over a broad range of network loads. It is interesting to observe that the Q-values were not relearned and retrained, indicating that the system possesses some generalization capabilities.

## 4.2 Non-uniform Distribution

In this case we used the policy learned in the training period but the traffic densities in terms of calls/hour are inhomogeneously distributed among 36 cells (for both classes $C_1$ and $C_2$) as shown in Fig. 6. The averaging arrival call rate is 100 calls/h for $C_1$ calls and 50 calls/h for $C_2$ calls.

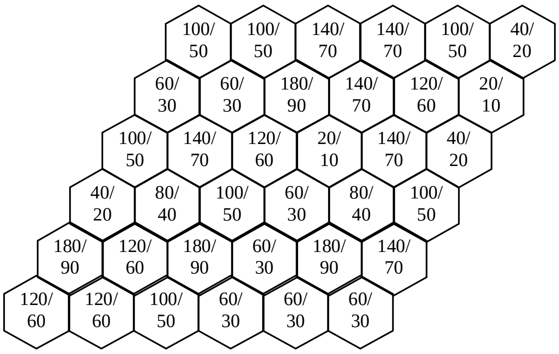


*Fig.*6. Nonuniform traffic distribution ($C_1$ traffic load/$C_2$ traffic load).

Fig. 8 shows the handoff blocking probabilities of using our method against the arrival rates which were increased by 100%, 120%, 140%, 160%, 180%, and 200% over the base rates given in Fig. 6. Figures 7 to 9 indicate significant improvements of the Q-learning algorithm over the greedy scheme.

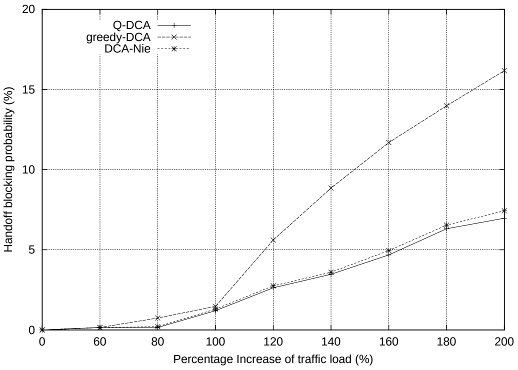
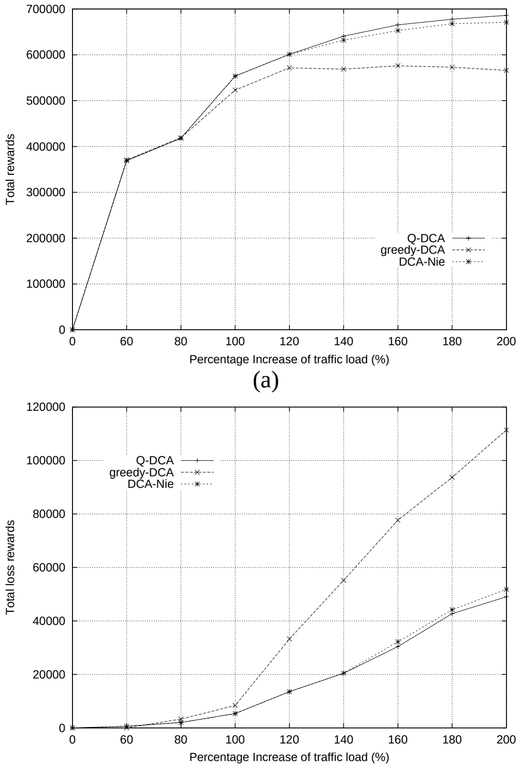


*Fig.*7 . Handoff blocking probability

[7] The rewards associated to $C_1$ calls are more important than those associated to $C_2$ calls (cf. table I)



(a)

(b)

*Fig.* 8 (a) Total rewards per 1 hour (b) Total Loss rewards per 1 hour.

It is also clear from the Fig. 9 that the handoff blocking probabilities of $C_1$ calls calculated using (8) are less than the handoff blocking probability of $C_2$ calls for the Q-DCA algorithm (Fig. 9).

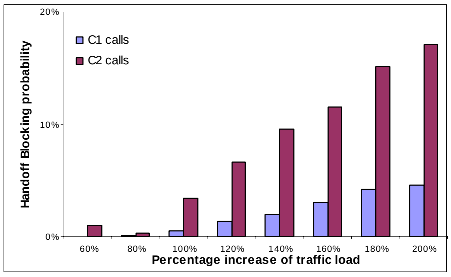


*Fig.* 9. Handoff blocking probability of $C_1$ vs. $C_2$ using Q-DAC.

## 4.3 Time-Varying Traffic Load

The traffic load in a cellular system is typically time varying. In this case, we always use the policy learned in training period and we use, as in [10], the pattern given in *Fig*. 10 concerning arrivals during a typical 24-h business day. The peak hours occur at 11:00 a.m. and 4:00 p.m. *Fig*. 11 gives the simulation results under the assumption that the two traffic classes were spatially uniformly distributed and followed the same time-varying pattern given in *Fig*. 10. The maximum traffic load is set to be 120 calls/h for $C_1$ class and 60 calls/h for $C_2$ class. The blocking probabilities were calculated on an hour-by-hour basis. The improvements of the proposed reinforcement learning algorithms over the greedy policy are apparent specially when the traffic is heavy (at 11:00 a.m. and 4:00 p.m.). The gains due to RL are about 61% for Q-DCA.

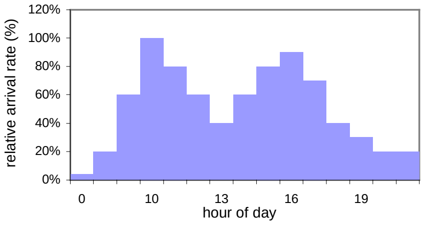


*Fig.*10. A traffic pattern of a typical business day.

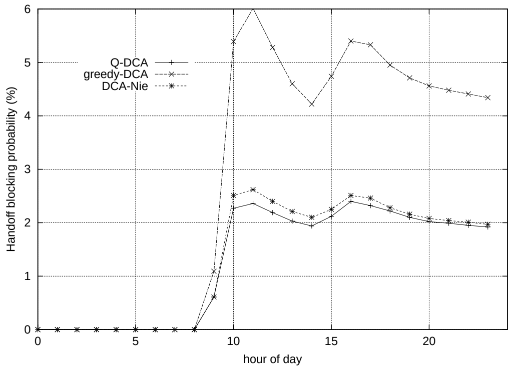


*Fig.* 11. Performance with time-varying traffic load and spatial uniform traffic.

We also examined the case in which the traffic loads were both spatially nonuniformally distributed and temporally varying. *Fig*. 12 gives the simulation results under the same assumption of the uniform case. The improvements of the proposed reinforcement learning algorithms over the greedy policy are apparent but a more significant improvement was seen in the uniform case. The gains due to RL are about 44% for Q-DCA.

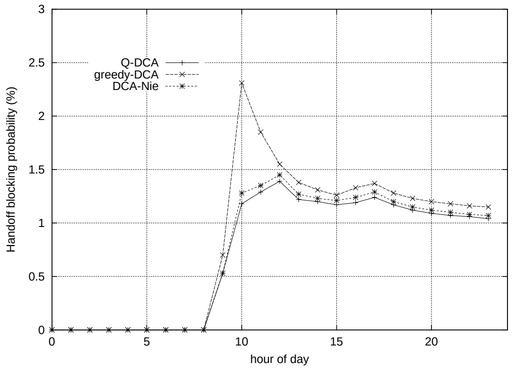


*Fig.* 12. Performance with time-varying traffic load and spatial nonuniform traffic.

## 4.4 Equipment Failure in DCA systems

We simulate, as in [10], an equipment failure by some frequency channels being temporally unavailable. There were initially 70 channels available in the system. Between 10:00 and 15:00 o'clock, we temporally shut down 0, 3, 5 or 7 channels. Figure 13 shows the effect of channel failure on the system using Q-DCA algorithm in term of handoff blocking probability. The two classes of traffic were spatially uniformly distributed and followed the same parameters given in table 2. We can clearly see that Q-DCA channel assignment algorithm possesses certain robustness to channel failure situations.

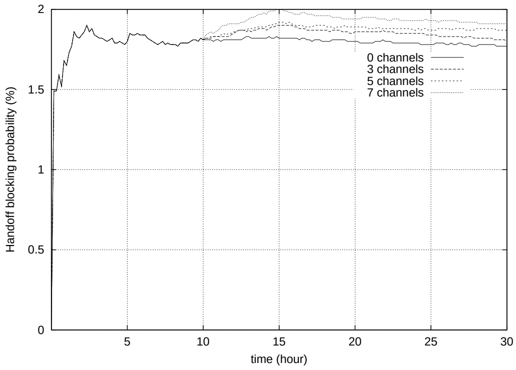


*Fig.* 13. Performance of Q-DCA with channel failure.

## 5 Conclusion

In this paper, we presented a new approach to solve the channel assignment combined with call admission control problems in DCA system. We formulate the problem as an average reward dynamic programming problem (SMDP), but with a very large state space. Traditional SMDP methods are computationally infeasible for such large-scale problems. So, the optimal solutions are obtained by using a self-learning scheme based on Q-Learning algorithm. The benefits gained by using Q-DCA can be summarized as follows. First, the learning approach provides a realistic and simple way to obtain an approximate optimal solution for which an optimal solution can be very difficult to find using traditional methods. Second, since the proposed scheme is performed in a real-time environment, it is possible to carry out online learning while performing the real channel assignment and admission control. Compared to other schemes, the system offers some generalization capabilities. So, any unforeseen event due to significant variations in the environment conditions can be considered as a new experience for improving the adaptation and the learning quality of the system. Third, the channel assignment policy can be determined with very little computational effort. Q-DCA algorithm is quite sophisticated compared to other Q-learning channel allocation schemes (DCA-Nie) [10] since it combines the notions of call admission control and channel allocation. It is, also, shown that the proposed algorithm results in significant savings than alternative heuristics.

## References

[1] I. Katzela, M. Naghshineh, “Channel assignement schemes for cellular mobile telecommunications systems,” *IEEE Personal Communications Magazine*, Juin 1996.

[2] M. Zhang and T.S. Yum, “Comparisons of channel assignment strategies in cellular mobile systems,” *IEEE Trans. Vehicular Technology*, Vol. 38, pp. 211-215, 1989.

[3] M. Zhang and T.S. Yum, “The nonuniform compact pattern allocation algorithm for cellular mobile systems,” *IEEE Trans. Vehicular Technology*, Vol 40, pp. 387-391, 1991.

[4] D. C. Cox and D. O Reudink, “Dynamic channel assignment in two dimensional large mobile radio systems”, *Bell Syst. Tech. J.*, Vol. 51, pp. 1611-1627, 1972.

[5] E. Del Re, R. Fantacci, and L. Ronga, “A dynamic channel allocation technique based on Hopfield neural networks”, *IEEE Trans. Vehicular Technology*, Vol. 45, pp. 26-32, 1996.

[6] D. D. Dimitrijevic and J. Vucetic, “Design and performance analysis of the algorithms for channel allocation in cellular networks”, *IEEE Trans. Vehicular Technology*, Vol. 42, pp. 526-534, 1993.

[7] K. N. Sivarajan, R.J McEliece, and J.W.Ketchum, “Dynamic channel assignment in cellular radio”, *Proc. IEEE 40th Vehicular Technology Conf.*, pp. 631-637, 1990.

[8] G. Barto, S. J. Bradtke, and S. P. Songh, “Learning to act using real-time dynamic programming”, *Artificial Intelligence*, vol. 72, pp. 81-138, 1995.

[9] C. J. C. H. Watkins and P. Dayan, “Q-learning”, *Machine Learning*, vol. 8, pp. 279-292, 1992.

[10] J. Nie and S. Haykin, “A Q-Learning based dynamic channel assignment technique for mobile communication systems”, *IEEE Transactions on Vehicular Technology*, vol. 48, N°. 5, September 1999.

[11] H. Tong and T. X. Brown, “Adaptive Call Admission Control under Quality of Service Constraint: a Reinforcement Learning Solution”, *IEEE Journal on Selected Areas in Communications (JSAC’2000)*, vol. 18**,** N°. 2, pp. 209-221, February 2000.

[12] H. Tong, Adaptive Admission Control for Broadband Communications, Ph.D. thesis, University of Colorado, Boulder, Summer 1999.

[13] T. M. Mitchell, “Machine Learning”, *McGraw-Hill companies, Inc.*, 1997.

[14] L. Breiman, J.H. Friedman, R.A. Olsen, and C.J. Stone, “Classification and Regression Trees”, *Chapman & Hall*, 1984.

[15] S. Senouci, A.-L. Beylot, Guy Pujolle, “Call Admission Control for Multimedia Cellular Networks Using Neuro-Dynamic Programming”, Networking 2002, Pisa, Italy, May 2002.

[16] C. J. C. H. Watkins, “Learning from delayed rewards,” *PhD. thesis*, University of Cambridge, Psychology Department, 1989.