IJICTR International Journal of Information & Communication Technology Research

Volume 6- Number 4- Autumn 2014 (49-56)

Quality Improvement of Interactive Tutorial-Like Systems by Use of Cellular Learning Automata

Hajar Hajary
Department of Computer
Science and Research Branch Islamic azad university
Tehran, Iran
Hajar_Hajary@yahoo.com

Ali Ahmadi
Department of Electrical and Computer
K.N.Toosi University of Technology (K.N.Toosi)
Tehran, Iran
Ahmadi@kntu.ac.ir

Received: December 9, 2013-Accepted: November 3, 2014

Abstract—**Determining the best way of learning and acquiring knowledge, especially in intelligent tutoring systems have drawn researchers' attention during past years. With regard to studies conducted on E-learning systems and strategies proposed to improve the quality of these systems, it can be said that the interactions play a vital role in the educational systems. Therefore, the learners are not only affected by the teacher in a learning environment but also significantly learn important materials through the interaction with other learners. In this article, a new modeling approach is presented for improving learning/teaching models as well as interaction among learners, from which the most benefit can be derived by learners. The proposed model uses cellular learning automata in order to model behavior of the learners as well as interactions between the learners for knowledge acquisition. This algorithm also deals with the process of teaching as well as education of the learners. The results indicate that relationship between the learners can improve their knowledge and also increase their learning speed compared to previous methods.**

Keywords-learning automata; cellular learning automata; interactive learning; tutorial-like system

I. INTRODUCTION

The Intelligent Educational Systems are new generation of educational systems which employ artificial intelligence techniques to obtain knowledge. These systems aimed to improve both teaching and learning abilities in human beings [1,2]. There are diverse architectures and components for such systems according to different interpretations extracted from the intelligent training concept. In general, these systems have three main factors (sometime user interface factor is also added to these main factors [3,4]). The main factors include the domain model, the student model, and the educational model, where the main focus lies on the student model. This model is basis of the student behavior as well as status which shows his attitude and state [5]. Self defined these components as the tripartite architecture for a ITS (Intelligent Tutoring System): the what (domain model), the who (student model), and the how (tutoring model) [6]. The application of machine learning techniques is greatly important in these systems and it has been investigated in a number of studies in the field that such techniques could be applied to improve teaching. Machine learning can be roughly classified into three categories according to the primary approach that they take to inference, induction, deduction, and analogy. Holland et al. defined induction as "all inferential processes that take place in the face of uncertainty". Induction is concerned with inferring knowledge from an incomplete set of observations. It is inherently uncertain because the resulting knowledge is based on incomplete information. Deduction learning (or compilation), on the other hand, works on existing facts and knowledge, and deduces new knowledge from the old knowledge. Assuming that the exiting knowledge is complete and correct, deduction therefore guarantees inference and the truth of this inference. Analogical learning can be viewed as a combination of the first two types. Machine learning algorithms can also be classified as supervised learning or unsupervised learning. In supervised learning, training examples consist of pairs of input objects and desired output. The task of the learning

algorithm is to learn how to predict the output values of new examples, based on their input values. In unsupervised learning, training examples contain only the input objects with no explicit target output. The learning algorithm needs to generalize from the input patterns to discover the output values. Machine learning is also used in different parts of ITSs. **It** is used in the construction of background knowledge. Beck et al. used machine learning to improve tutoring strategy. Sision and Shimura believed that analogical learning is more appropriate for learning-level analysis, while reinforcement learning, is apparently more appropriate for tutoring. Reinforcement learning can be used to train an agent to comply with the needs of a student [6]. Sision and Legaspi modeled the learning process by application of reinforcement learning as a major learning method in 2000 [7]. Baffes and Mooney implemented ASSERT in 1996 [8]. This model applied reinforcement learning in the student modeling in order to find the errors which the new students may make by only using an appropriate knowledge domain. Lelouche used a series of interactive factors in order to model the process of learning in intelligent educational systems in 2000 [9]. Finally, Oommen and Hesham used learning automata in intelligent education system in order to model the student's learning process as well as the interactions between them, respectively in 2007 and 2010[10, 11, 12]. Thus, the interactions between students were stated based upon a specified strategy by these researchers.

The interactions between students are considered as a source of learning in real-life teaching environments. While students usually consider the teacher as the main source of information, they also are dependent on each other to adjust their learning. The traditional fundamental principles of teaching/ learning as exemplified in the field of tutorial systems, assume that the learning achieved by the student(s) is a consequence of his/their learning from a teacher or a set of teachers. The intention of this paper is to demonstrate that this paradigm can be generalized so as to also permit the student to learn from a so-called classroom of students learning at different rates and abilities.

The main objective of the proposed system in this study is introducing a new method based on the cellular learning automata to model the interaction between students in a tutorial-like- system. In other words, we tried to introduce a new approach which can explain how a student learns new material in a tutorial like system based on the stochastic learning automata theory. In this new model a student is a member of a classroom of students, so that not only learns from the teacher, but also obtain information from other students. In our system a student simulator is used to mimic the behavior of real-life students during the learning process. Students are divided into three categories based on their mental model which include slow, normal and fast learners. This classification is in accordance with the real educational system. In this model, each student is considered as a learning automaton in a cell, so that the current model can be compared with the model presented in [10]. The interactions between students are modeled as the interactions between different learning automata (neighboring cells), and also the student-teacher interaction is simulated as the interaction of each learning automaton with the environment. The purpose of this model is to accelerate the learning process of each student as well as overall learning of the students and enhancing the quality of the student's learning.

It is worth noting that other learning mechanisms including neural network, Bayesian, Markovian and reinforcement learning models can also be used in this model. Although we realized that generalization of our case study into other learning models would not be so difficult, some problems we faced in presentation of each model are listed as below [11]:

1- A neural network model can be used if it consists of a large number of neurons. The network topology should be accurately specified and its weights identified as the network parameters.

2- If the student is presented by either Bayesian or Markov model, there should be some states whose relationship with the transfer function is specified.

3- If the model is designed based on reinforcement learning model, the state vector of the model should not only include probability distribution of the states, but also relevant information about the environment.

In the real environment, students can successfully communicate with others and obtain information from them. As the real environment, each cell (the student) can communicate with the neighboring cells and be affected by state of the neighboring cells. In other words, the relationship between the student and his smarter classmate, accelerates the process of learning relevant to the student (increases his curriculum development) in a real class. Then, as a real class, neighborhood can affect internal behavior of each cell (LA), or students' information/knowledge. As a result, this will increase the speed of learning.

The rest of the paper is organized as follows: in section II and III the concept of learning automata as well as cellular learning automata is described. Tutorial-like system is explained in Section IV. Sections V and VI are dealing with introduction of the proposed method and its evaluation. Finally, the conclusion is given in Section VII.

II. LEARNING AUTOMATA

Learning automata are used in the systems that have incomplete information about their environment [13,14]. automata behaves in a way that can result in performing few actions. Once this machine selects an action, the selected action is evaluated by the environment and a feedback of this evaluation is sent to the learning automata as either a positive feedback signal (if the action was done properly) or a negative one (in case the action was done improperly). The value of this signal determines which actions should be chosen in the following steps. This process aims to make the automata tend to the most appropriate action

desired by the environment after a while. In other words, this process aims to make the automat learn to do what is the best action. "Fig.1", illustrates how a stochastic automata works in feedback connection with a random environment.

This machine may act randomly in a probabilistic environment. As already mentioned, it can update the probability related to its action performance based on the inputs received from the environment. The learning automata are classified into two classes including the fixed structure (FSSA) and the variable structure (VSSA) type [15, 16]. A variable structure automata is defined by the quadruple {α, β, p, T} in which $\alpha = \{ \alpha_1 , \alpha_2 ,\ldots, \alpha_n \}$ represents the action set of the automata, $\beta = \{ \beta_1 , \beta_2 , \ldots, \beta_r \}$represents the input set, $p=\{p_1 , p_2 , \ldots, p_r \}$ represents the action

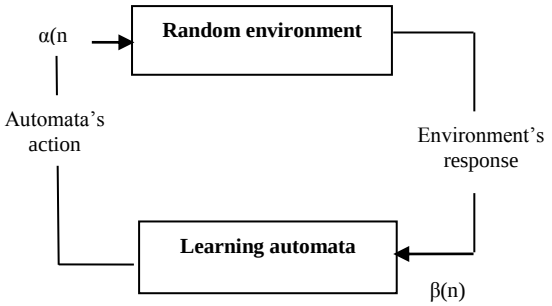

Fig. 1 The Interaction between learning automata and environment

probability set, and finally p(n+1) = T[α(n), β(n),p(n)] represents the learning algorithm[15,16]. Where $p_i(t)$ (i=1,2,…,r) is the probability that the automaton will select the action α_i at time "t", i.e., $p_i(t) = Pr[\alpha(t)= \alpha_i]$, i=1,2,…r. and it satisfies $\sum p_i(t) = 1 \quad \forall t$. This automaton operates as follows. Based on the action probability set *p*, automaton randomly selects an action α_i, and performs it on the environment. Having received the environment's reinforcement signal, automaton updates its action probability set based on equation (1) for favorable responses, and on equation (2) for unfavorable ones.

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

$$[2] \qquad \begin{aligned} p_i(t+1) &= (1-b).p_j(t)) \\ p_j(t+1) &= b/ r-1 + (1-b)p_i(t) \qquad \forall j \quad j\neq i \end{aligned}$$

In these two equations, a and b are reward and penalty parameters, respectively. For a=b, learning algorithm is called L_{R-P}, for a << b, it is called $L_{R\varepsilon P}$, and for b=0 it is called L_{R-I}.

It is worth noting that cellular automata are mathematical models for systems consisting of large numbers of simple identical components with local interactions. Cellular automata and Learning automata are combined to create a new model called Cellular Learning Automata (CLA). This model is superior to cellular automata because of its ability to learn and also is superior to single learning automata because it is a collection of learning automatons which can interact with each other.

III. CELLULAR LEARNING AUTOMATA

Cellular automata introduced by Von Neumann [17] are mathematical models for defining systems that consist of a large number of simple identical components with local interactions. Researchers, scientists and practitioners from different fields have exploited the CA paradigm of local information, decentralized control and universal computation for modeling different applications The combination of cellular automata and learning automata results in cellular learning automata (CLA) which is superior in performance to cellular automata due to its learning ability and also is superior to single learning automaton due to its distributed processing ability which is provided by employing a set of interacting learning automata. The reason behind the popularity of cellular automata can be traced to their simplicity, and to the enormous potential they hold in modeling complex systems.

Cellular Learning Automata (CLA) is a mathematical model for dynamical complex systems that includes large number of simple components. Cellular automata can be viewed as a simple model of a spatially extended decentralized system made up of a number of individual components (cells). The communication between constituent cells is limited to local interaction. Each individual cell is in a specific state which changes over time depending on the states of its local neighbors. The overall structure can be viewed as a parallel processing device. However, this simple structure when iterated several times produces complex patterns displaying the potential to simulate different sophisticated natural phenomena.

A cellular learning automata is a cellular automata in which a learning automata is assigned to its every cell [18]. The learning automaton residing in each cell determines the state of the cell on the basis of its action probability vector. The operating rule in CLA and the actions selected by the neighboring cells determine the reinforcement signal to the learning automata residing in that cell. In CLA, the neighboring learning automatons of any cell constitute its local environment.

The state of every cell is determined on the basis of action probability vector of the learning automata residing in that cell. The initial value of this state may be chosen based on the past experience or at random. In the second step, the rule of CLA determines the reinforcement signal to each learning automaton. 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 "Fig.2", [18,19].

IV. TUTORIAL-LIKE SYSTEM: AN OVERVIEW

Tutorial-like Systems provide a demonstration of status of a student during learning. In these systems, the student can learn and test without any need to

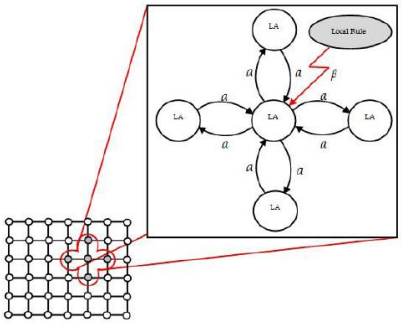

Fig. 2 Cellular Learning Automata [18]

presence of real person, even without presence of real-life students, but rather each student can be replaced by a simulated student that mimic a real-life student. The teacher attempts to provide the training materials to school of student simulators. Moreover, the students are allowed to share their information with each other, so that they can learn from each other more than what is provided in the traditional way of learning. Therefore, the learning environment allows the students to learn not only from the teacher but also from their other classmates. In our model, components of the tutorial-like system of education follow the scholastic model. The students obtain knowledge through questions which are designed in the form of multiple choices questions. These questions, in our present paradigm, include several items with different level of confidence. Then, the student learns to choose the answer which has the highest level of confidence in a gradual way [6].

A. *Characteristics of Tutorial-Like Systems*

Tutorial-like systems have similarities with the well developed field of tutorial systems. For example they model the teacher, the student, and the domain knowledge. However, they differ from "traditional" tutorial systems in some aspects as follows [10]:

1) Different Type of Teacher: In tutorial systems, as they are developed so far, the teacher is assumed to have perfect information about the material to be taught. Also, built into the model of the teacher are the knowledge of how the domain material is to be taught and a plan of how it will communicate and interact with the students. The teacher in our Tutorial-like system possesses different features. First, and as a fundamental difference is that the teacher is uncertain of the teaching material. Second, the teacher does not initially possess any knowledge about "how to teach" the domain subject. Rather, the teacher himself is involved in a "learning" process, and he "learns" what teaching material has to be presented to a particular student. To achieve this, the teacher follows the Socratic model of learning by teaching the material using questions that are presented to the students. He then uses the feedback from the students and their corresponding LAs to suggest new teaching materials. Although removing the "how-to-teach" knowledge from the teacher would take away the "bread-and-butter" premise of the teaching process in a tutorial system, in a Tutorial-like system, removing this knowledge allows the system to be modeled without excessive complications and renders the modeling of knowledge less burdensome. The success of our proposed methodology would be beneficial to systems in which any domain knowledge that is pertinent to tutoring the teaching material could be merely plugged into the system without the need to worry about "how to teach" the material.

2) No Real-Student: A tutorial system is intended for the use of real-life students. In our Tutorial-like system, there is no real-life students who use the system. The system could be used by either of the following:

a. Student simulators, which mimic the behavior and actions of real-life students using the system. Finally, would themselves simulate how the students improve their knowledge and their interaction with the teacher and with other students.

b. An artificial entity which, in itself, could be another software component that needs to "learn" specific domain knowledge.

3) Uncertain Course Material: Unlike the domain knowledge of "traditional" tutorial systems where the knowledge is typically well defined, the domain-knowledge teaching material presented in our Tutorial-like system contains a material that has some degree of uncertainty. The teaching material contains questions, each of which having a probability that refers the certainty of whether the answer to the question is in the affirmative.

4) Testing Versus Evaluation: Sanders differentiates between the concepts of "teaching evaluation" and "teaching testing." He defines "teaching evaluation" as an "interpretive process," in which the teacher "values and determines the merit or worth of the Student performance and their needs." He also defines "teaching testing" as a "data-collection process." In a tutorial system, an evaluation is required to measure the performance of the Student while using the system and acquiring more knowledge. In our Tutorial-like system, the Student(s) acquire knowledge using a Socratic model, where it gains knowledge from answering questions without having any prior knowledge about the subject material. In our model, the testing will be based on the performance of the set of Student simulators.

V. TUTORIAL-LIKE SYSTEM BASED ON CELLULAR LEARNING AUTOMATA

The tutorial-like system simulation was done based on cellular learning automata in this research. We used a student simulator to mimic real-life behavior of the students during their learning process. In this model learners use their classmates' knowledge to improve their own by communicating with them through CLA. As it was previously mentioned, each student was considered as a learning automaton in a cell and each automata and it's behavior in order; represents a student and his knowledge and since the number of iterations to converge in LA depends on behavior automata and this behavior of his knowledge is obtained. whatever the number of iteration's converge to reduce, so it caused to improve the quality of learning.In other words, the quality of the learning parameter in LA is the number of iteration of automata to converging. Whatever reducing the number of iterations compared to the previous amount iteration of automata which mentioned before is reduced by means of the automata is to gain knowledge of more and better. So it can be proved that improve the quality of learning would cause reducing iterations of automata.The interactions between students were done through (were simulated as) the interactions between different LAs (neighboring cells). These interactions can accelerate the learning process of each student and enhance the overall quality of the students' learning. Structure of the proposed model is illustrated in "Fig.3".

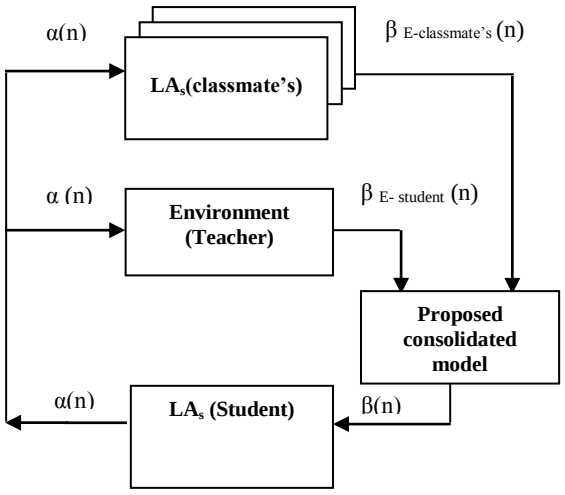

Fig.3 The proposed structure of Tutorial-like System Based on Cellular Learning Automata.

A. *Modeling a Student*

In this system, the model of each student shows behavior of the student and the student's state of mind as well as the method he chooses in order to obtain knowledge. The following models were selected for each student. In addition, all actions and overall performance of the student were recorded during the period of using system by students.

1) A model based on FSSA (fixed-structure stochastic automata) or slow VSSA (variable-structure stochastic automata) which represents a slow student.
2) A model based on VSSA which represents a normal student.
3) A model based on estimator automata (Pursuit) which represents a fast student.

Stochastic automata are generally divided into two groups including the automata with a fixed structure and the automata with a variable structure. The action's probability vector of the automata is fixed in the automata with a fixed structure. On the other hand, these probabilities vary step by step in the automat with a variable structure. Given that the automata's actions are based on its actions' probability vector in each step, the automata can select better actions with probably higher rewards for the next moment (step). As a result, it is evident that the automata with a fixed structure act significantly slower than the automata with a variable structure. Then, the automata with a fixed structure need multiple iterations for the convergence. It is also noteworthy that the change rate of the probability vector is dependent on both reward and penalty function. This can greatly influence the speed of learning process. In recent years, other models of learning automata were proposed which called as pursuit automata [16, 20]. As is clear from its name, this model always seeks the action that is already estimated as the optimal action. This method increases the probability of the optimal action, even if the selected action was rewarded or penalized. The main advantage of the automata is increasing the speed of learning process. We believe that these three families of automata are distinct kinds of students based on the way they learn new materials. These three families also represent the students' mental models.

VI. EXPERIMENTAL RESULTS

In this section, we present the experimental results obtained by testing the prototype implementation of the student-classroom interaction. To obtain these results, we performed numerous simulations to accurately simulate how a student interacts with other classmates.

To comply with the work done in [10], the simulation was done based on an environment with nine students. Thus, the cellular automaton model consisted of 9 cells in the proposed model. A learning automaton was placed in each cell depending on type of the student, i.e. whether he is a slow, normal or fast learner.

This classroom included the following types of student.

1) Fast-learning student: To mimic this type of student, the student simulator used a pursuit PL_{RI} scheme, with λ being in the range of

0.0041 – 0.0127. In this scheme, each LA will update its action-probability vector if it obtained a reward.

2) Normal-learning students: To simulate students of this type, the student simulators used VSSA. In particular, it utilized the L_{RI} scheme, with λ being in the range of 0.0182–0.0192.

3) Slow-learning student: The student simulators also used VSSA to simulate learners of this type. Again, our model used the L_{RI} scheme, but with a lower value of λ, which was between 0.0142 and 0.0152.

In the simulations, we used the majority-minority rule for neighborhood effect [21]. If the cell selected α_i action and the number of the neighbors who selected this action was equal to at least five, the cell under study has probably chosen the correct action according to its neighbors. Then, the response of the neighbors can be considered as a favorable response (neighbor=0). If the number of neighbors who have chosen this action were less than five, the response of the neighbors is considered as undesirable (neighbor=1). A model for integration of both neighborhood and environmental factors is also considered here which is shown in "Table 1".

The results were obtained from 75 times experiments. These results were compared with the model presented in [10]. In that work, the student only learns from the teacher and has no interaction with anybody. As it was previously mentioned, educational materials in the tutorial-like system included multiple-choice questions. The students should learn the questions and answer them correctly. The teaching problems in these experiments have been represented by two different type of environments, namely, two four-action environment and two ten-action environments, both of which have been earlier used as benchmarks in the field of LA. In all these simulations, the convergence of learning

Table 1: Model for integration neighborhood and environmental factors

Neighborhood rule	Environmental factors	Result of α_i
Neighbor=0	Reward	Reward
Neighbor=0	Penalty	Reward=0.2 penalty=0.8
Neighbor=1	Reward	Reward=0.85 penalty=0.15
Neighbor=1	Penalty	Penalty

automata is considered as when the probability of the selected action goes upper than a threshold value T and very closed to 1 (here T=0.99). In an educational environment, according to different knowledge level of the learners and their interactions, the benefits related to each learner or a group of learners is different from each other.

For example, when the number of fast learners is more than the number of slow learners in the learning group, therefore, there is a high probability that the slow learner may interact with individuals who are smarter than him and have a superior knowledge. Thus, he may have a significant progress in his relevant process of learning providing the interactions he has with smarter students and vice versa.

A. Results for Four and Ten-Action Environments

The experiments were done using two sets of environments, namely, two four-action environments ($E_{4,A}$ and $E_{4,B}$) and two ten-action benchmark environments ($E_{10,A}$ and $E_{10,B}$). The nine students in the simulations needed to learn the responses for the questions, and were asked to determine the corresponding answers, which, in this model, represented the actions that possessed the minimum penalty probability. The simulation of the student–classroom interaction would reveal if the students benefited from their interaction with other students according to their interaction strategy. For $E_{4,A}$ and $E_{10,A}$ the λ of the student simulators LA were set to be as follows:

1) 0.0127 for the fast-learning student;
2) 0.0192 for the normal-learning student;
3) 0.0142 for the below-normal-learning student.

Also, for $E_{4,B}$ and $E_{10,B}$, the λ of the student simulators LA were set to be as follows:

1) 0.0041 for the fast-learning student;
2) 0.0182 for the normal-learning student;
3) 0.0152 for the below-normal-learning student.

For the four-action environments, the two settings for the reward probabilities were

$E_{4,A} = \{0.7\ 0.5\ 0.3\ 0.2\}$

$E_{4,B} = \{0.1\ 0.45\ 0.84\ 0.76\}$.

Similarly, for the ten-action environments, the reward probabilities were

$E_{10,A} = \{0.7\ 0.5\ 0.3\ 0.2\ 0.4\ 0.5\ 0.4\ 0.3\ 0.5\ 0.2\}$

$E_{10,B} = \{0.1\ 0.45\ 0.84\ 0.76\ 0.2\ 0.4\ 0.6\ 0.7\ 0.5\ 0.3\}$.

The results of this simulation are presented in "Table 2". The experimental results showed that the knowledge of the slow student had a significant progress in the proposed model compared to the model of the students only interacted with his teacher and only learnt from him. Moreover, the number of iterations required to reach to convergence was decreased. For example, in the four-action $E_{4,A}$ environment, the number of iterations needed for the slow-learning student LA to converge decreased to 1110 from 1382. This indicates the effective relationship of the slow student with his other classmates because here three slow students communicated with six other students who had superior knowledge and learn faster than the three slow ones. Also, the learning process of the fast student has slowed down. This deterioration in the learning process of the fast student is due to the fact

that there are 3 fast students and 6 normal and slow students in this experiment. Therefore, when the fast student seeks help for improvement, he may find eight other students two of whom are in the same knowledge as himself and six others knowledge are lower than him including three normal students and three slow learner students. In other words, there are no genius help for the fast student in this group, so it is clear that his learning process slows down in this case.

For normal students, the interaction with their other fast and slow learner students can be beneficial. For example, in the four and ten-action $E_{4,A}$ and $E_{10,A}$ environments, the number of iterations needed for convergence has decreased from 996 to 696 and from 1297 to 1059. On the other hand, in the four and ten-action $E_{4,B}$ and $E_{10,B}$ environments, the number of iterations needed for convergence has decreased from 2201 to 1286 and from 2114 to 1419.

The results of this simulation suggest that the difference between iterations as well as the rate of improvement for slow students is more than normal students. This is due to the fact that there are two superior groups of students for slow students whose knowledge are higher than slow students (to get help from in order to improve in their learning process). However, there are one superior group and one lower group of students for the normal students. Therefore, while this interaction may improve learning status of the normal student, oscillations between these two groups will slow down at some points. Furthermore, the results shown in the "Table 2" indicate that the convergence time in $E_{4,B}$ and $E_{10,B}$ environments is greater than of $E_{4,A}$ and $E_{10,A}$ environments. This reflects the fact that the set of E_B environments was more difficult because of the proximity of the underlying penalty/reward probabilities. Also, the results showed that the ten-action environments were more difficult than the four-action environments. The iterations required for the LA convergence increased from the four-action environments to the ten-action Environments. The rates of learning for a slow, normal and fast student are shown in "Fig.4".

Table 2. Convergence of proposed model and comparison with presented model in [10]

		Fast student (λ: 0.004 - 0.0127)		**Normal student** (λ: 0.0182 - 0.0192)		**Slow student** **(λ: 0.0142 - 0.0152)**	
		# of Iteration to converge					
Env.	**# of Actions**	**Old model**	**New model**	**Old model**	**New model**	**Old model**	**New model**
E_A	4	572	662	996	696	1382	1110
E_B	4	1482	1564	2201	1286	2633	1442
E_A	10	686	704	1297	1059	1804	1424
E_B	10	1655	1642	2114	1419	2859	1479

Reward probabilities for 4-action environment are :
$E_{A,4}$: 0.7 0.5 0.3 0.2 $E_{B,4}$: 0.1 0.45 0.84 0.76

Reward probabilities for 10- action environment are:
$E_{A,10}$: 0.7 0.5 0.3 0.2 0.4 0.5 0.4 0.3 0.5 0.2 $E_{B,10}$: 0.1 0.45 0.84 0.76 0.2 0.4 0.6 0.7 0.5 0.3

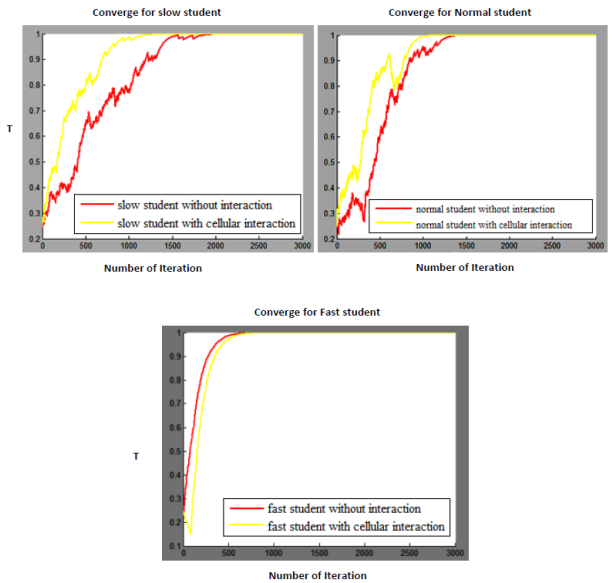

Fig. 4. Rate of Learning for Students

VII. CONCLUSION

In this paper, a new approach for modeling tutorial-like systems and improving the student modeling method was proposed. The cellular learning automata were used for simulation of neighborhood in the proposed model. We have demonstrated that tutorial-like systems can be generalized so as to permit the student to learn from a so-called classroom of students who are learning at different rates and abilities. From the simulation results, we conclude that the interaction between the different students was most beneficial to slow students, and fast students showed either minimal gains or deterioration in learning, mainly because they more often interacted with those students whose knowledge are lower than them. Finally, the simulation results showed that this method is a feasible and appropriate mechanism to be implemented for the learning process. Using cellular learning automata can properly implement the interactions between the students, increase their knowledge level and improve the learning process of the students as well as their learning speed compared to previous methods.

REFERENCES

[1] V.J.Shute ; J.Psotka., "Intelligent Tutoring Systems: Past, Present, and Future", Handbook of Research on Educational Communications and Technology, Scholastic Publications1995.

[2] E.Wenger, "Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge" ,Los Altos, CA: Morgan Kaufmann Publishers, Inc1987.

[3] E.Fischetti, A.Gisolfi., "From Computer-Aided Instruction to Intelligent Tutoring Systems", Educ. Technol, 1990, vol, 30, no. 8, pp. 7–17.

[4] R.Winkels ; J.Breuker., " What's in an ITS? a functional Decomposition, in New Directions for Intelligent Tutoring Systems", E. Costa, Ed.Berlin, Springer-Verlag, Germany, 1990.

[5] J.Self, "The Defining Characteristics of Intelligent Tutoring Systems Research: ITSs Care, Precisel", Int. J. Artif. Intell. Educ,1990, vol. 10, pp. 350–364.

[6] M. K. Hashem; "*Learning automata based intelligent tutorial-like systems*", Ph.D. dissertation, School Comput. Sci., Carleton Univ., Ottawa, ON, Canada, 2007.

[7] R.S.Legaspi, R.C.Sison, "Modeling the tutor using reinforcement learning". in Proc. PCSC , 2000, pp. 194–196.

[8] P.Baffes, R.Mooney., "Refinement-based student modeling and automated bug library construction". J.Artif. Intell. educ, 1996, vol. 7, no. 1, pp. 75–116.

[9] R.Lelouche., "A Collection of pedagogical agents for intelligent educational systems in intelligent tutoring systems". 6th international Conference, ITS, Lecture Notes in computer Science, Springer-verlag, Berlin, 2000, pp. 143-152.

[10] Hashem. M. K , B. J. Oommen. "Modeling a Student-Classroom Interaction in a Tutorial-Like System Using Learning Automata," IEEE Trans.Syst., Man, Cybern. B, Cybern. Feb. 2010, vol.40, no.1.

[11] M.K.Hashem; Modeling a Student's Behavior in a Tutorial-*Like* System Using Learning Automata", Ph.D. dissertation, School Comput. Sci., Carleton Univ., Ottawa, ON, Canada, 2010.

[12] Hashem. M. K , B. J. Oommen. "Modeling the Learning process of the teacher in a Tutorial-Like System Using Learning Automata," IEEE Trans.Syst., Man, Cybern. B, Cybern, Dec. 2013, vol.43, no.6, pp. 2010-2031.

[13] S.Lakshmivarahan, "Learning Algorithms: Theory and Applications". Springer-Verlag: Berlin, Germany, 1981.

[14] K.Najim, A.S.Poznyak, "Learning Automata: Theory and Applications". Oxford, U.K 1994: Pergamon.

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

[16] M.A.L.Thathachar, P.S.Sastry, "Varieties of Learning Automata: An Overview", IEEE Trans. Syst., Man, Cybern. B, Cybern. 2007, vol. 32, no. 6, pp. 711–722.

[17] J.V. Neumann, "The theory of self-reproducing automata" Burks AW (ed), Univ. of Illinois Press, Urbana and London (1966).

[18] M.R.Meybodi., H.Beigy, " A Mathematical Framework for Cellular Learning Automata" Advance on Complex Syst. Sep./Dec 2004 Nos. 3-4.

[19] M.R.Meybodi., H.Beigy, M.Taherkhani, "Cellular Learning Automata and its Applications", Sharif Journal of Science and Technology, 2003,vol. 19, no.25, pp.54-77.

[20] B.J.Oommen., M.Agache, "Continuous and Discretized Pursuit Learning Schemes: Various Algorithms and Their Comparison", IEEE Trans. Syst., Man. Cyrern. Jun 2001 B, vol. 31, no. 3, pp. 277-278.

[21] J.L. Schiff, "Cellular Learning Automata: A Discrete View of the World, A John Wily & Sons, 2007,Inc., Publication1807.

Hajar Hajary received her B.Sc. in Software Engineering from Mobarakeh Azad University, Esfahan, Iran and her M.Sc. in the same field from Science and Research Branch Islamic Azad university , Tehran, Iran. Her research interests include cloud computing and intelligent systems.

Ali Ahmadi received his B.Sc. in Electrical Engineering from Amirkabir University,Tehran, Iran in 1991 and his M.Sc. and Ph.D. degrees in Artificial Intelligence and Soft Computing from Osaka Prefecture University, Japan in 2001 and 2004, respectively.He worked as a researcher in Research Center for Nanodevices and Systems in Hiroshima University, Japan from 2004 to 2007. He has been with K.N. Toosi University of Technology, Tehran, Iran as assistant professor from 2007. His research interests include distributed intelligent systems, human-computer interaction computational intelligence, adaptation and interactive learning models, virtual reality and artificial life.