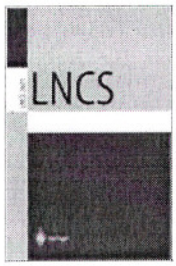
Chapter



Lecture Notes in Computer Science
Publisher: Springer-Verlag Heidelberg
ISSN: 0302-9743
Volume: Volume 2690 / 2003
Date: August 2003
Pages: 119 - 126
Title: Intelligent Data Engineering and Automated Learning
ISBN: 3-540-40550-X

Automated Learning

A Self-Organizing Channel Assignment Algorithm: A Cellular Learning Automata Approach

Hamid Beigy[1] and M.R. Meybodi[1]

(1) Soft Computing Laboratory, Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran,

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 introduce the model of cellular learning automata in which learning automata are used to adjust the state transition probabilities of cellular automata. Then a cellular learning automata based self-organizing channel assignment algorithm is introduced. The simulation results show that the micro-cellular network can self-organize by using simple channel assignment algorithm as the network operates.

A Self-Organizing Channel Assignment Algorithm: A Cellular Learning Automata Approach

Hamid Beigy and Mohammad Reza 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 introduce the model of cellular learning automata in which learning automata are used to adjust the state transition probabilities of cellular automata. Then a cellular learning automata based self-organizing channel assignment algorithm is introduced. The simulation results show that the micro-cellular network can self-organize by using simple 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

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

J. Liu et al. (Eds.): IDEAL 2003, LNCS 2690, pp. 119–126, 2003.

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 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]. 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. In HCA schemes, channels are divided into *fixed* and *dynamic* sets. 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 introduce 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). Then we propose a self-organizing channel assignment algorithm based on the CLA. The proposed algorithm reduces the time needed by the algorithms that use exhaustive search for finding optimal solution of FCA. 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 rest of this 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 presents the proposed algorithm and numerical example, 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* [3]. 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

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

when $\beta(k) = 0$ and

$$p_j(k+1) = \begin{cases} p_j(k) \times (1-b) & \text{if} \quad i = j \\ \frac{b}{r-1} + p_j(k)(1-b) & \text{if} \quad i \neq j \end{cases} \tag{2}$$

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 (1) and (2) is called *linear reward penalty*(L_{R-P}) algorithm and if $b = 0$, then the recurrence equations (1) and (2) is called *linear reward inaction*(L_{R-I}) algorithm. LA have been used successfully in many applications such as telephone and data network routing [4], solving NP-Complete problems [5], capacity assignment [6] and neural network engineering [7,8] 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 [9]. 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.

A *cellular learning automata* (CLA) is a CA in which an LA (or multiple LA) is assigned to its every cell. The LA residing in a particular cell determines its state on the basis of its action probability vector. Like CA, there is a rule that CLA operate under it. The rule of CLA and state of neighboring cells of any particular cell determine the reinforcement signal to the LA residing in that cell. In CLA, the neighboring cells of any particular cell constitute the environment for that cell. Because the neighboring cells produce the reinforcement signal to the LA residing in that cell. The operation of CLA could be described as follows: At the first step, the internal state of every cell is specified. The internal state of every cell is determined on the basis of action probability vectors of LA residing in that cell. The initial value of the action probability vectors may be chosen on the basis of past experience or at random. In the second step, the rule of CLA determines the reinforcement signal for each LA. Finally, each LA updates its action probability vector on the basis of supplied reinforcement signal and the action chosen by the cell. This process continues until the desired state reached. 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 [10], an asynchronous CLA is proposed and used as an adaptive controller. In his model, the state space of the environment(system under control) is uniformly discretized into cells, and each cell contain a number of LA. The actions of each LA corresponds to discritized values of the corresponding control variable. Based on the state of system (S_0) one cell in CLA is activated. Every LA of the activated cell chooses an action based on its action probability vector. These actions are applied to the system and the system changes its state from S_0 to S_1. The environment then passes a reinforcement signal to the LA of the activated cell. Depending on the reinforcement signal, LA in activated cell and its neighboring cells update their action probability vectors. This process continues until termination state is reached. In [11], a model of synchronous CLA has been proposed in which each cell can hold one LA. This model of CLA have been used in several applications such as image processing and rumor diffusion, to mention a few. In what follows, we extend the idea given in [11].

A CLA, as shown in figure 1, consists of following major components: neighboring cells, local rule, and a set of LA. The operation of CLA can be described as follows: Without loss of generality assume that an automaton is assigned to every cell. At instant n, the LA A associated to a particular cell u selects one of its actions, say $\alpha(n)$ based on its action probability vector. The reinforcement signal $\beta(n)$ is produced by a local rule R from the state of neighborhood cells.

Every LA updates its action probability vector based on the reinforcement signal $\beta(n)$, and selected action $\alpha(n)$. This process continues until the average received penalty is minimized.

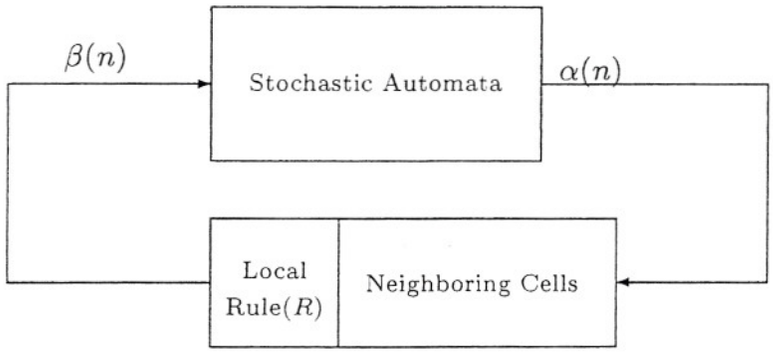


Fig. 1. Block diagram of one cell of CLA

Definition 1. A CLA mathematically can be described by a tuple $\{d, \Phi, N, A, R\}$, where

1. d is a positive integer indicating the dimension of CLA. When d equals to one (two) the CLA is referred to as *one-(two-)dimensional CLA*.
2. $\Phi = \{\phi_1, \cdots, \phi_k\}$ is the set of internal state, where $\phi_i \in \{0, 1, \ldots, L-1\}$ and L is an integer.
3. N is a neighborhood vector $N = (\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_m)$ of different elements in Z^d, which determines the relative position of neighboring sites from any given site u in the lattice Z^d. The neighbors of cell u are cells $\{u + \bar{x}_i | i = 1, 2. \cdots, m\}$. There is a neighborhood function $\bar{N}(u) = (u + \bar{x}_1, u + \bar{x}_2, \cdots, u + \bar{x}_m)$ that maps a site u to set of its neighbors. For the sake of simplicity, we assume that the first element of neighborhood vector (i.e. $\bar{x}_1$) is equals to d-tuple $(0, 0, \cdots, 0)$. $\bar{N}(u)$ satisfies in the two following conditions:
   - $u \in \bar{N}(u)$ for all $u \in Z^d$.
   - $u_1 \in \bar{N}(u_2) \Longleftrightarrow u_2 \in \bar{N}(u_1)$ for all $u_1, u_2 \in Z^d$.
4. A is a set of LA associated to a particular cell.
5. $R : \Phi^m \to \beta$ is the local rule of the CLA, which produces the reinforcement signal $\beta(n)$ from the state of neighboring cells.

4 The Proposed Channel Assignment Algorithm

In this section, we introduce a self-organizing channel assignment algorithm based on the CLA introduced in the previous section. In this algorithm, at first a CLA is build from the cellular network with neighborhood function $\bar{N}(.)$ and then channels are assigned to the cells according to the algorithm given in figure 2. The neighborhood function $\bar{N}(u)$ represents the set of cells in the reuse

distance of cell u; i.e. the set of cells in a cluster with center u. In the proposed algorithm, a set of LA $\underline{A} = \{A_1, \ldots, A_\sigma\}$ each with two actions are associated to each cell, where σ denotes the number of channels assigned to the network. In this algorithm, the automaton A_i is used for allocation of channel i. The action α_1 of automaton A_i represents that the channel i is a candidate for assignment to that cell and the action α_2 of automaton A_i represents that the channel i isn't a candidate for assignment to that cell. The state of each cell, Φ, is a set with σ elements $\Phi = \{\phi_1, \ldots, \phi_\sigma\}$, where the element $\phi_i \in \{0, 1\}$ (for $i = 1, \ldots, \sigma$) represents whether the suggested allocation of channel i for that cell is feasible or not. When $\phi_i = 0$ ($\phi_i = 1$), it means that the suggested assignment of channel i is feasible (infeasible) for that cell. The local rule of CLA determines whether the given channel is assigned to any cell in the cluster or not. For example the used rule is: if channel i is assigned to any cell in the cluster and it isn't assigned to cell u, then the result of the rule is 0; otherwise the result of the rule is 1.

```
Algorithm
1.    Build a CLA and initialize it.
2.    for k = 1 to N do in parallel
3.       for i = 1 to σ do in parallel
4.          Select action of automaton A_i and denote it as α^i
5.          if α^i = α_1 then
6.             if channel i is used in the cluster then
7.                Penalize action α_1.
8.             else
9.                Reward action α_1.
10.            end if
11.         else
12.            if channel i is used in the cluster then
13.               Reward action α_2.
14.            else
15.               Penalize action α_2.
16.            end if
17.         end if
18.      end for
19.   end for
end Algorithm
```

Fig. 2. The CLA based self-organizing channel assignment algorithm.

The description of the proposed algorithm for M cells one dimensional cellular network is shown algorithmically in figure 2, which can be described as follows. At first a CLA is build and initialized based on the topology of the cellular network. Then each cell do the following operation for each automaton A_i (for $i = 1, \ldots, \sigma$) associated to it. Automaton A_i (for $i = 1, \ldots, \sigma$) selects its action and then updates its action probability vector based on the result of local

channel assignment algorithm is self-organizing and converges to the optimal assignment. For more simulation results, the reader may refer to [12].

Figure 3 shows the convergence behavior of a typical LA in a cell for a typical run for different values of learning parameter, a. This figure shows that increasing the learning parameter,a, increases the speed of convergence of CLA.

6 Conclusions

In this paper, cellular learning automata is introduced. The cellular learning automata is a model in which learning automata are used to adjust the state transition probabilities of cellular automata. Then as application of cellular learning automata, a self-organizing channel assignment algorithm based on cellular learning automata is given. In order to show the power of the proposed method, the computer simulations are conducted.

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. K. S. Narendra and K. S. Thathachar, *Learning Automata: An Introduction.* New York: Printice-Hall, 1989.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. N. H. Packard and S. Wolfram, "Two-Dimensional Cellular Automata," *Journal of Statistical Physics*, vol. 38, pp. 901–946, 1985.
10. K. Krishna, "Cellular Learning Automata: A Stochastic Model for Adaptive Controllers," Master's thesis, Department of Electrical Engineering, Indian Institue of Science, Banglore, India, June 1993.
11. M. R. Meybodi, H. Beigy, and M. Taherkhani, "Cellular Learning Automata and its Applications," *Accepted for Publication in Journal of Sharif.*
12. H. Beigy and M. R. Meybodi, "Cellular Learning Automata Based Self-Organizing Channel Assignment Algorithms," Tech. Rep. TR-CE-2002-008, Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran, 2002.

Jiming Liu
Yiuming Cheung
Hujun Yin (Eds.)

LNCS 2690

Intelligent Data Engineering and Automated Learning

4th International Conference, IDEAL 2003
Hong Kong, China, March 2003
Revised Papers

Springer