WITID

ITRC

Proceedings

2nd

February 24-26, 2004
Kish Island- Iran

Workshop on Information Technology & Its Disciplines (WITID 2004)

The 2[nd] Workshop on Information Technology & Its Disciplines (WITID 2004)
CSI

A Dynamic Channel Assignment Algorithm: A Cellular Learning Automata Approach

Hamid Beigy and M. R. Meybodi*

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

ABSTRACT. Introduction of micro-cellular networks offer a potential increase in capacity of cellular networks, but they create problems in management of the cellular networks. A solution to these problems is self-organizing channel assignment algorithm with distributed control. In this paper, we first present the model of cellular learning automata and then a cellular learning automata based dynamic channel assignment algorithm is introduced. The simulation results show that the micro-cellular network can self-organize itself by using the proposed channel assignment algorithm as the network operates.

1 INTRODUCTION

With increasing popularity of mobile computing, demand for channels is on the rise. Since the number of channels allocated to the cellular network is limited, efficient management and sharing of channels among numerous users become an important issue. The limited number of channels means that channels have to be reused as much as possible in order to support the many thousands of simultaneously calls that may arise in any typical mobile communication environment. In order to support wireless communication for mobile hosts, geographical area covered by mobile network is divided into smaller regions called *cells*. Each cell has a fixed server computer called *base station* (BS), which is located at its center. A number of BSs are linked to a fixed computer called *mobile switching center* (MSC) which also acts as a gateway of the mobile network to the existing wired-line networks. The BSs are connected to the wired-line network and communicate with mobile hosts through wireless links and with MSCs through wired-line links. A mobile host communicates with any other node in the network, fixed or mobile, only through the BS of its cell using wireless communication. If a channel is used concurrently by more than one communication sessions in the same cell or in the neighboring cells, the signal of communicating units will interfere with others. Such interference is called *co-channel interference*. However, the same channel can be used in geographically separated cells such that their signal do not interfere with each other. The minimum distance at which co-channel can be reused with acceptable interference is called *co-channel reuse distance*. The set of all neighboring cells that are in co-channel interference range of each other form a *cluster*. At any time, a channel

* This work is partially supported by Iranian Telecommunication Research Center (ITRC), Tehran, Iran.

can be used to support at most one communication session in each cluster. The problem of assigning channels to communication sessions is called *channel assignment problem*.

There are several schemes for assigning channels to communication sessions, which can be divided into a number of different categories depending on the comparison basis. For example, when channel assignment algorithms are compared based on the manner in which co-channels are separated, they can be classified as *fixed channel assignment* (FCA), *dynamic channel assignment* (DCA), and *hybrid channel assignment* (HCA) schemes [1]. In FCA schemes, a set of channels are permanently allocated to each cell, which can be reused in another cell, at sufficiently distance, such that interference is tolerable. FCA are formulated as generalized graph coloring problem and belongs to class of NP-Hard problems [2–5]. In DCA schemes, there is a global pool of channels from where channels are assigned on demand and the set of channels assigned to a cell varies with time. After a call is completed, the assigned channel is returned to the global pool [6–11]. In the next paragraph, we describe a more detailed description of two DCA schemes: channel segregation [6] and reinforcement learning based channel assignment [7] which will be compared with the schemes proposed in this paper.

The *channel segregation* is a distributed self-organized channel assignment scheme in which each base station selects a channel with an acceptable co channel interference by scanning all channels. The scanning order is formed independently for each cell on the basis of a channel selection priority vector p. Each base station keeps the current value of channel selection priority $p(i)$ for channel i and selects a channel with highest priority value. In order to determine co channel interference, the received power level of the selected channel is measured. If the measured power level is above (below) of a threshold value, the channel is determined to be busy (idle). If the selected channel is idle, the base station starts communication using that channel and its priority is increased. If the channel is busy, the priority of the channel is decreased and the next highest priority channel is tried. If all channels are busy, the call is blocked. In channel segregation scheme, the value of the priority vector is updated in the following manner.

$$p(i) = \begin{cases} \frac{p(i)N(i)+1}{N(i)+1} & \text{if channel } i \text{ is idle.} \\ \\ \frac{p(i)N(i)}{N(i)+1} & \text{if channel } i \text{ is busy.} \end{cases} \quad (1)$$

where $N(i)$ is the number of times channel i is selected. The simulation results show that the channel segregation strategy uses channels efficiently and decreases the number of intra cell handoff, that is, the reassignment of channels to avoid interference. Singh and Bertsekas have been formulated the dynamic channel assignment using dynamic programming and reinforcement learning is used for solving it. In their formulation, state transition occurs when a channel becomes free due to call departures or when a call arrives at a given cell or when there is handoff. In their approach, at each instant, the state of system consists of two components: the lists of occupied and unoccupied channels at each cell, which referred to as configuration of cellular system and the event list (call arrival, call termination, or handoff) that causes the state transition. When a channel becomes free, due to the call departure in a particular cell, channels at a given cell is rearranged. The aim of this rearrangement is creating a more favorable channel packing pattern among cells. The objective of this formulation is to assign channels to calls which results minimization of the expected number of blocked calls over an infinite horizon.

In HCA schemes, channels are divided into *fixed* and *dynamic* sets [7,12]. Fixed set contains a number of channels that are assigned to cells as in the FCA schemes. The fixed channels of a particular cell are assigned only for calls initiated in that cell. Dynamic set of channels is shared between all users in network to increase flexibility. When a request for service is received by a base station, if there is a free channel in fixed set then the base station assigns a channel from fixed set and if all channels in the fixed set are busy, then a channel is allocated from dynamic set. Any DCA strategies can be used for assigning channels from dynamic set.

In this paper, we first present cellular learning automata (CLA) model. The basic idea of CLA is to use learning automata (LA) to adjust the state transition probability of cellular automata (CA) [13, 14]. Then we propose a self-organizing channel assignment algorithm. In order to show the feasibility of the proposed algorithm, computer simulations are conducted. The simulation results show that the cellular network can self-organize the assignment of channels by using simple channel assignment algorithm as network operates. The proposed algorithm has been compared with two existing methods: channel segregation algorithm reported by Furuya and Akaiwa, and reinforcement learning algorithm reported by Singh and Bertsekas. This algorithm segregates the channels among cells of the network in such a way that the blocking probability is in an acceptable range. Even the blocking probability of the proposed algorithm is slightly higher than the blocking probability for channel segregation and reinforcement learning algorithms, but it requires lesser number of messages to be exchanged among the cells.

The rest of the paper is organized as follows. In section 2, a brief review of learning automata is given and in section 3, the cellular learning automata is presented. Sections 4 and 5 present the proposed algorithm and numerical results, respectively and section 6 concludes the paper.

2 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 emits a response. This response is used by automaton to select its next action. By continuing this process, the automaton learns to select the action with the best response. The learning algorithm used by automaton to determine the selection of next action from the response of environment. An automaton acting in an unknown random environment and improves its performance in some specified manner, is referred to as *learning automaton.* LA can be classified into two main families: *fixed structure LA* and *variable structure LA* [15]. Variable structure LA 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 LA, is learning algorithm for updating the action probabilities. Various learning algorithms have been reported in the literature. In what follows, two learning algorithms for updating the action probability vector are given. Let α_i be the action chosen at time k as a sample realization from probability distribution $p(k)$. In linear reward-ϵpenalty algorithm ($L_{R-\epsilon P}$) scheme the recurrence equation for updating p is defined as

$$p_j(k+1) = \begin{cases} p_j(k) + a \times [1 - p_j(k)] & \text{if } i = j \\ p_j(k) - a \times p_j(k) & \text{if } i \neq j \end{cases} \quad (2)$$

when $\beta(k) = 0$ and

$$p_j(k+1) = \begin{cases} p_j(k) \times (1 - b) & \text{if } i = j \\ \frac{b}{r-1} + p_j(k)(1 - b) & \text{if } i \neq j \end{cases} \quad (3)$$

when $\beta(k) = 1$. Parameters $0 < b \ll a < 1$ represent *step lengths* and r is the number of actions for LA. The $a(b)$ determines the amount of increase(decreases) of the action probabilities. If $a = b$, then the recurrence equations (2) and (3) is called *linear reward penalty*(L_{R-P}) algorithm and if $b = 0$, then the recurrence equations (2) and (3) is called *linear reward inaction*(L_{R-I}) algorithm. LA have been used successfully in many applications such as telephone and data network routing [16], solving NP-Complete problems [17], capacity assignment [18] and neural network engineering [19, 20] to mention a few.

3 CELLULAR LEARNING AUTOMATA

Cellular automata are mathematical models for systems consisting of large numbers of simple identical components with local interactions. The simple components act together to produce complicated patterns of behavior. CA perform complex computation with high degree of efficiency and robustness [21]. CA are non-linear dynamical systems in which space and time are discrete. A CA consists of a finite dimensional lattice of cells whose states are restricted to a finite set of integers $\phi = \{0, 1, \cdots, k-1\}$. The state of each cell at any time instant is determined by a rule from states of neighboring cells at the previous time instant. Given a finite set ϕ and a finite dimension d, CA can be considered as a d-dimensional lattice Z^d in which every point has a label from set ϕ.

Cellular learning automata (CLA) is a mathematical model for dynamical complex systems that consists of large number of simple components [13]. The simple components, which have learning capability, act together to produce complicated behavioral patterns. A CLA is a CA in which a learning automaton (multiple automata) will be assigned to its every cell. The learning automaton residing in each cell determines the state of the cell on the basis of its action probability vector. Like CA, there is a rule that CLA operate under it. The rule of CLA and the actions selected by the neighboring LAs of any cell determine the reinforcement signal to the LA residing in that cell. In CLA, the neighboring LAs of any cell constitute its local environment. This environment is a nonstationary environment because of the fact that it changes as action probability vectors of neighboring LAs vary.

The operation of cellular learning automata could be described as follows: At the first step, the internal state of every cell is specified. The state of every cell is determined on the basis of action probability vectors of the learning automata residing in that cell. The initial value of this state may be chosen on the basis of past experience or at random. In the second step, the rule of cellular automata determines the reinforcement signal to each learning automaton residing in that cell. Finally, each learning automaton updates its action probability vector on the basis of supplied reinforcement signal and the chosen action. This process continues until the desired result is obtained.

In [14], a mathematical methodology to study the behavior of the synchronous CLA is developed and its convergence properties has been investigated. It is shown that the synchronous CLA converges to a globally stable state for a class of rules called commutative rules. The CLA can be classified into *synchronous* and *asynchronous* CLA. In synchronous CLA, all cells are synchronized with a global clock and executed at the same time. In [22], asynchronous cellular learning automata is introduced and its behavior is studied. It is shown that the synchronous CLA converges to a globally stable state for commutative rules. A CLA is called asynchronous if at a given time only some LAs are activated independently from each other, rather than all together in parallel. The learning automata may be activated in either *time-driven* or *step-driven* manner. In time-driven asynchronous CLA, each cell is assumed to have an internal clock which awaken the learning automaton associated to that cell while in step-driven asynchronous CLA, a cell is selected in a fixed order or at random. The asynchronous CLA in which cells are selected randomly is of more interest to us because of its application to cellular mobile networks. Recently a number of applications for CLA have been developed such as image processing [23], rumor diffusion [24], image processing [25, 26], modelling of commerce networks [27], fixed channel assignment in cellular networks [5], and VLSI Placement [28].

Formally a $d-$dimensional asynchronous step-driven CLA is given below.

Definition 1. *A $d-$dimensional asynchronous step-driven cellular learning automata is a structure $A = (Z^d, \Phi, A, N, \mathcal{F}, \underline{\rho})$, where*

1. *Z^d is a lattice of $d-$tuples of integer numbers.*
2. *$\underline{\Phi}$ is a finite set of states.*
3. *A is the set of LAs each of which is assigned to each cell of the CA.*
4. *$N = \{\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_{\bar{m}}\}$ is neighborhood vector.*
5. *$\mathcal{F} : \underline{\Phi}^{\bar{m}} \rightarrow \underline{\beta}$ is the local rule of the cellular automata.*
6. *$\underline{\rho}$ is an $n \times n$ diagonal activation matrix and represents the step driven vector, where ρ_{ii} is the activation probability of learning automata in cell i. When $\rho_{ii} = 1$, for $i = 1, \ldots, n$, the CLA is synchronous.*

4 THE PROPOSED CHANNEL ASSIGNMENT ALGORITHM

In this section, we propose a CLA based dynamic channel assignment algorithm for cellular network. In the proposed algorithm, we use an asynchronous cellular learning automata for dynamic assignment of channels in cellular networks. We assume that the cellular network has n cells and m full duplex and interference free channels. In order to assign channels to calls dynamically, we use an asynchronous cellular learning automata with n cells and m learning automata of L_{R-I} type in each cell. Each learning automaton in any particular cell corresponds to a channel. Let A_i (for $i = 1, \ldots, n$) be the learning automaton corresponding to channel i. The action set for all learning automata in the cellular learning automata is $\underline{\alpha} = \{0, 1\}$, where 0 means that the channel corresponding to this learning automaton is not a candidate channel for using in this cell while 1 means that the corresponding channel is a candidate channel for using in this cell. The neighboring cells of each cell are cells in its interference region. That is cells in a cluster of cells are the neighbors of that cell. The local rule determines whether or not the channel selected by a cell interferes with channels used in the other cells in cluster. The result of rule is 1 if the chosen channel doesn't interfere with other channels in the cluster and 0 otherwise.

To compute the result of the local rule, any base station needs to know whether or not a candidate channel is being used in the neighboring cells. This is usually implemented by transmission of messages among the base station of the activated cell and the base stations of its neighboring cells. The transmission of such messages wastes the bandwidth of wire-line network and also increases the response time of the channel assignment algorithm. The burden of message transmission among the neighboring base stations can be eliminated by using interference detection hardware in each base station. By using such a hardware, each base station can determine the number of used channels in its neighboring cells without any message transmission. Using the statistical data of interference detection, each learning automaton in a given cell learns the interference map among the base stations. The proposed algorithm causes the base station to give higher priority to some channels (by increasing the probability of action 1 of the corresponding learning automaton) and low priority to other channels.

The activation matrix $\underline{\rho}$ for the asynchronous cellular learning automata is obtained by taking expectation from product of two $nm \times nm$ diagonal binary matrices $\underline{\pi}_1$ and $\underline{\pi}_2$, i.e. $\underline{\rho} = \mathrm{E}[\underline{\pi}_1 \underline{\pi}_2]$. The matrix $\underline{\pi}_1$ is called *cell activation matrix* and determines when a cell is activated. Elements (i, i) (for $(j-1)m < i \leq jm$) of matrix $\underline{\pi}_1$ become equal to 1 when cell j (for $j = 1, \ldots, n$) is activated. It is apparent that when a call arrives to any particular cell, that cell is activated. Thus $\mathrm{E}[\pi_1(i, i)] = 1/\lambda_j$ (for $(j-1)m < i \leq jm$) and $j = 1, 2, \ldots, n$. The matrix $\underline{\pi}_2$ is called *learning automata activation matrix* and determines when a learning automaton in the activated cell is activated. In other words, the matrix $\underline{\pi}_2$ determines the order according to which learning automata in the activated cells are activated. Elements (i, j) (for $(j-1)m < i \leq jm$) of matrix $\underline{\pi}_2$ are equal to 1 when learning automaton A_k (for $k = i - (j-1)m$) in cell j

(for $j = 1, \ldots, n$) is activated. Thus $E[\pi_2(i,i)|\pi_1(i,i) = 1]$ (for $(j-1)m < i \leq jm$ and $j = 1, 2, \ldots, n$) is the frequency of activation of the learning automaton A_k (for $k = i-(j-1)m$) in cell j. The matrix $\underline{\pi}_1$ is determined by the call arrival rate while the matrix $\underline{\pi}_2$ can be obtained in various ways. In the rest of this section, we consider some strategies for obtaining the matrix $\underline{\pi}_2$.

Fixed sweep strategy : Suppose that a call arrives in cell j (for $j = 1, \ldots, n$). Then, the learning automata in cell j are activated according to matrix $\underline{\pi}_2$, which is recomputed every time a learning automaton is activated. When a call arrives at cell j, and there is no candidate channel to be assigned, then the incoming call will be blocked.

The recomputation of $\underline{\pi}_2$ is done in the following way. Suppose that cell j is activated, then as indicated in the previous paragraph $\pi_1(i,i)$ becomes 1 for $i = (j-1)m + 1, \ldots, jm$ and $\pi_2((j-1)m+1, (j-1)m+1)$ becomes 1. The remaining elements of $\underline{\pi}_2$ are computed according to the following rule.

$$\pi_2(i,i) = \begin{cases} 1 & \text{if } \pi_1(i,i) = 1 \text{ and } \pi_2(i-1, i-1) = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

for $i = (j-1)m + 1, \ldots, jm$.

In other words, in this strategy, the learning automata in cell j are activated sequentially in increasing order of their numbers in the list of learning automata for cell of j until a channel is found for the assignment.

Maximum usage strategy : Suppose that a call arrives in cell j (for $j = 1, \ldots, n$). In maximum usage strategy, the learning automata in cell j are activated in decreasing order of their usage of their corresponding channels in the cell j until a channel is found for the assignment. When there is no channels to be assigned to the incoming call, the incoming call is blocked. In this strategy, the learning automaton A_i is activated in the kth stage of the activation of cell j if u_i^j is the kth largest element in usage vector $\underline{u}^j = \{u_1^j, u_2^j, \ldots, u_m^j\}$, where u_i^j is the number of times that the channel i is assigned to the calls in cell j.

Minimum usage strategy : Suppose that a call arrives in cell j (for $j = 1, \ldots, n$). In minimum usage strategy, the learning automata in cell j are activated in increasing order of their usage of their corresponding channels in cell j until a channel is found for the assignment. When there is no channels to be assigned to the incoming call, the incoming call is blocked. In this strategy, the learning automaton A_i is activated in the kth stage of the activation of cell j if u_i^j is the kth smallest element in usage vector $\underline{u}^j = \{u_1^j, u_2^j, \ldots, u_m^j\}$, where u_i^j is the number of times that the channel i is assigned to the calls in the cell j.

Random sweep strategy : In this strategy, the learning automata in cell j are activated in random order. First a sequence of indices are generated randomly and then learning automata are activated according to the generated order.

5 NUMERICAL EXAMPLE

To verify the proposed channel assignment algorithm, computer simulations are performed. It is assumed that seven base stations, which are organized in a linear array, shares 5 full duplex and interference free channels. The interference constraints between any pair of cells is represented by an integer, which prescribes the minimum gap that exist between channels assigned to cells in order to avoid interference. The interference constraints for all cells are represented by matrix C. The element $c(i,j)$ of matrix C represents the interference constraint between cells i and j. The interference constraint $c(i,j)$ for the problem used in our simulation is defined as follows. Let $d(i,j)$ represents the normalized distance between the centers of cells i and j, where the distance between centers of adjacent cells are unity. Then we have

$$c(i,j) = \begin{cases} 1 & \text{if } d(i,j) \leq 2 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

If there is no interference between a pair of cells, then the corresponding elements in matrix C for this pair is defined as zero.

For the simulations reported in this section, we assume that the arrival of calls is Poisson process with rate λ and channel holding time of calls is exponentially distributed with mean $\mu = 1/3$. We also assume that no handoff occurs during the channel holding time. The results of simulations that are reported in this section are obtained from 120,000 seconds simulations. Figure 1 shows the average blocking probability of calls for different strategies reported in the previous section and compares with the results obtained for the channel segregation and reinforcement learning algorithms. Figure 2 shows the evolution of the blocking probability for different strategies proposed in the previous section for a typical run. Figure 3 shows the evolution of the number of interfering channels (the number of candidate channels that interfere with the channels being used in neighboring cells) as the cellular learning automata operates. This figure shows that the number of interfering channels is decreased as the result of learning. That is the cellular learning automata segregates channels among the cells of the network.

Figure 4 shows the number of messages transmitted (normalized to the number of calls arrived to the network) among the cell and its neighboring cells. Figure 5 shows the changes in the number of messages versus time for a typical run. This figure shows that the number of message transmission among the cell and its neighboring cells is decreased as the network operates. Figure 6 shows the number of candidate channels before a channel is assigned to calls. Figure 7 shows the number of candidate channels before a channel is assigned to calls as the network operates. The figures 8 through 11 show the probability of assigning different channels to different cells.

Remark 1. An important issue in the CLA based channel assignment algorithms is the speed of converges. An example of the convergence process is shown in figures 2 and 3. These figures show that the proposed approach quickly reaches to a sub-optimal allocation. Convergence to a solution which is an optimal allocation, if CLA can reach it, takes large amount of time. The time period required for convergence may be an order of months, which is long time in real applications. This time may be reduced by using other learning algorithms such as stochastic estimator [29] or pursuit learning [30] algorithms.

Remark 2. The final probability vectors of learning automata determines the performance of the CLA based channel assignment algorithm. The algorithm may be evolve in such a way that each probability vector converge to a unit vector or to a nonunit vector. In case that each probability vector converges to a unit vector, the final configuration of the CLA determines a pattern of allocation, which is fixed and cannot be changed during the operation of the system. This situation is useful when the traffic is stationary. In non-stationary traffic conditions, the convergence of each probability vector to a nonunit vector is more suitable. This is because, the allocation pattern will not be fixed and can adapt itself to incoming traffic.

Remark 3. If the results of the interference detection hardware are correct and there are no change in the propagation conditions during channel holding time, then the co-channel interference will never happen. Interference detection results, however, are not always correct because of fading phenomenon. Even, if the interference detection results are correct at the time of observation, the channel may suffer interference from changes in the propagation due to the movement of the mobile user. Therefor, it is impossible to perfectly avoid the interference so far as the channel assignment is based on the result of interference detection hardware. There is also another source of interference, which happen when two base stations try to use the same channel at the same time since their corresponding interference detection results show that this channel is idle. The proposed algorithm due to its learning capability decreases the probability of selection of the same channel by two base station simultaneously. This is because as the time goes by the pattern of allocation of channels among the cells becomes more stable. Of course if the traffic conditions change, a new pattern of allocation will be adapted by the algorithm.

6 CONCLUDING REMARKS

In this paper, an application of asynchronous cellular learning automata to channel assignment to cellular mobile system has been presented. In order to show the power of the proposed algorithm, which is a self-organizing channel assignment, computer simulations were conducted. Simulations showed that the proposed algorithm segregates the channels among cells of the network in such a way that the blocking probability is in an acceptable range. Even the blocking probability of the proposed algorithm is slightly higher than the blocking probability for channel segregation and reinforcement learning algorithms, but it requires lesser number of messages to be exchanged among the cells.

References

1. I. Katzela and M. Naghshineh, "Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A Comprehensive Survey," *IEEE Personal Communications*, pp. 10-31, June 1996.
2. W. K. Hale, "Frequence Assignment:Theory and Applications," *Proceedings of IEEE*, vol. 68, pp. 1497-1514, Dec. 1980.
3. J. A. Zoellner and C. L. Beall, "A Breakthrough in Spectrum Conserving Frequency Assignment Technology," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-12, pp. 313-310, Aug. 1977.
4. J. Zerovnik, "Experiments With a Randomized Algorithm for Frequency Assignment Problem," Tech. Rep. 97-27, Ecole Normale Superieure du Lyon, Lyon, France, Sept. 1997.
5. H. Beigy and M. R. Meybodi, *A Self-Organizing Channel Assignment Algorithm: A Cellular Learning Automata Approach*, vol. 2690 of *Springer-Verlag Lecture Notes in Computer Science*, pp. 119-126. Springer-Verlag, 2003.
6. Y. Furuya and Y. Akaiwa, "Channel Segregation– A Distributed Channel Allocation Scheme for Mobile Communication Systems," *IEICE Trans.* vol. 74, no. 6, pp. 1531-1537, 1991.
7. S. Singh and D. P. Bertsekas, "Reinforcement learning for dynamic channel allocation in cellular telephone systems," in *Advances in Neural Information Processing Systems: Proceedings of the 1996 Conference, Cambridge, MA. MIT Press*, 1997.
8. X. Dong and T. H. Lai, "Distributed Dynamic Carrier Allocation in Mobile Cellular Networks: Search vs. Update," in *Proceedings of the 17th International Conference on Distributed Computer Systems, Baltimore, Maryland*, pp. 108-115, May 1997.
9. J. Kind, T. Niessen, and R. Mathar, "Theory of Maximum Packing and related channel assignment strategies for cellular Radio Networks," *Mathematical Methods of Operations Research*, vol. 48, pp. 1-16, 1998.
10. R. Prakash, N. G. Shivaratri, and M. Singhal, "Distributed Dynamic Channel Allocation for Mobile Computing," in *Proceedings of ACM Symposium on Priciples of Distributed Computing*, Aug. 1995.
11. R. Prakash, N. G. Shivaratri, and M. Singhal, "Distributed Dynamic Fault-Tolerant Channel Allocation for Mobile Computing ," *To appear in IEEE Transactions on Vehicular Technology*.
12. S. K. Das, S. K. Sen, and R. Jayaram, "Dynamic Load Balancing Strategies for Channel Assignment Using Selective Borrowing in Cellular Mobile Environments," in *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 73-84, Nov. 1996.
13. M. R. Meybodi, H. Beigy, and M. Taherkhani, "Cellular Learning Automata and its Applications," *Accepted for Publication in Journal of Sharif*.
14. H. Beigy and M. R. Meybodi, "A Mathematical Framework to Study the Evolution of Cellular Learning Automata," Tech. Rep. TR-CE-2003-004, Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran, 2003.
15. K. S. Narendra and K. S. Thathachar, *Learning Automata: An Introduction.* New York: Printice-Hall, 1989.
16. P. R. Srikantakumar and K. S. Narendra, "A Learning Model for Routing in Telephone Networks," *SIAM Journal of Control and Optimization*, vol. 20, pp. 34-57, Jan. 1982.
17. 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.
18. 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.
19. M. R. Meybodi and H. Beigy, "A Note on Learning Automata Based Schemes for Adaptation of BP Parameters ," *Journal of Neuro Computing*, vol. 48, pp. 957-974, Nov. 2002.
20. M. R. Meybodi and H. Beigy, "New Learning Automata Based Algorithms for Adaptation of Backpropagation Algorithm Parameters," *International Journal of Neural Systems*, vol. 12, pp. 45-68, Feb. 2002.
21. N. H. Packard and S. Wolfram, "Two-Dimensional Cellular Automata," *Journal of Statistical Physics*, vol. 38, pp. 901-946, 1985.
22. H. Beigy and M. R. Meybodi, "Asynchronous Cellular Learning Automata," Tech. Rep. TR-CE-2003-006, Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran, 2003.
23. M. R. Meybodi, H. Beigy, and M. Taherkhani, "Application of Cellular Learning Automata to Image Processing," in *Proceedings of First Conference in Mathematics and Communication, Iranian Telecommunication Research Center, Tehran, Iran*, pp. 23.1-23.10, Oct. 2000.
24. M. R. Meybodi and M. Taherkhani, "Application of Cellular Learning Automata in Modeling of Rumor Diffusion," in *Proceedings of 9th Conference on Electrical Engineering, Power and Water Institute of Technology, Tehran, Iran*, pp. 102-110, May 2001.
25. M. R. Kharazmi and M. R. Meybodi, "Image Restoration Using Cellular Learning Automata," in *Proceedings of the Second Iranian Conf. on Machine Vision, Image Processeing and Applications, Tehran, Iran*, pp. 261-270, 2003.
26. M. R. Meybodi and M. R. Kharazmi, "Application of Cellular Learning Automata to Image Processing," *Journal of Amirkabir, to be published*, 2003.
27. M. R. Meybodi and M. R. Khojasteh, "Application of Cellular Learning Automata in Modelling of Commerce Networks," in *Proceedings of 6th Annual International Computer Society of Iran Computer Conference CSICC-2001, Isfehan, Iran*, pp. 284-295, Feb. 2001.
28. M. R. Meybodi and F. Mehdipour, "VLSI Placement Using Cellular Learning Automata," in *Proceedings of 8th Annual International Computer Society of Iran Computer Conference CSICC-2001, Mashad, Iran*, pp. 195-203, Mar. 2003. (An Extended Version of this Paper will Appear in Journal of Tarbiat Modares, University of Tarbiat Modares, Tehran, Iran).

29. M. A. L. Thathachar and P. S. Sastry, "A New Approach to the Design of Reinforcement Schemes for Learning Automata," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, pp. 168–175, Jan. 1985.
30. M. Agache and B. J. Oommen, "Generalized Pursuit Learning Schemes: New Families of Continuous and Discretized Learning Automata," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, pp. 738–749, Dec. 2002.

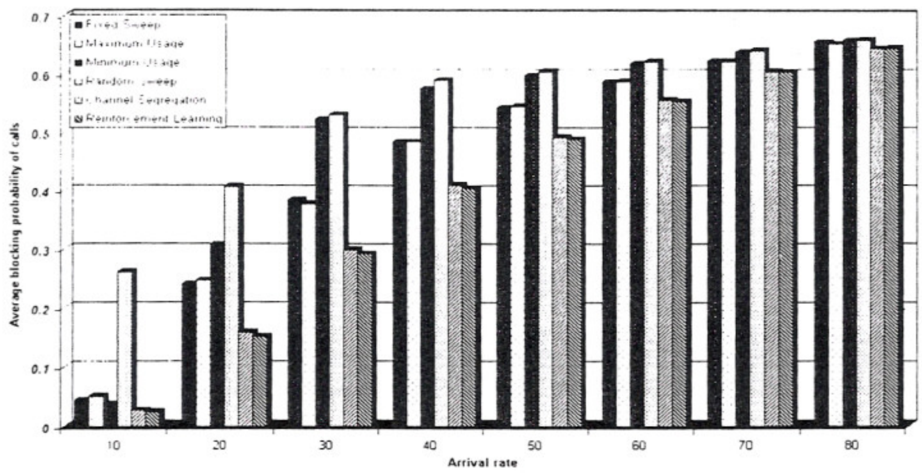

Fig. 1. The average blocking probability of calls.

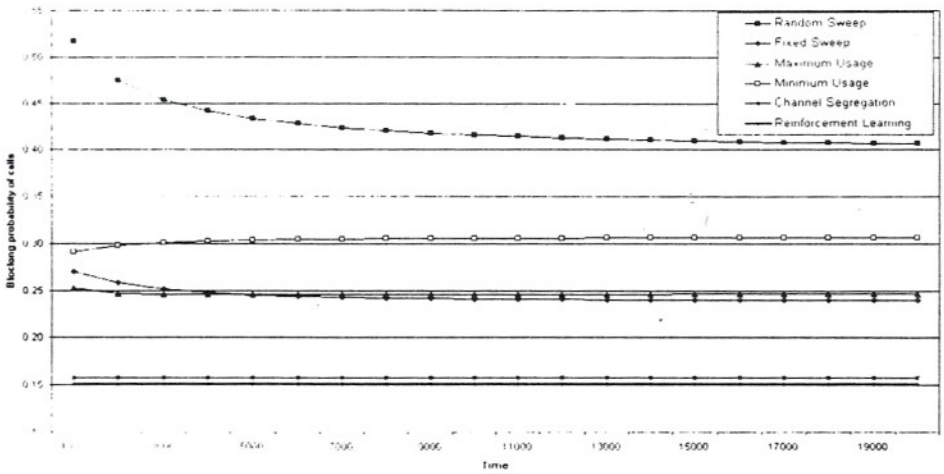

Fig. 2. The evolution of blocking probability of calls for a typical run.

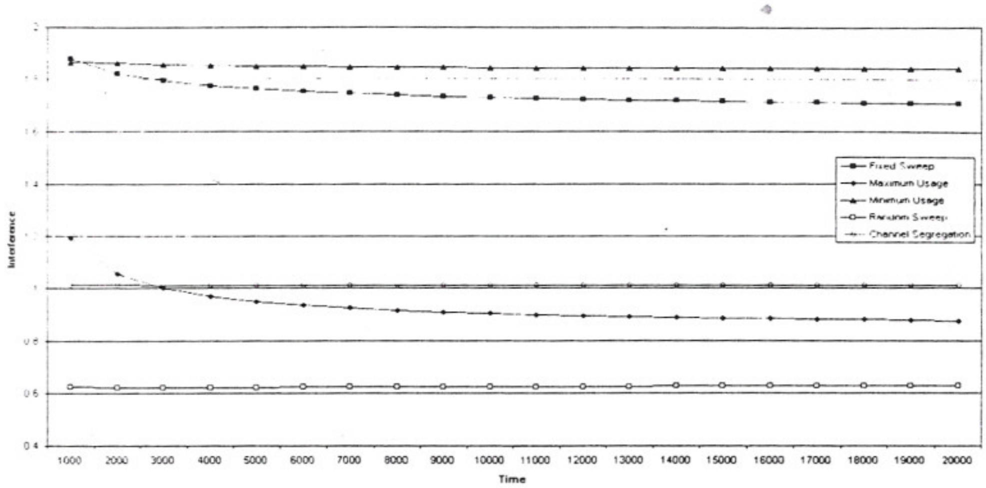

Fig. 3. The evolution of the number of interfering channels.

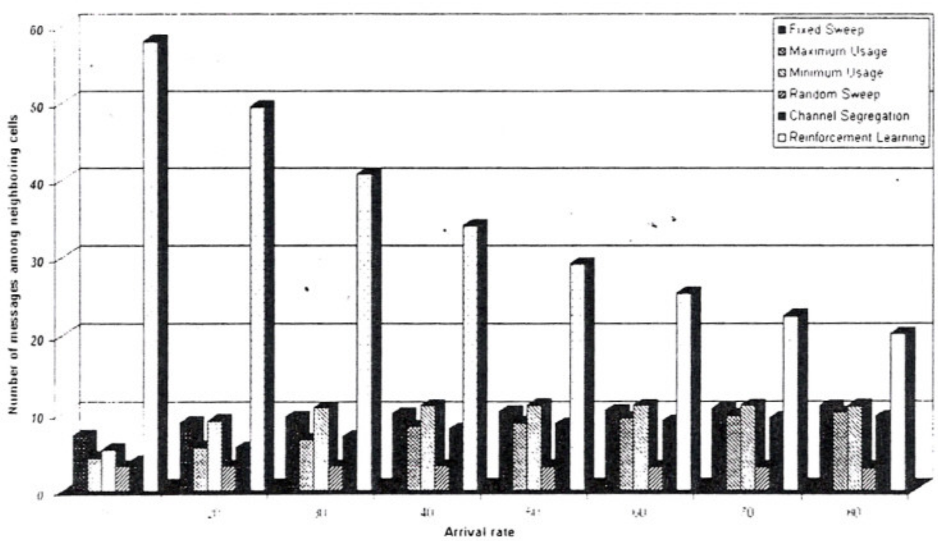

Fig. 4. The number of messages among the cell and its neighboring cells.

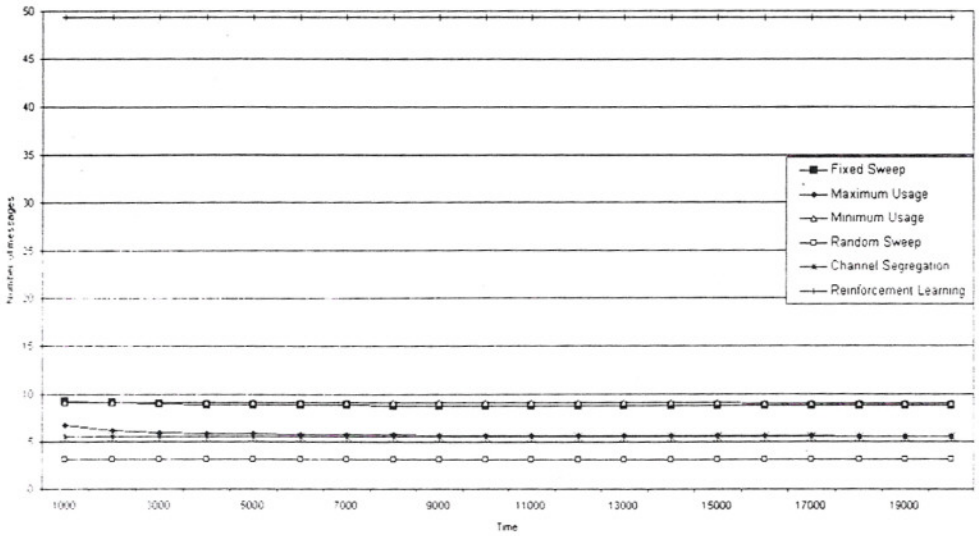

Fig. 5. The changes in the number of messages between transmitted among the cell and its neighboring cells.

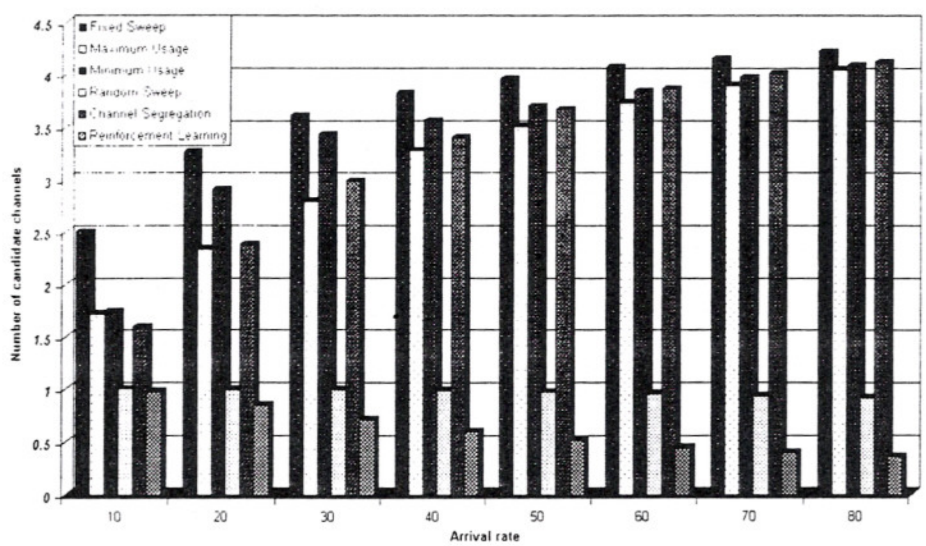

Fig. 6. The number of candidate channels before a channel is assigned to calls.

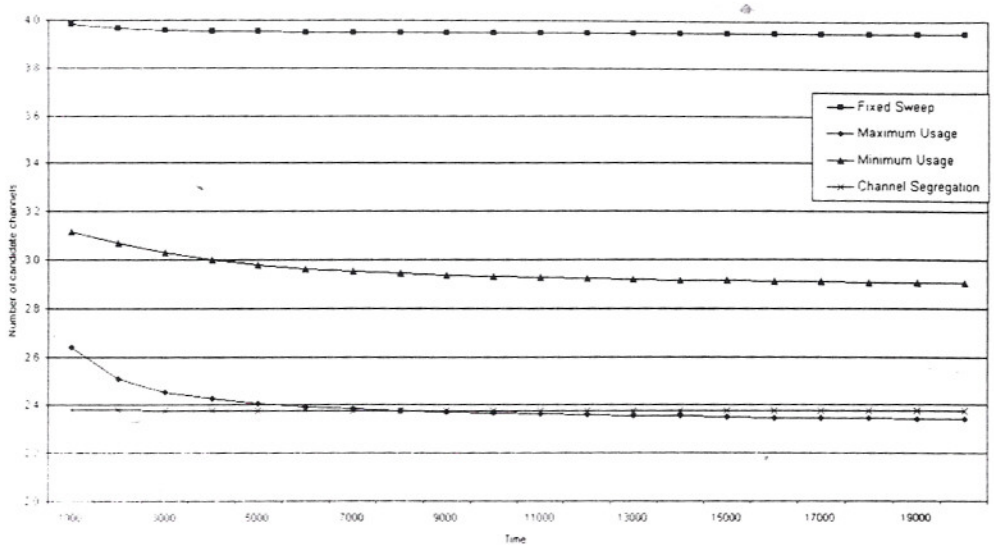

Fig. 7. The changes in the number of candidate channels before a channel is assigned to calls.

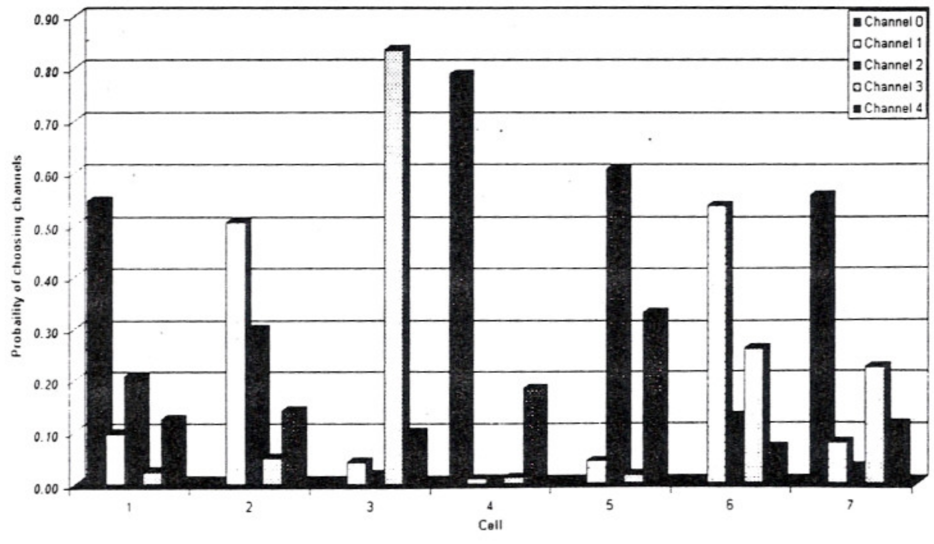

Fig. 8. The probability of assigning different channels to different cells for fixed sweep strategy.

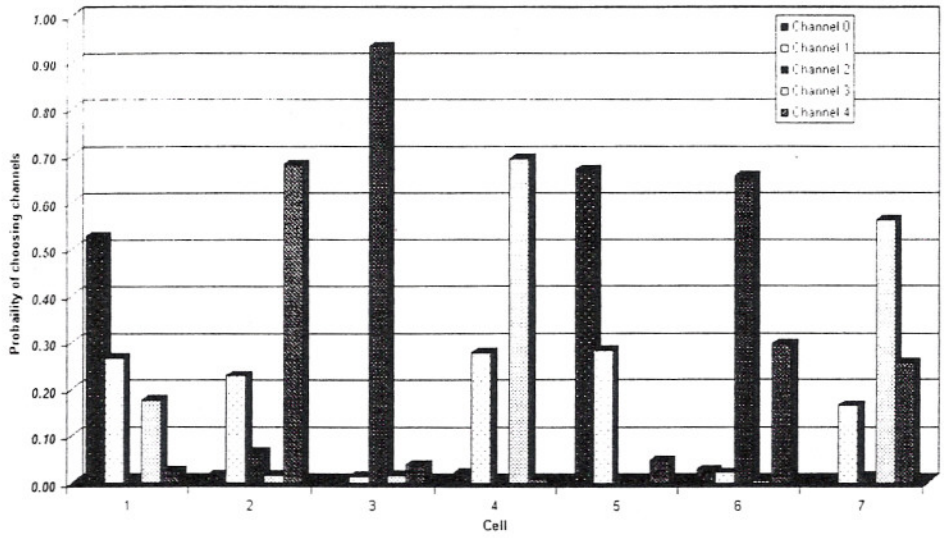

Fig. 9. The probability of assigning different channels to different cells for maximum usage strategy.

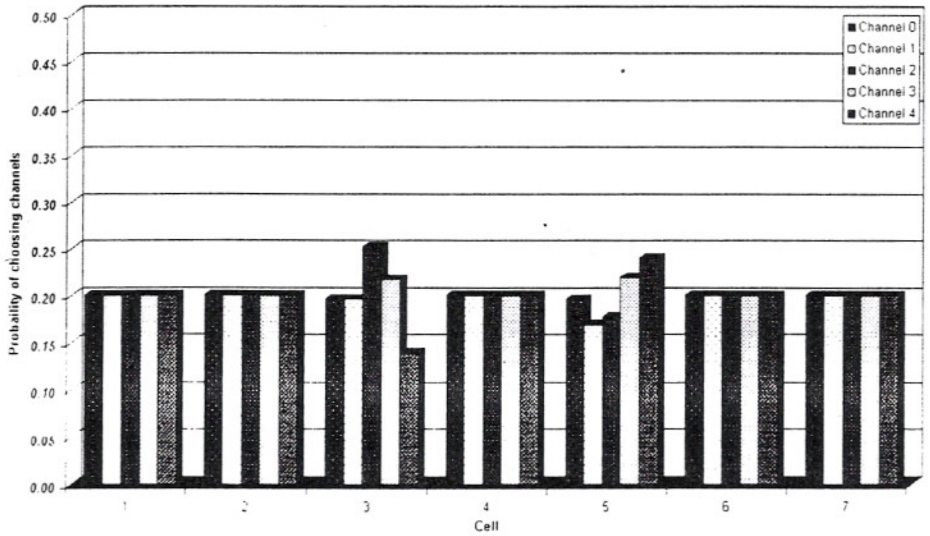

Fig. 10. The probability of assigning different channels to different cells for minimum usage strategy.

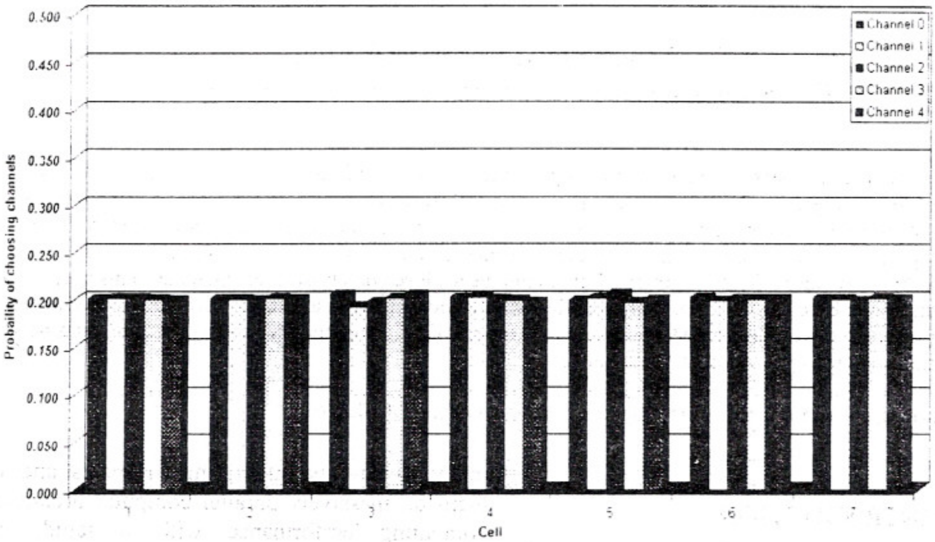

Fig. 11. The probability of assigning different channels to different cells for random sweep strategy.

WITID

Iran Telecom Research Center, Kargar Ave.,
Tehran, Iran. P.O.Box:14155-3961
Tel: +98 21 8005516- 8002455 -8497871- 8497870
Fax: +98 21 8005516
E-mail:witid2004@itrc.ac.ir
Website:www.itrc.ac.ir\witid2004