A Q-Learning-Based Dynamic Channel Assignment Technique for Mobile Communication Systems

Junhong Nie and Simon Haykin, *Fellow, IEEE*

***Abstract*—This paper deals with the problem of channel assignment in mobile communication systems. In particular, we propose an alternative approach to solving the dynamic channel assignment (DCA) problem through a form of real-time reinforcement learning known as Q learning. Instead of relying on a known teacher, the system is designed to learn an optimal assignment policy by directly interacting with the mobile communication environment. The performance of the Q-learning-based DCA was examined by extensive simulation studies on a 49-cell mobile communication system under various conditions including homogeneous and inhomogeneous traffic distributions, time-varying traffic patterns, and channel failures. Comparative studies with the fixed channel assignment (FCA) scheme and one of the best dynamic channel assignment strategies (MAXAVAIL) have revealed that the proposed approach is able to perform better than the FCA in various situations and capable of achieving a similar performance to that achieved by the MAXAVAIL, but with a significantly reduced computational complexity.**

***Index Terms*— Dynamic channel assignment, dynamic programming, neural networks, Q-learning.**

I. Introduction

SINCE THE number of channels allocated to a mobile communication system is limited, efficient utilization of these available channels by using efficient channel assignment strategies has been one of the main concerns in designing a cellular mobile communication system. The channel assignment problem involves assigning channels to each radio cell (or call) in such a way that the probability that incoming calls are blocked and the probability that the signal-to-interference ratio (SIR) due to channel reuse falls below a prespecified value are sufficiently low.

The existing channel assignment methods may be roughly classified into fixed and dynamic schemes [15], [18]. In the fixed channel assignment (FCA) scheme, a set of channels is allocated to each cell permanently by a frequency planning process. A channel can be associated with many cells as long as the cochannel interference constraint is satisfied or equivalently two cells are located at least a cochannel reuse distance D away. In other words, two cells at distance D or more are allocated the same subset of F channels. The number of assigned channels F can be determined by $F = M/K$, where M is the number of allocated channels in the system and K is the reuse factor. A number of FCA approaches exist, ranging from simple heuristic ones to more mathematically involved ones in which various conventional or nonconventional optimization schemes are applied, including neural networks, genetic algorithm, and simulated annealing [8], [12], [14].

In contrast to FCA, in dynamic channel assignment (DCA) schemes all the channels can be used in all the cells as long as cochannel constraints are satisfied and channels are assigned to cells only when they are required; there are no fixed relationships between cells and channels. In other words, channel assignment is carried out on a call-by-call basis in a dynamic manner. Therefore, traffic variability can be automatically adapted. This can potentially lead to improved performance, particularly if the spatial traffic profile is unknown, poorly known, or varies according to time. A number of DCA algorithms have been proposed [3]–[7], [17], [18]. Depending on the form of information used, we may identify two classes of DCA schemes: 1) interference-adaptive schemes, where actual field signal strength measurements are used as the basis for channel assignment and 2) traffic-adaptive schemes, where only traffic conditions in neighboring cells are taken into account. The first class of DCA schemes is described in [10] and [16]. The DCA scheme described in this paper belongs to the second class.

Among the proposed traffic adaptive schemes, one class of the strategies called exhaustive searching DCA [5]–[7], [17], [21], [22] are of particular interest to us. The basic idea is that each available channel is associated with a cost (reward). When a new call is attempted, 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. The maximum availability strategy, known as MAXAVAIL [17], has been claimed to produce best performance in the case of no intracell handovers being involved. The idea is to select channel k from a set of available channels $B(i,t)$ in cell i at time t which maximizes the total number of channels available in the entire system $\mathcal{S}(k)$ defined by

$$\mathcal{S}(k) = \sum_{i \in X} \{B(i,t) | k \text{ is assigned to } i\}. \tag{1}$$

Here, it is assumed that channel k is assigned to i, where X is the set of cells in the system.

This paper proposes an alternative approach to solving the dynamic channel assignment problem. The optimal dynamic assignment policy is obtained through a form of real-time reinforcement learning [1] known as Q learning [19]. Instead

Manuscript received January 18, 1997; revised April 10, 1997. This work was supported by Motorola, Vancouver, B.C., Canada.

The authors are with the Communications Research Laboratory, McMaster University, Hamilton, Ont., L8S 4K1, Canada.

Publisher Item Identifier S 0018-9545(99)05745-X.

of relying on a known teacher, the system is designed to learn an optimal policy by directly interacting with the environment with which it works, a mobile communication environment in our case. Learning is accomplished progressively by appropriately utilizing the past experience which is obtained during real-time operation. The performance of the Q-learning-based DCA was examined by extensive simulation studies on a 49-cell mobile communication system under various conditions including homogeneous and inhomogeneous traffic distributions, time-varying traffic patterns, and channel failures. Also, we carried out some comparative studies with the FCA scheme and one of the best DCA strategies, MAXAVAIL [17].

II. Proposed Technique

Conventional DCA strategies ignore completely the experience or knowledge that could be gained during real-time operation of the system. Although the neural network-based approach [3] does have a training stage, it is crucial to have a good teacher (a known DCA algorithm) to guide the training. On the other hand, exhaustive searching approaches are generally time-consuming to find a solution and are thus inefficient. Here, we propose an alternative approach to solving the channel assignment problem. The approach is based on the judgment that DCA can be regarded as a *large-scale constrained dynamic optimization problem* embedded in a *stochastic environment*, and learning is one of the effective ways to find a solution to this problem. A particular learning paradigm we have adopted is known as *reinforcement learning (RL)* [1]. In RL, a learner aims at learning an optimal control policy π by repeatedly interacting with the controlled environment in such a way that its performance evaluated by a scalar reward (cost) obtained from the environment is maximized (minimized). The RL algorithms developed so far are closely related to the well-known dynamic programming (DP) procedure developed some decades ago by R. Bellman [2]. There exists a variety of RL algorithms. A particular algorithm that appears to be suitable for the DCA task is called Q learning [19]. In what follows, we first describe the algorithm briefly and then present the details of how the DCA problem can be solved by means of Q learning.

A. Q-Learning Strategy

Assume that the environment, which a learner interacts with, is a finite-state discrete-time stochastic dynamical system as shown in Fig. 1. Let X be the set of possible states $X = \{x_1, x_2, \cdots, x_n\}$ and A be a set of possible actions $A = \{a_1, a_2, \cdots, a_m\}$.

The interaction between the learner and the environment at each time instant consists of the following sequence.

- The learner senses the state $x_t \in X$.
- Based on x_t, the learner chooses an action $a_t \in A$ to perform.
- As a result, the environment makes a transition to the new state $x_{t+1} = y \in X$ according to probability $P_{xy}(a)$ and thereby generates a return (cost) r_t.
- The r_t is passed back to the learner and the process is repeated.

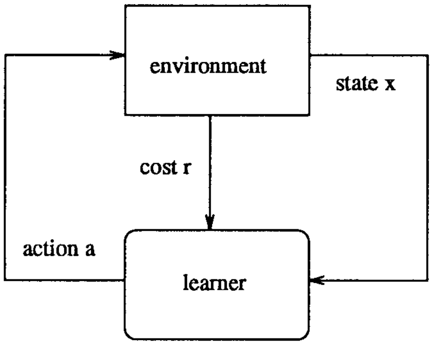


Fig. 1. An illustration of learner-environment interaction.

The objective of the learner is then to find an optimal policy $\pi^*(x) \in A$ for each x, which minimizes some cumulative measure of the costs $r_t = r(x_t, a)$ received over time. A particular measure, which is referred to as the total expected discounted return (cost) over an infinite time horizon, is given by

$$V^\pi(x) = \boldsymbol{E}\left\{\sum_{t=0}^{\infty} \gamma^t r(x_t, \pi(x_t)) | x_0 = x\right\} \tag{2}$$

where $\boldsymbol{E}$ stands for the expectation operator and $0 \leq \gamma < 1$ is a discount factor. $V^\pi(x)$ is often called the value function of the state x.

Equation (2) can be rewritten as [19]

$$V^\pi(x) = R(x, \pi(x)) + \gamma \sum_{y \in X} P_{xy}(\pi(x)) V^\pi(y)$$

where $R(x, \pi(x)) = \boldsymbol{E}\{r(x, \pi(x))\}$ is the mean value of $r(x, \pi(x))$. The optimal policy π^* satisfies Bellman's optimality criterion

$$\begin{aligned} V^*(x) &= V^{\pi^*}(x) \\ &= \min_{a \in A}\left[R(x, a) + \gamma \sum_{y \in X} P_{xy}(a) V^*(y)\right]. \end{aligned} \tag{3}$$

The task of Q learning is to determine a π^* without knowing $R(x, a)$ and $P_{xy}(a)$, which makes it well suited for the DCA problem. This is achieved by reformulating (3). For a policy π, define a Q value (or state-action value) as

$$Q^\pi(x, a) = R(x, a) + \gamma \sum_{y} P_{xy}(a) V^\pi(y)$$

which is the expected discounted cost for executing action a at state x and then following policy π thereafter.

Let

$$Q^*(x, a) = Q^{\pi^*}(x, a) = R(x, a) + \gamma \sum_{y} P_{xy}(a) V^{\pi^*}(y).$$

We then get

$$V^*(x) = \min_{a \in A}[Q^*(x, a)].$$

Thus, the optimal value function V^* that satisfies Bellman's optimality criterion can be obtained from $Q^*(x, a)$ and in turn

$Q^*(x, a)$ may be expressed as

$$Q^*(x, a) = R(x, a) + \gamma \sum_y \{P_{xy}(a)[\min_{b \in A} Q^*(y, b)]\}.$$

The Q-learning process tries to find $Q^*(x, a)$ in a recursive manner using available information (x_t, a_t, y_t, r_t), where x_t and $y_t(= x_{t+1})$ are the states at time t and $t+1$, respectively; and a_t and r_t are the action taken at time t and the immediate cost due to a_t at x_t, respectively. The Q-learning rule is

$$Q_{t+1}(x, a) = \begin{cases} Q_t(x, a) + \alpha \Delta Q_t(x, a), & \text{if } x = x_t \text{ and } a = a_t \\ Q_t(x, a), & \text{otherwise} \end{cases} \quad (4)$$

where α is the learning rate, and

$$\Delta Q_t(x, a) = \{r_t + \gamma \min_b [Q_t(y_t, b)]\} - Q_t(x, a).$$

It has been shown [20] that if the Q value of each admissible (x, a) pair is visited infinitely often, and if the learning rate is decreased to zero in a suitable way, then as $t \to \infty$, $Q_t(x, a)$ converges to $Q^*(x, a)$ with probability 1.

B. Learning DCA Policy by Q Learning

The mobile communication system can be considered as a discrete-time event system. Without considering handovers the major events which may occur in a cell include new call arrivals and call departures due to the completion of the call. These events are modeled as stochastic variables with appropriate probability distributions. In particular, new call arrivals in cell i are independent of all other arrivals and obey a Poisson distribution with a mean arrival rate λ. Call holding time τ_{holding} is assumed to be **exponentially** distributed with a mean call duration $1/\mu$. To utilize the Q-learning scheme, it is necessary to formulate the DCA into a dynamic programming problem, or equivalently, to identify the system state x, action a, associated cost r, and the next sate y.

1) State: Assume that there are N cells and M channels available in the mobile communication system. We define state x_t at time t as

$$x_t = (i, A(i))_t$$

where $i \in \{1, 2, \cdots, N\}$ is the cell index specifying there is an event, either call arrival or departure, occurring in cell i. $A(i) \in \{1, 2, \cdots, M\}$ is the number of available channels in cell i at time t, which depends on the channel usage conditions in cell i and in its interfering cells $I(i)$, where $I(i)$ is the set of cells interfering with i, i.e., those neighborhood cells that lie at a distance less than a reuse distance D.

To obtain $A(i)$ at time t, we define the channel status for cell $q, q = 1, 2, \cdots, N$ as a M-dimensional vector

$$\boldsymbol{u}_q = (u_{q1}, u_{q2}, \cdots, u_{qM})$$

where

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

where $q = 1, 2, \cdots, N$ and $k = 1, 2, \cdots, M$.

Furthermore, an availability vector $\boldsymbol{s_q} \in \{0, 1\}^M$ is formed

$$\boldsymbol{s}_q = (s_{q1}, s_{q2}, \cdots, s_{qM})$$

with each component s_{qk} being defined as

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

Once channel status in cell i and in its interfering cells $j \in I(i)$ are known, availability vector $\boldsymbol{s_i}$ can be formed easily with the corresponding components being obtained from

$$s_{ik} = \max\{u_{qk} | q \in \overline{I}(i)\}$$

where $k = 1, 2, \cdots, M$ and the set $\overline{I}(i)$ is the combination of $I(i)$ previously defined and i itself.

By using $s_{ik}, A(i)$ can be easily obtained from

$$A(i) = \sum_{k=1}^{M} \overline{s}_{ik} \quad (5)$$

where $\overline{s}_{ik}$ is the logical negation of s_{ik}.

2) Actions: Applying an action a is to assign a channel k from $A(i)$ available channels to the current call request in cell i. Here, a is defined as

$$a = k \quad k \in \{1, 2, \cdots, M\} \quad \text{and} \quad s_{ik} = 0.$$

3) Costs: The cost $r(x, a)$ assesses the immediate cost incurred due to the assignment of a at state x. More specifically, it is a cost of choosing channel k to serve the currently concerned call attempt in cell i. There are many possibilities to define r. Here, we assess the cost of applying action $a = k$ by evaluating usage conditions in cochannel cells associated with cell i. The basic idea is to assign higher costs to those usages in which cochannel cells are located further away from cell i. Thus, the lower costs are associated with those usages in which cochannel cells have minimum compact distance. More specifically, $r(x, k)$ is calculated by the following weighted sum:

$$r(x, k) = n_1(k)r_1 + n_2(k)r_2 + n_3(k)r_3. \quad (6)$$

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 [22]. 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 subcosts associated with the above-mentioned conditions related to $n_1(k), n_2(k)$, and $n_3(k)$, respectively. The cost defined above (6) is of a similar form to that used in [21] for a locally optimized dynamic assignment strategy where a distance measure was established by using three steps. However, the calculation of the cost in our case is somewhat simpler and more explicit.

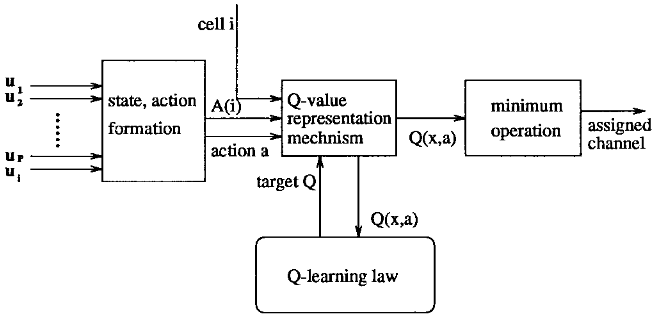


Fig. 2. Structure of Q-learning-based DCA.

4) Next State: According to the definition of state x_t described before, the state transition from x_t to x_{t+1} is determined by two stochastic events, call arrivals and call departures. Therefore, the next state $y = x_{t+1}$ can be obtained whenever one of these events occurs. However, in this paper only call arrivals are treated explicitly as sources to trigger the state transition in which actions, i.e., channel assignments, are required to be taken. Although call departures do alter the number of available channels, we will not carry out any actions for them (here no intracell handover is considered) except to release the channel on which a call is just completed.

C. Algorithm Implementation

Having specified the state, action, cost, and next state, we are ready to describe a detailed implementation of the Q-learning algorithm for solving the DCA problem. Fig. 2 illustrates the structure of the Q-learning-based DCA system. We notice that Q learning is an online learning scheme. In our case, it means that the task of learning a good assignment policy and assigning a channel to a call attempt can be performed simultaneously. The system using Q learning, however, may work in a fashion consisting of two successive procedures, learning and assigning. The Q value is first online learned with a sufficiently long time period, with the learned Q values being stored in a representation mechanism. Then the task of online assignment is carried out by using the learned Q values. Here, an important issue arises as to how to store the Q values.

There exists a variety of approaches to representing the Q values [1]. The lookup table is the most straightforward method. It has the advantage of being both computationally efficient and completely consistent with the structure assumption made in proving the convergence of the Q-learning scheme. However, when the input space consisting of state-action pairs is large or the input variables are continuous, using lookup tables can be prohibitive because memory requirement may be huge. In this case, some function approximators such as neural networks [11] may be used in an efficient manner. As expected, a second learning (or training) procedure will be involved in which the network parameters such as weights are determined. In this paper, both the lookup table and the neural network are considered as the representational mechanism.

Now the steps concerning learning and assigning corresponding to Fig. 2 are given as follows.

1) *State-Action Construction:* Construct current state $x_t = (i, A(i))$ by identifying the current cell number i and using channel usage information associated with i and its interfering cells. Also, find a list of m_x available channels denoted by the set $L(m_x)$ is.
2) *Q-Value Retrieval:* Form a set of m_x-argumented inputs $x_a = (x_t, k), k \in L(m_x)$ and feed them into the Q-value representation mechanism, thereby deriving a set of m_x $Q(x_t, k)$ values.
3) *Channel Assignment:* According to the definition of the Q values, the optimal action, i.e., the optimal channel k^*, is the one with minimum Q values
$$k^* = \min_{k \in L(m_x)} \{Q(x_t, k)\} \tag{7}$$
as indicated in Fig. 2.
4) *Q-Value Update:* Update the Q values, once the next state y and the instant cost $r(x_t, k^*)$ become available. The target Q value denoted by $Q^*(x_t, k^*)$ according to (4) is
$$Q^*(x_t, k^*) = r(x_t, k^*) + \gamma \min_b [Q_t(y, b)]$$
where $b \in L(m_y)$ are available channels at state y. The Q value of $Q(x_t, k^*)$ is updated according to the difference $Q^*(x_t, k^*) - Q(x_t, k^*)$.
5) *Network Parameters Update:* If the Q value is stored in a neural network or any type of approximator, the second learning procedure (training) is necessary to learn the weight parameters associated with the network. In this case, $Q^*(x_t, k^*) - Q(x_t, k^*)$ is served as an error signal which is backpropagated.

It can be seen that if the Q values are learned and represented faithfully, the task of assignment with learning being stopped can be very efficient, since in this case only the first four steps are involved.

It should be pointed out that the current implementation of channel assignment described above belongs to the centralized DCA schemes where all channels and their usage information are kept in a central pool, thereby resulting in a high centralization overhead. The algorithm proposed here may be implemented in a distributed fashion where each base station is responsible for assigning a channel to a call initiated within that cell. In this case, the cell index i in state x can be dropped and the only information needed is the channel status in neighboring cells, thereby simplifying the complexity of the algorithm. However, exchange of channel-status information between neighboring base stations must be taken into account.

III. Simulation Results and Discussions

A. Simulation Model and Procedures

The performance of the proposed DCA algorithm was evaluated by simulating a mobile communication system consisting of 49 hexagonal cells. With the reuse distance $D = \sqrt{21}R$, 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.

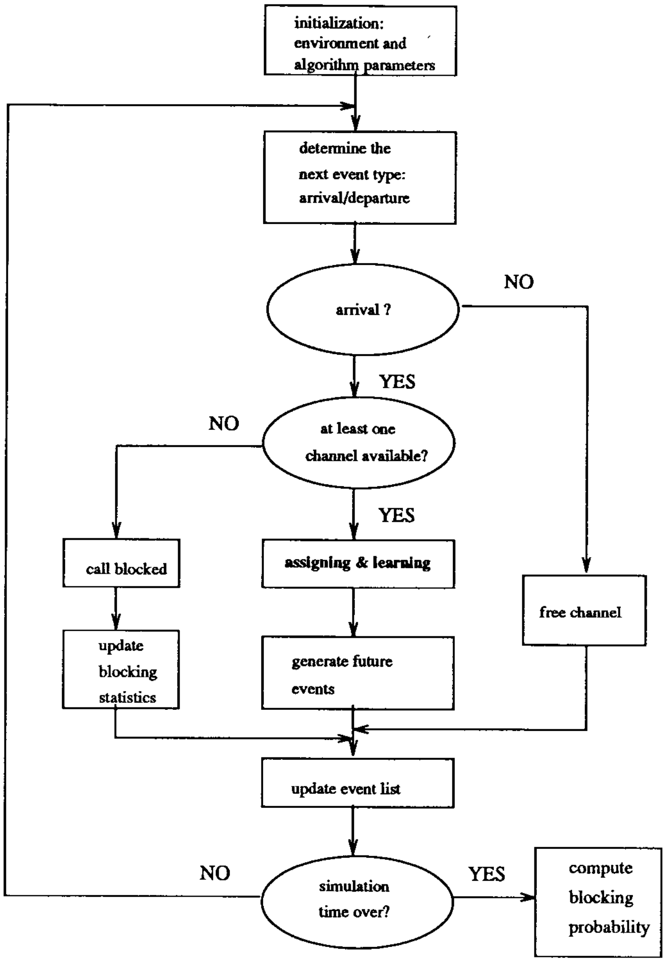

initialization: environment and algorithm parameters
determine the next event type: arrival/departure
arrival ?
NO
YES
at least one channel available?
NO
YES
call blocked
update blocking statistics
assigning & learning
generate future events
free channel
update event list
simulation time over?
NO
YES
compute blocking probability

Fig. 3. Flowchart for simulations.

The assumptions and the parameters used in the simulation include the following.

- New call arrivals obey Poisson distributions with uniform and nonuniform mean interarrive times among the cells. The mean arrival rate λ can be from 20 to 250 calls/h in each cell.
- The call-holding time obeys an exponential distribution with a mean call-duration $1/\mu$. Throughout this paper, $1/\mu = 180$ s for all calls was used.
- The offered traffic ρ_i in cell i is given by
$$\rho_i = \frac{\lambda}{\mu}.$$
- There are $M = 70$ channels available in the system, although the number of channels can vary.
- Blocked new and handover calls are dropped and cleared (Erlang B).

The performance of a channel assignment algorithm at a particular traffic loading was assessed by measuring the new call blocking probability P_n given by

$$P_n = \frac{\text{number of blocked calls in a cell}}{\text{number of new call arrivals to that cell}}. \tag{8}$$

To simulate the mobile communication system as a discrete-event dynamic system, a simulation clock is maintained. It gives the current value of simulated time of the whole system. The simulation clock is advanced according to the time of occurrence of the most imminent future event, which can be a call arrival or a call departure. To this end, it is necessary to maintain dynamically a list of future events. If the event occurring is a call arrival, a set of steps described in Section II-C is performed, resulting in either the call being blocked or served by a channel. If necessary, learning is carried out. On the other hand, if the event occurring is a call departure, the occupied channel is released. After the event is processed accordingly, the channel usage information in each cell is updated and the time clock is advanced. To calculate the system performance, the number of new call arrivals and the number of blocked calls are recorded.

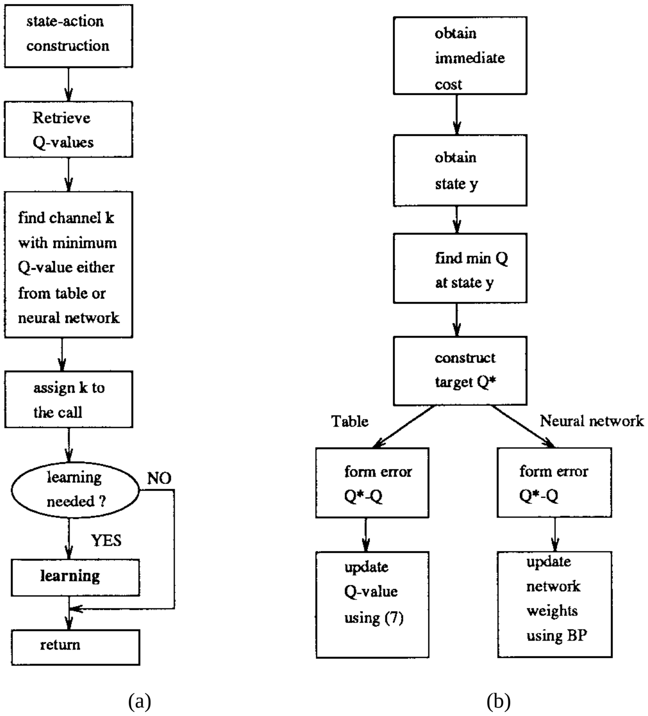


Fig. 4. (a) Assigning and learning process and (b) learning procedures.

The major procedures involved in the simulations are summarized in Fig. 3. The flowchart for the subroutine **assigning** and **learning** is shown in Fig. 4(a). If the system in the stage of learning optimal Q values, another subroutine **learning** is called as shown in Fig. 4(b). Depending on the form representing the Q values, i.e, a table or a neural network, the Q values are either updated directly in the case of lookup table, or updated indirectly in the case of neural network through the adjustment of weights using the well-known backpropagation (BP) algorithm [11].

B. Results

A set of simulations were carried out, including the cases of homogeneous and inhomogeneous traffic distributions, time-varying traffic patterns, and channel failures. For the purpose of comparison, the results due to the FCA and the maximum availability-based DCA algorithm, MAXAVAIL [17], were included. The reason for selecting the MAXAVAIL is that it has been claimed to be one of the best DCA algorithms in the

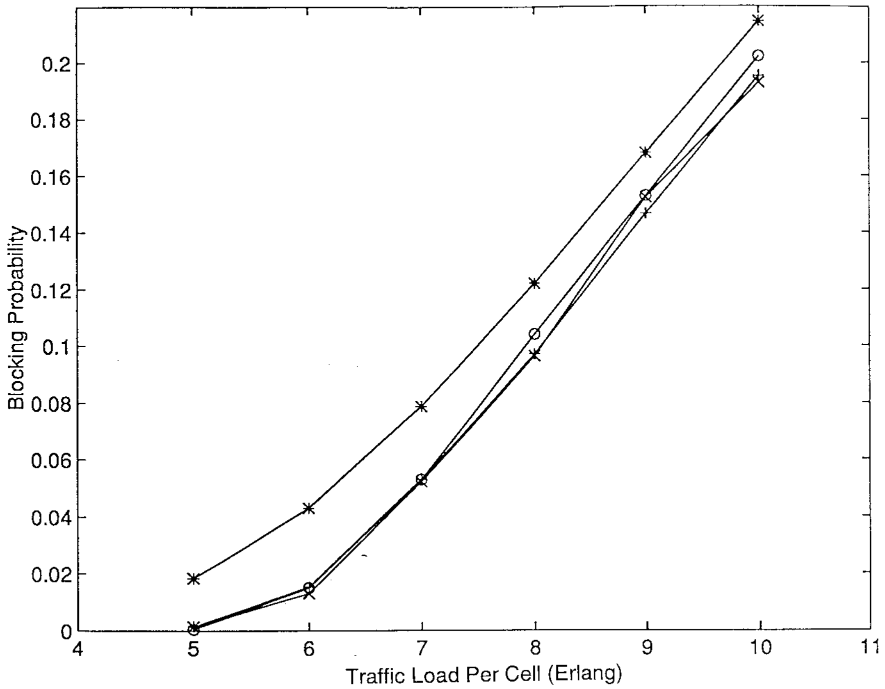


Fig. 5. Performance comparison with uniform traffic: FCA by $*$, MAXAVAIL by $\circ$, Q learning with table by $+$, and Q learning with neural network by x.

sense that its performance is close to the best achievable in this class of channel assignment algorithms where no intracell handovers are involved.

1) Uniform Distribution: In this case, the traffic load ρ was assumed to be the same among all 49 cells. Six different ρ's in Erlangs were used, being 5, 6, 7, 8, 9, and 10 which are equivalent, respectively, to call arrival rates of 100, 120, 140, 160, 180, and 200 calls/per h. Two Q-value representation mechanisms were considered. In the first place, a three-dimensional (3-D) lookup table was used. The Q values were learned by running the simulated mobile communication system for 30 simulated hours with a constant arrival rate being 120 calls/per h. The discount factor γ was chosen to be 0.5 and the learning rate α was designed to be state-action dependent varying with time. More specifically, each state action (x, a) was associated with a learning rate $\alpha_t(x, a)$ which was inversely proportional to the frequency of the (x, a) being visited up to the present time. The parameters in cost evaluation of (6) were $r_1 = -5$, $r_2 = -1$, and $r_2 = +1$. Such a setting would result in a situation in which the channels being used in the compact cells (associated with r_1) have minimum Q values and thus become the most favorable candidates to be chosen. The learned table was then used to assign the desired channel in the same communication system, but with six different traffic load conditions.

The same procedures were applied to the situation where a multilayer neural network [1] was used to represent the Q values. The network with 3 inputs, 8 hidden units, and 1 output unit was trained online for 30 simulated hours by using the BP algorithm in conjunction with the Q learning. The learning rate and momentum gains for network training were 0.3 and 0.9, respectively. The trained network was then used to select a desired channel in response to a call attempt.

Fig. 5 shows the blocking probabilities of using the Q learning with the table structure (marked by "$+$"), and with the neural network structure (marked by "$\times$"). The results due to FCA (marked by "$*$"), and MAXAVAIL (marked by "$\circ$") are also shown. For the FCA scheme, each cell was assigned $70/7 = 10$ channels because a seven-cell cluster pattern was assumed. The testing time for all the algorithms was five simulated hours.

It can be seen from Fig. 5 that the Q-learning-based DCA can perform better than the FCA although the improvement degree gained by the DCA decreases slightly with the increase in traffic load. For the interesting range of blocking probability 2%–6%, an increase in carried traffic of 20% can be obtained. Compared with the MAXAVAIL scheme, we conclude that the Q-learning-based DCA strategies are able to achieve a performance similar to that achieved by the MAXAVAIL. However, the computational complexities are quite different. This issue will be discussed in some details in Section IV-C.

It is noteworthy that when the traffic load is very heavy, the performance advantage of the DCA over FCA may become invisible. In our case, when the traffic load is 20 Erlang (400 calls per h), the blocking probability is 0.5358 for the DCA and it is 0.5368 for the FCA. This can be easily explained by the fact that when the traffic load is heavy, in general only very few channels are available and thus the optimal selection of a channel from few channels (for instance, only one channel being available) becomes meaningless.

2) Nonuniform Distribution: Fig. 6 shows a case [21] in which the traffic densities in terms of calls/per hour are inhomogeneously distributed among 49 cells. The averaging arrival call rate is 91.83 calls/per h. Fig. 7 shows the blocking probabilities of using the four methods described in the uniform case against the arrival rates which were increased

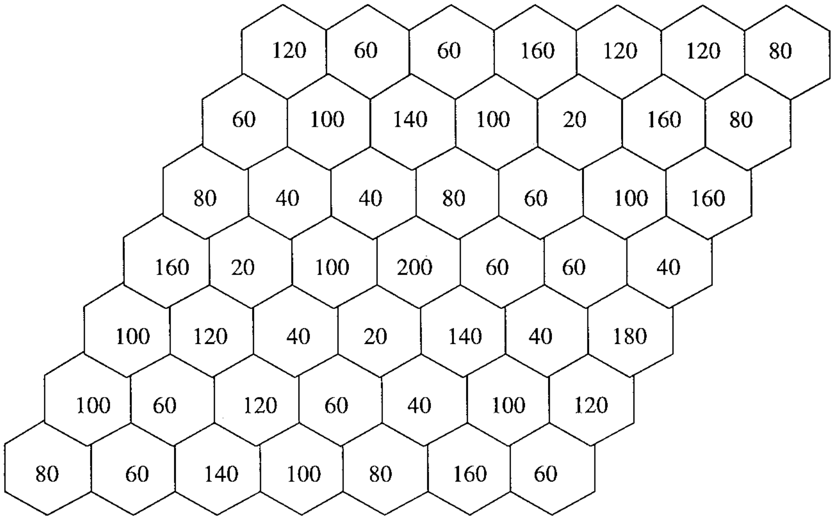


Fig. 6. Nonuniform traffic distribution: case 1.

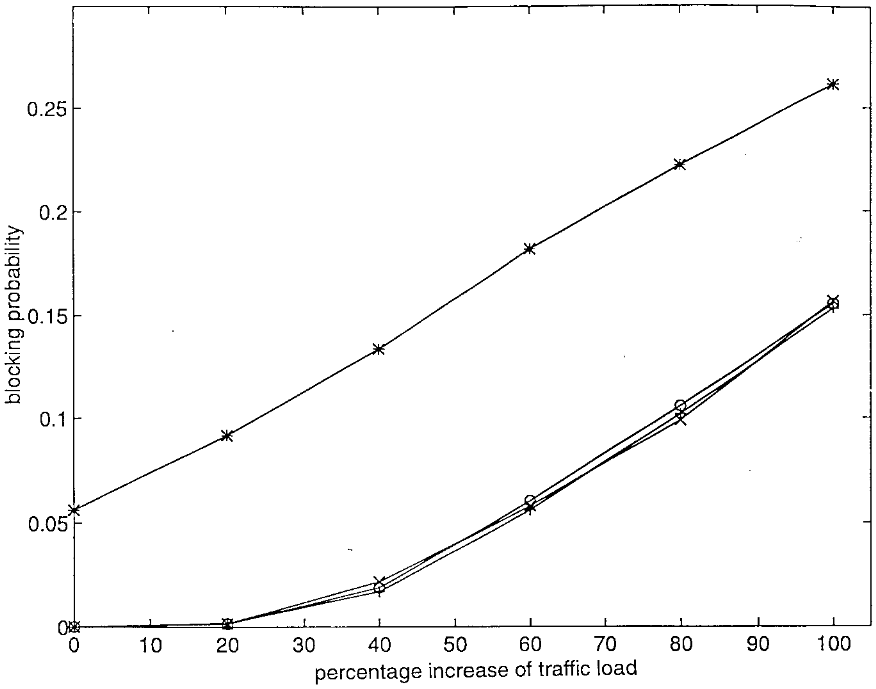


Fig. 7. Performance comparison with nonuniform traffic distribution (case 1): FCA by $*$, MAXAVAIL by o, Q learning with table by $+$, and Q learning with neural network by x.

by 0%, 20%, 40%, 60%, 80%, and 100% over the base rates given in Fig. 6. Fig. 7 indicates some significant improvements of the DCA algorithm over the FCA scheme, namely, about a 50% increase in the traffic load at the same blocking probabilities. This is somewhat expected because the DCA scheme is on a call-by-call basis and thus is able to adapt to the spatial nonuniform situations. However, for the FCA to perform better, the traffic in the system should be as homogeneous as possible.

We notice that the Q-learning-based DCA, whether using the table or the neural network, again performed as well as the MAXAVAIL did. It is interesting to observe that neither the table nor the neural network was relearned and retrained, indicating that the system possesses some generalization capability. While the generalization property of the neural networks is well known, the generalization property of the Q learning may be explained partially by the fact that the state-action knowledge embedded in the Q values is just an approximate reflection of the traffic adaptive property of the DCA.

Fig. 8 gives another example where the base traffic loads are given in Fig. 9 [21] with averaging arrival rate 106.53 calls/per h. As expected, the DCA schemes in this case did not perform as well as in the case of Fig. 7 in terms of the improvement degree over that obtained by the FCA approach.

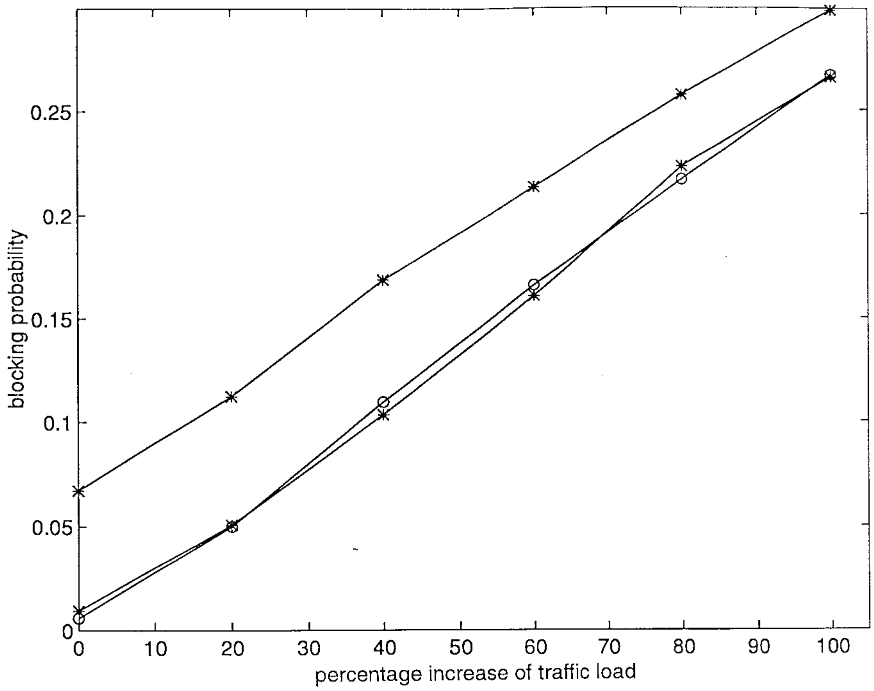
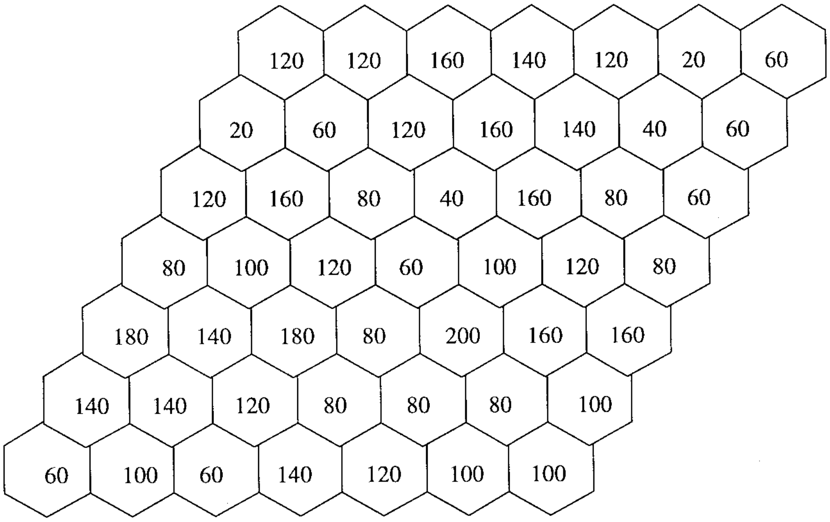


Fig. 8. Performance comparison with nonuniform traffic distribution (case 2): FCA by $*$, MAXAVAIL by $\circ$, Q learning with table by $+$, and Q learning with neural network by x.



120	120	160	140	120	20	60
20	60	120	160	140	40	60
120	160	80	40	160	80	60
80	100	120	60	100	120	80
180	140	180	80	200	160	160
140	140	120	80	80	80	100
60	100	60	140	120	100	100

Fig. 9. Nonuniform traffic distribution: case 2.

This is partly because the traffic loads were higher than those of Fig. 7.

3) Time-Varying Traffic Load: The traffic load in telephony systems is typically time varying. Fig. 10 shows a pattern concerning call arrivals during a typical 24-h business day, beginning at midnight [9]. It can be seen that the peak hours occur around 11:00 a.m. and 4:00 p.m. Fig. 11 gives the simulation results under the assumption that the traffic load was spatially uniformly distributed among 49 cells (maximum 165 calls/per h), but followed the time-varying pattern given in Fig. 9. The blocking probabilities were calculated on an hour-by-hour basis. The result obtained using the Q learning with the table structure is shown in Fig. 11(a), whereas that due to the FCA approach is shown in Fig. 11(b). The improvement of the Q-learning-based DCA over the FCA is apparent. For example, the number of hours at which the blocking probabilities were over 4% is two in Fig. 11(a), whereas that number is four in Fig. 11(b).

We also examined the case in which the traffic loads were both spatially nonuniformally distributed and temporally varying. Fig. 12 gives the results due to the Q learning with the table structure [Fig. 12(a)] and the FCA [Fig. 12(b)]. The spatial distribution was in accordance with that given in Fig. 6, and the temporal distribution was consistent with that given in Fig. 9. As expected, a more significant improvement in terms of blocking probability was seen in this case than that

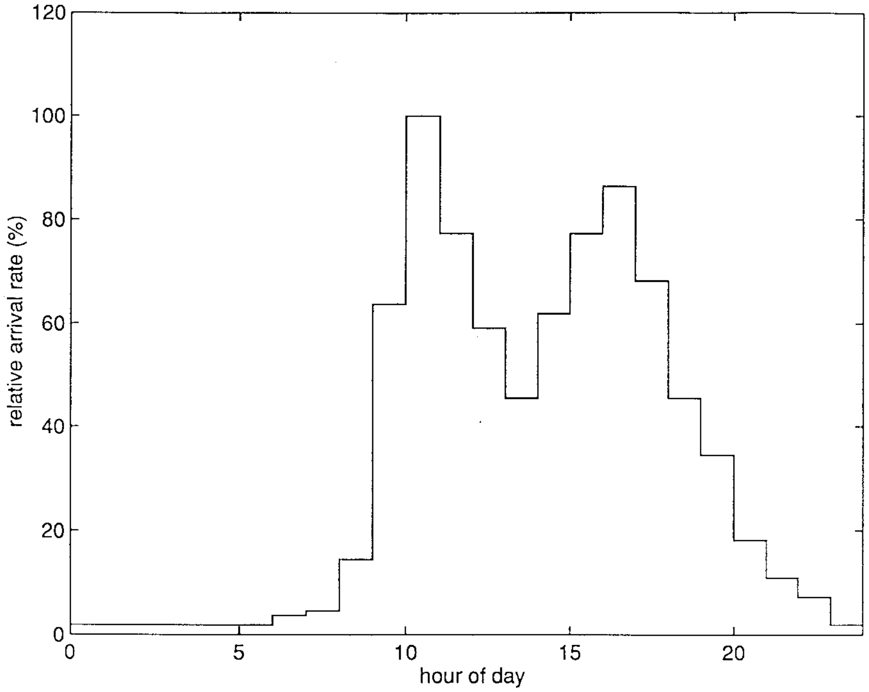
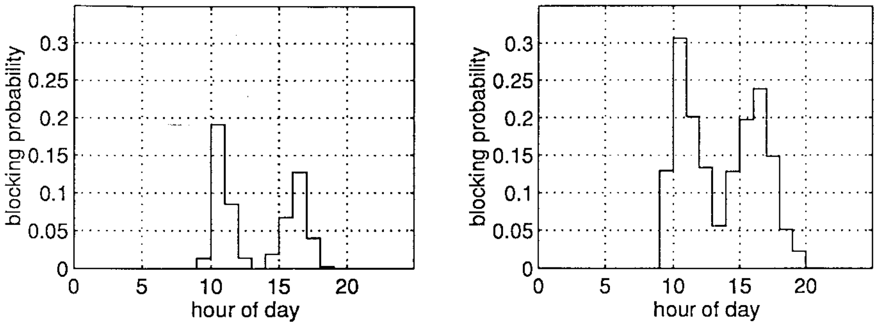
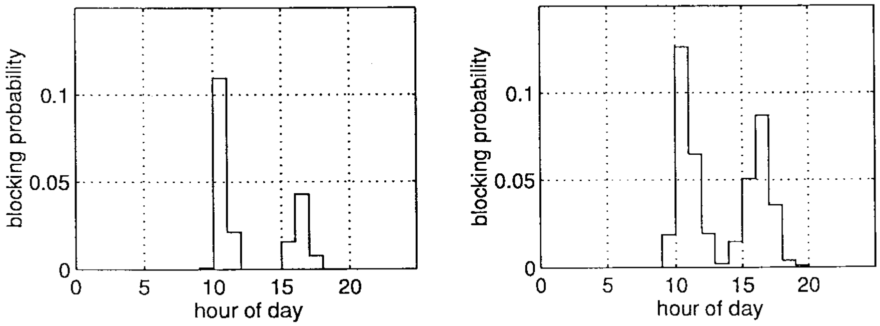


Fig. 10. A traffic pattern of a typical business day.



(a) (b)

Fig. 11. Performance with temporal varying and spatial uniform traffic: (a) Q learning and (b) FCA.



(a) (b)

Fig. 12. Performance with temporal varying and spatial nonuniform traffic: (a) Q learning and (b) FCA.

in the uniform distribution case. In particular, if again a 4% blocking probability is set to be a threshold, the number of hours exceeding that threshold is four in Figs. 12(a) and 10 in Fig. 12(b).

4) Equipment Failure and Online Behavior: In a mobile communication system, equipment failure during the normal operating hours may occur. To simulate this situation, we assumed that the various equipment failure cases will result

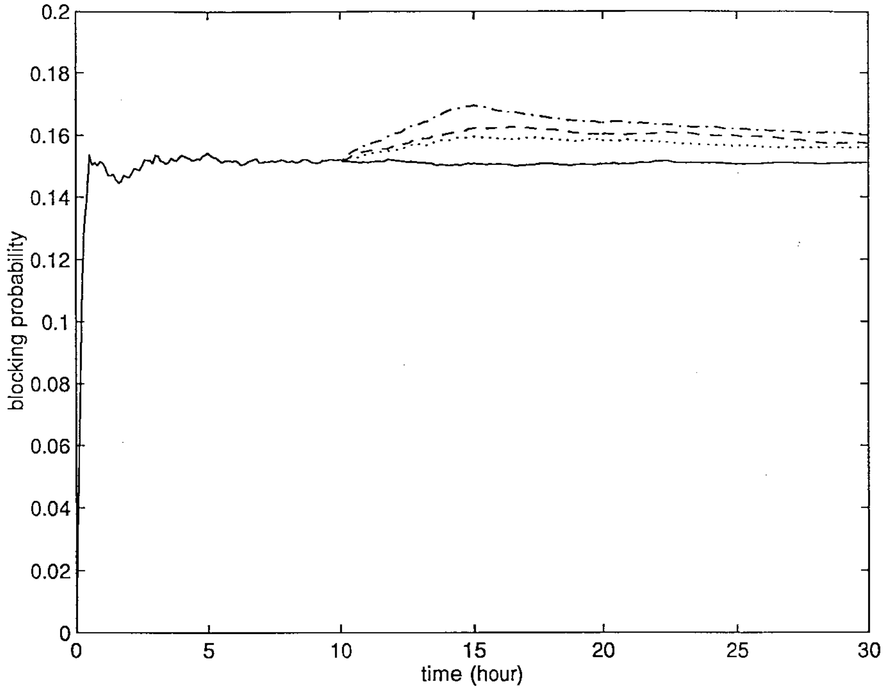
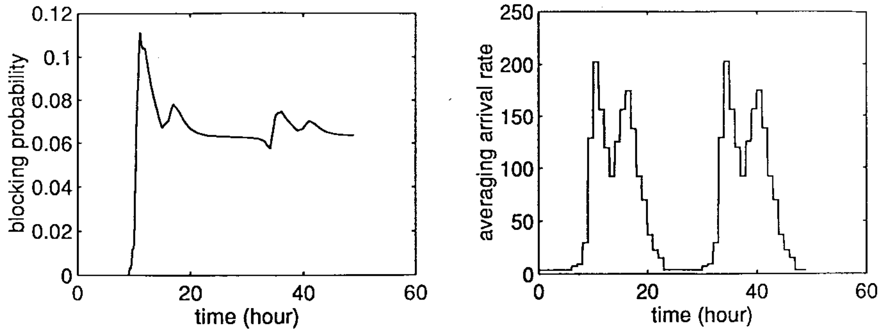


Fig. 13. Robustness to channel failure: zero channel (solid line); three channel (dotted line); five channel (dashed line); seven channel (dash-dotted line).



(a) (b)

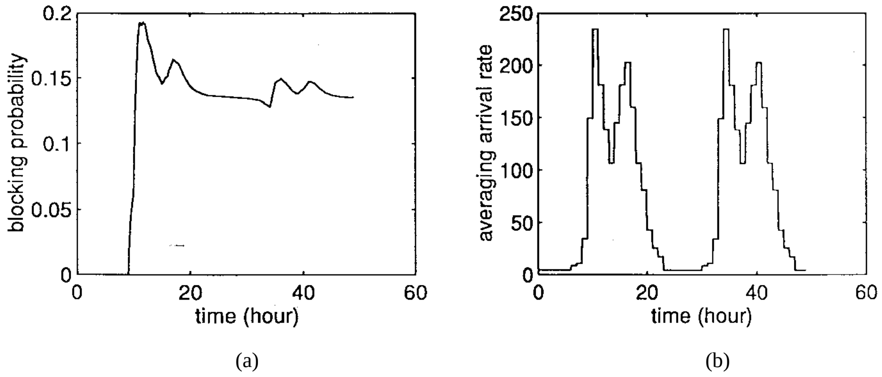
Fig. 14. Online behavior of the Q learning. (a) Blocking probability curve. (b) Averaging arrival rate with nonuniform distribution of case 1.

in some frequency channels being temporally unavailable. Fig. 13 gives an example where the effect of channel failure on the system blocking probability was demonstrated under the Q-learning-based scheme with the table representation structure. The call arrival rate was 180 call/per h in all the cells. There were 70 channels available initially and, during ten to fifteen o'clock, zero (solid line), three (dotted line), five (dashed line), or seven (dash-dotted line) channels were temporally shut down and thus not available for use. By comparing the results, it seems that the channel assignment algorithm possesses certain robustness to channel failure situations, particularly when the number of the failed channels is small, e.g., three–five.

Finally, we examined the online behavior of the Q-learning-based DCA in the sense that both learning and assigning operations were carried out simultaneously. Fig. 14(a) shows one of the results where the blocking probability was computed accumulatively over two days (48 h). The call arrival rates were nonuniformally distributed as shown in Fig. 8 with the average varying according to Fig. 14(b). Some improvement due to online learning can be seen in Fig. 14(a) in the sense that the accumulated blocking probabilities during the second day were generally lower than those during the first day. A similar behavior was observed in another case as shown in Fig. 15(a) where the call arrival rates were nonuniformally distributed as shown in Fig. 9 with the average varying according to Fig. 15(b).

C. Computational Issues

The results given in Figs. 5, 7, and 8 suggest that Q-learning-based DCA strategies are able to achieve a performance similar to that achieved by the MAXAVAIL. However, the computational complexities are quite different. In the process of assigning a channel, the complexity of using a table or neural network depends primarily on the number of channels, or more precisely, the number of available channels N_{ava}.



(a)

(b)

Fig. 15. Online behavior of the Q learning. (a) Blocking probability curve. (b) Averaging arrival rate with nonuniform distribution of case 2.

TABLE I
Number of Operations Required for Three DCA Schemes

Method	Number of Operations	Numerical Example
Table	$(N_{ava} - 1) + N_{ava} \times 1$	19
Neural network	$(N_{ava} - 1) + N_{ava} \times 64$	649
MAXAVAIL	$(N_{ava} - 1) + N_{ava} \times N \times (N' \times (M + M))$	1.23×10^6

$N_{\text{ava}} - 1$ comparisons with respect to N_{ava} Q's are needed to make a decision. To obtain individual Q's, in the case of table representation, it is a matter of index addressing which can be very fast. In the case of neural network representation, it depends on the size of the network. In our case, approximately $2 \times (3+1) \times 8 = 64$ operations (multiplications or additions) were required. Notice that network size is independent of the number of channels M and the number of cells N. Therefore, the total number of operations needed to assign a channel are $(N_{\text{ava}} - 1) + N_{\text{ava}} \times 1$ for the table representation and $(N_{\text{ava}} - 1) + N_{\text{ava}} \times 64$ for the neural network case as shown in Table I. As an example, 19 and 649 operations (comparisons, additions, or multiplications) will be needed in the table and neural network cases, respectively, if we assume that ten channels are available.

The complexity of the MAXAVAIL scheme depends on the number of channels, the number of the cells, and the number of interfering cells. Besides N_{ava} comparisons, for each available channel the availability of that channel in each of N cells is checked. For each cell, N' interfering cells (in our case N' can be 18) have to be visited to determine the channel status in that cell, requiring roughly M *Or* operations and M *addition* operations for each visit. Thus, the total number of operations needed to assign a channel is $(N_{\text{ava}} - 1) + N_{\text{ava}} \times N \times (N' \times (M + M))$ as given in Table I. If we assume again that ten channels are available, the number of operations using the MAXAVAIL scheme would be $9 + 10 \times 49 \times (18 \times (70 + 70)) \approx 1.23 \times 10^6$.

In terms of storage requirement, however, the MAXAVAIL method possesses lowest number of memory units since it does not need to memorize much knowledge. The table-based Q learning requires a higher number of memory units, the maximum of which in our case is $70 \times 70 \times 49 = 240100$, whereas $(3 + 1) \times 8 = 32$ memory units are needed to story the weights in the case of the neural network-based Q-learning approach. It should be mentioned that it is highly possible to reduce the storage requirement of the table-based Q learning by using some localized network such as CMAC, CPN, or RBF network.

IV. Conclusion

We have presented a novel approach to solving the problem of dynamic channel assignment. The optimal assignment policy is obtained by using a self-learning scheme based on Q learning. The real-time simulation studies carried out in a 49-cell mobile communication system have demonstrated that the proposed approach is a practical alternative to existing schemes. In particular, the benefits gained by using the Q-learning-based approach are as follows. First, the learning approach provides a realistic, systematic, and simple way to obtain an approximate optimal solution to the channel assignment problem for which an optimal solution can be very difficult to find using traditional methods. Second, the learned knowledge can be stored in a knowledge representation mechanism such as a neural network or a lookup table which is able to perform the DCA task efficiently in the sense that the desired channel can be determined with very little computational effort. Third, since the proposed learning scheme is performed in a real-time environment, it is possible to carry out online learning while performing the real-time assignment task. In this way, any unforeseen event occurring due to significant variations in the environment conditions, such as traffic or interference conditions can be considered as new experiences that the system could utilize for improving its learning quality. Finally, comparative studies with the FCA and the MAXAVAIL-based DCA algorithm have shown that the Q-learning-based DCA is able to perform better than the FCA in different situations, including the traffic load being spatially uniformly and nonuniformly distributed, and being time varying. Also, the new approach is capable of achieving a performance similar to that achieved by the one of the best known DCA algorithms, MAXAVAIL. However, the online computational efficiency of the proposed approach is far better than that of the MAXAVAIL. This is a definite advantage of our approach since the time efficiency can be a critical issue in real-time implementation.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their valuable comments and suggestions, which have helped improve the quality of this paper.

REFERENCES

[1] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using real-time dynamic programming," *Artificial Intelligence*, vol. 72, pp. 81–138, 1995.

[2] R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.

[3] P. T. Chen, M. Palaniswami, and D. Everitt, "Neural network-based dynamic channel assignment for cellular mobile communication systems," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 279–288, 1994.

[4] J. Chuang, "Performance issues and algorithms for dynamic channel assignments," *IEEE J. Select. Areas Comm.*, vol. 11, pp. 955–963, 1993.

[5] 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.

[6] E. Del Re, R. Fantacci, and L. Ronga, "A dynamic channel allocation technique based on Hopfield neural networks," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 26–32, 1996.

[7] D. D. Dimitrijevic and J. Vucetic, "Design and performance analysis of the algorithms for channel allocation in cellular networks," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 526–534, 1993.

[8] M. Duque-Anton, D. Kunz, and B. Ruber, "Channel assignment for cellular radio using simulated annealing," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 14–21, 1993.

[9] R. L. Freeman, *Telecommunication System Engineering*, 3rd ed. New York: Wiley, 1996.

[10] Y. Furuya and Y. Akaiva, "Channel segregation: A distributed adaptive channel allocation scheme for mobile communications systems," in *DMR II*, Stockholm, Sweden, 1987, pp. 311–315.

[11] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.

[12] D. Kunz, "Channel assignment for cellular radio using neural networks," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 188–193, 1991.

[13] W. C. Y. Lee, *Mobile Cellular Telecommunications*. New York: McGraw-Hill, 1995.

[14] W. K. Lai and G. G. Coghill, "Channel assignment through evolutionary optimization," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 91–96, 1996.

[15] V. H. Macdonald, "The cellular concept," *Bell Syst. Tech. J.*, vol. 58, pp. 15–41, 1979.

[16] R. W. Nettleton and G. R. Schloemer, "A high capacity assignment method for cellular mobile telephone systems," in *Proc IEEE 39th Veh. Technol. Conf.*, 1989, pp. 359–367.

[17] K. N. Sivarajan, R. J. McEliece, and J. W. Ketchum, "Dynamic channel assignment in cellular radio," in *Proc. IEEE 40th Veh. Technol. Conf.*, 1990, pp. 631–637.

[18] S. Tekinary and B. Jabbari, "Handover and channel assignment in mobile cellular networks," *IEEE Commun. Mag.*, pp. 42–46, Nov. 1991.

[19] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Cambridge Univ., Cambridge, U.K., 1989.

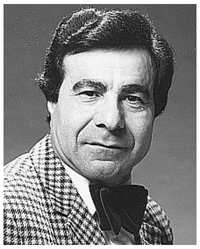
[20] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[21] M. Zhang and T. S. Yum, "Comparisons of channel assignment strategies in cellular mobile systems," *IEEE Trans. Veh. Technol.*, vol. 38, pp. 211–215, 1989.

[22] ——, "The nonuniform compact pattern allocation algorithm for cellular mobile systems," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 387–391, 1991.

Junhong Nie received the Ph.D. degree in systems and control engineering from the University of Sheffield, U.K., for work on fuzzy-systems and control.

He served as a Lecturer at the Northwest Telecommunication Engineering Institute of China from 1985 to 1989. During 1993–1996, he was a Research Scientist in the Department of Electrical Engineering, National University of Singapore. Since February 1996, he has been with the Communication Research Laboratory, McMaster University, Hamilton, Ont., Canada. He is the author of the book *Fuzzy-Neural Control: Principles, Algorithms, and Applications* (U.K.: Prentice-Hall). His research interests include fuzzy and neural computing and learning and adaptive approaches with applications to signal processing and mobile communications.



Simon Haykin (F'82) received the B.Sc. (first-class honors) degree in 1953, the Ph.D. degree in 1956, and the D.Sc. degree in 1967, all in electrical engineering, from the University of Birmingham, Birmingham, U.K.

He is the Editor for *Adaptive and Learning Systems for Signal Processing, Communications and Control*, a new series of books for Wiley-Interscience. He is the Founding Director of the Communications Research Laboratory, McMaster University. His research interests include nonlinear dynamics, neural networks, adaptive filters, and their applications in radar and communication systems.

Dr. Haykin was elected Fellow of the Royal Society of Canada in 1980. He received the McNaughton Gold Medal (IEEE Region 7) in 1986. He is the recipient of the Canadian Telecommunications Award from Queen's University and in 1996 was awarded the title "University Professor."