Fusion of Learning Automata to Optimize Multi-constraint Problem

Sara Motamed*
Department of Computer Engineering, Fuman Branch, Islamic Azad University, Fuman, Iran
Samotamed@yahoo.com

Ali Ahmadi
Department of Computer Engineering, K.N. Toosi University of Technology, Tehran, Iran
ahmadi@eetd.kntu.ac.ir

Received: 23/Sep/2013 Revised: 11/Oct/2014 Accepted: 12/Nov/2014

Abstract

This paper aims to introduce an effective classification method of learning for partitioning the data in statistical spaces. The work is based on using multi-constraint partitioning on the stochastic learning automata. Stochastic learning automata with fixed or variable structures are a reinforcement learning method. Having no information about optimized operation, such models try to find an answer to a problem. Converging speed in such algorithms in solving different problems and their route to the answer is so that they produce a proper condition if the answer is obtained. However, despite all tricks to prevent the algorithm involvement with local optimal, the algorithms do not perform well for problems with a lot of spread local optimal points and give no good answer. In this paper, the fusion of stochastic learning automata algorithms has been used to solve given problems and provide a centralized control mechanism. Looking at the results, is found that the recommended algorithm for partitioning constraints and finding optimization problems are suitable in terms of time and speed, and given a large number of samples, yield a learning rate of 97.92%. In addition, the test results clearly indicate increased accuracy and significant efficiency of recommended systems compared with single model systems based on different methods of learning automata.

Keywords: Stochastic Automata with Fixed and Variable Structures; Discrete Generalized Pursuit Automata; Fusion Method; Parallel Processing**.**

1. Introduction

Learning automata is one of the important models of learning and the goal of this method is to determine the optimal actions in unpredictable situation. The most important application of learning automata is to estimate parameters [3] while it is used in identification of pattern and play theory [4, 5, 6, 7]. All researches up to late 1980 have been reviewed and discussed at this book (Narendra and Thathachar) [1, 10]. Numerous examples and applications of learning automata have been presented in [9]. Another important feature of learner systems is their potential to improve efficiency with time. According to [2], the objective of the learner system is to improve a function which is not totally identified. Therefore, an approach to the problem is to decrease the objectives of the learner system which is defined on a set of parameters which aim at finding the optimum parameters set; reinforcement learning is a common method in finding the optimum objective. Main benefit of reinforcement learning compared with other learner methods are that it requires no information from the environment (than reinforcing signal) [1]. Stochastic learning automata are one of the reinforcement learning models which try to increase their efficiency and are divided into fixed structure stochastic automata (FSSA) and variable structure stochastic automata (VSSA) [8]. Alonso and Mondragon introduced a framework of limitations according to Markov decision making and optimization methods [16]. Similarly [17], reinforcement learning algorithms have been used for multi-factor systems and elements. In the later section, the paper presents an optimum method by the use of a learning method by using constraints based on Markov hypothesis. The reinforcement learning is defined in an environment using Markov model, by multi-objective functions and long term rewarding [18]. To prove their claim they applied this method to statistical plays with reward average limitations and obtained significant results. Also they explained the limited reinforcement learning method by stochastic and true rewards with a new view for optimization problems. The objective of this paper is to analyze the object by parallel computations and learning automata by heuristic algorithms. Cardei considered a special algorithm of learning automata which is called pursuit algorithm [7]. This algorithm pursues the current optimal action and if this action is not the one with the minimum penalty probability, this algorithm pursues a wrong action. This algorithm presents with two models of continuous and discrete. The continuous pursuit algorithm that used a reward and penalty learning paradigm, denoted CP_{RP}. Later [19] introduced the first discretized

* Corresponding Author

Pursuit estimator algorithm by presenting a discretized version, denoted DP_{RI} that uses a Reward-Inaction learning paradigm. The discretized generalized pursuit algorithm (DGPA) is another algorithm that generalizes the concepts of the Pursuit algorithm. In this algorithm all the actions have higher estimates than the current chosen action. This paper is used this DGPA to predict the best actions in unstable situations. In section 2, an effective model of partitioning is introduced by multi-constrained problems and also in sections 3 reviews the stochastic learning automata method. In the next parts of this section, fixed and variable structures and discrete generalized pursuit automata (DGPA) are considered. The fusion of DGPA is defined in section 4, to remove the problem of convergence in a very big space as a recommended method. Experimental results and conclusion are given in sections 5 and 6.

2. Partitioning by Multi-constraint

The main goal of this algorithm is to allocate a set of elements to elected classes and similar groups which are hopefully placed in the same class. This paper introduced an optimal model of classification in special situations. Also this paper is explained the proper model of partitioning to solve the multi constraint problems. Object allocation is a problem of partitioning a set of P with |P| elements in N classes. These elements have a certain capacity of limitations in each class [12]. This means that each class has limited capacity considering constraints for connecting joints. In the problem, it is supposed the relationship between P elements is weighted in a certain way and there is no connection of the object with itself. The aim in placing the processes on nodes is to find the shortest completion time in parallel applications. It is obvious that if ($|P| \leq |N|$), each node would host in more than one process. Partitioning the process set would be possible where ($|P|>|N|$) and is grown in a combinatory manner and computations of processes connection are defined by Narendra and Thathachar [12]. To express limitations and relations this part should apply conventions for external and internal relations. $X_{i,n}$ is a typical alternative of $X_{i,n} \in \{0,1\}$ and its P_i process is allocated to N_n node so the quantity is one, otherwise it is zero. It assumed that the external relation is formed from a node with the supposition that P_i has been allocated to this certain node [12]. Then $\sum_{j=1}^{|P|} (1 - X_{i,n}) w_{i,j}$ the external connection of P_i processes with all other processes of P_j would be formed. If each P_j is allocated to N_n node, then $X_{j,n} = 1$ and this process has no participation with the above sum. To find the external relation of the node, the sum of similarities should be added and then multiplied and rearranged. In the external relation limit calculated by Eq. 1 [12]:

$$\sum_{i=1}^{|P|}\sum_{j=1}^{|P|}(X_{i,n} - X_{j,n}X_{j,n})W_{i,j} \leq 1 \quad n = 1,\ldots,|N| \qquad (1)$$

The above equation shows the set of processes and their connections from a subset to another. The only guiding quantity is a $w_{i,j}$ relation which indicates the P_i process on the node to P_j distant process which is not on the node. Therefore, the internal connection may be obtained by a similar formula, but by adding the connection for going from the distant process to the process on $w_{j,i}$, the result changes to Eq. 2 [12]:

$$\sum_{i=1}^{|P|}\sum_{j=1}^{|P|}(X_{i,n} - X_{j,n}X_{j,n})W_{j,i} \leq 1 \quad n = 1,\ldots,|N| \qquad (2)$$

Set of constraints 3 limits total computation time for the processes allocated to each node with the node normalized capacity. Set of constraints 4 assures that a process only be allocated to a single node [12].

$$\sum_{i=1}^{|P|} X_{i,n}T_i \leq 1 \qquad n = 1,\ldots,|N| \qquad (3)$$

$$\sum_{i=1}^{|N|} X_{i,n} \leq 1 \qquad n = 1,\ldots,|P| \qquad (4)$$

Along a cycle, automaton picks a behavior and according to performance receives a response from the environment. The response may be a penalty or award [10]. Automaton obtains this response and knowledge form former behaviors and so selects its next move. The goal of the learning automaton is an optimum measure beyond the set of permitted behaviors. Automaton adjusts itself with the environment through learning optimum operation selection. Modeled learning sample is found in systems with insufficient knowledge about the environment and their startup by learning automaton applications. In FSSA, there is the property that output and transfer functions are not changing with time. The problem is based on the fact that static map of a subclass is obtained from learning automaton solutions and is used for solving object partitioning problems. For pairing and computation of w_i external connection for all processes, the node is selected with most violation from constraint 3. For instance P_A, which has been randomly selected among the processes on the node according to experimental distribution form their τ_i average weights, is allocated to this node. Then P_A process randomly selects another P_B process according to W_A distribution probability. A set of $\langle P_A, P_B \rangle$ processes are considered as a pair where pairing is said to be successful. If two processes belong to the same node, then these two processes receive a reward, unless the pairs are unsuccessful and are both penalized [12].

3. Stochastic Learning Automata

Learning automata are one of the important models of learning on unknown random environment. Each automaton has a finite set of input and certain probability of reward or penalty from environment. So the most important application of learning automata is to estimate parameters, while it is used in classification subjects, play theory and identification of pattern [10]. Learning automata are classified into the two groups of fixed and variable structure learning automata. In stochastic automata one action is selected randomly. Then the response of environment to this action is calculated by probabilities. Now the new action is selected again and according to updated action probabilities, and the processors is repeated.

A stochastic automaton is defined by the sextuple {x, φ, α, p, A, G} and each parameter is:

- X is a input set.
- $\{\varphi_1, \varphi_2, \ldots, \varphi_r\}$ is a finite set of internal state.
- $\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is a finite set of output or response set.
- $p = \{p1, p2, \ldots, pr\}$ represents the action probability set.
- A is an algorithm which generates $p(n + 1)$ from $p(n)$.
- G is the output function.

For execution of training, the feedback signal from the environment, which triggers the updating of the action probabilities by the automaton, can be given by specifying an appropriate "error" function.

3-1- Fixed structure stochastic automata (FSSA)

Along a cycle, automata would pick a behavior and according to performance receives a response from the environment in a manner that the response may be a penalty or reward [10]. Automata obtain this response and knowledge from former behavior for expression of the next measure. The goal of the learning automaton is an optimum measure beyond the set of permitted behaviors. Automaton adjusts itself with the environment through learning optimum operation selection. Modeled learning sample by learning automaton applications is found in systems with insufficient knowledge about the environment and startup. In FSSA there is the property that output and transfer functions would change with time. The problem is that stable mapping a subclass of the learning automaton solutions and is used for solving the object partitioning problems. For pairing and calculation of external connection of w_i for all processes, first a node with most violation from region 3 is selected. A process allocated to this node, for instance P_A which has randomly selected among the processes on the node according to experimental distribution form their τ_i average weights. Then P_A process would randomly select another P_B process according to the possibility of W_A distribution. Set of $\langle \mathbf{P_A}, \mathbf{P_B} \rangle$ process is considered as a pair and called successful pairing. If the two processes belong the same node then these two processes would receive a reward unless the pairs are unsuccessful and both penalized [12].

3-2- Variable structure stochastic automata (VSSA)

VSSA is a replacement for FSSA and its transfer and output matrixes are changed in time [10, 13, 14 and 15]. They are defined as a possible operation vector P(K) where $P_i(K)$ is the possibility of i^{th} operation in the set of A operation, which is selected in K time from available $|A|$ operations. As $\sum_i P_i(K) = 1$, for all Ks, updating the law for possibility vector would be continuous or discontinuous. The quickest learning automaton convergence belongs to the VSSA family. Adaptability of the family with automata to solve the specific problems may hopefully improve the speed of obtaining a solution. Thathachar and Sastry [16] introduced what is known as estimating algorithms. The main feature of these algorithms is that they maintain estimates of possible rewards for each operation and use them in possibility updating equations. In the first stage of functional, automaton selects an operation and the environment produces a response to this operation. According to the estimating algorithm answer, estimation of possible rewards for that operation is updated. Changes in possibility vector is P(K) operation based on $\hat{d}(k)$ and the vector being executed is estimated from the rewards possibility which is updated according to feedback from the environment. A model of random automata learning is DGPA, referred to in section 3-3.

3-3- Discrete generalized pursuit automata (DGPA)

Pursuit algorithm is a special type of estimator algorithm and it converged in statistic space rapidly. This model of pursuit algorithm works based on reward inaction learning paradigm and it updates the action of probability vectors if the environment rewards the chosen action. One of the problems in standard learning algorithms is their relatively slow convergence in selection of optimum operation in static environments for the removal of which various solutions have been introduced. One of the first solutions is disconnection of possibility space [10] in which possibility of operation selection can pick only certain quantities in the range of [0, 1]. On this basis, most standard algorithms are discrete. One of the existing problems in new models is premature convergence of learning algorithms with non-optimized operations. The root of these problems is in limiting their probability space. Thatachar and Sastry opened a new rout in their research by introducing estimator algorithms in line with their endeavors to improve learning algorithms convergence. The most important feature of such algorithms is in maintaining a continuous estimation from the possibility of receiving the reward for each operation and using it in updating automata equations. In other words, in the first stage of operation cycle, automaton selects an operation and then environment produces a response for it. According to this response, the estimating algorithm updates the reward possibility estimation for that operation. Pursuit automata are a group of estimating algorithms. As it is clear from its name, such algorithms are identified based on the fact that operation possibility vector encourages the operation which is currently considered as the best operation according to estimations. This is made with increasing the possibility of an operation wherein the reward possibility estimation is highest than other operations. Pursuit automata are divided in two continuous and discrete classes [11, 12]. The difference between these two algorithms is in updating the law for operation possibilities. According to the results, partitioning on the basis of pursuit automata with variable structure is effective for only small classes [10]. Therefore, the above problems cannot possibly be used as an ideal method for solving such problems. Our goal in this paper was to solve these problems by using the fusion of DGPAs. This model is called fusion of DGPAs (FDGPA) and is defined in section 4 to solve multi-constraint problems.

4. Fusion of DGPA (FDGPA)

An important problem in automata learning is their learning rate or their convergence speed equivalence. This is highly important for learning as it mostly changes slowly in the environment and learning processes should be completed in the environment before main changes appear, unless learning is ineffective. One idea is to use parallel operations because this method is considered as increased convergence speed. But if we have parallel learning automaton operations, each automaton would produce its operation and according to the signal, the environment would produce reinforcement in time. As there are multi responses, convergence speed would be raised. FDGPA is considered to be parallel instead of single learning automaton. Fig. 1 indicates our recommended algorithm. In Fig. 1, partitioning algorithm is applied to the input data for allocation of |P| elements to |N| classes with the constraints and it goes on to the next step. In that stage, data are divided to n classes so that they are given to processors for DGPAs which are applied to a parallel manner. As seen in Fig. 1, n parameters are the total number of operations in n={1,…,n} and P is the number of processors while in this case, each processor is an automaton. In our parallel model indexes are fixed and Current Primes are changing each moment while input vectors are divided by each processor in a certain number. Processors output is the automaton input to which DGPA learning is applied and the output for this step is the environment input while a cycle is repeated for the permitted number. In fusion section the set of operations is given by α and operation possibility vector p(K) is common by all n automata.

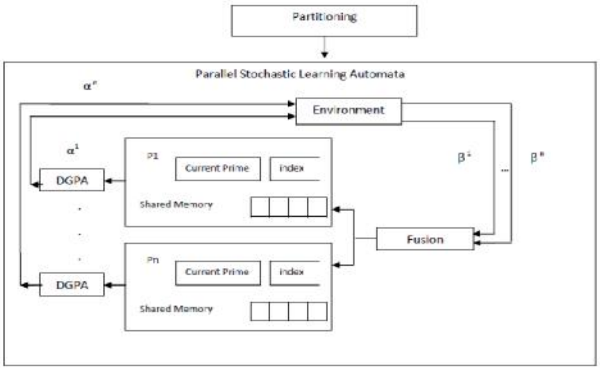

Fig.1. FDGPA model

For instance, each DGPA is based on the common operation possibility vector which is selected from ($\alpha^i(k)$). This vector obtains its own reinforcement signal (($\beta^i(k)$). Possibility vector for updated common operation is obtained based on all selected operations and obtained reinforcement learning. Fusion method uses all information obtained by the possibility of updating. The supposition is that $\beta^j(k) \in [0,1]$ is obtained for all i, k and the fusion vector is computed by the below equations [11]. To calculate the total response to α^j in Eq. 5:

$$q_i(k) = \sum_{j=1}^{n} {}^{j}(k) I\{\sim^{j}(k) = \sim_i\} \qquad (5)$$

In Eq. 6 is calculated the response to obtain the sum total:

$$q(k) = \sum_{j=1}^{n} {}^{j}(k) = \sum_{j=1}^{n} q_i(k) \qquad (6)$$

Output from fusion step is obtained from Eq.7:

$$p_i(k+1) = p_i(k) + \tilde{\lambda} \left(q_i(k) - q(k)p_i(k) \right)$$
$$i = 1, \dots, r \qquad (7)$$

Where $\lambda \epsilon (0, 1]$ is learning parameter and $\tilde{\lambda} = \lambda / n$ is its normalized value. According to computations in eq.7, p(k) has been updated only once and does not require to be updated in each learning. Updated value p(k+1) is shared by all automata for selection of the next operation. This algorithm is fit for n sizes and for speeding up the rate of convergence.

5. Experimental Results

This paper studies the problem of partitioning a P set of **|P|** elements (or objectives) in **|N|** multiple discordant levels with the objective of having similar cluster elements. In this same level, objects can be connected in a multi-constraint (possible or discordant) method. This method has been formed by static mapping from allocation of a set of processes in parallel application to the set of computing nodes. In order to compare our fusion method with FSSA, multi-constraint partitioning is run on processes. We first presented a FSSA algorithm to solve multi-constraint problems. Solution is applicable but requires some centralized collaborations. A solution should be free from a centralized control mechanism. For this reason VSSA was presented. But this method is not ideal for such problems for sets with scattered data and is not applicable for big sets. There are two different models of pursuit learning algorithm which are called discrete and continuous. The differences between two models of pursuit algorithm are the updating rules for the actions probabilities.

For this reason is used the DGPAs fusion method. The software used for simulating the data is Matlab. Table 1 and 2 indicate learning rate of our recommended method in 200 iterations. Results indicate that fusion of stochastic learning automata algorithms give better results than single method stochastic learning automata algorithms. Plus by increasing the number of processors the rate of learning has increased.

Table 1: The results of single FSSA and fused FSSAs.

Number of nodes	***Number of processes***	***Fusion of DGPAs with two parallel processors***	***Fusion of DGPAs with three parallel processors***
2	4 6 8	100 99.90 99.12	100 100 99.50
4	8 12 16	98.66 98.51 98.43	99.14 99.10 99.03
6	12 18 24	98.40 98.37 98.05	99.06 98.48 98.12
16	64	97.03	97.92

Table1 indicates learning rate in single FSSA and fusion FSSAs and it can be understood that fusion of FSSAs give better result than single FSSA.

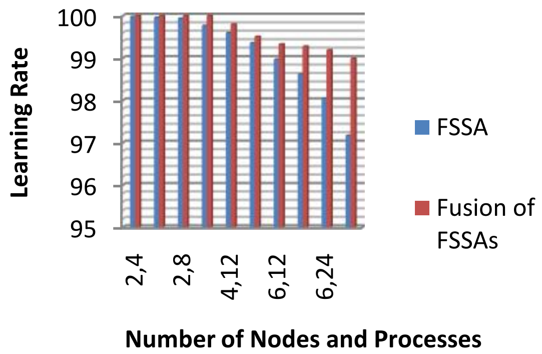

Fig.2. Comparison of single FSSA and fused FSSAs results on different nodes and processes

FSSAs. This figure indicates that fusion of FSSAs give better result than single FSSA.

Table 2: The results of fused DGPAs with two and three parallel processors.

Number of nodes	*Number of processes*	*FSSA*	*Fusion of FSSAs*
2	4 6 8	99.98 99.96 99.94	100 100 100
4	8 12 16	99.77 99.60 99.36	100 99.80 99.51
6	12 18 24	98.97 98.62 98.05	99.33 99.28 99.19
16	64	97.17	98.99

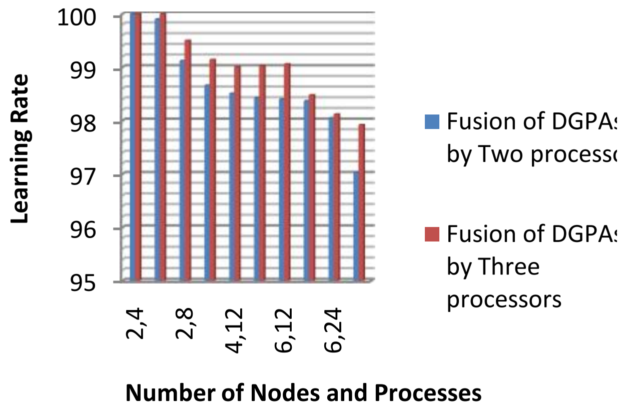

Fig.3. Comparison of the results from fused DGPAs with different processors

Fig. 3. Indicate the learning rate in fused algorithms of DGPAs with two and three parallel processors. As deduced, any more the number of processors, learning rate would be higher.

Fig. 4, 5 and 6 indicate partitioning of P elements in N class by applying proposed algorithm.

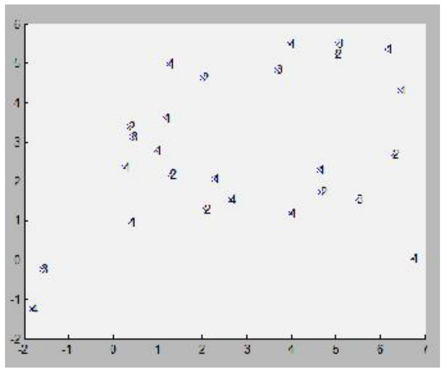

Fig.4. Partitioning the data with FDGPA for 4 nodes and 8 processors

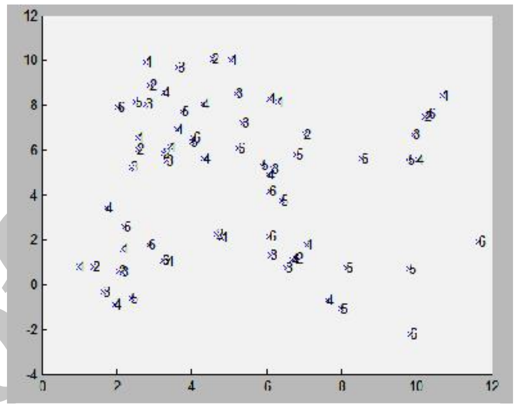

Fig.5. Partitioning the data with FDGPA for 6 nodes and 12 processors

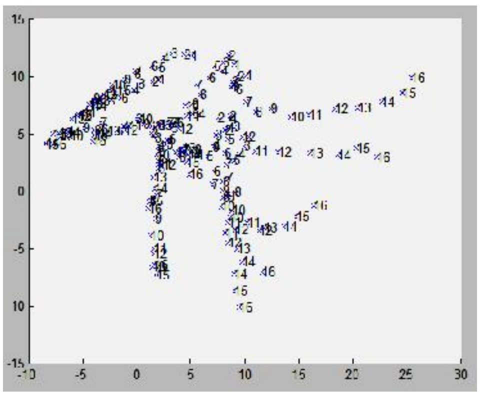

Fig.6. Partitioning the data with FDGPA for 16 nodes and 64 processors

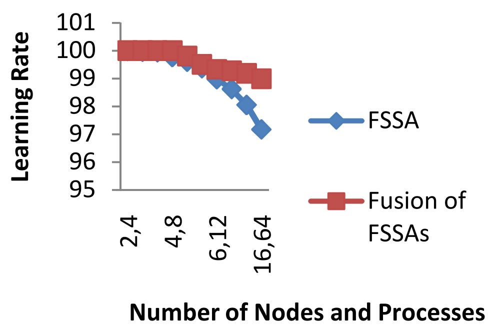

Fig.7. The results of fused and single model algorithms

This section presents a comparison of the performance of the different model of learning automata for using the partitioning the data by multi constraint rules. Looking at Fig. 7, it found that the fused of automata's algorithms have a higher learning rate than single model automata's algorithm. Furthermore, by adding the number of processors in FDGPA, our model's performance rises.

6. Conclusion

Over the last two decades, many new families of learning automata have emerged, with the class of estimator algorithms being among the fastest ones. All of those algorithms are divided into the two main groups of variable stochastic automata and fixed structure automata. One of the big advantages of variable structure automata has been the efficiency of this model in dynamic environment. So this paper is used a special model of variable structure automata is called DGPA algorithm. In contrast to the GPA algorithm, the DGPA always increases the probability of all the actions with higher estimates.

Then this paper is introduced a new model of variable stochastic learning automata by name of FDGPA. This model is used the fusion of DGPA to solve the partitioning of multi-constraint problems.

Review of the results in table 1 and 2 indicate that when there are 16 nodes and 64 processes fused learning rate would be 98.99 in fixed structure automata while the fused learning rate in stochastic learning automata with variable structure would be 97.92% using three processors with 200 iterations.

Overall, the proposed algorithm proved to be faster than the other algorithms in environments with more than two actions. When convergence speed is very low in stochastic automata learning algorithms with fixed and variable structures, and the number of samples increases, stochastic automata learning algorithm with variable structure would not be efficient.

Therefore, it can be claimed that the fused model outperforms the single model. Also, the number of processors increases in our parallel model as the learning rate climes. This method can be used in structural identification of pattern recognitions as a powerful classifier in further studies.

References

[1] K. S. Narendra, and M. A. L. Thathachar," Learning automata: An introduction", Prentice Hall, 1989.

[2] K.S. Narendra, and M. A. L Thathachar, "Learning automata a survey", IEEE Transactions on Systems, Man and Cybernetics, vol. 4, no. 4, 1974.

[3] E. Mance, and S. H. Stephanie, "Reinforcement learning: A tutorial", Wright Laboratory, 1996.

[4] M. R. Meybodi, and M. Taherkhani, "Application of Cellular Learning Automata to Modeling of Rumor Diffusion", Proceedings of 9th Conference on Electrical Engineering, 2001, pp. 102-110.

[5] H. Beigy, and M. R. Meybodi, "A Mathematical Framework for Cellular Learning Automata", Advances on Complex Systems, Vol. 7, No. 3, 2004, pp. 1-25.

[6] V. Raghunathan, C. Schurgers, V. Park, and M. B. Srivastava, "Energy-Aware Wireless Microsensor Networks", IEEE Signal Processing Magazine, Vol 19, 2002, pp 40-50.

[7] M. Cardei, and J. Wu, "Coverage inWireless Sensor Networks", Department of Computer Science and Engineering Florida Atlantic University Boca Raton, FL 33431, 2004.

[8] B.J. Oommen, and M. Agache, "A Comparison of Continuos and Discretized Pursuit Learning Scheme", IEEE, Carleton University, Ottawa, 1988.

[9] J. Winter, Y. Xu, and W. C. Lee, "Energy Efficient Processing of K Nearest Neighbor Queries in Locationaware Sensor Networks", Second International Conference on Mobile and Ubiquitous Systems, 2005.

[10] M.A.L. Thatachar, P.S. Sastry, "Varieties of Learning Automata: An Overview", IEEE transaction on system, man, and cybernetics- part B, 2002.

[11] G. Horn, B.J. Oommen, "Solving Multiconstraint Assignment Problems Using Learning Automata", IEEE transaction on system, man, and cybernetics- part B, 2002.

[12] K. S. Narendra, and M. A. L. Thathachar, "Learning Automata", Englewood Cliffs. NJ: Prentice-Hall, 1989.

[13] V. I. Varshavskii, and I. P. Vorontsova, "On the behaviour of stochastic automata with a variable structure", Remote Control, vol. 24, 1963, pp. 327–333.

[14] M. A. L. Thathachar, and P. S. Sastry, "A new approach to designing reinforcement schemes for learning automata", IEEE Trans, vol. SMC-15, no. 1, 1985, pp. 168–175.

[15] N. Abe, P. Melville, C.K. Reddy, C. Pendus , and D.L. Jensen, "Integrationg Data Modeling and Dynamic Optimization Using Constrained Reinforcement Learning", Mathematical Science Department, 2005.

[16] E. Alonso, E. Mondragon, "Associative Reinforcement Learning A Proposal to Build Truly Adaptive Agent and Multi-agent System", International Conference on Agents and Artificial Intelligence, 2013.

[17] E. S. Mannor, N. Shimkin, "A Geometric Approach to Multi-Criterion reinforcement Learning", Journal of Machine Learning Research 5, 2004.

[18] E. Euchibe, K. Doya, "Constrained Reinforcement Learning from Intrinsic and Extrrinsic Rewards". Okinava Institue Science and Technology, Japan, 2009.

[19] B.J. Oommen, and J. K. Lanctôt, "Discretized Pursuit Learning Automata", IEEE Trans. Syst. Man. Cybern., vol. 20, No.4, 1990, pp.931-938.

Sara Motamed received the B.Sc. from Kerman Azad University, Kerman, Iran and M.Sc. degree from Qazvin Azad University, Qazvin, Iran. Since 2012, she started her PhD in the field of artificial intelligence in Science and Research University, Tehran, Iran. She is interested in speech, image recognition and cognitive science. Now, she is lecturer at the faculty of computer engineering in Fuman Azad University, Fuman, Iran.

Ali Ahmadi received his B.Sc. in Electrical Engineering from Amirkabir University, Tehran, Iran and M.Sc. and Ph.D. in Artificial Intelligence and Soft Computing from Osaka Prefecture University, Japan in 2001 and 2004, respectively. He worked as a researcher in Hiroshima University, Japan during 2004–2007. He is currently assistant professor at K.N. Toosi University of Technology. His research interests are interactive learning models, virtual reality and artificial life, semantic data mining and information fusion, image search engines, and Software–hardware system co-design.