books@ocg.at

Behrouz Homayoun Far, M. Hassan Shafazand,
Makoto Takizawa, Roland Wagner (eds.)

EurAsia-ICT 2002

Advances in Information and Communication Technology

First Eurasian Conference
Shiraz, Iran, October 2002

Workshop Proceedings

OESTERREICHISCHE COMPUTER GESELLSCHAFT
AUSTRIAN COMPUTER SOCIETY

Limited Fractional Guard Channel Scheme Using Learning Automata

Hamid Beigy and M. R. Meybodi

Soft Computing Laboratory
Computer Engineering Department
Amirkabir University of Technology
Tehran, Iran
{beigy, meybodi}@ce.aku.ac.ir

Abstract

In this paper, we propose a new continuous action learning automaton and an adaptive and autonomous call admission algorithm, which uses the proposed learning automaton. The proposed algorithm minimizes blocking probability of new calls subject to constraint on dropping probability of handoff calls. The simulation results show that the performance of this algorithm is close to the performance of limited fractional guard channel policy for which we need to know all traffic parameters in advance. Two advantages of the proposed policy are that it is fully autonomous and adaptive. The first implies that, the proposed policy doesn't require any exchange of information between the neighboring cells and the second implies that, the proposed policy doesn't need any priori information about input traffic.

1. Introduction

Call admission (CAC) policies determine whether a new call should be admitted or blocked. Both blocking probability of new calls (B_n) and dropping probability of handoff calls (B_h) are affected by CAC policies. Blocking more new calls generally improves dropping probability of handoff calls and admitting more new calls generally improves blocking probability of new calls. Suppose that the given cell has C full duplex channels. The simplest CAC policy, called *guard channel* policy (GC),reserves a subset of channels allocated to a cell for sole use of handoff calls (say $C-T$ channels) [1]. These channels are called *guard channels.* Whenever the channel occupancy exceeds the 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. As the number of guard channels increased, the dropping probability of handoff calls will be reduced while the blocking probability of new calls will be increased [2]. It has been shown that there is an optimal threshold T^* in which the blocking probability of new calls is minimized subject to the hard constraint on the dropping probability of handoff calls [3]. Algorithms for finding the optimal number of guard channels are given in [3, 4]. These algorithms assume that the input traffic is a stationary process with known parameters. If the parameter B_h is considered, the guard channel policy gives very good performance, but the parameter B_n is degraded to great extent. In order to have more control on blocking probability of new calls and the dropping probability of handoff calls, *limited fractional guard channel policy* (LFG) is introduced [3]. The LFG policy uses an additional parameter π. The LFG policy is same to the GC policy except when T channels are occupied in the cell. In such situations, the LFG policy accepts new calls with probability π. 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 the dropping probability of handoff calls [3]. The algorithm for finding such optimal parameters is given in [3]. Since the input traffic is not a stationary process and its parameters are unknown a priori, the optimal number of guard channels cannot determined. In such cases the *dynamic guard channel* policy can be used in which number of guard channels varies during the operation of the cellular network.

Learning automaton (LA) has been used successfully in many applications such as telephone and data network routing [5], solving NP-Complete problems [6, 7, 8] and capacity assignment [9], to mention a few. In this paper, we propose new continuous action LA and an adaptive and autonomous call admission algorithm, which uses the proposed continuous action learning automata. This algorithm uses only the current channel occupancy of the given cell and dynamically adjusts number of guard channels. The proposed algorithm minimizes blocking probability of new calls subject to constraint on dropping probability of handoff calls. Since learning automaton starts its learning without any priori knowledge about its environment, the proposed algorithm does not need any a priori information about input traffic. One of the most important advantage of the proposed algorithm is that no status information will be exchanged between neighboring cells. Of course exchange of such status information can be used to increase the performance of the proposed algorithm. The simulation results show that performance of the proposed algorithm without any priori information is close to performance of LFG policy that knows all traffic parameters.

The rest of this paper is organized as follows: Section 2 presents blocking performance of LFG policy. LA is given in section 3 briefly. The new continuous action learning automaton and the proposed algorithm are given in section 4. Simulation results is given in section 5 and section 6 concludes the paper.

2. The Blocking Performance of LFG

Blocking performance of LFG policy is computed based on the following assumptions.

1. The arrival process of new and handoff calls is poisson process with rate λ_n and λ_h, respectively. Let $\lambda = \lambda_n + \lambda_h$, $\rho = \lambda/\mu$, and $\alpha = \lambda_h/\lambda$.
2. The call holding time for both types of calls is exponentially distributed with mean μ^{-1}.
3. The time interval between two calls from a mobile host is much greater than mean call holding time.
4. Only mobile to fixed calls are considered.
5. The network is homogenous.

The above first three assumptions have been found to be reasonable as long as the number of mobile hosts in a cell is much greater than the number of channels allocated to that cell. The fourth assumption makes our analysis easier and the fifth one allows us to examine the performance of a single network cell in isolation. Suppose that the given cell has a limited number of full duplex channels, C, in its channel pool. In the LFG policy, when the channel occupancy exceeds threshold T, new calls are rejected until the channel occupancy goes below the threshold and when the channel occupancy equals to T, new calls are accepted with probability of π. The LFG policy accepts handoff calls as long as channels are available. We define the state of a particular cell at time t to be the number of busy channels in that cell, which is represented by $c(t)$. The $\{c(t)|t \geq 0\}$ is a continuous-time Markov chain with states $0, 1, \ldots, C$. The state transition rate diagram of a cell with C full duplex channels and LFG policy is shown in figure 1. Define the steady state probability

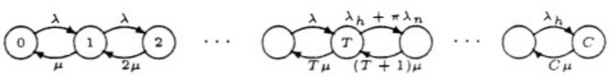

Fig. 1. Markov chain model of cell

$P_n = \lim_{t\to\infty} \text{Prob}[c(t) = n]$ for state $n = 0, 1, \ldots, C$. Then, the following expression can be derived for P_n.

$$P_n = \frac{\rho^n \prod_{i=1}^{n} \gamma_i}{n!} P_0 \tag{1}$$

where γ_i is arrival rate in state i equals to and given by the following equation.

$$\gamma_i = \begin{cases} \lambda & \text{if } i < T \\ \lambda_h + \pi\lambda_n & \text{if } i = T \\ \lambda_h & \text{if } i > T \end{cases}$$

P_0 is the probability that all channels are free and calculated by the following expression.

$$P_0 = \left[\sum_{k=0}^{C} \frac{\rho^k \prod_{i=1}^{k} \gamma_i}{k!} \right]^{-1} \tag{2}$$

Hence, the dropping probability of handoff calls using C channels and $C - T$ guard channels is equal to

$$B_h(C, T) = \frac{\rho^C \alpha^{C-T}}{C!}. \tag{3}$$

Similarly, the blocking probability of new calls is given by the following expression.

$$\begin{aligned} B_n(C, T) &= \sum_{k=T+1}^{C} P_k + (1 - \pi) P_T \qquad (4) \\ &= (1 - \pi)\frac{\rho^T}{T!} + \sum_{k=T+1}^{C} \frac{\rho^k \alpha^{k-T}}{k!} \end{aligned}$$

The objective of our call admission policy is to find a $T^* + \pi^*$ that minimizes the $B_n(C, T^* + \pi^*)$ given the constraint $B_h(C, T^* + \pi^*) \leq p_h$. The value of p_h specifies by the quality of service of the network. In order to find the optimal value of $T^* + \pi^*$, in [3] a binary search algorithm is given. This algorithm assumes that all parameters of traffic are known in advance.

3. Learning Automata

The automata approach to learning involves determination of an optimal action from a set of allowable actions. Automaton selects an action from its finite set of actions and applies it to environment which in turn emits a stochastic response $\beta(n)$ at the time n. $\beta(n)$ is an element of $\beta = \{0, 1\}$ and is the feedback response of the environment to the automaton. The automaton is penalized by the environment with probability c_i, which is action dependent. On the basis of response $\beta(n)$, state of the automaton is updated and a new action chosen at $(n+1)$. Note that the c_i's are unknown initially and it is desired that as a result of interaction between the automaton and the environment arrives at the action which presents it with the minimum penalty response in an expected sense. An automaton acting in an unknown environment and improves its performance in some specified manner, is called *learning automaton* (LA). Since LA has finite number of actions, for continuous action variable, a corresponding discrete action will be created by discretizing the continuous variable at some points. For precise discretization, number of actions must be large which decreases the convergence speed. To increase precision of discritization, *continuous action set learning automaton* (CALA) is introduced [10]. The CALA chooses actions from the real line $\Re$. The action probability distribution at instant n is a normal distribution with mean $\mu(n)$ and standard deviation $\sigma(n)$. At each instant, the CALA updates its action probability distribution by updating $\mu(n)$ and $\sigma(n)$, which is analogous to updating the action probabilities by the LA with finite set of actions. Since action set is continuous, instead of reward probabilities for various actions, we now have a reward probability function, $f : \Re \rightarrow \Re$, defined by $f(\alpha) = E[\beta(n)|\alpha(n) = \alpha]$. It is assumed that $f(\alpha)$ is unknown to the automaton. The objective for CALA is to learn value of α at which $f(\alpha)$ attains a maximum. That is, we want the action probability distribution, $N(\mu(n), \sigma(n))$ to converge to $N(\alpha_0, 0)$ where α_0 maximizes $f(\alpha)$. Learning algorithm for CALA can be described below. Unlike finite action set LA, CALA interacts with the environment through choice of two actions at each instant. At each instant n, the CALA chooses $\alpha(n) \in \Re$ at random from its current distribution $N(\mu(n), \phi(\sigma(n)))$ and pair pair $(\mu(n), \alpha(n))$ is applied to the environment. Then it gets the reinforcement from the environment for pair $(\mu(n), \alpha(n))$. Based on these reinforcement signals, the action probability distribution is updated. For this algorithm it is proved that with arbitrary large probability, $\mu(n)$ will converge close to a maximum of $f(.)$ and $\sigma(n)$ will converge close to zero [10].

Continuous action reinforcement learning automata (CARLA) is introduced in [11], independently and can be described as follows. Let output $\alpha(n)$ of automaton be a bounded continuous random variable defined over interval $[\alpha_{min}, \alpha_{max}] \in \Re$. In the CARLA, action probability vector $p(n)$ is replaced by a continuous probability density function $f_\alpha(n)$. 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. This learning automaton selects action $\alpha(n)$ at instant n and applies to the environment which emits a response $\beta(n) \in [0, 1]$. Based on $\beta(n)$, $f_\alpha(n)$ is updated according to the following rule: When $\alpha(n) \in [\alpha_{max}, \alpha_{min}]$, $f_\alpha(n)$ is set to $a[f_\alpha(n) + (1 - \beta(n))H(\alpha, \alpha(n))]$ and otherwise $f_\alpha(n)$ is set to 0, where a is a normalization factor and $H(\alpha, r)$ is a symmetric Guassian neighborhood function centered on $r = \alpha(n)$ and spreads the reward for neighboring actions. The asymptotic behavior of the continuous action learning automata is not known.

4. Proposed Algorithm

In this section, we first introduce a new continuous learning automaton and then purpose a LA based algorithm for finding the optimal value of $T + \pi$. The proposed algorithm uses the proposed continuous action learning automaton and adjusts the value of $T + \pi$ as the network operates.

4.1 A New Continuous Action Learning Automaton

In some applications, such as call admission control, when value of a parameter, say x, is large (small), then all values of $y > x$ ($y < x$) are also large (small). In such situations, if x is penalized (rewarded), all values of $y > x$ or $y < x$ also must be penalized (rewarded). Since the standard LA don't have this feature, a new continuous action LA having this feature is introduced. For learning algorithm studied in this paper, we use Gaussian distribution $N(\mu, \sigma)$, which is completely specified by the first and second order momentums: μ and σ. The learning algorithm updates μ and σ of Gaussian distribution at any instant using the reinforcement signal, β. In the proposed learning automaton, $\sigma(n)$ and $\mu(n)$ are adjusted by the learning algorithm. The proposed LA chooses its actions with equal probability at the beginning which is achieved by choosing a large value for variance. The proposed LA can be described algorithmically as follows.

Step 1: Set $\sigma(0)$ to a large and $\mu(0)$ to a random number.
Step 2: Choose action $\alpha(n) \sim N(\mu(n), \sigma(n)$.
Step 3: Apply $\alpha(n)$ to the environment and get response $\beta(n) \in [-1, +1]$.
Step 4: Update $\mu(n)$ and $\sigma(n)$ by following equations.

$$\mu(n+1) = \mu(n) + a(n)\beta(n) \tag{5}$$

$$\sigma(n+) = f(\sigma(n)) \tag{6}$$

Step 5: Go to Step 2.

Fig. 2. Learning algorithm used by the proposed continuous action learning automaton

From the above algorithm, it is apparent that $\mu(n)$ pursues the optimal value. In this algorithm, f is any arbitrary function that has property $x > f(x)$. This property results a narrower probability density function and converges to zero as n increases to infinity. Equation (5) moves the mean to its optimal value. In order to have a stable equilibrium point, the proposed algorithm uses a monotonically decreasing learning parameter a.

4.2 LA Based LFG Algorithm

In this section, we introduce a new learning automata based algorithm (figure 3) to determine the optimal value of $T + \pi$ for LFG algorithm when parameters λ_n, λ_h, and μ are unknown and possibly time varying. Assume that the cell has C full duplex channels. Let $x(n) = T(n) + \pi(n)$ be the parameter of LFG algorithm at instant n and $x^* = T^* + \pi^*$ be the optimal value of x, which minimizes the blocking probability of new calls subject to constraint that the dropping probability of handoff calls is at most p_h. Let $x(n)$ is in interval $[g_{\min}, g_{\max}]$, where $0 \leq g_{\min} \leq g_{\max} \leq C$. In the proposed algorithm, each base station uses a continuous action learning automaton A described in the previous section to adjust x. The objective of our call admission control policy is to minimize $B(C, x)$ subject to the hard constraint $B_h(C, x) \leq p_h$. Since $x(n)$ must be in the interval $[g_{\min}, g_{\max}]$, the proposed continuous action learning automata cannot be applied directly and we use a projected version of it. In the projected version of continuous action learning automata, the constraint set H is $\{\mu | g_{\min} \leq \mu \leq g_{\max}\}$. Thus the step 4 of the proposed learning automata is replaced by $\mu(n+1) = \Pi_H(\mu(n) + a(n)\beta(n))$, where Π_H is the projection on to the constraint set H and is defined by the following relation.

$$\Pi_H(x) = \begin{cases} g_{\min} & \text{if } x < g_{\min} \\ x & \text{if } g_{\min} \leq x \leq g_{\max} \\ g_{\max} & \text{if } x > g_{\max} \end{cases}$$

Since $B_n(C, x)$ is a decreasing function of x, it doesn't have minimum. Therefore, we define the following objective function.

$$\tilde{B}_n(C, x) = B_n(C, x)\mathrm{I}\{x \leq x^*\} + B_n(C, 2x^* - x)\mathrm{I}\{x > x^*\}$$

where I(.) is indicator function. The above objective function has a minimum at x^*, which occurred when $\tilde{B}_h(C, x^*) = p_h$. Thus we define an indicator of error $\beta = \hat{B}_h(C, x) - p_h$, which acts as the response of the environment. Now, we describe the proposed LA based LFG algorithm. Since the controlling handoff calls is not beneficial as it leads to idling of some channels, the handoff calls are accepted as long as channels are available and the proposed algorithm is used only when new calls arrive. When a new call arrives at the given cell, learning automaton associated to that cell selects one of its action, say $\alpha(n) = T + \pi$. If the number of busy channels of cell is less than T, then the incoming call is accepted; when cell has T busy channels the call is accepted with a certain probability of π; otherwise the incoming call is blocked. Then the base station computes the current estimate of dropping probability of handoff calls ($\hat{B}_h$), indicator of error, and μ, σ, and a are updated by the learning algorithm.

```
set x ← LA.action ()
if (c(t) < T )then
    accept call
else if (c(t) = T and  rand(0,1) ≤ π )then
    accept call
else
    reject call
end if
set β ← B̃_h(C,x) − p_h
LA.update (β).
```

Fig. 3. LA based dynamic guard channel algorithm

By careful inspection of the above algorithm, it is evident that when β is negative (positive), $T + \pi$ is decreased (increased) and when β is zero, $T + \pi$ is unchanged. The decreasing of learning rate ensures stability of the proposed learning algorithm at the optimal point.

5. Simulation Results

In this section, we compare performance of the limited fractional guard channel [3] and the proposed algorithm. The results of simulations are summarized in table 1. The simulation is based on the single cell of homogenous cellular network system. In such network, each cell has 8 full duplex channels ($C = 8$). In the 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 table 1 are obtained by averaging 10 runs from 2,000,000 seconds simulation of each algorithm. The objective is to minimize the blocking probability of new calls subject to the constraint that the dropping probability of handoff calls is less than 0.01. The optimal parameters of limited fractional guard channel policy is obtained by algorithm given in [3].

Table 1. Minimize B_n such that $B_h \leq 0.01$

		LFG		Proposed Algorithm	
Case	λ_h	B_n	B_h	B_n	B_h
1	2	0.031609	0.023283	0.050784	0.010063
2	4	0.051414	0.020675	0.067582	0.009794
3	6	0.071632	0.018707	0.084571	0.010059
4	8	0.092138	0.016706	0.103642	0.009394
5	10	0.114445	0.015572	0.135579	0.009333
6	12	0.147902	0.014044	0.181468	0.01009
7	14	0.204217	0.012675	0.227705	0.009718
8	16	0.250642	0.011554	0.257067	0.009617
9	18	0.294441	0.010877	0.30659	0.010053
10	20	0.384157	0.010182	0.385888	0.010052

By inspecting table 1, it is evident that 1) upper bound of dropping probability of handoff calls is maintained by our algorithm, while this upper bound in not maintained by the LFG. 2) performance of the proposed algorithm is close to the performance of LFG algorithm. One reason for the difference in performances of the LFG and the proposed policies is due to the fact that transient behavior of the proposed algorithm is included. Since, the blocking probability of new calls and the dropping probability of handoff calls in the early stages of simulation are far from their desire value, they affect the long-time calculation of the performance parameters. However, such effect can be removed by excluding the transient behaviors from our algorithm.

6. Conclusions

In this paper, a new continuous action learning automaton is introduced. Then a call admission control algorithm is given which uses the proposed LA. The simulation results show the the power of the proposed algorithm. The proposed algorithm has the following advantages: 1) Its performance is very close to the performance of the LFG policy. 2) The upper bound of dropping probability is maintained by our algorithm which is optimal when all information are available. 3) The proposed algorithm doesn't need any information about the input traffic.

References

[1] D. Hong and S. Rappaport, "Traffic Modelling and Performance Analysis for Cellular Mobile Radio Telephone Systems with Priotrized and Nonpriotorized Handoffs Procedure," *IEEE Transactions on Vehicular Technology*, vol. 35, pp. 77–92, Aug. 1986.

[2] S. Oh and D. Tcha, "Priotrized Channel Assignment in a Cellular Radio Network," *IEEE Transactions on Communications*, vol. 40, pp. 1259–1269, July 1992.

[3] R. Ramjee, D. Towsley, and R. Nagarajan, "On Optimal Call Admission Control in Cellular Networks," *Wireless Networks*, vol. 3, pp. 29–41, 1997.

[4] G. Haring, R. Marie, R. Puigjaner, and K. Trivedi, "Loss Formulas and Their Application to Optimization for Cellular Networks," *IEEE Transactions on Vehicular Technology*, vol. 50, pp. 664–673, May 2001.

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

[6] B. J. Oommen and E. V. de St. Croix, "Graph Partitioning Using Learning Automata," *IEEE Transactions on Commputers*, vol. 45, pp. 195–208, Feb. 1996.

[7] H. Beigy and M. R. Meybodi, "An Algorithm Based on Learning Automata for Determining the Minimum Number of Hidden Units in Three Layers Neural Networks," *Amirkabir Journal of Science and Technology*, vol. 12, no. 46, pp. 111–136, 2001.

[8] M. R. Meybodi and H. Beigy, "New Class of Learning Automata Based Schemes for Adaptation of Backpropagation Algorithm Parameters," *International Journal of Neural Systems*, vol. 12, pp. 45–68, Feb. 2002.

[9] B. J. Oommen and T. D. Roberts, "Continuous Learning Automata Solutions to the Capacity Assignment Problem," *IEEE Transactions on Commputers*, vol. 49, pp. 608–620, June 2000.

[10] G.Santharam, *Distribute Learning with Connectionist Models for Optimization and Control.* PhD thesis, Departement of Electrical Engineering, Indian Insititue of Science, Bangalore, India, 1994.

[11] G. P. Frost, *Stochastic Optimization of Vehicle Suspension Control Systems Via Learning Automata.* PhD thesis, Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK, Oct. 1998.