The Journal of
Mathematics and Computer Science

Available online at
http://www.TJMCS.com

The Journal of Mathematics and Computer Science Vol. 4 No.2 (2012) 159 - 171

A New Feature Selection Method based on CLA-EC in Face Recognition System

Akhtar hazrati bishak *[1], Karim faez[2], morteza hazrati bishak [3]

[1]Electrical, Computer and IT Engineering Department, Islamic Azad University of Ahar, AHAR, IRAN
Hazrati81@gmail.com
[2]Electrical Engineering Department, Amirkabir University of Technology, TEHRAN, IRAN, 15914
Kfaez@aut.ac.ir
[3]functional- Scientific University of Ahar
Hazrati2012@gmail.com

Received: January 2012, Revised: April 2012
Online Publication: June 2012

ABSTRACT

Feature selection is an important stage in pattern recognition systems. In this paper we propose a new method based on Cellular Learning Automata- Computing Evolutionary (CLA-EC). The CLA-EC algorithm is an Evolutionary algorithm that is obtained combining from Cellular Learning Automata (CLA) and Computing Evolutionary concept (CA). In this method classification accuracy and number of unselected feature (zeros), considered as fitness function. My Experiments on ORL databases show the effectiveness of the proposed method in compare with Genetic algorithm.

Keywords: Feature selection, Cellular Learning Automata, Computing Evolutionary, Genetic algorithm, face recognition.

1. INTRODUCTION

In most areas such as machine learning, data mining, pattern recognition and signal processing, often there are a large number of features. In these contexts, use of Feature

---

[*] Corresponding Author

selection methods is essential [1, 2]. In feature selection problem, optimal subset of features with size m is selected from a set of Features with size n (n>m) [3]. Reducing the search space for learning algorithms is the purpose of feature selection. In pattern recognition, feature selection can raise the accuracy rate and speed of classification [4].

Feature selection, usually is implement with filter and wrapper method [5, 6].Filter method; select the best features independent of any learning algorithm and power of each feature for classification. Instead, in the wrapper approach, the feature subset selection is done using the Induction algorithm as a black box (i.e., no knowledge of the algorithm is needed, just the interface). Although wrappers may produce better results, they are expensive to run and can break down with very large numbers of features. This is due to the use of learning algorithms in the evaluation of subsets, some of which can encounter problems when dealing with large datasets [6]. Filter and wrapper methods are divided into five groups [6]. In wrapper method, fitness function, evaluate the subset generated, if new subset be better than former candidate subset then new subset is replaced with candidate subset. At the end of each iteration, stop condition is tasted to determine, whether the feature selection process continue or not. Another, well know methods for feature selection is Methods based on ant colony algorithm [7], Methods based on genetic algorithms and Simulated Annealing (SA) [3]. Among the methods that have been suggested for feature selection, the population-based algorithms such as particles swarm optimization algorithm (PSO), Ant colony optimization algorithm and genetic algorithms have been more pay attention [7].

In this paper we introduced a new method that uses Cellular Learning Algorithm Based Evolutionary Computing (CLA-EC), to find the best subset. The proposed method used in face recognition systems on coefficients of discrete wavelet transform and classification accuracy and number of unselected feature (zeros), considered as quality measure. The rest of this paper organized as follows. In section 2 and 3, respectively Evolutionary algorithms and types of automata is briefly described. Section 4 presents the proposed method based on CLA-EC algorithm. Results of the implementation on ORL database are given in the Section 5. And finally, Section 6 concludes this paper with a conclusion and outlook on future work.

2. Evolutionary algorithms

Evolutionary algorithms form a class of random search algorithms in which principles of natural evolution are regarded as rules for optimization. They are often applied to optimization problem where specialized techniques such as gradient based algorithms, linear programming, dynamic programming, and etc., are not available or standard methods fail to give reasonable answers due to multimodality, nondifferentiability or discontinue of the problem at hand.

Genetic algorithms are a large group of Evolutionary algorithms that have used in solving complex problems with rough search space. These algorithm works based on selection and recombination. The set of solutions produced in per period of algorithm called a generation and each of the solutions called genome or chromosome. Each genome is combination of problem variables. Always correlation and information exchange among genomes is done through Selection and Re-combination of genomes in per generation. This information transportation causes that slight responses combined together and produce the high quality response. The poor behaviors of genetic algorithms in some problems, in which the designed operators of crossover and mutation do not guarantee that the building block hypothesis is preserved, have led to propose other approaches. However, with all the positive features that there are in standard Genetic algorithm, behavior of Genetic algorithm depends on definition of parameters such as: Mutation and Crossover operators, Mutation and Crossover probability, population size And Number of generations that produce. Therefore, sometimes in combination of genomes not only improve the quality of answers is not obtained, But the algorithm are trapped in local optimum points. In order to solve this problem, several versions of genetic algorithm have been created. More information about genetic algorithms can be found in [8-10].

One notable characteristic of genetic algorithm is that: each genome can relate to other genome in genomes population, and structure of relationship is complete graph; this matter recently has been considered from society of researchers. In this area attempt for computations parallelization has caused that, local relationships problem between the genomes are introduced seriously. Studies shows that localization problem of relationship between genomes can be simplify parallel computing and increase the quality of the solutions of algorithm [11]. One reasons for this improvement is gradual transfer of effect of a genome to others genomes, that cause to reduce the fanaticism in algorithm. Although this matter reduce the convergence speed but due to the parallel computation capability, its effect is greatly reduce. One of the models that by using its, localization process of genetic operators and relationship between the genomes have been done is Cellular Automata [11-14]. Cellular Automata in addition to having a simple structure for modeling complex systems due to its distributed nature, easily can be implemented on parallel processor and this issue has resulted in Cellular Automata be able to find your application in the field of Evolutionary Computation.

Cellular Learning Automata (CLA) model is obtained combining Cellular Automata (CA) and Learning Automata (LA) models**.** CLA is a CA in which the learning automata are assigned for all cells, to define the state of each cell. Like CA, there is a rule that CLA operate under it. In CA, local rule directly define the state of the cell, but in CLA, this rule determine that: whether the action selected by learning automata in cell should be rewarded or penalized. In order to get specified purpose, the reward or penalty action leads to update the structure of CLA. Dynamism of CLA is obtained by applying local rule on all cells of the LA, frequently. Learning is as an adaptive phenomenon in duration of survival of a living entity and Evolution of long-term process in several generations. Answers to this question that, how learning and Evolution affect each other, yet as an issue of research is open. At this field, different theories have been presented that two of the most important theory are Lamarckian and Baldwin. According to Lamarckian theory, during the Animal life, Environment effected on it and this effect is reflected on genetic structure of living and transmitted to the offspring. Versus From the perspective of Baldwin's theory, learning with encourage the successes that partially take place in the evolution process, lead to conduction and speed up of Evolution [15]. Although now, from researchers in natural sciences, wrong of some of these theories, fully is accepted, but the origin of things in the field of computing is inspired by the nature [16, 17]. One of the models that are obtained from the combining the concepts of evolution and learning is a CLA-EC model that is a combination of a model called cellular learning automata (CLA) and the evolutionary model[18].

2.1. Cellular Automaton

A *cellular automaton* consists of a regular array of cells, each in one of finite number of states. The array can be in any finite number of dimensions. The state of a cell at time t+1 in CA is a function of the states of a finite number of cells (called its neighborhood) at time *t*. The simplest CA is one-dimensional, with two possible states per cell, and a cell's neighbors defined to be the adjacent cells on either side of it. For more information about cellular automata the reader may refer to [19, 20].

2.2. Learning Automata

Learning Automata are adaptive decision making devices operating on unknown random environments. The Learning Automaton has a finite set of actions and each action has a certain probability (unknown for the automaton) of getting rewarded by the environment of the automaton. The aim is to learn to choose the optimal action (i.e. the action with the highest probability of being rewarded) through repeated interaction on the system. If the learning algorithm is chosen properly, then the iterative process of interacting on the environment can be made to result in selection of the optimal action.

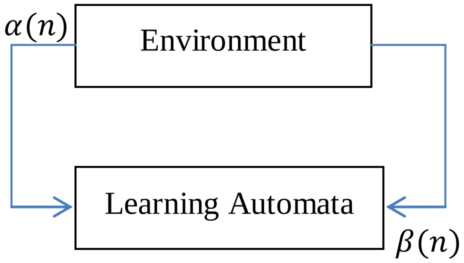

Fig. 1 The interaction between learning automata and environment

Figure 1 illustrates how a stochastic automaton works in feedback connection with a random environment. Learning Automata can be classified into two main families: fixed structure learning automata and variable structure learning automata (VSLA. In the following, the variable structure learning automata are described. A VSLA is a quintuple <α, β, p, T(α, β, p)>, where α, β, p are an action set with s actions, an environment response set and the probability set p containing s probabilities, each being the probability of performing every action in the current internal automaton state, respectively. The function of T is the reinforcement algorithm, which modifies the action probability vector p with respect to the performed action and received response. Let a VSLA operate in an environment with β= {0, 1}. Let n∈N be the set of nonnegative integers. A general linear schema for updating action probabilities can be represented as follows. Let action i be performed at instance n.

If *β (n) =0,*

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1-p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \qquad \forall j,\ \ j \neq i \end{aligned} \tag{1}$$

If *β (n) =1,*

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) \qquad \forall j,\ \ j \neq i \end{aligned} \tag{2}$$

Where *a* and *b* are reward and penalty parameters. When *a=b*, automaton is called L_{RP}. If *b=0* and *0<b<<a<1*, the automaton is called L_{RI} and $L_{R\varepsilon P}$, respectively. For more information about learning automata and its application is referred to [21, 22].

2.3. Cellular Learning Automata

Cellular Learning Automata is a mathematical model for dynamical complex systems that consists of large number of simple components [23, 24]. The simple components, which have learning capabilities, act together to produce complicated behavioral patterns. A CLA is a CA in which learning automaton (multiple learning automaton) is assigned to its every cell. The learning automaton residing in particular cell determines its state (action) 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 neighboring LAs of any particular LA determine the reinforcement signal to the LA residing in that cell. In CLA, the neighboring LAs of any particular LA constitute its local environment, which is nonstationary because it varies as action probability vector 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 specified. The state of every cell is determined on the basis of action probability vectors of learning automata residing in that cell. The initial value may be chosen on the basis of 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. For more information about cellular learning automata and its application is referred to [25-28].

3. CLA-EC algorithm

3.1. Concept of CLA-EC

CLA-EC model is a combination of a model called cellular learning automata (CLA) and the evolutionary model. In this model, each genome is assigned to a cell of cellular learning automata to each of which a set of learning automata is assigned. The set of actions selected by the set of automata associated to a cell determines the genome's string for that cell. Based on a local rule, a reinforcement signal vector is generated and given to the set learning automata residing in the cell. Based on the received signal, each learning automaton updates its internal structure according to a learning algorithm. The process of action selection and updating the internal structure is repeated until a predetermined criterion is met. In this paper to simplify the algorithm, we assume that sight search space is a binary finite search space.

3.2. Feature selection with CLA-EC

In this paper the binary form of CLA-EC is used to solve feature selection problem. In high space of features that each dimension represents the one type of feature, a subset of feature is a point in the search space. If n is a number of all feature then there are 2^n subset of feature, that each of them are different in size and content of features. Best points are subsets that contain minimal numbers of features (equal to most number of zeros in the chromosome) with high classification accuracy. In our encoding scheme, the genome is a bit string whose length (n) is determined by the number of coefficients of discrete wavelet transform (DWT). Each coefficient is associated with one bit in the string. If the i^{th} bit is 1, then the i^{th} coefficient is selected, otherwise, that coefficient is ignored. Each chromosome thus represents a different subset of coefficient.

3.3. Fitness function

The goal of feature subset selection is to use fewer features to achieve the same or better performance. Therefore, the fitness evaluation contains two terms: (i) classification accuracy and (ii) number of features used. Each feature subset contains a certain number of discrete wavelet transform. If two subsets achieve the same performance, while containing different number of features, the subset with fewer features is preferred. Between accuracy and feature subset size, accuracy is our major concern. Combining these two terms, the fitness function is given as:

$$\text{Fitness}=10^4 \text{ Accuracy} +0.5 \text{ Zeros} \tag{3}$$

Where *Accuracy* is the classification accuracy of one specific feature subset, and *Zeros* is the number of coefficients that not selected (i.e., the number of zeros in the chromosome). The accuracy ranges roughly from 0.5 to 0.99 and the number of zeros ranges from 0 to $n-1$ where n is the length of the chromosome.

Overall, higher accuracy implies higher fitness. Also, fewer features used imply a greater number of zeros, and as a result, the fitness increases. It should be noted that individuals with higher accuracy will outweigh individuals with lower accuracy, no matter how many features they contain.

3.4. The proposed algorithm

Cellular Learning Algorithm Based Evolutionary Computing, called CLA-EC is introduced as a new parallel model for evolutionary computing. In my offered algorithm, we use of CLA to solve a feature selection problem. Here, we describe the performance of CLA.

In CLA-EC, similar to other evolutionary algorithms, the parameters of the search space are encoded in the form of genomes. Each genome has two components, model genome and string genome. Model genome is a set of learning automata. The set of actions selected by the set of learning automata determines the second component of the genome called string genome. For each cell, based on a local rule, a reinforcement signal vector is generated and given to the set of learning automata residing in that cell. Each learning automaton based on the received signal update its internal structure according to a learning algorithm. Then, each cell in CLA-EC generates a new string genome and compares its fitness with the fitness of the string genome of the cell. If the fitness of the generated genome is better than the quality of the sting genome of the cell, the generated string genome becomes the string genome of that cell. This process of generating string genome by the cells of the CLA-EC is iterated until a termination condition is satisfied. The main issue involved in designing a CLA-EC for a problem is finding a suitable genome representation and fitness function, and the parameters of CLA such as the number of cells (population size), topology and the type of the learning automata for each cell.

Evolutionary algorithms as the one described algorithm in this paper can be used in any arbitrary finite discrete search space. To simplify the algorithm, we assume that sight search space is a binary finite search space. So the optimization problem can be presented as follows. Assume $f:\{0, 1\}^m \rightarrow \Re$ be a real function that is to be minimized. In order to use CLAEC for the optimization function f first a set of learning automata is associated to each cell of CLA-EC. The number of learning automata associated to a cell of CLA-EC is the number bits in the string genome representing points of the search space of f; the genome is a bit string whose length is determined by the number of coefficients of discrete wavelet transform (DWT). Each automaton has two actions called action 0 and 1. Then the following steps will be repeated until a termination criterion is met.

1- Every automata in a cell i chooses one of its actions using its action probability vector

2- Cell i generates a new string genome, new^i, by combining the actions chosen by the learning automata of cell i. The newly generated string genome is obtained by concatenating the actions of the automata (0 or 1) associated to that cell. This section of algorithm is equivalent to learning from previous self-experiences.

3- Every cell i computes the fitness value of string genome new^i; if the fitness of this string genome is better than the one in the cell then the new string genome new^i becomes the string genome of that cell. That is

$$X_{n+1}^{i} = \begin{cases} X_n^i & f(X_n^i) > f(new_{n+1}^i) \\ new_{n+1}^i & f(X_n^i) \le f(new_{n+1}^i) \end{cases} \tag{4}$$

4- *Se* cells of the neighboring cells of the cell *i* are selected. This Selection is based on the fitness value of the neighboring cells according to truncation strategy. This process is equivalent to mating in the nature. Note that mating in the context of proposed algorithm is not reciprocal, i.e., a cell selects another cell for mating but necessarily vice versa.

5- Based on selected neighboring cells a reinforcement vector is generated. This vector becomes the input for the set of learning automata associated to the cell. This section of algorithm is equivalent to learning from experiences of others. Let N_s *(i)* be set selected neighbors of cell *i*. Define,

$$N_{i,j}(k) = \sum_{l \in \mathrm{Ns}\,(\mathrm{i})} \delta_l (X_n^{l,j} = k) \tag{5}$$

Where

$$\delta(exp) = \begin{cases} 1 & \exp\ is\ true \\ 0 & otherwise \end{cases} \tag{6}$$

$\beta^{i,j}$, the reinforcement signal given to learning automaton *j* of cell *i,* is computed as follows,

$$\beta_n^{i,j} = \begin{cases} u\left(N_{i,j}(1) - N_{i,j}(0)\right) & if \quad X_n^{i,j} = 0 \\ u\left(N_{i,j}(0) - N_{i,j}(1)\right) & if \quad X_n^{i,j} = 1 \end{cases} \tag{7}$$

Where u (.) is a step function. The overall operation of CLAEC is summarized in the algorithm of figure of 2.

```
Initialize.
While not done do
    For each cell $i$ in CLA do in parallel
        Generate a new string genome
        Evaluate the new string genome
        If $f$(new string genome)> $f$(old string genome) then
            Accept the new string genome
        End if
        Select $Se$ cells from neighbors of cell $i$
        Generate the reinforcement signal vector
        Update LAs of cell $i$
    End parallel for
End while
```

Fig .2 Pseudocode of CLA-EC

4. Implementations and conclusions

In order to show the performance of the proposed algorithm, we compare the binary CLA-EC with genetic algorithm on ORL database. The ORL database [24, 25] contains ten different images of 40 distinct subjects in up-right, frontal position with tolerance for some tilting and rotation of up to 20 degrees. Moreover, the most variation of some image scale is close to 10%. Therefore, it is expected that this is a more difficult database to work with. For ORL database, the training set has been formed by using deferent samples of each individual. Sample images from this database are given in Figure 3. Public information of ORL database is illustrated in table 1.

Table .1 Public information of ORL database

Total Images	Dimensions of Image	Number of persons
400	92×112	40

Figure .3 Sample images from ORL database

After preprocessing the images (histogram equalization), Wavelet transform was applied on all images and DWT coefficients were considered as feature vectors. The images were discrete, so the Haar wavelet has used. To analysis the pre-processed images, three levels of the pyramid algorithm were used. Then by connecting approximation column, the images had built in level three of feature vectors. In ORL dataset, Images dimensions are 92x112, so after images analysis to level three the length of Wavelet features became 168. Suggested method tested by Different parameters and showed that by using values in the table 2, maximum efficiency is obtained.

Table .2 parameters of GA and CLA-EC algorithms

CLA-EC algorithm	GA algorithm	
35	35	Population size (number of cells)
100	100	Number of iteration
	0.7	crossover Probability
	0.03	mutation Probability
1	-	Neighborhood radius (r)
2	-	eS (Number of selected cells in each neighborhood)
0.03	-	Reward rate (a)
0.03	-	penalty rate (b)

Acquired features from both algorithms were classified by the Nearest Neighbor algorithm and classification rate is considered as evaluation measure. Classification rate for GA algorithm with running time of 1580 seconds was obtained 98.5%. While these rates for CLA-EC algorithm with running time of 1510 seconds, was achieved 99.5%. The results showed that CLA-EC algorithm works better than genetic algorithm, both in terms of speed and accuracy. In order to display the performance of GA algorithm and the proposed algorithm, the recognition rate chart based on number of features selected by both algorithms is shown in the following figure. As you seen only with select 39 Feature the proposed method has reached to %99.5 recognition rate.

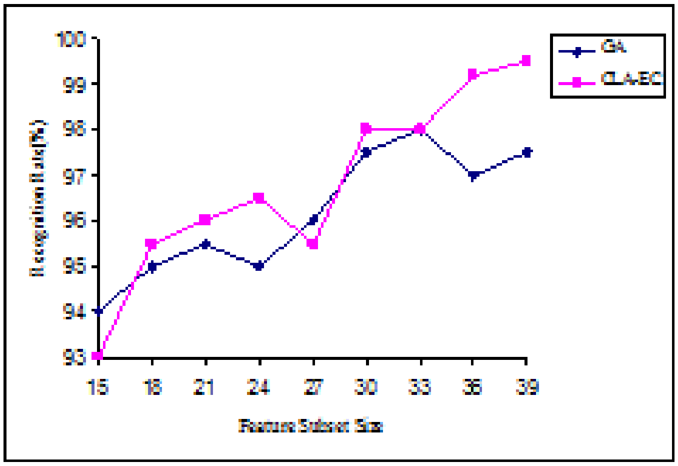

Fig .4 Recognition rate in terms of number of features that selected by GA algorithm and CLA-EC algorithm

5. CONCLUSION

In this paper a new method based on CLA-EC is presented for feature selection problem. CLA-EC algorithms converge to optimal solutions, quickly and have high ability in search of solutions space; also it can find the smallest subset of features. In this method classification accuracy and number of unselected feature (zeros), considered as evaluation measure. The results of experiments on ORL dataset proved the superiority of new methods in solving the feature selection problems compared with the Genetic algorithms. The advantage of this algorithm is its distribution ability. On the other hand the results shows, with a relatively small number of genomes (population or cells), problems with large numbers of feature are optimized. While in the case of genetic algorithms, such rule is not true.

REFERENCE

[1] Liu, H. and Motoda, H., Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, 1998.
[2] Guyon, I. and Elisseeff, A., “An introduction to variable and feature selection”, *J. Mach. Learning Res.* 3, 1157–1182, 2003.
[3] Jensen, R., “Combining rough and fuzzy sets for feature selection”, Ph.D. Thesis, University of Edinburgh, 2005.
[4] Kudo, M. and Sklansky, J., “Comparison of algorithms that select features for pattern classifiers,” Pattern Recognition 33 (1), 25–41, 2000.
[5] Duda, R.O. and Hart, P.E., *Pattern Recognition and Scene Analysis*, Wiley, New York, 1973.

[6] Kohavi, R. and John, G., “Wrappers for feature selection”, *Artificial intelligence,* Vol. 97, No. (1-2), pp. 273-324, 1997.

[7] Kanan, H.K., Faez, K. and Hosseinzadeh, M., “Face Recognition System using Ant Colony Optimization- Based Feature Selection”, Proceeding of the CISDA, 2007.

[8] William M. Spears, An Overview of Evolutionary Computation. proceedings of the 1993 European Conference on Machine Learning

[9] A.E.Eiben J.E.Smith ,Introduction to Evolutionary Computing (Springer2003)

[10] Genetic Algorithms Archive, Repository for GA related information, http://www.aic.nrl.navy.mil/galist.

[11] Rudolph, G. and Joachim, S. “A Cellular Genetic Algorithm with Self-Adjusting Acceptance Threshold”, Genetic Algorithms in Engineering Systems: Innovations and Applications, Conference Publication, no. 414, pp. 365-372, 1995.

[12] Alba, E. and Troya, J. M. “Analyzing Synchronous and Asynchronous Parallel Distributed Genetic Algorithms” , Future Generation Computer Systems, vol. 17, pp. 451-465, 2001

[13] Lessani, M. and Meybodi, M. R. “Genetic Cellular Automata”, in proceedings of the 5th conference on Intelligent Systems, Mashhad, Iran, 2003.

[14] Tomassini, M. “The Parallel Genetic Cellular Automata: Application to Global function Optimization”, in proceedings of International Artificial Neural Nets and Genetic Algorithms Conference, Austria, pp. 385-391, Springer, Wein,1993.

[15] Riopka, T. P. and Bock, P. “Intelligent Recombination Using Individual Learning in a Collective Learning Genetic Algorithm”, In Proc. of the Genetic and Evolutionary Computation Conf. (GECCO-2000), pp. 104-111, Morgan Kaufmann, 2000.

[16] Whitley, D., Gordon, V. S. and K. Mathias, K. “Lamarckian Evolution, The Baldwin Effect and Function Optimization”, Parallel Problem Solving from Nature III, Springer-Verlag, 1994.

[17] Julstrom, B. A. “Comparing Darwinian, Baldwinian, and Lamarckian Search in Genetic Algorithm for 4- Cycle Problem”, Technical Report, Department of Computer Science, St. Cloud State University, St. Cloud, USA, 1999.

[18] Rastegar, R. and Meybodi, M. R. “A New Evolutionary Computing Model based on Cellular Learning Automata”, to appear in proceedings of IEEE International Conference on Cybernetics and Intelligent Systems 2004, Singapore, 2004.

[19] Mitchell,M, James P. Crutch_eld, Rajarshi Das" Evolving Cellular Automata with Genetic Algorithms: "Proceedings of the First International Conference on EvolutionaryComputation and Its Applications (EvCA'96) Moscow, Russia: Russian Academy of Sciences.

[20] Wolfram, S. Cellular Automata and Complexity, Perseus Books Group, 1994.

[21] Narendra, K. S. and Thathachar, M. A. L. Learning Automata: An introduction, Prentice Hall, 1989.

[22] Najim, K. and Poznyak, A. S., editors, Learning automata: theory and application, Tarrytown, New York: Elsevier Science Publishing Ltd., 1994.

[23] Masoodi, B., Meybodi, M. R. and M. Hashemi, " Cooperative CLA-EC", Proceedings of 12th Annual CSI Computer Conference of Iran, Shahid Beheshti University, Tehran, Iran, pp. 558-559, Feb. 20-22, 2007.

[24] Masoodifar, B., Meybodi, M. R. and Rastegar, R., "Asynchronous CLA-EC", Proceedings of 11th Annual CSI Computer Conference of Iran, Fundamental Science Research

Center (IPM), Computer Science Research Lab., Tehran, Iran, pp. 447-458, Jan. 24-26, 2006.

[25] Meybodi, M. R., Beigy, H. and Taherkhani, M. "Cellular Learning Automata and Its Applications", Journal of Science and Technology, University of Sharif, No. 25, pp.54-77, Autumn/Winter 2003-2004.

[26] Rastegar, R. and Meybodi, M. R. "Experiments with Cellular Learning Automata", Technical Report, Computer Engineering Department, Amirkabir University, Tehran, Iran, May 2005.

[27] Kharazmi, M. R. and Meybodi, M. R. "Application of Cellular Learning Automata to Image Segmentation", Proceedings of Tenth Conference on Electrical Engineering (10th ICEE), University of Tabriz, Vol 1, pp. 298-306, May 2002.

[28] Kharazmi, M. R. and Meybodi, M. R. "An Algorithm Based on Cellular Learning Automata for Image Restoration", Proceedings of The First Iranian Conference on Machine Vision & Image Processing, University of Birjand, pp. 244 –254, March 2001.