



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Computers and Electrical Engineering 31 (2005) 132–151

Computers and
Electrical Engineering

www.elsevier.com/locate/compeleceng

An adaptive call admission algorithm for cellular networks ☆

Hamid Beigy ^{a,b,*}, M.R. Meybodi ^{b,c}

^a Department of Computer Engineering, Sharif University of Technology, Hafez Avenue, Tehran 15914, Iran

^b Institute for Studies in Theoretical Physics and Mathematics (IPM), School of Computer Science, Tehran, Iran

^c Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

Received 29 May 2003; received in revised form 1 October 2004; accepted 9 December 2004

Abstract

In this paper, we first propose a new continuous action-set learning automaton and theoretically study its convergence properties and show that it converges to the optimal action. Then we give an adaptive and autonomous call admission algorithm for cellular mobile networks, which uses the proposed learning automaton to minimize the blocking probability of the new calls subject to the constraint on the dropping probability of the handoff calls. The simulation results show that the performance of the proposed algorithm is close to the performance of the limited fractional guard channel algorithm for which we need to know all the traffic parameters in advance.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Learning automata; Continuous action learning automata; Adaptive call admission control; Guard channel policy; Limited fractional guard channel policy

☆ This research was in part supported by a grant from Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran.

* Corresponding author. Address: Department of Computer Engineering, Sharif University of Technology, Hafez Avenue, Tehran 15914, Iran. Tel.: +98 216 419411; fax: +98 216 495521.

E-mail addresses: beigy@ce.sharif.edu (H. Beigy), meybodi@ce.aut.ac.ir (M.R. Meybodi).

1. Introduction

In cellular networks, geographical area covered by a mobile network is divided into smaller regions called *cells*. Each cell has a *base station*, which is located at its center. A number of base stations are connected to a *mobile switching center*, which also acts as a gateway of the mobile network to the existing wired-line networks. In order for a mobile user to be able to communicate with other user(s), a connection usually must be established between the users. When a mobile user needs a connection, sends his 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 low.

Introduction of micro cellular networks leads to efficient use of channels but increases the expected rate of handovers per call. As a consequence, some network performance parameters such as *blocking probability of new calls* and *dropping probability of handoff calls* are affected. Call admission algorithms control both the blocking probability of the new calls (B_n) and the dropping probability of the handoff calls (B_h) by putting some restrictions on the allocation of channels to the incoming calls. Since the dropping probability of handoff calls is more important than the blocking probability of new calls, call admission algorithms usually put restriction on the acceptance of new calls. Assume that the given cell has C full duplex channels. The simplest call admission algorithm, which is called *guard channel algorithm*, reserves a subset of the channels allocated to the cell, called *guard channels*, for sole use of the handoff calls. In the guard channel algorithm, when the channel occupancy exceeds certain threshold T , then the new calls are rejected until the channel occupancy goes below threshold T [1]. The guard channel algorithm accepts the handoff calls as long as channels are available. It has been shown that there is an optimal threshold T^* for which the blocking probability of the new calls is minimized subject to the constraint on the dropping probability of handoff calls [2]. Algorithms for finding T^* are given in [2–4]. Although, the guard channel algorithm decreases the dropping probability of the handoff calls, but the blocking probability of the new calls may be degraded to a great extent.

In order to have more control on the blocking probability of the new calls and the dropping probability of the handoff calls, the *limited fractional guard channel algorithm* (LFG) is introduced [2]. The LFG algorithm uses an additional parameter π and is the same as to the guard channel algorithm except that when T channels are occupied in the cell, it accepts the new calls with probability π . It has been shown that there is a threshold T^* and a value π^* for parameter π such that the blocking probability of the new calls is minimized subject to the constraint on the dropping probability of handoff calls [2]. An algorithm for finding T^* and π^* is given in [2]. *Uniform fractional channel policy* (UFC) is introduced in [5,6], which accepts the new calls with probability π independently of channel occupancy. It is shown that there is an optimal π^* , which minimizes the blocking probability of the new calls subject to the constraint on the dropping probability of the

handoff calls. In [5,6], an algorithm for finding π^* is given and the conditions under which the UFC policy performs better than the guard channel policy is derived and finally it is concluded that, the UFC policy performs better than the guard channel policy under the low handoff traffic conditions. All of the above mentioned call admission algorithms assume that the input traffic is a stationary process with known parameters. Since in reality, the input traffic is not a stationary process, the optimal number of guard channels cannot be kept fixed and needs to be adapted as traffic conditions change. In such cases *adaptive call admission algorithms* can be used to adapt the required number of guard channels as the network operates.

The learning automata are adaptive decision making devices that operating in an unknown random environment and progressively improve their performance via a learning process. Learning automata are divided into two main groups: finite action-set learning automata (FALA) and continuous action-set learning automata (CALA) based on whether the action set is finite or continuous [7]. For an r -action FALA, the action probability distribution is represented by an r -dimensional probability vector and is updated by the learning algorithm. In many applications a large number of actions may be needed. A FALA with too large number of actions converges too slowly. In such applications CALA, whose actions are chosen from real line, are very useful [8–10].

In this paper, we first propose a new continuous action-set learning automaton. This learning automaton uses a Gaussian distribution $N(\mu, \sigma)$ for choosing its actions. We state and prove a strong convergence theorem that implies the optimal performance of CALA. Then we introduce an adaptive and autonomous call admission algorithm, which uses the proposed continuous action-set learning automaton. This algorithm uses only the current channel occupancy of the given cell and dynamically adjusts the number of guard channels in order to minimize the blocking probability of the new calls subject to the constraint on the dropping probability of the handoff calls. Since the learning automaton starts its learning without any priori knowledge about its environment, the proposed algorithm does not need any a priori information about the input traffic. One of the most important advantages of the proposed algorithm is that no status information will be exchanged among neighboring cells. The simulation results show that the performance of the proposed algorithm is near to the performance of the LFG algorithm that knows all the traffic parameters.

The rest of the paper is organized as follows: Section 2 presents the performance parameters of the LFG algorithm. Section 3 presents a brief review of learning automata. In Section 4, a new continuous action-set learning automaton is given and its behavior is studied. In Section 5, an adaptive call admission control algorithm, which uses the proposed continuous action-set learning automaton, is given. The simulation results are presented in Section 6 and Section 7 concludes the paper.

2. Blocking performance of LFG

In the limited fractional guard channel algorithm (LFG), a fractional number of channels is reserved in each cell exclusively for the handoff calls [2]. The LFG algorithm uses two parameters T and π and operates the same as the guard channel algorithm except that when T channels are occupied in the cell, the new calls are accepted with probability π . Since in the LFG algorithm,

both T and π control the acceptance of the new calls, we consider $T + \pi$ as a control parameter. In what follows, we study the blocking performance of the LFG algorithm. We consider a homogenous wireless network where all the cells have the same number of channels, C , and experience the same new and handoff calls arrival rates. In each cell, the arrival of the new calls and the handoff calls are Poisson distributed with arrival rates λ_n and λ_h , respectively and the channel holding time of new and handoff calls are exponentially distributed with the same 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. These 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. Define the state of a cell at time t by the total number of occupied channels, $c(t)$. Thus, the cell channel occupancy can be modelled by a continuous time Markov chain with states $0, 1, \dots, C$. Fig. 1 shows the state transition diagram of a system with C channels for the LFG algorithm.

Define the steady state probability $P_n = \lim_{t \rightarrow \infty} \text{Prob}[c(t) = n]$ as the probability of n channels being occupied. Given this, it is straight forward to derive probability P_n (for $n = 0, 1, \dots, C$). These state probabilities are given below:

$$P_n = \begin{cases} \frac{\rho^n}{n!} & \text{if } n \leq T, \\ \gamma a^{-(T+1)} \frac{(\rho a)^n}{n!} & \text{if } T < n \leq C, \end{cases} \tag{1}$$

where

$$P_0 = \left[\sum_{n=0}^T \frac{\rho^n}{n!} + \gamma a^{-(T+1)} \sum_{n=T+1}^C \frac{(\rho a)^n}{n!} \right]^{-1} \tag{2}$$

and $a = \lambda_h/\lambda$, $\lambda = \lambda_n + \lambda_h$, $\rho = \lambda/\mu$, $\gamma = [a + (1 - a)\pi]$, and $\lambda = \lambda_n + \lambda_h$. Given these state probabilities vector, we can find the dropping probability of the handoff calls, $B_h(C, T, \pi)$, and the blocking probability of the new calls, $B_n(C, T, \pi)$, as

$$B_h(C, T, \pi) = \gamma a^{-(T+1)} \frac{(\rho a)^C}{C!} P_0, \tag{3}$$

$$B_n(C, T, \pi) = (1 - \pi)P_T + \sum_{n=T+1}^C P_n. \tag{4}$$

Below we study some of the useful properties of $B_n(C, T, \pi)$ and $B_h(C, T, \pi)$. These properties will be used later in this paper.

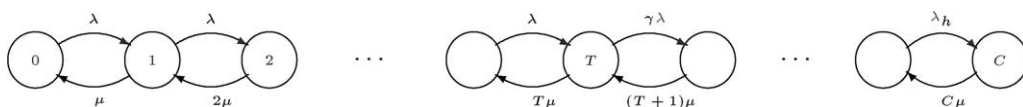


Fig. 1. Markov chain model of cell using LFG algorithm.

Property 1. $B_h(C, T, \pi)$ is a monotonically increasing function of both T and π .

Corollary 1. $B_h(C, T, \pi)$ is a monotonically increasing function of $T + \pi$.

Proof. Since $B_h(C, T, \pi)$ is a monotonically increasing function of both T and π , it is a monotonically increasing function of $T + \pi$. \square

Property 2. $B_h(C, T, \pi)$ is a monotonically decreasing function of both T and π provided that $\rho < (T + 1)$ and $\lambda_n/\lambda < \min\{\frac{1}{T+1}, \frac{1}{C-T}\}$.

Corollary 2. $B_h(C, T, \pi)$ is a monotonically decreasing function of $T + \pi$.

Proof. Since $B_h(C, T, \pi)$ is a monotonically decreasing function of both T and π , it is a monotonically decreasing function of $T + \pi$. \square

3. Learning automata

Learning automata (LA) are adaptive decision making units that can learn to choose the optimal action from a set of actions by interaction with an unknown random environment. At each instant n , the LA chooses an action α_n from its action probability distribution and applies it to the random environment. The random environment provides a stochastic response, which is called reinforcement signal, to the LA. Then the LA uses the reinforcement signal and a learning algorithm to update the action probability distribution. Based on the nature of the reinforcement signal, the random environment could be classified into three classes: P-, Q-, and S-model environments. The reinforcement signal in P-model environments has two values while in Q-model environments can take a finite number of values in the interval $[0, 1]$. In S-model environment, the reinforcement signal is a bounded continuous random variable. The LA can be classified into two main groups: finite action-set learning automata (FALA) and continuous action-set learning automata (CALA) [7]. The action-set of FALA is finite, for example for an r -action ($2 \leq r < \infty$) FALA, the action probability distribution is represented by an r -dimensional probability vector and is updated by a learning algorithm. When the FALA is used for solving optimization problems, we need to discretize the parameter space, so that actions of LA can be possible values of the corresponding parameter. The accuracy of the solution is increased by choosing the finer discretization and hence increasing the number of actions of LA. However, increasing the number of actions of LA leads to slow convergence of the learning algorithm. In order to provide a higher rate of convergence, hierarchical structure LA [11], discretized LA [12], estimator algorithms [13–15], and pursuit algorithms [16–19] are introduced. A more satisfying solution would be to employ the LA model where the action-set can be continuous. Such model of LA is called continuous action-set learning automata in which the action set is the real line. Like FALA, the CALA also uses a probability distribution function to choose an action and the learning algorithm updates this function based on the reinforcement signal.

In [10], a CALA is given, in which the action probability distribution at instant n is a normal distribution with mean μ_n and standard deviation σ_n . At each instant, the CALA updates its

action probability distribution (based on its interaction with the environment) by updating μ_n and σ_n . Since the action set is continuous, instead of penalty probabilities for various actions, we now have a penalty probability function, M , defined by

$$M(\alpha) = E[\beta(\alpha_n)|\alpha_n = \alpha].$$

CALA has no knowledge of the penalty function $M(\cdot)$. The objective of the automaton is to identify the optimal action, which results in the minimum value of $M(\cdot)$. This is to be achieved through the learning algorithm that updates the action probability distribution using the most recent interaction with the random environment. We shall denote the reinforcement signal in response to action α as $\beta(\alpha)$ and thus

$$M(\alpha) = E[\beta(\alpha)].$$

The objective for CALA is to learn value of α at which $M(\alpha)$ attains a minimum. That is, we want the action probability distribution, $N(\mu_n, \sigma_n)$, to converge to $N(\alpha^*, 0)$ where α^* is the minimum of $M(\alpha)$. However, for some technical reasons σ_n cannot converge to zero [10]. So, another parameter, $\sigma_L > 0$, is used in which the objective of learning is the convergence of σ_n and μ_n to σ_L and α^* , respectively. By choosing σ_L sufficiently small, asymptotically CALA will choose actions sufficiently close corresponding to the minimum of M with probability sufficiently close to unity [10].

The learning algorithm for CALA is described below. Since the updating given for σ_n does not automatically guarantee that $\sigma_{n+1} \geq \sigma_n$, always a projected version of σ_n is used, denoted by $\phi[\sigma_n]$. Also, unlike FALA, this CALA interacts with the environment through choice of two actions at each instant. At each instant n , the CALA chooses $\alpha_n \in \mathfrak{R}$ at random from its current distribution $N(\mu_n, \sigma_n)$. Then it gets the reinforcement from the environment for the two actions: μ_n and α_n . Let these reinforcements be $\beta(\mu)$ and $\beta(\alpha)$. Then, the action probability distribution is updated as

$$\begin{aligned} \mu_{n+1} &= \mu_n + af_1[\mu_n, \sigma_n, \alpha_n, \beta(\alpha), \beta(\mu)], \\ \sigma_{n+1} &= \sigma_n + af_2[\mu_n, \sigma_n, \alpha_n, \beta(\alpha), \beta(\mu)] - Ca[\sigma_n - \sigma_L], \end{aligned} \tag{5}$$

where $f_1(\cdot)$, $f_2(\cdot)$, and $\phi(\cdot)$ are defined as below:

$$\begin{aligned} f_1(\mu, \sigma, \alpha, \beta(\alpha), \beta(\mu)) &= \left[\frac{\beta(\alpha) - \beta(\mu)}{\phi(\sigma)} \right] \left[\frac{\alpha - \mu}{\phi(\sigma)} \right], \\ f_2(\mu, \sigma, \alpha, \beta(\alpha), \beta(\mu)) &= \left[\frac{\beta(\alpha) - \beta(\mu)}{\phi(\sigma)} \right] \left[\left(\frac{\alpha - \mu}{\phi(\sigma)} \right)^2 - 1 \right], \end{aligned} \tag{6}$$

$$\phi(\sigma) = (\sigma - \sigma_L)I\{\sigma > \sigma_L\} + \sigma_L$$

and σ_L , $C > 0$, and $a \in (0, 1)$ are parameters of the algorithm. For this algorithm it is shown that with arbitrary large probability, μ_n will converge close to a minimum of $M(\cdot)$ and σ_n will converge close to σ_L , if we choose μ and σ_L sufficiently small and C sufficiently large [10].

Another continuous action-set learning automata called *continuous action reinforcement learning automata* (CARLA) is given in [20,21], independently. This automaton can be described as follows. Let actions-set of the automaton be a bounded continuous random variable defined over the

interval $[\alpha_{\min}, \alpha_{\max}] \in \mathfrak{R}$. The CARLA uses a continuous probability density function, $f(n)$, to choose its actions. It is assumed that no information about the actions is available at the start of learning and therefore the probabilities of actions are initially equal. Thus, the initial distribution is chosen as uniform distribution. The CARLA selects action α_n at instant n from distribution $f(\cdot)$ and applies it to a S-model random environment that emits a response $\beta_n \in [0, 1]$. Based on the response β_n , the continuous probability density function $f(n)$ is updated according to the following rule:

$$f(n+1) = \begin{cases} a[f(n) + (1 - \beta_n)H(\alpha, \alpha_n)] & \text{if } \alpha_n \in [\alpha_{\max}, \alpha_{\min}], \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where a is a normalization factor and $H(\alpha, r)$ is a symmetric Gaussian neighborhood function centered on $r = \alpha_n$ given by Eq. (8). The function $H(\alpha, r)$ has the effect of spreading the reward for neighborhood actions of the selected action:

$$H(\alpha, r) = \lambda \exp\left(-\frac{1}{2}\left(\frac{\alpha - r}{\sigma}\right)^2\right), \quad (8)$$

where λ and σ are parameters that affect the height and the width of the neighborhood function. The asymptotic behavior of this continuous action learning automata is not known.

Learning automata have been used successfully in many applications such computer network [22–24], solving NP-complete problems [25–27], capacity assignment [28,29], neural network engineering [30–33], and cellular networks [34–37] to mention a few.

4. A new continuous action-set learning automaton

In this section, we introduce a new continuous action-set learning automaton (CALA), which will be used later in Section 5 for designing an adaptive call admission algorithm. For the proposed CALA, we use the Gaussian distribution, $N(\mu, \sigma)$ for selection of actions, which is completely specified by the first and second-order moments, μ and σ . The learning algorithm updates the mean and the variance of the Gaussian distribution at any instant using the reinforcement signal β , obtained from the random environment. The reinforcement signal, $\beta \in [0, 1]$, is a noise-corrupted reinforcement signal, which indicates a noise-corrupted observation of function $M(\cdot)$ at the selected action. The reinforcement signal β is a random variable whose distribution function coincides almost with the distribution $H(\beta|\alpha)$ that belongs to a family of distributions which depends on the parameter α . Let

$$M(\alpha) = E[\beta(\alpha)|\alpha] = \int_{-\infty}^{\infty} \beta(\alpha) dH(\beta|\alpha)$$

be a penalty function with bound \mathcal{M} corresponding to this family of distributions. We assume that $M(\cdot)$ is measurable and continuously differentiable almost everywhere. The CALA has to minimize $M(\cdot)$ by observing $\beta(\alpha)$. Let μ_n and σ_n be the mean and the standard deviation of the Gaussian distribution, respectively. Using the learning algorithm, we ideally want that $\mu_n \rightarrow \mu^*$ and $\sigma_n \rightarrow 0$ as time tends to infinity.

The interaction between the CALA and the random environment takes place as iterations of the following operations. Iteration n begins by selection of an action α_n by the CALA. This action is generated as a random variable from the Gaussian distribution with parameters μ_n and σ_n . The selected action is applied to the random environment and the learning automaton receives an evaluative signal $\beta(\alpha_n)$, which has the mean value $M(\alpha_n)$, from the environment. Then the learning automaton updates the parameters μ_n and σ_n . Initially $M(\cdot)$ is not known and it is desirable that with interaction of learning automaton and the random environment, the μ and σ converges to their optimal values which results the minimum value of $M(\cdot)$. The learning automaton uses the following rule to update the parameters μ_n and σ_n , thus generating a sequence of random variables μ_n and σ_n :

$$\begin{aligned}\mu_{n+1} &= \mu_n - a\beta(\alpha_n)\sigma_n(\alpha_n - \mu_n), \\ \sigma_{n+1} &= f(\sigma_n),\end{aligned}\tag{9}$$

where a is the learning rate and $f(\cdot)$ is a function that produces a sequence of σ_n (described later). Eq. (9) can be written as

$$\mu_{n+1} = \mu_n - a\sigma_n^2 y_n(\alpha_n),\tag{10}$$

where

$$y_n(\alpha_n) = \beta(\alpha_n) \left(\frac{\alpha_n - \mu_n}{\sigma_n} \right).\tag{11}$$

An intuitive explanation for the above updating equations is as follows. We can view the fraction in Eq. (11) as the normalized noise added to the mean. Since a , β and σ all are positive, the updating equation changes the mean value in the opposite direction of the noise. If the noise is positive, then the learning automaton should update its parameter, so that mean value increased and vice versa. Since $E[\beta|\alpha]$ is close to unity when α is far from its optimal value and it is close to zero when α is near to the optimal value, thus the learning automaton updates μ with large steps when α is far from its optimal value and with small steps when α is close to its optimal value. This causes a finer quantization of μ near its optimal value and a grain quantization for points far away its optimal value. Thus, we can consider the learning algorithm as a random direction search algorithm with adaptive step sizes.

In what follows, we state and prove the convergence of the proposed learning automaton in stationary environments. The convergence is proved based on the following assumption:

Assumption 1. The sequence of real numbers $\{\sigma_n\}$ is such that $\sigma_n \geq 0$, $\sum_{n=1}^{\infty} \sigma_n^3 = \infty$, $\sum_{n=1}^{\infty} \sigma_n^4 < \infty$.

Note that these conditions imply that $\sigma_n \rightarrow 0$ as $n \rightarrow \infty$. Therefore, in limit, σ_n of Gaussian distribution tends to zero and the action of learning automaton becomes equal to the mean. The condition $\sum_{n=1}^{\infty} \sigma_n^3 = \infty$ ensures that the sum of increments to the initial mean, μ_0 , can be arbitrary large, so that any finite initial value of μ_0 can be transformed into the optimal value μ^* . At the same time, the condition that $\sum_{n=1}^{\infty} \sigma_n^4 < \infty$ ensures that the variance in μ_n is finite and the mean cannot diverge to infinity.

Assumption 2. The $M(\cdot)$ has a unique minimum μ^* , a finite number of minima inside a compact set, and bounded first and second derivatives with respect to α .

Let $R(\alpha) = \frac{\partial M(\alpha)}{\partial \alpha}$ and $S(\alpha) = \frac{\partial^2 M(\alpha)}{\partial \alpha^2}$ be the first and the second derivative of $M(\alpha)$, respectively. Suppose that \mathcal{R} and \mathcal{S} be the bounds on R and S , respectively.

Assumption 3. $M(\alpha)$ is linear near μ^* , that is, there is an $\epsilon > 0$ such that the following condition is satisfied:

$$\sup_{\epsilon \leq |\alpha - \mu^*| \leq \frac{1}{\epsilon}} (\alpha - \mu^*)R(\alpha) > 0. \quad (12)$$

Assumption 3 is restrictive, because it requires $M(\cdot)$ to be linear in the neighborhood of the optimal action. However, it may be possible to weaken this assumption as has been done in stochastic approximation algorithms, but only at complicating the cost of the analysis, which we do not consider it here. Assumptions 2 and 3 mean that the function being optimized has a unique global minimum and is quadratic in the vicinity of the global minimum.

Assumption 4. The noise in the reinforcement signal $\beta(\cdot)$ has a bounded variance, that is

$$E\left\{[\beta(\alpha) - M(\alpha)]^2\right\} \leq K_1 \left[1 + (\alpha - \mu^*)^2\right] \quad (13)$$

for some real number $K_1 > 0$.

Given the above assumptions, in what follows, we study the behavior of the proposed CALA. The following theorem states the convergence of the random process defined by (9). The proof of this theorem is given in Appendix B.

Theorem 1. *Suppose that Assumptions 1–4 hold, μ_0 is finite and there is an optimal value of μ^* for μ . Then if μ_n and σ_n are evolved according to the given learning algorithm, then $\lim_{n \rightarrow \infty} \mu_n = \mu^*$ with probability 1.*

5. An adaptive call admission algorithm

The LFG algorithm assumes that the traffic is a stationary process with known parameters. But in reality, the traffic is a non-stationary process. Even if we assume the traffic is stationary, its parameters may not be known a priori. In either case, the optimal value for the parameters of the LFG algorithm is not known a priori and may vary with time. In non-stationary traffic, an adaptive LFG algorithm which adapts the parameter of the LFG as the network operates has superiority over the LFG algorithm.

In this section, we consider the call admission problem for two classes of calls: new and handoff calls and present a learning automata based LFG algorithm (Fig. 2) to adapt the value of $T + \pi$ for LFG algorithm. This algorithm can be used particularly when λ_n , λ_h and μ are unknown and possibly time varying. The objective of this algorithm is to adapt parameter $T + \pi$ in such a way that minimizes the blocking probability of new calls subject to the constraint that the dropping probability of handoff calls be at most p_h . Since $T + \pi$ is a continuous parameter, the algorithm

```

set  $\sigma(0)$  to a large value and  $\mu(0)$  to a random number in interval  $[0, C]$ .
loop
  if NEW CALL then
    choose an action of CALA from  $N(\mu(n), \sigma(n))$  and call it  $x$ 
     $T \leftarrow \lfloor x \rfloor, \pi \leftarrow x - \lfloor x \rfloor$ 
    if  $c(t) < T$  then
      accept the call
    else if  $(c(t) = T)$  and  $(rand(0, 1) \leq \pi)$  then
      accept the call
    else
      reject the call
    end if
    upon the arrival of the new call, compute  $\beta$ .
    update the Gaussian distribution using  $\beta$ .
  else if HANDOFF CALL then
    if  $c(t) < C$  then
      accept the call
    else
      reject the call
    end if
  end if
end loop

```

Fig. 2. Learning automata based adaptive limited fractional guard channel algorithm.

uses a continuous action-set learning automaton (CALA) for adaptation of the value of parameter $T + \pi$. Let $x(n) = T(n) + \pi(n)$ be the parameter of the LFG algorithm at instant n , and $x(n)$ takes values in the interval $[x_{\min}, x_{\max}]$, where $0 \leq x_{\min} < x_{\max} \leq C$. The CALA uses the real line as the action-set and uses the Gaussian distribution, $N(\mu; \sigma)$, to choose its actions. This Gaussian distribution is updated using the reinforcement signal, β , which is emitted from the environment. Initially, the CALA chooses one of its actions with equal probability using a Gaussian distribution with a large variance.

Since $x(n)$ and $\mu(n)$ must be in the interval $[x_{\min}, x_{\max}]$, the above mentioned CALA cannot be used directly to adapt the value of $T + \pi$, and hence a projected version of CALA will be used. In the projected version of CALA, a constraint set $H = \{y | x_{\min} \leq y \leq x_{\max}\}$ is used for updating μ as well as choosing actions of CALA. In the projected version, when the updated value of μ goes outside of the constraint set H , then μ is pushed into H and when the action, x , chosen by the CALA does not belong to H , then x is pushed into H .

The proposed algorithm can be described as follows. Each base station is equipped with a CALA for adapting $T + \pi$. When a new call arrives at a given cell, the learning automaton associated to that cell chooses one of its actions, say $x(n)$. Let $T(n) = \lfloor x(n) \rfloor$ and $\pi(n) = x(n) - \lfloor x(n) \rfloor$. If the number of busy channels of a cell is less than $T(n)$, then the incoming call will be accepted; when the cell has $T(n)$ busy channels, then a call will be accepted with probability $\pi(n)$; otherwise the incoming call will be blocked. On the arrival of a new call the base station computes the current estimate of the dropping probability of the handoff calls and based on the result of the comparison of this quantity with the specified level of QoS, p_h , a reinforcement signal to the CALA will be produced.

In this algorithm, the reinforcement signal at instant n is produced using the following expression:

$$\beta(n) = |\hat{B}_h - p_h|, \quad (14)$$

where \hat{B}_h is the current estimate of the dropping probability of the handoff calls. \hat{B}_h is calculated using the current statistics of the network. The algorithm does not require a priori information about the parameters of the traffic distribution. In fact at any stage, each base station uses an estimate of the dropping probability of the handoff calls by counting the number of the dropped handoff calls in the cell. It is evident that when \hat{B}_h is close to p_h , then β becomes small and approaches zero and when \hat{B}_h far from p_h , then β becomes large. The variance is updated independently of the reinforcement signal in such a manner that $\sigma(n)$ is a decreasing function of n .

The following theorem states the fact that this algorithm adapts x in such a way that the blocking probability of new calls will be minimized and at the same time the constraint on the dropping probability of the handoff calls is satisfied.

Theorem 2. *If sequence $\{\sigma(n)\}$ satisfies Assumption 1, i.e., $\sigma(n) \geq 0$, $\sum_{n=1}^{\infty} \sigma^3(n) = \infty$, and $\sum_{n=1}^{\infty} \sigma^4(n) < \infty$ and the number of channels allocated to the cell is large enough, then the proposed call admission algorithm minimizes the blocking probability of the new calls subject to the constraint on the dropping probability of the handoff calls.*

Proof. Since the number of channels allocated to the cell is large enough, the constraint set H can be ignored. From (14), it is clear that $\beta \in [0, 1]$, and β and $E[\beta]$ satisfy the following conditions:

- The penalty function $E[\beta]$ has a unique minimum at p_h .
- The penalty function $E[\beta]$ can be approximated linearly near p_h .
- The noise in β has bounded variance.

Using Theorem 1 and (14), we can conclude that the proposed algorithm attains its minimum when $\hat{B}_h = p_h$. This implies that the constraint on the dropping probability of the handoff calls will be satisfied. Property 1 states that the dropping probability of the handoff calls (B_h) is a monotonically increasing function of $T + \pi$, and hence the maximum value for $T + \pi$ subject to the constraint on the dropping probability of the handoff calls will be obtained at p_h . Property 2 indicates that the blocking probability of new calls is a monotonically decreasing function of $T + \pi$. From these two properties and the fact that the algorithm attains its minimum at p_h , we can conclude that the blocking probability of new calls is minimized by the algorithm at p_h , as shown in Fig. 3. \square

6. Simulation results

In this section, we compare the performance of the adaptive uniform fractional channel algorithm [34], the dynamic guard channel algorithm [36], the LFG algorithm [2] and the proposed algorithm. The results of simulations are summarized in Table 1. The simulation is based on

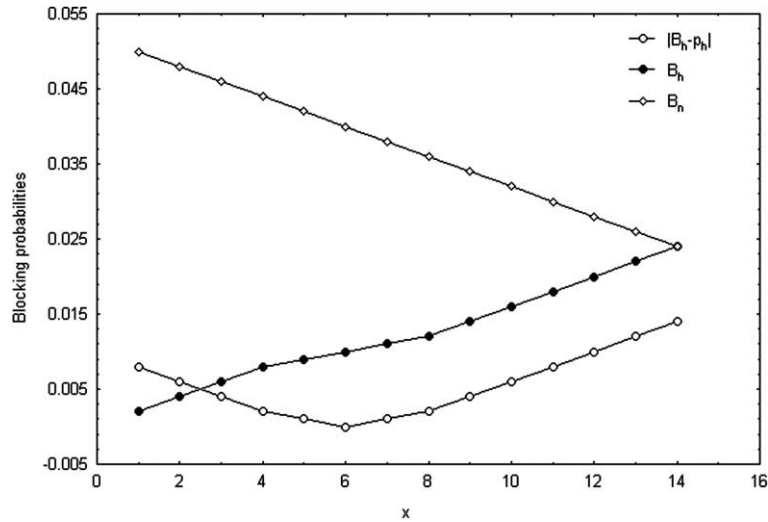


Fig. 3. Reinforcement signal for adaptive fractional guard channel algorithm.

Table 1
Comparison of the proposed call admission algorithm with other algorithms

Case	λ_h	LFG		Adaptive UFC		Dynamic guard channel		The proposed algorithm	
		B_n	B_h	B_n	B_h	B_n	B_h	B_n	B_h
1	2	0.031609	0.023283	0.208524	0.010001	0.053433	0.010619	0.050541	0.010170
2	4	0.051414	0.020675	0.260971	0.010001	0.080966	0.010039	0.066736	0.010130
3	6	0.071632	0.018707	0.318610	0.010003	0.125500	0.009964	0.083931	0.009956
4	8	0.092138	0.016706	0.365239	0.010003	0.154861	0.010031	0.104699	0.010034
5	10	0.114445	0.015572	0.337246	0.015549	0.207490	0.010067	0.134831	0.010028
6	12	0.147902	0.014044	0.469893	0.010003	0.245842	0.010017	0.181290	0.010015
7	14	0.204217	0.012675	0.515486	0.010290	0.290619	0.009960	0.228663	0.009794
8	16	0.250642	0.011554	0.550553	0.011164	0.331478	0.009983	0.259676	0.009838
9	18	0.294441	0.010877	0.589336	0.012027	0.377334	0.009953	0.306801	0.010030
10	20	0.384157	0.010182	0.623895	0.013519	0.427894	0.010005	0.378593	0.010251

the single cell of homogenous cellular network system. Each cell has eight full duplex channels ($C = 8$). We use the following function to update the variance in the proposed algorithm:

$$\sigma(n) = \frac{1}{\lfloor \frac{n}{10} \rfloor^{1/3}}, \tag{15}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. It can be easily verified that sequence $\{\sigma_n\}$ satisfies the conditions of Assumption 1, i.e., $\sigma(n) \geq 0$, $\sum_{n=1}^{\infty} \sigma^3(n) = \infty$, and $\sum_{n=1}^{\infty} \sigma^4(n) < \infty$.

In the simulations, we assume that the arrival of the new calls is Poisson process with rate λ_n fixed at 30 calls/min. The arrival of handoff calls is Poisson process with rate λ_h varied between 2 and 20 calls/min. We also assume that the duration of calls are exponentially distributed with

mean $\mu = 1/6$. We set p_h to 0.01. The results listed in Table 1 are obtained by averaging over 10 runs. Each run takes 2,000,000 s. The optimal parameters of LFG algorithm is obtained using the algorithm given in [2].

By inspecting Table 1, it is evident that (1) unlike the LFG algorithm, the upper bound of the dropping probability of the handoff calls is maintained by the proposed algorithm; (2) the blocking of the new calls for the proposed algorithm is close to the blocking probability of the new calls for the LFG algorithm that knows the traffic parameters.

7. Conclusions

In this paper, a new continuous action-set learning automaton was introduced and its convergence was studied. We stated and proved a strong convergence theorem that implies the optimality of the proposed CALA. Then using the proposed CALA, an adaptive call admission algorithm, which minimizes the blocking probability of the new calls subject to the constraint on the dropping probability of the handoff calls was given. The algorithm does not need to know any a priori knowledge about the traffic parameters. The computer simulations were conducted to show the effectiveness of the algorithm.

Acknowledgement

The authors would like to thank the referees for their suggestions and comments.

Appendix A

In this section, we present the proofs of Properties 1 and 2. Before we give the proofs, we first give some notations and a property.

Proof (Property 1). In order to show that $B_h(C, T, \pi)$ is a monotonically increasing function of T , we need to show that $B_h(C, T, \pi) < B_h(C, T + 1, \pi)$. Using Eq. (3) and some algebraic simplifications, we obtain

$$B_h(C, T, \pi) - B_h(C, T + 1, \pi) = \gamma a^{-(T+1)} \frac{(\rho\alpha)^C}{C!} \left[\frac{\pi \frac{(\rho\alpha)^{T+1}}{(T+1)!} + \sum_{n=0}^T \frac{\rho^n}{n!}}{D(C, T + 1, \pi)D(C, T, \pi)} \right] (1 - a^{-1}) < 0,$$

where

$$D(C, T, \pi) = \sum_{n=0}^T \frac{\rho^n}{n!} + \gamma a^{-(T+1)} \sum_{n=T+1}^C \frac{(\rho\alpha)^n}{n!}.$$

In order to show that $B_h(C, T, \pi)$ is a monotonically increasing function of π , we need to show that $\frac{\partial B_h(C, T, \pi)}{\partial \pi} > 0$. Differentiating $B_h(C, T, \pi)$ with respect to π , we obtain

$$\frac{\partial B_h(C, T, \pi)}{\partial \pi} = \frac{\rho^C}{C!} \left[\frac{\sum_{n=0}^T \frac{\rho^n}{n!}}{D^2(C, T, \pi)} \right] (1 - a) > 0. \quad \square$$

Proof (Property 2). In the first part of the proof, we show that $B_h(C, T, \pi)$ is a monotonically decreasing function of T . The blocking probability of new calls can be written as

$$B_n(C, T, \pi) = \frac{(1 - \pi)Z_T + \gamma D_2(C, T, \pi)}{D_1(C, T, \pi) + \gamma D_2(C, T, \pi)},$$

where

$$D(C, T, \pi) = \sum_{n=0}^T \frac{\rho^n}{n!} + \gamma a^{-(T+1)} \sum_{n=T+1}^C \frac{(\rho a)^n}{n!},$$

$$D_2(C, T, \pi) = a^{-(T+1)} \sum_{n=T+1}^C \frac{(\rho a)^n}{n!},$$

$$Z_T = a^{-T} \frac{(\rho a)^T}{T!} \quad \text{and} \quad D_1(C, T, \pi) = \sum_{n=0}^T \frac{\rho^n}{n!}.$$

In order to show that $B_h(C, T, \pi)$ is a monotonically decreasing function of T , we need to show that $B_n(C, T, \pi) > B_n(C, T + 1, \pi)$. Using Eq. (4) and some algebraic simplifications, we obtain

$$\begin{aligned} B_n(C, T + 1, \pi) - B_n(C, T, \pi) &= \frac{(1 - \pi)Z_{T+1} + \gamma D_2(C, T + 1, \pi)}{D(C, T + 1, \pi)} \\ &\quad - \frac{(1 - \pi)Z_T + \gamma D_2(C, T, \pi)}{D(C, T, \pi)} \\ &= (1 - \pi)Z_T \left[\frac{\rho}{(T + 1)D(C, T + 1, \pi)} - \frac{1}{D(C, T, \pi)} \right] \\ &\quad + a^{-1}\gamma \frac{D_2(C, T, \pi) - Z_T}{D(C, T + 1, \pi)} - \gamma \frac{D_2(C, T, \pi)}{D(C, T, \pi)} \\ &< (1 - \pi) \frac{Z_T}{D(C, T + 1, \pi)} \left[\frac{\rho}{T + 1} - 1 \right] \\ &\quad + \gamma \frac{a^{-1}D_2(C, T, \pi) - a^{-1}Z_T - D_2(C, T, \pi)}{D(C, T, \pi)}. \end{aligned} \tag{A.1}$$

Since $\rho < (T + 1)$, the first term in the above inequality is negative and therefore we have

$$\begin{aligned} B_n(C, T + 1, \pi) - B_n(C, T, \pi) &< \gamma \frac{a^{-1}D_2(C, T, \pi) - a^{-1}Z_T - D_2(C, T, \pi)}{D(C, T, \pi)} \\ &< \gamma \frac{D_2(C, T, \pi)(a^{-1} - 1) - a^{-1}Z_T}{D(C, T, \pi)}. \end{aligned} \tag{A.2}$$

Since $\lambda_n/\lambda < 1/(T + 1)$, we have

$$(\alpha^{-1} - 1) = \frac{\lambda_n}{\lambda} < \frac{1}{T + 1} < \frac{1}{1 + \dots + \left(\frac{\rho\alpha}{T+1}\right)^{C-T-1}}.$$

Since $\rho < (T + 1)$, the above inequality becomes

$$\alpha^{-1} - 1 < \frac{1}{1 + \dots + \frac{(\rho\alpha)^{T+1}}{(T+1)\times\dots\times C}} = \frac{\frac{(\rho\alpha)^T}{T!}}{\sum_{n=T+1}^C \frac{(\rho\alpha)^n}{n!}} < \frac{\alpha^{-(T+1)} \frac{(\rho\alpha)^T}{T!}}{\alpha^{-(T+1)} \sum_{n=T+1}^C \frac{(\rho\alpha)^n}{n!}} < \alpha^{-1} \frac{Z_T}{D_2(C, T, \pi)}.$$

Using the above inequality and inequality (A.1), we conclude that $B_n(C, T + 1, \pi) - B_n(C, T, \pi) < 0$ and hence $B_n(C, T, \pi)$ is a decreasing function of T .

In the second part of the proof, we show that $B_n(C, T, \pi)$ is a monotonically decreasing function of π . Differentiating $B_n(C, T, \pi)$ with respect to π , we obtain

$$\begin{aligned} \frac{\partial B_n(C, T, \pi)}{\partial \pi} &= \frac{D_1(C, T, \pi)[D_2(C, T, \pi) - Z_T] - D_2(C, T, \pi)[aD_1(C, T, \pi) + Z_T]}{[D_1(C, T, \pi) + \gamma D_2(C, T, \pi)]^2} \\ &< \frac{D_1(C, T, \pi)}{[D_1(C, T, \pi) + \gamma D_2(C, T, \pi)]^2} [(1 - a)D_2(C, T, \pi) - Z_T]. \end{aligned} \tag{A.3}$$

Since $a^{-1} - 1 < 1/(C - T)$ and $\rho\alpha < T + 1$, we have

$$a^{-1} - 1 = \frac{\lambda_n}{\lambda} < \frac{1}{C - T} < \frac{1}{\frac{\rho\alpha}{T+1} + \dots + \left(\frac{\rho\alpha}{T+1}\right)^{C-T}} < \frac{1}{\frac{\rho\alpha}{T+1} + \dots + \frac{(\rho\alpha)^{C-T}}{(T+1)\times\dots\times C}} < \frac{\frac{(\rho\alpha)^T}{T!}}{\alpha^{-T} \sum_{n=T+1}^C \frac{(\rho\alpha)^n}{n!}}, \tag{A.4}$$

$$\begin{aligned} \frac{1 - a}{a} &< \frac{\frac{(\rho\alpha)^T}{T!}}{\alpha^{-T} \sum_{n=T+1}^C \frac{(\rho\alpha)^n}{n!}}, \\ 1 - a &< \frac{Z_T}{D_2(C, T, \pi)}. \end{aligned} \tag{A.5}$$

Hence, $D_2(C, T, \pi)(1 - a) < Z_T$. Using (A.3) and (A.5), we can conclude that $B_n(C, T, \pi)$ is a monotonically decreasing function of π . \square

Appendix B

In this section, we give the proof of Theorem 1. Before we prove this theorem, we first prove the following two supplementary lemmas:

Lemma 1. *Under Assumptions 1–4, we have $E[y_n(\alpha_n)|\mu_n] = \sigma_n R(\mu_n)$.*

Proof. Let β_n to denote $\beta(\alpha_n)$, where α_n is a random variable chosen from Gaussian distribution with mean μ_n and variance σ_n^2 . Using the total expectation law, we have

$$E[y_n(\alpha_n)|\mu_n] = E\left[\beta_n\left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)\middle|\mu_n\right] = E\left[M(\alpha_n)\left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)\middle|\mu_n\right]. \tag{B.6}$$

Replacing $M(\alpha_n)$ with its second-order Taylor series expansion around μ_n , we obtain

$$\begin{aligned} E[y_n(\alpha_n)|\mu_n] &= E\left\{\left[M(\mu_n) + R(\mu_n)(\alpha_n - \mu_n) + S(\xi)\frac{(\alpha_n - \mu_n)^2}{2}\right]\left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)\middle|\mu_n\right\} \\ &= M(\mu_n)E\left[\left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)\middle|\mu_n\right] + R(\mu_n)E\left[\left(\frac{(\alpha_n - \mu_n)^2}{\sigma_n}\right)\middle|\mu_n\right] \\ &\quad + \frac{1}{2}E\left[S(\xi)\left(\frac{(\alpha_n - \mu_n)^3}{\sigma_n}\right)\middle|\mu_n\right], \end{aligned} \tag{B.7}$$

where ξ lies between α_n and μ_n . The first and last terms on the right-hand side of the above equation is zero because the odd moments of a Gaussian random variable, α_n , are zero. Since $M(\cdot)$ has bounded second derivative, thus the above equation can be written as

$$E[y_n(\mu_n)|\mu_n] = R(\mu_n)E\left[\left(\frac{(\alpha_n - \mu_n)^2}{\sigma_n}\right)\middle|\mu_n\right] = \sigma_n R(\mu_n). \quad \square \tag{B.8}$$

Lemma 2

$$E[y_n^2(\alpha_n)|\mu_n] \leq K_2 [1 + (\mu_n - \mu^*)^2].$$

Proof

$$\begin{aligned} E[y_n^2(\alpha_n)|\mu_n] &= \int_{-\infty}^{\infty} E\left\{\left[\beta_n\left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)\right]^2\middle|\mu_n\right\} dN(\alpha_n|\mu_n, \sigma_n) \\ &= \int_{-\infty}^{\infty} E\{\beta_n^2|\mu_n\} \left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)^2 dN(\alpha_n|\mu_n, \sigma_n). \end{aligned} \tag{B.9}$$

Since the noise in the reinforcement signal has a bounded variance and $\beta^2(\alpha)$ is bounded by a quadratic function for all α , we obtain

$$\begin{aligned} E[y_n^2(\alpha_n)|\mu_n] &= \int_{-\infty}^{\infty} E\{\beta_n^2|\mu_n\} \left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)^2 dN(\alpha_n|\mu_n, \sigma_n) \\ &\leq K_1 \int_{-\infty}^{\infty} \left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)^2 dN(\alpha_n|\mu_n, \sigma_n) + K_1 \int_{-\infty}^{\infty} M^2(\alpha_n) \left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)^2 dN(\alpha_n|\mu_n, \sigma_n) \\ &\quad + K_1 \int_{-\infty}^{\infty} [(\alpha_n - \mu_n)^2 + 2(\alpha_n - \mu_n)(\mu_n - \mu^*) \\ &\quad + (\mu_n - \mu^*)^2] \left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)^2 dN(\alpha_n|\mu_n, \sigma_n). \end{aligned}$$

Since the odd moments of a Gaussian random variable is zero, the above equation can be written as

$$E[y_n^2(\alpha_n)|\mu_n] \leq K_1 + 3K_1\sigma_n^2 + K_1(\mu_n - \mu^*)^2 + K_1 \int_{-\infty}^{\infty} M^2(\alpha_n) \left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)^2 dN(\alpha_n|\mu_n, \sigma_n). \tag{B.10}$$

Replacing $M(\alpha_n)$ with its second-order Taylor series expansion around μ_n , the last term of the above equation equals to

$$\begin{aligned} & \int_{-\infty}^{\infty} M^2(\alpha_n) \left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)^2 dN(\alpha_n|\mu_n, \sigma_n) \\ &= \int_{-\infty}^{\infty} \left[M(\mu_n) + R(\mu_n)(\alpha_n - \mu_n) + S(\xi) \frac{(\alpha_n - \mu_n)^2}{2} \right]^2 \left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)^2 dN(\alpha_n|\mu_n, \sigma_n) \\ &\leq \int_{-\infty}^{\infty} \left[\mathcal{M} + \mathcal{R}(\alpha_n - \mu_n) + \mathcal{S} \frac{(\alpha_n - \mu_n)^2}{2} \right]^2 \left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)^2 dN(\alpha_n|\mu_n, \sigma_n), \end{aligned} \tag{B.11}$$

where ξ lies between α_n and μ_n and \mathcal{R} and \mathcal{S} are bounds of the first and the second derivative of $M(\cdot)$, respectively. Using the fact that the odd moments of a Gaussian random variable is zero, the above inequality can be written as

$$\int_{-\infty}^{\infty} M^2(\alpha_n) \left(\frac{\alpha_n - \mu_n}{\sigma_n}\right)^2 dN(\alpha_n|\mu_n, \sigma_n) \leq \mathcal{M}^2 + 3\mathcal{R}^2\sigma_n^2 + 3\mathcal{M}\mathcal{S}\sigma_n^2 + \frac{15}{4}\mathcal{S}^2\sigma_n^4.$$

Substituting the above inequality in Eq. (B.10), we obtain

$$E[y_n^2(\alpha_n)|\mu_n] \leq K_1 + 3K_1\sigma_n^2 + K_1(\mu_n - \mu^*)^2 + K_1\mathcal{M}^2 + 3K_1\mathcal{R}^2\sigma_n^2 + 3K_1\mathcal{M}\mathcal{S}\sigma_n^2 + \frac{15}{4}K_1\mathcal{S}^2\sigma_n^4.$$

Since σ_n , \mathcal{M} , \mathcal{R} and \mathcal{S} are bounded random variables, there is a constant $K_2 > 0$ such that the above inequality can be written as $E[y_n^2(\alpha_n)|\mu_n] \leq K_2(1 + (\mu_n - \mu^*)^2)$. \square

Proof of Theorem 1. Let

$$e_n = \mu_n - \mu^*. \tag{B.12}$$

Then using Eq. (10), the following recursive formula can be written for e_n :

$$e_{n+1} = e_n - a\sigma_n^2 y_n(\alpha_n). \tag{B.13}$$

Squaring, taking conditional expectation given μ_0, \dots, μ_n of both sides of the above equation and then using Lemmas 1 and 2, we obtain

$$\begin{aligned} E[e_{n+1}^2|\mu_0, \dots, \mu_n] &= E\left[(e_n - a\sigma_n^2 y_n(\alpha_n))^2 \middle| \mu_0, \dots, \mu_n \right] \\ &\leq e_n^2 + a^2 K_2 \sigma_n^4 (1 + e_n^2) - 2a^2 K_2 \sigma_n^3 e_n R(\mu_n) \\ &\leq e_n^2 + a^2 K_2 \sigma_n^4 (1 + e_n^2) \\ &= e_n^2 (1 + a^2 K_2 \sigma_n^4) + a^2 K_2 \sigma_n^4. \end{aligned} \tag{B.14}$$

Let

$$Z_n = e_n^2 \prod_{j=n}^{\infty} (1 + a^2 K_2 \sigma_j^4) + a^2 K_2 \sum_{j=n}^{\infty} \sigma_j^4 \prod_{i=j+1}^{\infty} (1 + a^2 K_2 \sigma_i^4). \tag{B.15}$$

Then using the above equations, it is easy to show that $e_n^2 \leq Z_n$ and

$$E\{Z_{n+1} | \mu_0, \dots, \mu_n\} \leq Z_n. \tag{B.16}$$

Taking conditional expectations given Z_1, \dots, Z_n on both sides of the above inequality, we obtain

$$E\{Z_{n+1} | Z_0, \dots, Z_n\} \leq Z_n, \tag{B.17}$$

which shows that Z_n is a non-negative super-martingale. Thus, we have

$$E\{Z_{n+1}\} \leq E\{Z_n\} \leq \dots \leq E\{Z_1\} \leq \infty. \tag{B.18}$$

Therefore, using the martingale convergence theorems [38], Z_n converges with probability 1. Since $e_n^2 \leq Z_n$, hence we conclude that e_n^2 converges to η with probability 1, where $\eta < \infty$ is a random variable. Taking expectation on both sides of (B.14), we obtain

$$E[e_{n+1}^2] - E[e_n^2] \leq a^2 K_2 \sigma_n^4 (1 + E[e_n^2]) - 2a^2 K_2 \sigma_n^3 e_n R(\mu_n).$$

Adding the first n of these inequalities, we get

$$E[e_{n+1}^2] - E[e_1^2] \leq a^2 K_2 \sum_{j=1}^n \sigma_n^4 (1 + E[e_j^2]) - 2a^2 K_2 \sum_{j=1}^n \sigma_j^3 e_j R(\mu_j).$$

Adding $E[e_1^2]$ to both sides of the above inequality, we obtain

$$E[e_{n+1}^2] \leq E[e_1^2] + a^2 K_2 \sum_{j=1}^n \sigma_n^4 (1 + E[e_j^2]) - 2a^2 K_2 \sum_{j=1}^n \sigma_j^3 e_j R(\mu_j).$$

Since $E[e_{n+1}^2]$ is positive, the above inequality becomes

$$E[e_1^2] + a^2 K_2 \sum_{j=1}^n \sigma_n^4 (1 + E[e_j^2]) - 2a^2 K_2 \sum_{j=1}^n \sigma_j^3 e_j R(\mu_j) \geq 0.$$

From the above inequality, using the boundness of $E[e_j^2]$ (for $j > 0$), and Assumption 1, it follows that

$$2a^2 K_2 \sum_{j=1}^n \sigma_j^3 e_j R(\mu_j) \leq E[e_1^2] + \sum_{j=1}^n a^2 K_2 \sigma_n^4 (1 + E[e_j^2]) < \infty.$$

Since $\sum_{j=1}^{\infty} \sigma_j^3$ diverges and by Assumption 3, the quantity of $e_j R(\mu_j)$ is positive, we can conclude that for some sequence $\{n_j\}$, we have

$$e_{n_j} R(\mu_{n_j}) \rightarrow 0 \tag{B.19}$$

with probability 1. The fact that e_n^2 converges with probability 1 to some random variable η together with Assumptions 1–4 and the above equation imply that $\eta = 0$ with probability 1. Hence, μ_n converges to μ^* with probability 1. \square

References

- [1] Hong D, Rappaport S. Traffic modelling and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoffs procedure. *IEEE Trans Vehicular Technol* 1986;35(August):77–92.
- [2] Ramjee R, Towsley D, Nagarajan R. On optimal call admission control in cellular networks. *Wireless Networks* 1997;3(March):29–41.
- [3] Oh S, Tcha D. Prioritized channel assignment in a cellular radio network. *IEEE Trans Commun* 1992;40(July):1259–69.
- [4] Haring G, Marie R, Puigjaner R, Trivedi K. Loss formulas and their application to optimization for cellular networks. *IEEE Trans Vehicular Technol* 2001;50(May):664–73.
- [5] Beigy H, Meybodi MR. Uniform fractional guard channel. In: *Proceedings of sixth world multiconference on systemmics, cybernetics and informatics, Orlando, USA; July 2002*.
- [6] Beigy H, Meybodi MR. A new fractional channel policy. *J High Speed Networks* 2004;13(Spring):25–36.
- [7] Thathachar MAL, Sastry PS. Varieties of learning automata: an overview. *IEEE Trans Syst Man Cyb B* 2002;32(December):711–22.
- [8] Gullapalli V. Reinforcement Learning and its Application on Control. PhD thesis. Department of Computer and Information Sciences, University of Massachusetts, Amherst, MA, USA; February 1992.
- [9] Vasilakos A, Zikidis K. Adaptive stochastic algorithm for fuzzy computing/function estimation. In: *Proceedings of third IEEE international conference on fuzzy systems (FUZZ-IEEE'04)*. 1994. p. 1087–92.
- [10] Santharam G, Sastry PS, Thathachar MAL. Continuous action set learning automata for stochastic optimization. *J Franklin Inst B* 1994;331(5):607–28.
- [11] Thathachar MAL, Ramakrishnan KR. A heirarchical system of learning automata. *IEEE Trans Syst Man Cyb* 1981;SMC-11(March):236–48.
- [12] Oommen BJ, Hansen E. The asymptotic optimality of discretized linear reward-inaction learning automata. *IEEE Trans Syst Man Cyb* 1984;SMC-14(May):542–5.
- [13] Thathachar MAL, Sastry PS. A new approach to the design of reinforcement schemes for learning automata. *IEEE Trans Syst Man Cyb* 1985;SMC-15(January):168–75.
- [14] Papadimitriou GI. A new approach to the design of reinforcement schemes for learning automata: stochastic estimator learning algorithm. *IEEE Trans Knowledge Data Eng* 1994;6(August):649–54.
- [15] Lanctôt JK, Oommen BJ. Discretized estimator learning automata. *IEEE Trans Syst Man Cyb* 1992;22(November):1473–83.
- [16] Oommen BJ, Lanctôt JK. Discretized pursuit learning automata. *IEEE Trans Syst Man Cyb* 1990;20(July):931–8.
- [17] Papadimitriou GI. Hierarchical pursuit nonlinear automata with rapid convergence and high accuracy. *IEEE Trans Knowledge Data Eng* 1994;6(August):654–9.
- [18] Oommen BJ, Agache M. Continuous and discretized pursuit learning schemes: various algorithms and their comparison. *IEEE Trans Syst Man Cyb B* 2001;31(June):277–87.
- [19] Agache M, Oommen BJ. Generalized pursuit learning schemes: new families of continuous and discretized learning automata. *IEEE Trans Syst Man Cyb B* 2002;32(December):738–49.
- [20] Frost GP. Stochastic Optimization of Vehicle Suspension Control Systems via Learning Automata. PhD thesis. Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire, LE81 3TU, UI; October 1998.
- [21] Howell MN, Frost GP, Gordon TJ, Wu QH. Continuous action reinforcement learning applied to vehicle suspension control. *Mechatronics* 1997;7(3):263–76.
- [22] Nedzelnitsky OV, Narendra KS. Nonstationary models of learning automata routing in data communication networks. *IEEE Trans Syst Man Cyb* 1987;SMC-17(November):1004–15.
- [23] Obaidat MS, Papadimitriou GI, Pomportsis AS, Laskaridis HS. Learning automata-based bus arbitration for shared-medium ATM switches. *IEEE Trans Syst Man Cyb B* 2002;32(December):815–20.
- [24] Papadimitriou GI, Obaidat MS, Pomportsis AS. On the use of learning automata in the control of broadcast networks: a methodology. *IEEE Trans Syst Man Cyb B* 2002;32(December):781–90.
- [25] Oommen BJ, de St. Croix EV. Graph partitioning using learning automata. *IEEE Trans Comput* 1996;45(February):195–208.

- [26] Meybodi MR, Beigy H. Solving stochastic shortest path problem using distributed learning automata. In: Proceedings of sixth annual Iran computer society of Iran computer conference CSICC-2001, Isfahan, Iran. February 2001. p. 70–86.
- [27] Beigy H, Meybodi MR. Solving the graph isomorphism problem using learning automata. In: Proceedings of fifth annual international computer society of Iran computer conference, CISCC-2000, Tehran, Iran. January 2000. p. 402–15.
- [28] Oommen BJ, Roberts TD. Continuous learning automata solutions to the capacity assignment problem. *IEEE Trans Comput* 2000;49(June):608–20.
- [29] Oommen BJ, Roberts TD. Discretized learning automata solutions to the capacity assignment problem for prioritized networks. *IEEE Trans Syst Man Cyb B* 2002;32(December):821–31.
- [30] Meybodi MR, Beigy H. Neural network engineering using learning automata: determining of desired size of three layer feedforward neural networks. *J Faculty Eng* 2001;34(March):1–26.
- [31] Meybodi MR, Beigy H. A note on learning automata based schemes for adaptation of BP parameters. *J Neurocomput* 2002;48(November):957–74.
- [32] Meybodi MR, Beigy H. New learning automata based algorithms for adaptation of backpropagation algorithm parameters. *Int J Neural Syst* 2002;12(February):45–68.
- [33] Beigy H, Meybodi MR. Backpropagation algorithm adaptation parameters using learning automata. *Int J Neural Syst* 2001;11(June):219–28.
- [34] Beigy H, Meybodi MR. Adaptive uniform fractional channel algorithms. *Iran J Electr Comput Eng* 2004;3(Winter–Spring):47–53.
- [35] Beigy H, Meybodi MR. Call admission in cellular networks: a learning automata approach. Springer-Verlag lecture notes in computer science, vol. 2510. New York: Springer-Verlag; 2002. p. 450–7.
- [36] Beigy H, Meybodi MR. A learning automata based dynamic guard channel scheme. Springer-Verlag lecture notes in computer science, vol. 2510. New York: Springer-Verlag; 2002. p. 643–50.
- [37] Beigy H, Meybodi MR. An adaptive uniform fractional guard channel algorithm: a learning automata approach. Springer-Verlag lecture notes in computer science, vol. 2690. New York: Springer-Verlag; 2003. p. 405–9.
- [38] Doob JL. Stochastic processes. New York: John Wiley; 1953.



Hamid Beigy received the B.S. and M.S. degrees in Computer Engineering from Shiraz University in Iran, in 1992 and 1995, respectively. He also received the Ph.D. degree in Computer Engineering from Amirkabir University of Technology in Iran, in 2004. Currently, he is an assistant professor in Computer Engineering Department at Sharif 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, USA, 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 to 1985 as an assistant professor at Western Michigan University, and from 1985 to 1991 as an associate professor at Ohio University, USA. His research interests include channel management in cellular networks, learning systems, parallel algorithms, soft computing and software development.