Adaptive Uniform Fractional Channel Algorithms

H. Beigy and M. R. Meybodi

***Abstract*—Dropping probability of handoff calls and blocking probability of new calls are two important QoS measures for cellular networks. Since the dropping probability of handoff calls is more important, call admission policies are used to maintain the upper bound of dropping probability of handoff calls. The fractional guard channel policy (FG) is a general call admission policy and includes most prioritized channel allocation schemes such as guard channel (GC), limited fractional guard channel (LFG), and uniform fractional channel policy (UFC) policies. Since the input traffic is not a stationary process and its parameters are unknown a priori, the optimal value of UFC parameter is not known in advance and possibly varies as traffic conditions change. In this paper, we propose two adaptive algorithms based on learning automata for finding the optimal value of UFC parameter. To evaluate the proposed algorithms, the computer simulations are conducted. The simulation results show that for some range of input traffic, the performance of the proposed algorithms is close to the performance of the uniform fractional channel policy which knows the traffic parameters a priori.**

***Index Terms*—Cellular mobile networks, guard channel policy, uniform fractional channel policy, learning automata, adaptive uniform fractional guard channel**

I. Introduction

In recent years, there has been a rapid growth in the number of mobile communication networks users. However, the bandwidth allocated to the mobile communication networks is very limited [1]. This limitation means that the frequency channels, or simply channels, have to be reused as much as possible in order to support a numerous number of simultaneous calls that may arise in any typical mobile communication network. Thus, the efficient management and sharing of channels among numerous users become an important issue. In order to reuse channels, micro cellular networks are introduced. In these networks, the geographical area covered by the network is divided into smaller regions, which are called cells. Each cell has a base station, located at its center, which is used to serve the users located at that cell. In order to enable a mobile user to communicate with other user(s), a connection usually must be established between the users. When a mobile user needs a connection, sends his/her request to the base station of the cell residing it. Then, the base station determines whether it can meet the requested quality of service (QoS) requirements and, if possible, allocates a channel to the incoming call and establishes a connection.

When a call gets a channel, it will keep the channel until its completion, or until the mobile user moves out of the cell, in which case the used channel will be released. When the mobile user moves into a new cell while its call is ongoing, a new channel needs to be acquired in the new cell for further communication. This process is called *handoff* and must be transparent to the mobile user. During the handoff, if there is no channel available in the new cell for the ongoing call, it is forced to terminate (dropped) before its completion. The disconnection in the middle of a call is highly undesirable and one of the goals of the network designer is to keep such disconnections as low as possible

Introduction of micro cellular networks leads to efficient use of channels but increases expected rate of handovers per call. As a consequence, some network performance parameters, such as *blocking probability of new calls* (B_n) and *dropping probability of handoff calls* (B_h), are affected. These two parameters are interdependent. For example, accepting more handoff calls increases the blocking probability of new calls and vice versa. As a result, there is a trade-off between these two performance parameters. In order to keep these performance parameters at a reasonable level, *call admission policies* are used. The call admission policy plays a very important role in the cellular networks because it directly controls B_n and B_h. Call admission policies control B_n and B_h by putting some restrictions on the channel allocation to calls. Since the dropping probability of handoff calls is more important than the blocking probability of new calls, call admission policies usually give the higher priority to handoff calls. This priority is implemented through allocation of more resources (channels) to handoff calls.

A general call admission policy, which is called *fractional guard channel policy* (FG), accepts new calls with a certain probability that depends on the current channel occupancy and accepts handoff calls as long as channels are available [2]. Suppose that the given cell has C full duplex channels. The FG policy uses a vector $\mathbf{\Pi} = [\pi_0, ..., \pi_{C-1}]$ to accept the new calls, where $0 \le \pi_k \le 1$ (for $k = 0,...,C-1$). This policy accepts new calls with probability π_k when k (for $k = 0,...,C-1$) channels are busy. Depending on $\mathbf{\Pi}$, we may have different call admission policies and some of which are reviewed below.

A restricted version of FG is called *guard channel policy* (GC) [3]. The guard channel policy reserves a subset of channels allocated to a cell, called *guard channels*, for handoff calls (say $C-T$ channels). Whenever the channel occupancy exceeds a certain threshold T, the guard channel policy rejects new calls until the channel

Manuscript received February 3, 2003; revised February 28, 2004. This work was partially supported by Iranian Telecommunication Research Center (ITRC), Tehran, Iran.

The authors are with the Soft Computing Laboratory, Computer Engineering Department, Amirkabir University of Technology, Tehran, I. R. Iran (e-mail: beigy@ce.aut.ac.ir, meybodi@ce.aut.ac.ir).

Publisher Item Identifier S 1682-0053(04)0197

1682-0053/04$10 © 2004 JD

occupancy goes below the threshold. The guard channel policy accepts handoff calls as long as channels are available. Note that the GC policy can be obtained from FG policy by setting $\pi_k = 1$, (for $k = 0,...,T-1$), and $\pi_k = 0$, (for $k = T,...,C-1$). It has been shown that there is an optimal threshold T^* at which the blocking probability of new calls is minimized subject to the hard constraint on the dropping probability of handoff calls and an algorithm for finding such optimal threshold is given in [4].

The GC policy reserves an integral number of guard channels for handoff calls. If performance parameter B_h is considered, the guard channel policy gives very good performance, but performance parameter B_n is degraded to a great extent. In order to have more control on blocking probability of new calls and dropping probability of handoff calls, *limited fractional guard channel policy* (LFG) is introduced [2]. The LFG can be obtained from FG policy by setting $\pi_k = 1$, (for $k = 0,...,T-1$), $\pi_T = \pi$ and $\pi_k = 0$, (for $k = T+1,...,C-1$). It has been shown that there is an optimal threshold T^* and an optimal value of π^* for which the blocking probability of new calls is minimized subject to the hard constraint on dropping probability of handoff calls. An algorithm for finding these optimal parameters is also reported in [2].

Another version of FG policy, called uniform fractional channel policy (UFC) is reported in [5]. The UFC policy accepts new calls with probability of π_k independent of channel occupancy and accepts handoff calls as long as free channels are available. The UFC can be obtained from FG by setting $\pi_k = \pi$ (for $k = 0,...,C-1$). It is shown that there is an optimal π^* which minimizes the blocking probability of new calls with the constraint on the upper bound on dropping probability of handoff calls and an algorithm for finding the value of optimal parameter is also given. It is also shown that the UFC policy performs superior than GC policy under the low handoff traffic conditions. There are some call admission policies which allow either handoff or new calls to be queued until free channels are obtained in the cell [6], [7].

All above mentioned call admission policies are static and assume that all parameters of traffic are known in advance. These policies are useful when input traffic is a stationary process with known parameters. Since the parameters of input traffic are unknown and possibly time varying, adaptive version of these policies need to be used.

In this paper, we propose two adaptive UFC algorithms. These algorithms use a learning automaton to accept/reject new calls and the pre-specified level of dropping probability of handoff calls is used to penalize/reward the action selected by the learning automaton. The simulation results show that the performance of the first adaptive UFC algorithm (AUFC I) is close to the performance of UFC algorithm in low handoff traffic conditions, but the constraint on the dropping probability of handoff calls is not maintained. The second adaptive algorithm (AUFC II) maintains the constraint on the dropping probability of handoff calls and its performance is close to the performance of UFC policy in high traffic conditions.

The rest of this paper is organized as follows. The learning automata are given in Section II. Section III presents the UFC policy, its performance parameters and

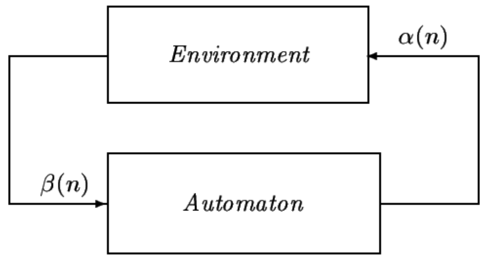

Fig. 1. The interaction of automata and its environment.

an algorithm to find the optimal value of its parameter. Two adaptive algorithms for finding the optimal value of parameter π are given in Section IV. The computer simulations are given in Section V and finally Section VI concludes the paper.

II. Learning Automata

The automata approach to learning involves determination of an optimal action from a set of allowable actions. An automaton can be regarded as an abstract object that has finite number of actions. It selects an action from its finite set of actions and applies to a random environment. The random environment evaluates the applied action and gives a grade to the selected action of automaton. The response from the environment (i.e. grade of action) is used by automaton to select its next action. By continuing this process, the automaton learns to select an action with the best grade. The learning algorithm used by automaton to determine the selection of next action from the response of environment. The interaction of an automaton with its environment is shown in Fig. 1.

An automaton acting in an unknown random environment and improves its performance in some specified manner, is referred to as *learning automaton* (LA). Learning automata can be classified into two main families: *fixed structure learning automata* and *variable structure learning automata* [8]. Variable structure learning automata are represented by triple $<\beta,\alpha,T>$, where β is a set of inputs, α is a set of actions, and T is learning algorithm. The learning algorithm is a recurrence relation and is used to modify action probabilities (p) of the automaton. It is evident that the crucial factor affecting the performance of the variable structure learning automata, is learning algorithm for updating the action probabilities. Various learning algorithms have been reported in the literature [8]. Let α_i be the action chosen at time k as a sample realization from distribution $p(k)$. In what follows, two learning algorithms for updating the action probability vector are given. In linear reward-εpenalty algorithm ($L_{R-\varepsilon P}$) scheme the recurrence equation for updating p is defined as

$$p_j(k) = \begin{cases} p_j(k) + a\left(1 - p_j(k)\right) & \text{if i = j} \\ p_j(k)(1-a) & \text{if i} \neq \text{j} \end{cases} \quad (1)$$

if $\beta(k) = 0$ and

$$p_j(k) = \begin{cases} p_j(k)\left(1 - \text{b}\right) & \text{if i = j} \\ \dfrac{b}{\text{r - 1}} + (1 - \text{b})\, p_j(k) & \text{if i} \neq \text{j} \end{cases} \quad (2)$$

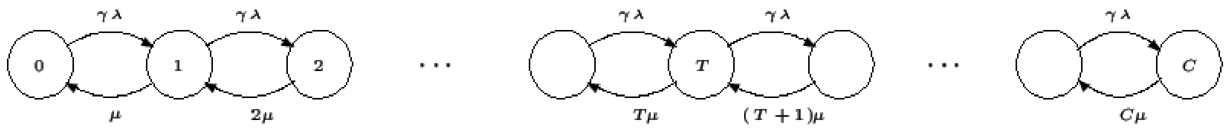

Fig. 2. Markov chain model of cell.

if $\beta(k)=1$. The parameters $0<b<<a<1$ represent *step lengths* and r is the number of actions for learning automaton. a and b determine the amount of increase and decreases of the action probabilities, respectively. If a equals to b, recurrence (1) and (2) are called (L_{R-P}) algorithm.

Learning automaton have been used successfully in many applications, such as telephone and data network routing [9], [10], solving NP-Complete problems [11]-[13], capacity assignment [14] and neural network engineering [15]-[18] to mention a few.

III. Uniform Fractional Channel Policy

In this section, we first review fractional guard channel, guard channel, limited fractional guard channel and uniform fractional channel policies and then compute the blocking performance of the UFC policy and finally give an algorithm for finding the optimal value of π. We assume that a given cell has a limited number of full duplex channels, C, in its channel pool. We define the state of a particular cell at time t to be the number of busy channels in that cell and is represented by $c(t)$.

A. Guard Channel Policy

The guard channel policy reserves a subset of channels allocated to a particular cell for handoff calls (say $C-T$ channels). Whenever the channel occupancy exceeds a certain threshold T, the guard channel policy rejects new calls until the channel occupancy goes below the threshold. The guard channel policy accepts handoff calls as long as channels are available.

B. Fractional Guard Channel Policy

The fractional guard channel policy is a general call admission policy. This policy accepts handoff calls as long as channels are available. This policy uses vector $\mathbf{\Pi}=[\pi_0,...,\pi_{C-1}]$ to accept new calls. That is, a new call will be accepted with probability π_k when k channels in the cell are occupied.

C. Limited Fractional Guard Channel Policy

The limited fractional guard channel policy can be obtained from FG policy by setting $\pi_k=1$, (for $k=0,...,T-1$), $\pi_T=\pi$ and $\pi_k=0$, (for $k=T+1,...,C-1$). In LFG a fractional number of channels is reserved in each cell exclusively for handoff calls. The LFG scheme uses an additional parameter π_T and operates same as the guard channel policy except when T channels are occupied in the cell, in which case new calls are accepted with probability π_T. The limited fractional guard channel policy accepts handoff calls as long as channels are available.

D. Uniform Fractional Channel Policy

The UFC policy uses new call admission probability π independent of channel occupancy to accept new calls. This policy accepts handoff calls as long as channels are available. This policy can be obtained from FG policy by setting $\pi_k=\pi$, (for $k=0,...,C-1$). UFC policy reserves non-integral number of guard channels for handoff calls by rejecting new calls with some probability. In the next subsection, we review the blocking performance of UFC and then a binary search algorithm to find the optimal value of parameter π. The analysis given in [5] has shown that the UFC policy has a lower blocking probability for new calls in low handoff/new calls traffic ratio.

1) Blocking Performance of UFC

In what follows, we study the blocking performance of the UFC policy. We consider a homogenous wireless network where all cells have the same number of channels C and experience the same new and handoff calls arrival rates. In each cell, the arrival of new calls and handoff calls are Poisson distributed with arrival rates λ_n and λ_h, respectively. In each cell, the channel holding time of new and handoff calls are exponentially distributed with mean μ^{-1}. Note that the same service rate for both types of calls implies that the base station of a cell does not need to discriminate between new and handoff calls, once they are connected. This set of assumptions have been found reasonable as long as the number of mobile users in a cell is much greater than the number of channels allocated to that cell. Let $\lambda=\lambda_n+\lambda_h$, $a=\lambda_h/\lambda$, and $\rho=\lambda/\mu$. We define the state of a particular cell at time t represented by $c(t)$ to be the number of busy channels in that cell. $\{c(t)\}_{t\geq 0}$ is a continuous-time Markov chain (birth-death process) with states $0,1,...,C$. The state transition rate diagram of a cell with C full duplex channels and UFC call admission policy is shown in Fig. 2.

At state n (for $0\leq n<C$), new calls are accepted with probability π (for $0\leq\pi\leq 1$) and handoff calls are accepted with probability 1. Both types of calls are blocked when the cell is in state C. Thus, the state dependent arrival rate in the birth-death process is equal to $[a+(1-a)\pi]\lambda$. Define the steady state probability

$$P_n=\lim_{n\to\infty}\text{Prob}[c(t)=n] \quad \text{for } n=0,1,...,C. \tag{3}$$

By solving the balance equations of the Markov chain, the following expression can be derived for P_n ($n=0,1,...,C$)

$$P_n=\frac{(\gamma\rho)^n}{n!}P_0, \tag{4}$$

where $\lambda=[a+(1-a)\pi]$ and P_0 is the probability that all channels are free and obtained using $\sum_{n=0}^{C}P_n=1$.
The value of P_0 is calculated by

$$P_0=\left[\sum_{n=0}^{C}\frac{(\gamma\rho)^n}{n!}\right]^{-1}. \tag{5}$$

Given the state probability vector, we can find the dropping probability of handoff calls, $B_h(C,\pi)$, by

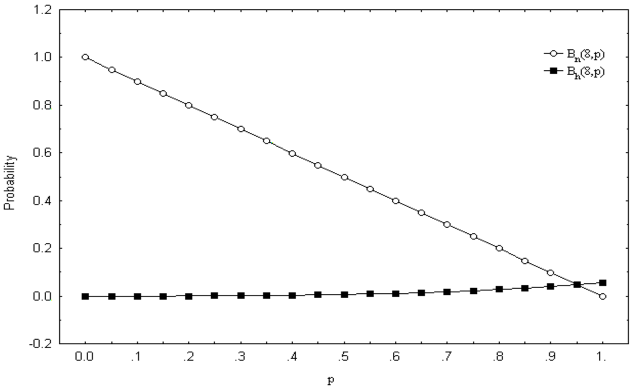

Fig. 3. The effect of π on the blocking probabilities ($\lambda_h = 10$).

$$B_h(C,\pi) = P_C = \frac{(\gamma\rho)^C}{C!} P_0 . \tag{6}$$

Similarly, the blocking probability of new calls, $B_n(C,\pi)$, is given by

$$B_n(C,\pi) = (1-\pi)\sum_{n=0}^{C-1} P_n + P_C = 1-\pi[1-B_h(C,\pi)] . \tag{7}$$

$B_h(C,\pi)$ and $B_n(C,\pi)$ have interesting properties, which are listed below. The proof for these properties can be found in [5].

Property 1: For any given constraint $\rho / C < 1$, $B_h(C,\pi)$ is a monotonically increasing function of π .

Property 2: For any given constraint $C < \lambda_h / \lambda p_h$, $B_n(C,\pi)$ is a monotonically decreasing function of π .

A graph of $B_h(C,\pi)$ and $B_n(C,\pi)$ functions versus π is shown in Fig. 3. The traffic parameters correspond to those of case 5 in Table I, which satisfies these properties.

In what follows, we consider the problem of finding the optimal value of new call admission probability (π^*) when the number of channels allocated to a particular cell is held fixed. Given C channels allocated to a cell, the objective is to find π^* that minimizes $B_n(C,\pi)$ subject to the hard constraint $B_h(C,\pi) \le p_h$. In order to find the solution for this problem, we first study the existence and uniqueness of the solution by the following theorem.

Theorem 1: Let $B_h(C,\pi) \le p_h$ and $B_h(C,1) \ge p_h$, then there exists a unique π^* that minimizes $B_h(C,\pi)$ while the constraint $B_h(C,\pi) \le p_h$ is satisfied.

Proof: Define $B(\pi) = B_h(C,\pi) - p_h$ and consider $B(\pi)$ at its two end points. Thus, we have

$$B(\pi) = \begin{cases} B_h(C,0) - p_h \le 0 & \pi = 0 \\ B_h(C,1) - p_h \ge 0 & \pi = 1 \end{cases} . \tag{8}$$

Since $B(.)$ is a continuous function of π, then there exists at least one π^* such that $B(\pi^*) = 0$. Uniqueness follows since $B(.)$ is a strictly increasing function of π (property 1). Since $B_n(C,\pi)$ is a strictly decreasing function of π (property 2), value π^* minimizes B_n subject to the hard constraint on $B_h(C,\pi) \le p_h$.

Condition $B_h(C,0) \le p_h$ in the above theorem implies that when all channels allocated to the cell are used for handoff calls, the level of QoS is satisfied and condition $B_h(C,1) \ge p_h$ implies that a fraction of channels must be reserved for handoff calls or in other words a higher priority must be given to the handoff calls.

```
set upper ←1; lower ← 0
if ( Bh(C,1) ≤ ph ) then
      return 1
end if
if ( Bh(C,0) ≥ ph ) then
      return 0
end if
while ((upper - lower) < 0.0001) do
     set π ←(upper + lower) / 2
     if ( Bh(C,π) > ph ) then
          set upper ← π
     else
          set lower ← π
     end if
end while
return π
```

Fig. 4. Algorithm for determination of π^*.

In what follows, we give an algorithm for finding π^*. This algorithm is given in Fig. 4 and can be described as follows. At first, the algorithm considers the case when all channels are shared between handoff and new calls. If the complete sharing does not satisfy the level of QoS, then the algorithm considers the case when all channels are exclusively used for handoff calls. If the exclusive use of channels for handoff calls does not satisfy the level of QoS, then the number of allocated channels to the cell is not sufficient and the algorithm terminates; otherwise the algorithm searches for the optimal value of π^*. The search method used in this algorithm is binary search.

The following theorem is concerned with the optimality of the solution found by the algorithm given in Fig. 4.

Theorem 2: Let $B_h(C,0) \le p_h$ and $B_h(C,1) \ge p_h$ then the algorithm given in Fig. 4 minimizes the value of $B_n(C,\pi)$ while the constraint $B_h(C,\pi) \le p_h$ is satisfied.

Proof: Using theorem 1, it follows that there is a unique π^* which satisfies the conditions of the theorem. Since $B_h(C,\pi)$ is strictly increasing and $B_n(C,\pi)$ is strictly decreasing, both with respect to π, the largest value of π that satisfies condition $B_h(C,\pi) \le p_h$ is the optimal solution. Hence the algorithm starts with the largest value of π, which is one, and uses binary search to find π^*.

IV. Adaptive Uniform Fractional Channel Algorithms

In this section, we introduce two adaptive uniform fractional channel algorithms. These algorithms are used to determine admission probability π when the parameters a and ρ (or equivalently λ_h, λ_n or μ) are unknown or probably time varying. The proposed algorithms adjust new call admission probability, π, as network operates. These algorithms use reward-penalty type learning automata with two actions in each cell. The action set of these automata correspond to {ACCEPT, REJECT}. Since values of a and ρ are unknown, initially the probability of selecting the actions of automaton are set to 0.5. In the rest of this section, we present these two algorithms and then study their behaviors.

A. Adaptive Uniform Fractional Channel Algorithm I

The first algorithm is shown in Fig. 5 and can be described as follows. When a handoff call arrives, it is accepted as long as there is a free channel. If there is no free channel, the handoff call will be dropped. When a new call arrives to a particular cell, the learning automaton associated to that cell chooses one of its actions. Let π be the probability of selecting the action ACCEPT. The learning automaton chooses action ACCEPT with probability π and action REJECT with probability $1-\pi$. If action ACCEPT is selected by the automaton and the cell has at least one free channel, then the incoming call is accepted and action ACCEPT is rewarded. If there is no free channel to be allocated to the arrived new call, the call is blocked and the action ACCEPT is penalized. When the automaton selects action REJECT, then the base station computes an estimation of the dropping probability of handoff calls (b_h) and uses it to decide whether or not accept new calls. If the current estimate of dropping probability of handoff calls is less than the given threshold p_h and there is a free channel, then the new call is accepted; otherwise, the new call is rejected. When the automaton selects action REJECT, then the base station waits until the arrival of the next new call. Then the algorithm computes an estimation of the dropping probability of handoff calls (b_h) and uses it to reward or punish the selected action. If the current estimate of dropping probability of handoff calls is less than the given threshold p_h and the new call is accepted then action REJECT is penalized; otherwise, action REJECT is rewarded.

1) Blocking Performance

Now, we study the blocking performance of adaptive UFC Algorithm I. The blocking performance of this algorithm is computed based on the assumptions given in subsection III.D.1. Define $\delta = \Pr ob[b_h < p_h]$ and $\sigma = a + (1-a)\pi + (1-\pi)(1-a)\delta$. The state dependent arrival rate in the birth-death process for AUFC I is equal to $\sigma\lambda$. Comparing σ of AUFC I with γ of UFC policy reveals the fact that AUFC I accepts more new calls than UFC. This results in an increase in $B_h(C,\pi)$ and a decrease in $B_h(C,\pi)$ for AUFC I. The state transition diagram of adaptive UFC Algorithm I is shown in Fig. 6.

By writing down the balance equations for the steady-state probabilities P_n ($n = 0,1,\ldots,C$), we obtain $\sigma\lambda P_{n+1} = n\mu P_n$. Then, the following expression can be derived for P_n ($n = 0,1,\ldots,C$)

$$P_n = \frac{(\sigma\rho)^n}{n!} P_0, \tag{9}$$

where

$$P_0 = \left[\sum_{n=0}^{C} \frac{(\sigma\rho)^n}{n!}\right]^{-1}. \tag{10}$$

Given the state probabilities, we can find the dropping probability of handoff calls, $B_h(C,\pi)$, by

$$B_h(C,\pi) = P_C = \frac{(\gamma\rho)^C}{C!} P_0. \tag{11}$$

Similarly, the blocking probability of new calls, $B_n(C,\pi)$, is given by

$$B_n(C,\pi) = (1-\pi)\sum_{n=0}^{C-1} P_n + P_C = 1 - \pi[1 - B_h(C,\pi)]. \tag{12}$$

```
if (NEW CALL) then
    if (action of learning automaton is ACCEPT) then
        if ( c(t) < C ) then
            accept call and reward action ACCEPT
        else
            reject call and penalize action ACCEPT
        end if
    else
        if ( c(t) < C  and  b_h < p_h ) then
            accept call
        else
            reject call
        end if
        upon the arrival of next new call compute b_h
        if (new call is accepted and b_h < p_h ) then
            penalize action REJECT
        else
            reward action REJECT
        end if
    end if
end if
```

Fig. 5. Adaptive uniform fractional channel Algorithm I.

B. Adaptive Uniform Fractional Channel Algorithm II

The simulation results for AUFC I show that this algorithm cannot maintain the specific level of QoS for the dropping probability of handoff calls. The AUFC II, which is shown in Fig. 7, aims to overcome this problem by lowering the acceptance rate of new calls. The main difference between this algorithm and AUFC I Algorithm is when the learning automaton selects REJECT as its action. This algorithm can be described as follows. When a handoff call arrives, it is accepted as long as there is a free channel. If there is no free channel, the handoff call is dropped. When a new call arrives to a particular cell, the learning automaton associated to that cell chooses one of its actions.

Let π be the probability of selecting the action ACCEPT. Thus, the learning automaton accepts new calls with probability π as long as there is a free channel and rejects new calls with probability $1-\pi$. If action ACCEPT is selected by automaton and the cell has at least one free channel, the incoming call is accepted and the selected action is rewarded. If there is no free channel to be allocated to the arrived new call, the call is blocked and action ACCEPT is penalized. When the automaton selects action REJECT, then the new call is rejected and the base station waits until the arrival of the next new call. Then the algorithm computes an estimation of the dropping probability of handoff calls (b_h) and uses it to reward or punish the selected action. If the current estimate of dropping probability of handoff calls is less than the given threshold p_h, then action REJECT is penalized; otherwise, action REJECT is rewarded.

V. Simulation Results

In this section, through simulation we compare the performance of the limited fractional guard channel, the guard channel and the uniform fractional channel policies, and the proposed adaptive uniform fractional channel algorithms. The results of simulations are summarized in Tables I and II. Simulation are conducted based on the single cell of a homogenous cellular network system. In the network, each cell has 8 full duplex channels ($C = 8$).

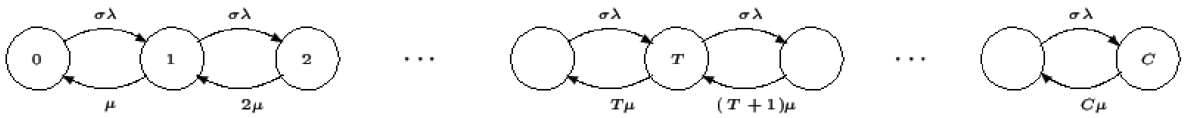

Fig. 6. Markov chain model of cell.

```
if (NEW CALL) then
    if (action of learning automaton is ACCEPT) then
        if ( c(t) < C ) then
            accept call and reward action ACCEPT
        else
            reject call and penalize action ACCEPT
        end if
    else
        reject call
        upon the arrival of next new call compute b_h
        if ( b_h < p_h ) then
            penalize action REJECT
        else
            reward action REJECT
        end if
    end if
end if
```

Fig. 7. Adaptive uniform fractional channel Algorithm II.

In all simulations, new call arrival rate is fixed to 30 calls per minute ($\lambda_n = 30$), channel holding time is set to 6 seconds ($\mu^{-1} = 6$), and the handoff call traffic is varied between 2 calls per minute to 20 calls per minute.

The results listed in Tables I and II are obtained by averaging 10 runs from 2,000,000 seconds simulation of each algorithm. The level of QoS for the dropping probability of handoff calls is set to 0.01. The optimal parameters for the limited fractional guard channel, the guard channel and the uniform fractional channel policies are obtained by using the algorithms given in [2], [4], [5], respectively. In Table I, the guard channel, the limited fractional guard channel and the uniform fractional channel policies are compared and in Table II, the uniform fractional channel policy and the two adaptive uniform fractional channel algorithms are compared. We use three performance measures for comparison: the blocking probability of new calls (B_n), the dropping probability of handoff calls (B_h), and normalized channel utilization (ξ). The normalized channel utilization is defined as the ratio of the average number of busy channels in a cell to the number of channels allocated to that cell, that is, the percentage of busy channels in each cell.

By inspecting Table I, it is evident that the blocking probability of new calls for uniform fractional channel policy is less than the blocking probability of new calls for guard channel and limited fractional guard channel policy when the handoff traffic is low. But its performance degrades as handoff traffic increases. The results of simulations given in Table I also show that the normalized channel utilization for UFC is greater than the normalized channel utilization for the guard channel policy when the traffic for handoff is low, but decreases monotonically as the handoff traffic increases.

Careful inspection of Table II, reveals the fact that AUFC I has lower blocking probability of new calls comparing to UFC at the expense of failing to maintain the level of QoS. This table also shows that at high handoff traffic conditions the performance of the AUFC II is close to the performance of the UFC policy. Since in the low

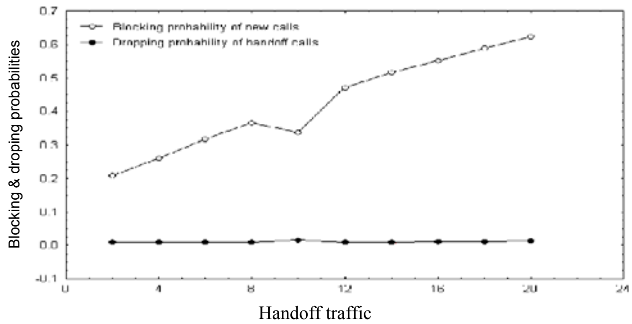

Fig. 8. Performance of the adaptive UFC Algorithm II for different handoff traffic.

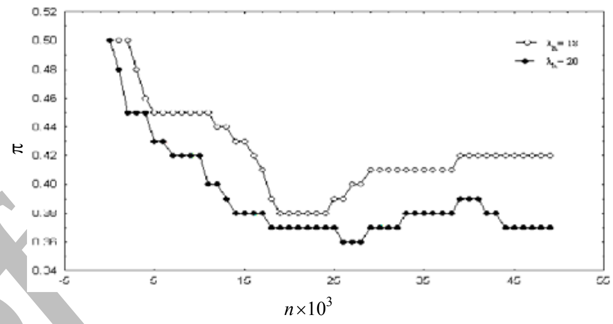

Fig. 9. Convergence of the proposed algorithm for different handoff traffic.

handoff traffic conditions, the UFC policy does not maintain the upper bound on the dropping probability of handoff calls, the blocking probability of new calls for the proposed algorithms is greater than the blocking probability of new calls for UFC. When the handoff traffic becomes high, the UFC policy maintains the upper bound on the dropping probability of handoff calls and the performance of UFC policy and that of the AUFC II is very close. Fig. 8 shows the performance of the adaptive UFC Algorithm II under different handoff traffics when the other parameters of the cell are fixed. Note that the level of QoS is maintained by the AUFC II for various handoff traffic conditions.

Fig. 9 shows B_n and B_h for the adaptive uniform fractional channel Algorithm II for different handoff traffic load. The traffic parameters used for Fig. 9 corresponds to cases 9 and 10 in Table II. Careful inspection of Fig. 9 shows that the admission probability, π, converges to its optimal value.

VI. CONCLUSIONS

In this paper, we introduced two learning automata based algorithms for finding the optimal value for the parameter of uniform fractional channel policy. The simulation results showed that for some range of input traffics, the performance of the proposed algorithms is close to the performance of the uniform fractional guard channel policy which knows the traffic parameters a priori.

TABLE I
COMPARISON OF SIMULATION RESULTS FOR CALL ADMISSION ALGORITHMS

		LFG			GC			UFC		
Case	λ_h	B_n	B_h	ξ	B_n	B_h	ξ	B_n	B_h	ξ
1	2	0.031609	0.023283	0.646076	0.063507	0.001525	0.627121	0.023935	0.024675	0.635954
2	4	0.051414	0.020675	0.660527	0.077080	0.003538	0.646112	0.089897	0.023639	0.623986
3	6	0.071632	0.018707	0.672639	0.091013	0.005923	0.662472	0.15725	0.022202	0.610485
4	8	0.092138	0.016706	0.684128	0.105002	0.008380	0.677965	0.223872	0.020367	0.597570
5	10	0.114445	0.015572	0.694984	0.120260	0.011877	0.692446	0.289849	0.019248	0.586468
6	12	0.147902	0.014044	0.699548	0.231559	0.004309	0.654597	0.356866	0.017607	0.575185
7	14	0.204217	0.012675	0.692473	0.255346	0.005975	0.666051	0.424072	0.016390	0.565458
8	16	0.250642	0.011554	0.688376	0.275489	0.007999	0.676061	0.489967	0.015076	0.554360
9	18	0.294441	0.010877	0.687161	0.296834	0.010518	0.686021	0.557026	0.013939	0.544336
10	20	0.384157	0.010182	0.664995	0.459183	0.006081	0.627354	0.623746	0.013318	0.536724

TABLE II
COMPARISON OF SIMULATION RESULTS OF UFC AND THE PROPOSED ADAPTIVE UFC ALGORITHMS

		UFC			AUFC I			AUFC II		
Case	λ_h	π^*	B_n	B_h	π^*	B_n	B_h	π^*	B_n	B_h
1	2	0.975952	0.023935	0.024675	0.925530	0.088148	0.019475	0.760948	0.208524	0.010001
2	4	0.909302	0.089897	0.023639	0.896014	0.119601	0.021635	0.704340	0.260971	0.010001
3	6	0.842407	0.157250	0.022202	0.682171	0.315232	0.010064	0.636622	0.318610	0.010003
4	8	0.775757	0.223872	0.020367	0.735226	0.272205	0.016551	0.617122	0.365239	0.010003
5	10	0.709106	0.289849	0.019248	0.655973	0.357947	0.014037	0.671225	0.337246	0.015549
6	12	0.642456	0.356866	0.017607	0.626881	0.386985	0.015565	0.547896	0.469893	0.010003
7	14	0.575806	0.424072	0.016390	0.538180	0.476427	0.012920	0.480809	0.515486	0.010290
8	16	0.509155	0.489967	0.015076	0.526355	0.483954	0.015869	0.451857	0.550553	0.011164
9	18	0.442505	0.557026	0.013939	0.503609	0.499460	0.018558	0.412855	0.589336	0.012027
10	20	0.375854	0.623746	0.013318	0.502960	0.501800	0.023435	0.367029	0.623895	0.013519

REFERENCES

[1] I. Katzela and M. Naghshineh, "Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey," *IEEE Personal Communications*, vol. 3, no. 3, pp. 10-31, Jun. 1996.

[2] R. Ramjee, D. Towsley, and R. Nagarajan, "On optimal call admission control in cellular networks," *Wireless Networks*, vol. 3, no. 1, pp. 29-41, Mar. 1997.

[3] D. Hong and S. Rappaport, "Traffic modeling and performance analysis for cellular mobile radio telephone systems with prioritized and non-prioritized handoffs procedure," *IEEE Trans. on Vehicular Tech.*, vol. 35, no. 3, pp. 77-92, Aug. 1986.

[4] G. Haring, R. Marie, R. Puigjaner, and K. Trivedi, "Loss formulas and their application to optimization for cellular networks," *IEEE Trans. on Vehicular Tech.*, vol. 50, no. 3, pp. 664-673, May 2001.

[5] H. Beigy and M. R. Meybodi, "A new fractional channel policy," *Journal of High Speed Networks*, Accepted for Publication, 2004.

[6] C. H. Yoon and C. Kwan, "Performance of personal portable radio telephone systems with and without guard channels," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 6, pp. 911-917, Aug. 1993.

[7] R. Guern, "Queuing-blocking system with two arrival streams and guard channels," *IEEE Transactions on Communications*, vol. 36, no. 2, pp. 153-163, Feb. 1988.

[8] K. S. Narendra and K. S. Thathachar, *Learning Automata: An Introduction*, New York, Printice-Hall, 1989.

[9] P. R. Srikantakumar and K. S. Narendra, "A learning model for routing in telephone networks," *SIAM Journal of Control and Optimization*, vol. 20, no. 1, pp. 34-57, Jan. 1982.

[10] O. V. Nedzelnitsky and K. S. Narendra, "Nonstationary models of learning automata routing in data communication networks," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 17, no. 6, pp. 1004-1015, Nov. 1987.

[11] B. J. Oommen and E. V. de St. Croix, "Graph partitioning using learning automata," *IEEE Trans. on Computers*, vol. 45, no. 2, pp. 195-208, Feb. 1996.

[12] M. R. Meybodi and H. Beigy, "Solving stochastic shortest path problem using distributed learning automata," in *Proc. of 6th Annual Int. CSI Computer Conf. CSICC-2001*, pp. 70-86, Isfehan, Iran, Feb. 2001.

[13] H. Beigy and M. R. Meybodi, "Solving the graph isomorphism problem using learning automata", in *Proc. of 5th Annual Int. CSI Computer Conf., CISCC-2000*, pp. 402-415, Tehran, Iran, Jan. 2000.

[14] B. J. Oommen and T. D. Roberts, "Continuous learning automata solutions to the capacity assignment problem," *IEEE Trans. on Computers*, vol. 49, no. 6, pp. 608-620, Jun. 2000.

[15] M. R. Meybodi and H. Beigy, "Neural network engineering using learning automata: determining of desired size of three layer feedforward neural networks," *Journal of Faculty of Engineering, University of Tehran*, vol. 34, no. 4, pp. 1-26, Mar. 2001.

[16] M. R. Meybodi and H. Beigy, "A note on learning automata based schemes for adaptation of BP parameters," *Journal of Neurocomputing*, vol. 48, no. 3, pp. 957-974, Nov. 2002.

[17] M. R. Meybodi and H. Beigy, "New learning automata based algorithms for adaptation of backpropagation algorithm parameters," *Int. Journal of Neural Systems*, vol. 12, no. 1, pp. 45-68, Feb. 2002.

[18] H. Beigy and M. R. Meybodi, "Backpropagation algorithm adaptation parameters using learning automata," *Int. Journal of Neural Systems*, vol. 11, no. 3, pp. 219-228, Jun. 2001.

Hamid Beigy received the B.Sc. and M.Sc. degrees in Computer Engineering from Shiraz University in Iran, in 1992 and 1995, respectively. Currently, he is a graduate student pursuing Ph.D. degree in Computer Engineering, Amirkabir University of Technology, Tehran, Iran. His research interests include, channel management in cellular networks, learning systems, parallel algorithms, and soft computing.

Mohammad Reza Meybodi received the B.S. and M.S. degrees in Economics from Shahid Beheshti University in Iran, in 1973 and 1977, respectively. He also received the M. S. and Ph.D. degree from Oklahoma University, US, in 1980 and 1983, respectively in computer science. Currently he is a Full Professor in Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran. Prior to current position, he worked from 1983-1985 as an Assistant Professor at Western Michigan University, and from 1985-1991 as an Associate Professor at Ohio University, US. His research interests include, channel management in cellular networks, learning systems, parallel algorithms, soft computing and software development.