

# A novel parallel framework for pursuit learning schemes

Hao Ge, Jianhua Li, Shenghong Li\*, Wen Jiang, Yifan Wang

Department of Electronic Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China

## ARTICLE INFO

### Keywords:

Learning automata  
Parallel framework  
Decentralized learning  
Centralized fusion  
Learning speed

## ABSTRACT

Parallel operation of learning automata (LA), which is proposed by Thathachar and Arvind, is a promising mechanism that can reduce the computational burden without compromising accuracy. However, as far as we know, this parallel mechanism has not been widely used due to two reasons: one is the fact that the environment can response to multi-actions simultaneously are few, the other is the relatively slow speed of the learning process.

In this paper, a novel parallel framework is presented to reduce the number of required interactions between the incorporated pursuit LA and the environment by introducing decentralized learning and centralized fusion. The philosophy is to learn various aspects of the problem at hand by taking advantage of the diverse exploration of decentralized learning and summarize the common knowledge learned by centralized fusion. Simulations are conducted to verify the effectiveness of our framework and demonstrate its outperforming. The proposed framework is further applied to the stochastic point location problem and obtains an attractive performance.

## 1. Introduction

Learning Automaton (LA), an important research area of Artificial Intelligence (AI), is a self-adaptive machine that can learn the optimal action from a random environment. LA was first investigated by Tsetlin to model the behavior of biological learning systems [1,2] and by now the study of LA has reached a relatively high level of maturity. Various successful applications utilizing LA have been reported in areas such as community detection [3], cooperative spectrum sensing [4], clustered wireless ad-hoc networks [5], tutorial-like systems [6,7], on-line event pattern tracking [8] and multi-class classification problems [9]. One intriguing property that popularize the learning automata based approaches in engineering is that LA can learn the stochastic characteristics of the external environment it interacts with, and maximize the long term reward it obtains through interacting with the environment. When the environment in which they operate provides noisy and incomplete information, LA's performance is significantly better than other methods.

In theoretical field, networks of LA are committed to solve the problems that are difficult for single LA to handle. By the synthesis of complex learning structures from simple learning automaton, networks demonstrate some new features.

The study of networks of LA was pioneered by Thathachar [10]. Through his efforts, systems consisting of several learning automaton such as hierarchical structure, games and parallel operation are constructed.

1. Hierarchical structure looks at larger aggregations of LA so that more complex learning problems can be handled. Such a system has several levels, each of whom is comprised of several LA. The automaton in upper level selects an action, which activates the corresponding automaton in the lower level. This procedure repeats from the top to the bottom. Thus, the hierarchy system has a tree like structure. Only those LA that at the bottom level (corresponds to leaf nodes in the tree) can interact with environment directly. Some new features are demonstrated by this new structure. Poznyak and Najim [11] showed that the use of hierarchical structure LA (HSLA) accelerates the learning process. And stochastic point location (SPL) can be solved by using hierarchical learning automata [12]. Meanwhile, [13] demonstrates the hierarchical structure can cope with problems of general non-stationary multi-teacher environment (NME).
2. "Games of LA" are multi-automata system constructed to overcome the high dimensionality of the decision space. In the case where the objective function has  $N$  variables, then the point where a maximum is attained would be a vector of  $N$  components. Using a single LA whose action set corresponds to points in  $R^N$  is unreasonable because the number of actions would be unacceptably large. It would be better to use one automaton for learning one component of the maximum point. These  $N$  LA constitute a multi-automata system, where each automaton would be viewed as a player involved in a game. In such a game, multiple automata are able to control a

\* Corresponding author.

E-mail addresses: [sjtu\\_gehao@sjtu.edu.cn](mailto:sjtu_gehao@sjtu.edu.cn) (H. Ge), [lijh888@sjtu.edu.cn](mailto:lijh888@sjtu.edu.cn) (J. Li), [shli@sjtu.edu.cn](mailto:shli@sjtu.edu.cn) (S. Li), [wenjiang@sjtu.edu.cn](mailto:wenjiang@sjtu.edu.cn) (W. Jiang), [wangyifan\\_1123@sjtu.edu.cn](mailto:wangyifan_1123@sjtu.edu.cn) (Y. Wang).

<http://dx.doi.org/10.1016/j.neucom.2016.09.082>

Received 18 February 2016; Received in revised form 20 July 2016; Accepted 3 September 2016

Available online xxxx

0925-2312/ © 2016 Elsevier B.V. All rights reserved.

finite Markov chain with unknown transition probabilities and rewards [14]. The collective wisdom of these  $N$  LA are utilized to locate the optimum point in solution space  $R^N$ . In pattern recognition field, [15] shows an example that  $N$  LA constitute a common payoff game to learn the underlying separating hyperplane, and each LA corresponds to one dimension of the hyperplane.

3. Parallel operation of LA [10] is presented with the objective of improving the speed of convergence via utilizing the parallel nature of the environment. The learning process of single LA can be viewed as essentially sequential, at a time only one action is selected and one feedback is elicited from the environment. In some cases where the environment could response to multi-actions simultaneously, several actions could be sent collectively as an input and all feedback signals can be used collectively to update the action probability. The philosophy of this parallel mechanism is to trade space for time.

However, unfortunately, despite the parallel operation looks promising, there are few literatures that exist in the field of LA for parallel applications. The reason is two-fold: 1) The classic parallel framework does not intend to decrease the number of interactions with environment, but only to reduce computational burden. Computation power is no longer the bottleneck for most scenarios nowadays, but getting a response from the environment may be time-consuming or energy-consuming sometimes. So a framework that can reduce the required number of interactions is desired. 2) In practical applications, the environment could response to multi-actions simultaneously are few.

In this paper, a novel parallel framework is presented. The parallel operation is divided into two steps, decentralized learning and centralized fusion. Several numerical simulations are also carried out to verify the effectiveness of the proposed framework. The results demonstrate this new framework can outperform the classic one, and reduce the number of interactions with environment. The contributions of this work are highlighted as follows:

1. Classic parallel framework is presented with the objective of reducing computational burden. The total number of interactions required does not vary with parallel scale. While the goal of our framework is to learn in parallel more efficiently, i.e., requires fewer interactions with the environment.
2. Two novel mechanisms are employed in our framework. One is the decentralized learning, which can take advantage of diverse exploration to learn different aspects of the problem at hand. The other is centralized fusion which will summarize the learned information into common knowledge.
3. The  $\varepsilon$ -optimality of the proposed framework is derived and simulation results demonstrate its superiority to the classic one.

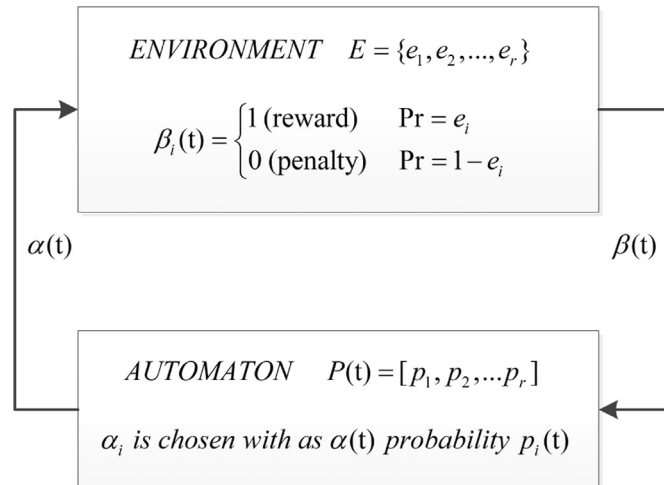


Fig. 1. Block diagram of a learning automaton.

## 2. Related works

### 2.1. Deterministic estimator based LA

In the history of single LA, various approaches have been proposed to speed up the learning process, among which discretization [16] and estimation [17] are two epoch-making concepts. The former is implemented by restricting the probability of choosing an action within a finite number of values in the interval  $(0, 1)$ , and the latter are modules that gather history information to estimate the reward probability of each possible action, in order to update action probability vector purposefully.

Deterministic estimator based LA, such as  $DP_{ri}$  [18], DGPA [19] and the newly presented LELA [20], DGCPA [21], are the major family of LA.  $SE_{ri}$  [22], a very fast LA scheme, has an extra tunable parameter to control the randomness imposed to the deterministic estimates. Its training stage need to traverse a 2-dimensional parameter space. After training, this extra parameter carries extra information about the environment. It is not fair to compare it with deterministic estimator based LA. Hence we only take deterministic estimator based LA into consideration in this paper.

As pursuit schemes are the most fundamental one of deterministic estimator based LA, we take the classic  $DP_{ri}$  as an example to describe the main features of a learning automaton. A block diagram depicting the automaton – environment interaction is shown in Fig. 1.

Environment, the aggregate of external influences of learning process, can be depicted as  $\{A, B, E\}$ . Automaton, the learning module, is defined as  $\{A, B, P, T, D\}$ . Among them,  $A$  and  $B$  are interaction information exchanged between automaton and the environment.

$A = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ , a finite set of  $r$  actions.  $\alpha(t) \in A$  is the output of the automaton and the input of the environment at time  $t$ .

$B$  is a feedback set.  $\beta(t) \in B$  denotes the reaction from the environment at time  $t$ . If  $B$  is a binary output set, e.g.  $\{0, 1\}$ , the environment is referred to as a P-model environment. All the schemes discussed within this paper are restricted to interacting with a P-model stationary environment.

$E = \{e_1, e_2, \dots, e_r\}$  is the set of reward probabilities. The feedback in response to each action  $\alpha_i$  is modeled as a Bernoulli distribution over  $B$ , that is  $e_i = \text{Prob}[\beta(t) = 1 | \alpha(t) = \alpha_i]$ . The challenge of the learning problem is that the reward probabilities are unknown to the automaton. The only information that can be utilized by the automaton is the stochastic reinforcement signal in response to each action choice made.

$P(t) = [p_1(t), p_2(t), \dots, p_r(t)]$  is the action probability vector, where  $p_i(t) = \text{Prob}[\alpha(t) = \alpha_i | P(t)]$ ,  $i = 1, \dots, r$ .

$D(t) = [d_1(t), d_2(t), \dots, d_r(t)]$  is the deterministic estimator vector.  $d_i(t)$  is the current deterministic estimates of  $e_i$  and is calculated as formula (1)<sup>1</sup>. Where  $Z_i(t)$  is the number of times action  $\alpha_i$  was selected up to  $t$ , and  $W_i(t)$  is the number of times action  $\alpha_i$  was rewarded during the same period.

$$d_i(t) = \frac{W_i(t)}{Z_i(t)}, \forall i \in \{1, \dots, r\} \quad (1)$$

$T$  is the updating rule so that  $P(t+1) = T(P(t), \cdot)$ . During a cycle, LA chooses an action  $\alpha(t)$  and then receives a stochastic response  $\beta(t)$  from the environment.

According to pursuit scheme with reward-inaction philosophy, the updating rule  $T$  is: If  $\beta(t) = 1$  then

$$p_j(t+1) = \max\{p_j(t) - \Delta, 0\}, \forall j \neq m \quad (2)$$

$$p_m(t+1) = 1 - \sum_{j \neq m} p_j(t+1) \quad (3)$$

<sup>1</sup> This kind of estimator is called Maximum Likelihood Estimator (MLE), there are also other kinds of estimators, such as Confidence Interval Estimator (CIE) proposed in [21].

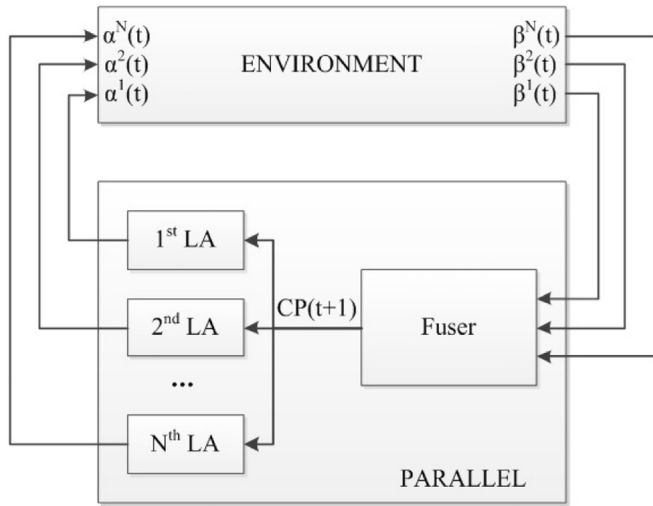


Fig. 2. Parallel framework for modules of LA.

where  $m = \arg\max \{d_i(t)\}$ .

Else,

$$p_j(t+1) = p_j(t), j = 1, \dots, r \quad (4)$$

## 2.2. Classic parallel framework

Thathachar and Arvind presented a parallel version of estimator based LA [10], which was intended to reduce computational burden. Assuming the parallel system consists of  $N$  identical LA, they have a common action set  $A = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ , a common probability vector  $CP(t) = [cp_1(t), cp_2(t), \dots, cp_r(t)]$  and a shared estimator vector  $D(t)$ . Fig. 2 shows the paradigm of the framework, and the procedure can be summarized briefly as follows.

**Step 1** Each LA, say  $m^{th}$  LA, selects an action  $\alpha^m(t)$  based on the  $CP(t)$  independently of all other LA and obtains its own reinforcement signal  $\beta^m(t)$ .

**Step 2** A fuser combines all the selected actions and feedbacks to update the common probability,  $cp_i(t+1) = cp_i(t) + \lambda [R_i(t) - cp_i(t) \sum_{i=1}^N R_i(t)]$ .

**Step 3** Update the common estimator

$$W_i(t+1) = W_i(t) + R_i(t) \quad (5)$$

$$Z_i(t+1) = Z_i(t) + C_i(t) \quad (6)$$

where  $R_i(t) = \sum_{j=1}^N \beta^j(t) \cdot I\{\alpha^j(t) = \alpha_i\}$  and  $C_i(t) = \sum_{j=1}^N I\{\alpha^j(t) = \alpha_i\}$ ;

**Step 4** If  $\max\{CP(t+1)\} = 1$  then converge. Else go to **Step 1**.

However, the parallel framework has not been widely used in practical areas because LA is not act as the learning module but action selector. The task of LA is just choosing the action based on the  $CP(t)$  and the job of update is done by the fuser. In [19] the authors pointed out that the question of studying parallel ensembles of LA schemes remains open.

## 3. A novel parallel framework

As pointed out previously, the slow speed of learning is one of the most important problems need to be addressed. In the context of original parallel framework, the parallel learning automata can reduce the convergence time roughly by a factor of  $N$ , which is the paralleliza-

tion scale. The reason is quite intuitive: multiple actions get feedbacks from the environment simultaneously during one iteration and probability vector are updated based on these multiple samples. However, this framework is not designed to decrease the number of interactions with environment, but to save computational power. In this paper, the key point is to reduce the number of interactions required with the environment through parallel operation of these LA.

In [23], cooperative team of LA has been proved to be effective on combining several single LA as a system, leading to a faster convergence than single LA. The rationale behind this is that, if the learning procedure of the single LA members in the team is not strongly correlated, individual learners can learn different aspect of the problem. The common things shared by diverse agents can reinforce the learning outcome. Inspired by this, we expect this parallel system has properties such as learning agents are performed independently, and different learning preferences are integrated appropriately. Therefore, the novel parallel system is proposed. The above two techniques are named as decentralized learning and centralized fusion. The details are introduced below.

### 3.1. Algorithm description

#### Parameters

$n$	resolution parameter
$\Delta=1/rn$	smallest step size
$W_i^m(t)$	in the module of $m^{th}$ LA, the number of times that the action $i$ has been rewarded up to time instant $t$ , for $i = 1, 2, \dots, r$ , $m = 1, 2, \dots, N$
$Z_i^m(t)$	in the module of $m^{th}$ LA, the number of times that the action $i$ has been selected up to time instant $t$ , for $i = 1, 2, \dots, r$ , $m = 1, 2, \dots, N$
$d_i^m(t)$	the estimate value of action $i$ in the $m^{th}$ LA at time instant $t$ . $d_i^m(t) = W_i^m(t)/Z_i^m(t)$
$D^m(t)$	$D^m(t) = [d_1^m(t), d_2^m(t), \dots, d_r^m(t)]$ is the estimator vector of the $m^{th}$ LA

#### Method

Initialize  $W_i^m(0) = 1$  and  $Z_i^m(0) = 1$  and set  $p_i^m(0) = 1/r$ , for  $i = 1, 2, \dots, r$ ,  $m = 1, 2, \dots, N$

#### Repeat

- Step 1** At time instant  $t$ , a set of actions  $\vec{\alpha} = \{\alpha^1(t), \alpha^2(t), \dots, \alpha^N(t)\}$  is selected by  $N$  LA members according to its probability vector  $P(t)$  independently.
- Step 2** Receive  $\vec{\beta} = \{\beta^1(t), \beta^2(t), \dots, \beta^N(t)\}$  from the environment as the response set.
- Step 3** Each LA updates its temporary vector  $P^m(t+1)$  based on its updating rule  $T$ , in a decentralized learning manner,  $P^m(t+1) = T(P(t), D(t), \alpha^m(t), \beta^m(t))$ .
- Step 4** The fuser collects the temporary vector of these components and calculates the fusion result. This is the centralized fusion,  $CP(t+1) = F(P^m(t+1))$ ,  $m = 1, 2, \dots, N$  is assigned to LA members as the probability vector for the next instant.
- Step 5** Using the action and feedback component  $\alpha^m(t)$ ,  $\beta^m(t)$  to update  $W^m(t)$  and  $Z^m(t)$

#### Until

This process terminates as one element of the fusion result is equal to one.

It is worth noting that two parts, decentralized learning (step 1 to

step 3 and step 5) and the centralized fusion (step 4) are involved in the learning process. The decentralized learning process, which uses the trial and error information to refine the probability vector, is implemented by deterministic estimator based LA independently. In centralized fusion process, the fuser function combines the updated probability vectors together into the common probability. The strategies of fusion are manifold, taking average is the most general one. In the proposed framework, we take averaging strategy as the default operation in centralized fusion stage. For every component in the common vector, averaging strategy is taking the average over all LA's temporary probability value at that position. That is

$$cp_i(t+1) = \frac{1}{N} \sum_{m=1}^N p_i^m(t+1) \quad (7)$$

Besides, the initialization of estimator is different from the existing framework. The technique of *optimistic initial values* is applied, which has been reported as a simple trick that can be quite effective on stationary problems. As clarified in [24], initial action values can be used as a simple way of encouraging exploration. In our framework,  $w^m(0)$  and  $Z^m(0)$  are initialized identically as 1.

### 3.2. Analysis of $\epsilon$ -optimal

Assuming the fusion strategy is taking average. In this section, we shall show that the proposed parallel framework with  $DP_{r,i}$  scheme incorporated can be proved to be  $\epsilon$ -optimal. The proposed framework with other pursuit learning schemes incorporated can be easily derived by the same way.

The updating rules of  $DP_{r,i}$  are described in Section 2. Notice that the  $N$  LA in the proposed parallel framework share a common probability vector  $CP(t)$ .

**Theorem 1.** For each action  $\alpha_i$ , if  $cp_i(0) \neq 0$ . Then for any given constants  $\delta > 0$  and  $M > 0$ , there exist  $n_0 < \infty$  and  $T_0 < \infty$  such that under the proposed parallel framework, for all learning parameters  $n > n_0$  and  $t > T_0$ ,  $Pr\{\text{each action chosen more than } M \text{ times at time } t\} \geq 1 - \delta$ .

**Proof.** A similar proof was given by Thathachar and Sastry in [25,26]. Then it was extended for the case of discretized learning automata by Oommen and Lanctot in [18]. Additional explanation is necessary to demonstrate the two upper bounds in the existing proof.

At time  $t$ ,

$$Pr\{\alpha_i \text{ is not chosen}\} \leq 1 - cp_i(0) + t\Delta \quad (8)$$

The left hand side expression is the situation that action  $\alpha_i$  is decreased every time in the first  $t$  iterations and in all the LA modules. Based on the averaging strategy in the fuser,  $cp_i(t) = cp_i(0) - t\Delta$ , then (8) holds.

During any of the first  $t$  iterations of the scheme, it is obvious that,

$$Pr\{\alpha_i \text{ is chosen}\} \leq 1 \quad (9)$$

$$Pr\{\alpha_i \text{ is not chosen}\} \leq 1 - cp_i(0) + t\Delta \quad (10)$$

The rest of the proof remains the same as Theorem 2.2 proved in [18].  $\square$

**Theorem 2.** Suppose there exists an index  $f$  and a time instant  $T_1 < \infty$  such that  $d_f^m(t) > d_i^m(t)$  for all  $i \neq f$ ,  $m = 1, 2, \dots, N$  and all  $t \geq T_1$ . Then there exists an integer  $n_0$  such that for all learning parameter  $n > n_0$ ,  $cp_f(t) \rightarrow 1$  with probability one as  $t \rightarrow \infty$ .

**Proof.** We shall illustrate that the sequence of random variable  $\{cp_f(t)\}_{t \geq 0}$  is a sub-martingale within this framework. The convergence will be reached from the sub-martingale convergence theorem.

The condition  $d_f^m(t) > d_i^m(t)$  for all  $i \neq f$  and  $m = 1, 2, \dots, N$  means that for all LA, the  $f^{th}$  action receive the highest estimate. No matter

what action the LA chooses, if a reward reinforcement signal is received from the environment,  $f^{th}$  action will gain its probability value based on updating Eqs. (2) and (3).

Define  $\delta(R(t)) = cp_f(t+1) - cp_f(t)$  as the increment of  $cp_f(t)$  under the condition that  $R(t)$  out of  $N$  LA receive reward from the environment.

$$cp_f(t+1) = \frac{1}{N} [R(t) \cdot (1 - \sum_{i \neq f} (cp_i(t) - \max\{cp_i(t) - \Delta, 0\})) + (N - R(t)) \cdot cp_f(t)] \quad (11)$$

$$\delta(R(t)) = \frac{R(t)}{N} \cdot \sum_{i \neq f} (cp_i(t) - \max\{cp_i(t) - \Delta, 0\}) \quad (12)$$

It is obvious that  $cp_i(t) - \max\{cp_i(t) - \Delta, 0\} \geq 0$  and  $\delta(R(t)) \geq 0$ .

Thus,  $E[cp_f(t+1) - cp_f(t) | Q(t)] \geq 0$  is a sub-martingale, where  $Q(t) = \langle p(t), \hat{d}(t) \rangle$  is the state vector of the estimator.

According to the sub-martingale convergence theorem [27],

$$E[cp_f(t+1) - cp_f(t) | Q(t)] \rightarrow 0 \quad w.p.1 \quad (13)$$

Hence  $cp_f(t) \rightarrow 1$  w.p.1 and this theorem is proved.  $\square$

Last, the final proof of  $\epsilon$ -optimal has no restriction in updating rules, therefore it can be deduced from Theorems 1,2 above proved and Theorem 3 proved in [16].

## 4. Experiments and evaluations

### 4.1. Simulation results

In this section, the proposed novel parallel framework is compared with the classic parallel framework, two pursuit LA schemes  $DP_{RL}$ , DGPA are incorporated in these two frameworks respectively. All of them have been proved to be  $\epsilon$ -optimal. Estimators of the LA that incorporated in the classic framework are initialized by sampling each actions ten times. The extra iterations imposed by initialization are also included in the results.

Two ten-action ( $r=10$ ) problems  $E_A, E_B$ , which are the well-known benchmark environments used in [16], are considered for simulation studies. The reward probabilities for the actions of the two benchmark environments are:

$$E_A = \{0.70, 0.50, 0.30, 0.20, 0.40, 0.50, 0.40, 0.30, 0.50, 0.20\}; E_B = \{0.10, 0.45, 0.84, 0.76, 0.20, 0.40, 0.60, 0.70, 0.50, 0.30\};$$

In all the simulations performed, the execution of an algorithm is considered to be converged if the probability of choosing an action is greater or equal to a threshold  $T$  ( $0 < T < 1$ ). If the converged action is the one with the highest value of being rewarded, this system was considered to have converged correctly.

Note that the comparisons among different schemes with different frameworks are considered to be fair, because that the best parameters for each environment of their own are used to evaluate the performance. The “best” parameters of a scheme are defined as the values that yielded the fastest convergence speed and guaranteed the system converged to the correct action in a sequence of  $NE$  experiments always. Specifically, in our experiment, simulations were performed with the same threshold  $T$  and the number of experiments  $NE$  in [22], that is,  $T=0.999$  and  $NE=750$ . After the tuning of “best” parameters, 250000 experiments were carried out with the tuned “best” parameter to evaluate the average convergence speed and accuracy.

Indicators for comparison are defined below. Accuracy is calculated as *correctly convergence*/ $NE$ . We also compare our novel parallel framework with classic parallel and single operated LA. The improvement is obtained by calculating:

$$(\text{Iteration}_{\{\text{classic parallel}\}} - \text{Iteration}_{\{\text{novel parallel}\}}) / \text{Iteration}_{\{\text{classic parallel}\}} \quad (14)$$



$$(\text{Iteration}_{\{\text{single}\}} - N * \text{Iteration}_{\{\text{novel parallel}\}}) / \text{Iteration}_{\{\text{single}\}} \quad (15)$$

It should be noted that the best resolutions for some cases are *decimals*, because it apparently will be a bottleneck if we restrict resolution parameters to integers inasmuch as we defined the smallest step size as  $\Delta = 1/r/n$ .

#### 4.2. Performance analysis

From the above simulation results shown in Tables 1, 2, we may draw the following conclusions.

Firstly, the novel parallel system is superior to the classic system with respect to number of iterations. Notice that there are  $N$  interactions taken place within one iteration. The highest improvement is up to 67.38%. For example, in  $E_A$  when  $DP_{ri}$  is incorporated in parallel framework and the parallel scale is 3, classic framework needs 328 iterations while the novel framework needs only 107 iterations. What we want to clarify here is that when parallel scale is 1, the proposed framework is outperformed by original one, but technically speaking, manipulating one LA every iteration cannot be deemed to be “parallel operation”.

Secondly, the novel parallel system need less communications with the environment. For example, in  $E_B$ , three DGPA incorporated in the novel parallel framework need 111 iterations to get converged. Hence the total number of interactions with environment is  $3 \times 111 = 333$ , which is 55.6% faster than the single operated DGPA. The improvement is significant, less iterations mean less number of interactions and less computation time, which is especially useful in environments where interacting with environments could be expensive, such as drug trials, destructive tests or financial investments.

Finally, we want to point out an interesting phenomenon that the improvement is not monotone increasing with the parallel scale, and experiments shows that three automata in parallel always yielded fastest convergence. Three DGPA in parallel operating in  $E_A$  is even faster than the state-of-art scheme [21].

#### 4.3. Application in stochastic point location

Stochastic point location (SPL) problem, first investigated in [28], is the problem that an entity, guided by the environment information, attempts to locate a particular point. The problem has been extensively investigated in literatures [29–33] and the procedure proposed in [32] can be introduced briefly as:

- (1) At first, we divide the  $\Delta_k$  space into  $d$  subintervals  $\Delta_k = [\Delta_k^1, \Delta_k^2, \dots, \Delta_k^d]$ ;
- (2) Each subinterval  $\Delta_k^i$  has an agent (which usually implemented by a LA), all the agents are working independently to communicate with

**Table 1**

Average convergence speed and accuracy over 250000 experiments of the discretized pursuit reward-inaction ( $DP_{ri}$ ) scheme working in different parallel frameworks.

$E_A$	Classic parallel			Novel parallel			Improvement	
	Param.	Itera.	Acc.	Param.	Itera.	Acc.	Classic	Single
1	216	783	0.995	176	743	0.995	5.1%	5.1%
2	108	442	0.995	14	207	0.995	53.17%	47.13%
3	72	328	0.995	1.1	107	0.995	67.38%	59%
4	54	273	0.995	0.5	136	0.999	50.18%	30.52%
$E_B$								
1	881	2364	0.995	900	2457	0.995	–3.93%	–3.93%
2	440	1161	0.995	171	858	0.995	26.1%	27.41%
3	294	789	0.995	56	431	0.995	45.37%	45.3%
4	220	578	0.995	19	414	0.995	28.37%	29.95%

**Table 2**

Average convergence speed and accuracy over 250000 experiments of the discretized generalized pursuit scheme (DGPA) working in different parallel frameworks.

$E_A$	Classic parallel			Novel parallel			Improvement	
	Param.	Itera.	Acc.	Param.	Itera.	Acc.	Classic	Single
1	28	753	0.997	38	920	0.997	–22.18%	–22.18%
2	10	331	0.997	9	299	0.997	9.67%	20.58%
3	5	250	0.997	0.3	111	0.997	55.6%	55.78%
4	3	169	0.997	0.2	140	1	17.16%	25.63%
$E_B$								
1	55	1445	0.997	58	1461	0.997	–1.1%	–1.1%
2	26	727	0.997	15	514	0.997	29.3%	28.86%
3	18	497	0.997	5.2	308	0.997	38.03%	36.06%
4	12	385	0.997	2	268	0.997	30.39%	25.81%

its own environment;

- (3) In each  $M$  iteration, if the agent converges to the left/right side, the learning algorithm concludes the target point is at the left/right of this interval. If the agent is not converging, the learning algorithm concludes the target point is inside the interval;
- (4) Based on these decision outputs, a new search interval is chosen. Then goto Step 1). The procedure repeats until the search interval is small enough.

The learning agent in each subinterval has three outcomes, i.e. converged to left, converged to right and not converged. Because they may not converged, a key step in the location problem is how to choose a proper  $M$  value. If  $M$  is too small, the learning algorithm may not interact with the environment fully. In this situation, the information to choose new search interval in Step 3) is inadequate and will lead to wrong direction. If  $M$  is too large, which means more time is spent in the learning process, thus reduce the flexibility of the system.

Then, to address this issue in point location problem, we can apply parallel framework in the learning step to accelerate the learning phase. We adopted notations and definitions identical to the ones in [32].

Environment set: Original search space  $\Delta_1 = [0, 1]$ , the target point  $\lambda^* = 0.9123$  and environment is informative with probability 0.9. For more details about the environment, please refer to [29].

Learning algorithm set: number of divided subintervals  $d=3$ , prefixed resolution of the estimation is  $\varepsilon=0.005$ , learning strategy is  $DP_{ri}$ , the resolution is the best parameter in [32]  $N=147$ .

Figs. 4, 5 and 6 show the converging progression of the different learning system with different  $M$  values, one is the single operated LA, second is two LA incorporated in the original parallel scheme and the third is two LA incorporated in the novel parallel framework. As mentioned before, due to insufficiently large  $M$ , the learning algorithm may not interact with the Fig. 3 environment fully, that will cause the agent can't converge to the correct action.

In Fig. 4, when  $M=400$ , the agent traces the point correctly, however, if we reduce the learning phase length  $M$ , the agent communicates inadequately with the environment. In Fig. 5, the situation is better. The feature of original parallel system is that it can't reduce the required iterations. The figure is clear that one LA need  $M=400$  iterations to converge while two LA need merely  $M=200$  iterations. It can be seen from Fig. 6 that when  $M=400/200/100$  LA converge to the correct point, which means this novel parallel framework need less communication with the environment. From this example, it is evident that the novel parallel framework can not only reduce the computation time, but also save the interaction expense. In fact, this novel parallel system can replace single LA in all practical fields or realistic applications, once.

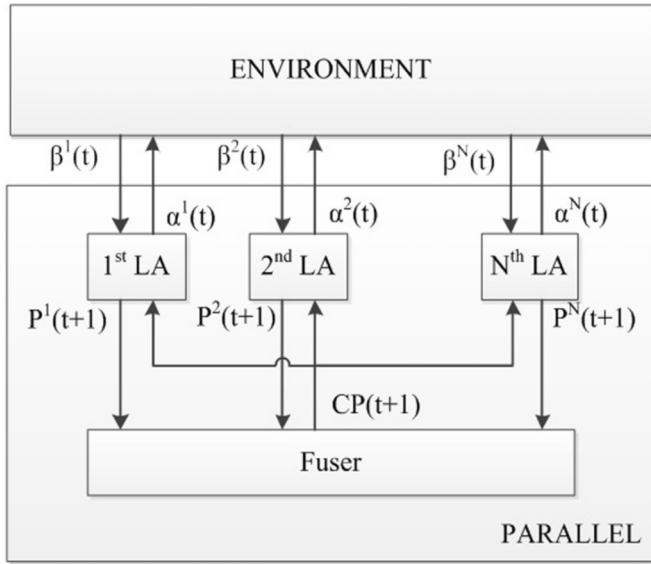


Fig. 3. Proposed structure of generalized parallel framework.

## 5. Conclusion

In this paper, we introduced a novel parallel framework for pursuit learning automata. By importing decentralized learning and centralized fusion, this framework can outperform the classic one. The philosophy of this paper is that decentralized learning can learn different aspect of the problem, and centralized fusion can reinforce the common learned knowledge. This novel parallel framework is better than the original one because it can save the communication expense with the environment. Simulation experiments have confirmed that the efficiency of this novel parallel system. Stochastic point location problem is a successful application of the proposed parallel framework. The question of studying a generalized parallel system which is suitable for all kinds of learning automata remains open. A parallel scheme that can unleash the full potential of large parallel scale operation deserves to be investigated further.

## Acknowledgment

This research work is funded by the National Science Foundation of China (61271316), Key Laboratory for Shanghai Integrated Information

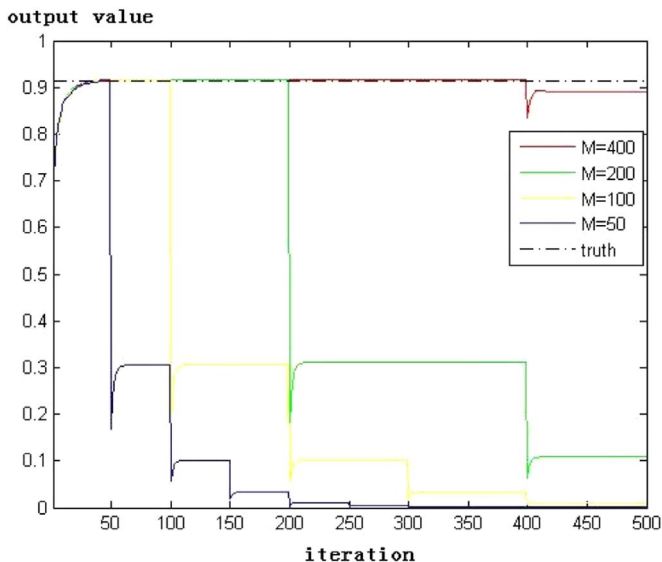


Fig. 4. Average Convergence process of the single  $DP_{ri}$  LA strategy with different  $M$  values in 750 experiments.

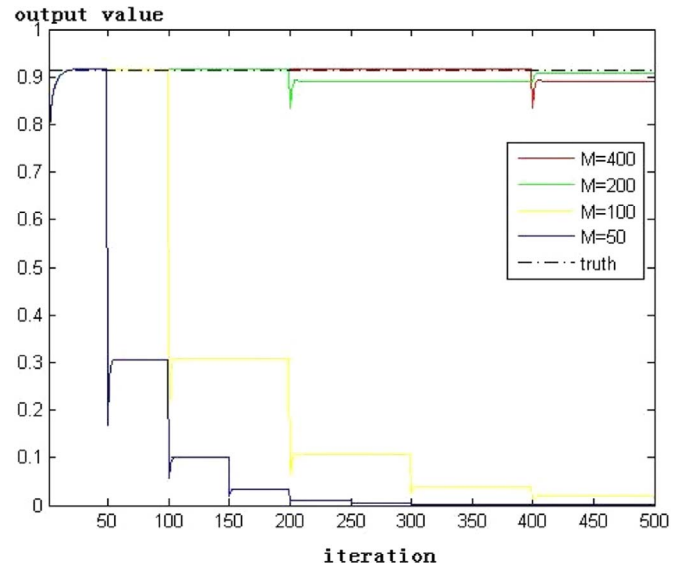


Fig. 5. Average Convergence process of the original parallel system using two  $DP_{ri}$  LA strategy with different  $M$  values in 750 experiments.

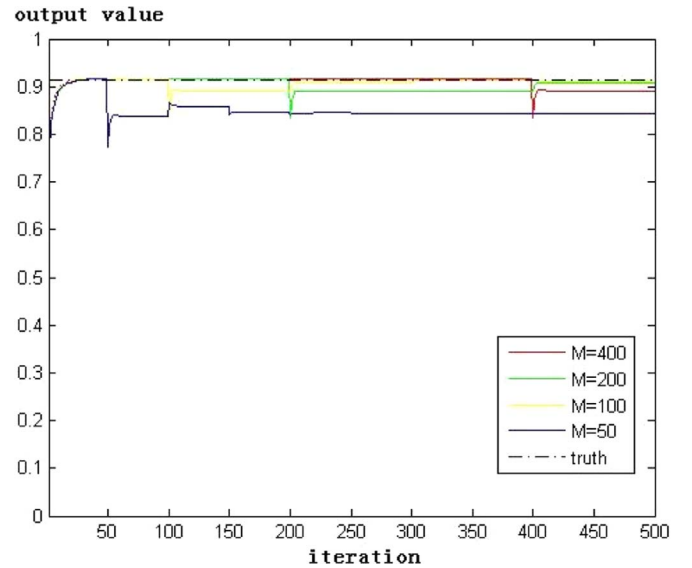


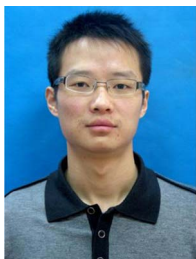
Fig. 6. Average Convergence process of the novel parallel system using two  $DP_{ri}$  LA strategy with different  $M$  values in 750 experiments.

Security Management Technology Research, and Chinese National Engineering Laboratory for Information Content Analysis Technology.

## References

- [1] M.L. Tsetlin, On the behavior of finite automata in random media, *Avtom. Telemekhanika* 22 (1961) 1345–1354.
- [2] M. Tsetlin, *Automaton Theory and Modeling of Biological Systems*, Academic Press, 1973.
- [3] Y. Zhao, W. Jiang, S. Li, Y. Ma, G. Su, X. Lin, A cellular learning automata based algorithm for detecting community structure in complex networks, *Neurocomputing* 151 (3) (2015) 1216–1226. <http://dx.doi.org/10.1016/j.neucom.2014.04.087>.
- [4] W. Yuan, H. Leung, W. Cheng, S. Chen, Optimizing voting rule for cooperative spectrum sensing through learning automata, *IEEE Trans. Veh. Technol.* 60 (7) (2011) 3253–3264.
- [5] J.A. Torkestani, Laap: a learning automata-based adaptive polling scheme for clustered wireless ad-hoc networks, *Wirel. Pers. Commun.* 69 (2) (2013) 841–855.
- [6] B.J. Oommen, M.K. Hashem, Modeling the learning process of the teacher in a tutorial-like system using learning automata, *IEEE Trans. Cybern.* 43 (6) (2013) 2020–2031.
- [7] H. Ge, Y. Wang, S. Li, C.L.P. Chen, Y. Guo, A cooperative framework of learning automata and its application in tutorial-like system, *Neurocomputing* 188 (2016) 1–10.

- 311–318. <http://dx.doi.org/10.1016/j.neucom.2015.04.122>.
- [8] W. Jiang, C.-L. Zhao, S.-H. Li, L. Chen, A new learning automata based approach for online tracking of event patterns, *Neurocomputing* 137 (2014) 205–211. <http://dx.doi.org/10.1016/j.neucom.2013.08.047>.
- [9] S. Afshar, M. Mosleh, M. Kheyranfar, Presenting a new multiclass classifier based on learning automata, *Neurocomputing* 104 (2013) 97–104. <http://dx.doi.org/10.1016/j.neucom.2012.10.005>.
- [10] M.A. Thathachar, P.S. Sastry, *Networks of Learning Automata: techniques for Online Stochastic Optimization*, Springer Science & \$2 Business Media, 2011.
- [11] A.S. Poznyak, K. Najim, *Learning Automata and Stochastic Optimization*, Springer-Verlag Ltd, London, 1997.
- [12] A. Yazidi, O.-C. Granmo, B.J. Oommen, M. Goodwin, A novel strategy for solving the stochastic point location problem using a hierarchical searching scheme, *IEEE Trans. Cybern.* 44 (11) (2014) 2202–2220.
- [13] N. Baba, Y. Mogami, A relative reward-strength algorithm for the hierarchical structure learning automata operating in the general nonstationary multitask environment, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 36 (4) (2006) 781–794.
- [14] P. Vrancx, K. Verbeeck, A. Nowé, Decentralized learning in markov games, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 38 (4) (2008) 976–981.
- [15] P. Sastry, G. Nagendra, N. Manwani, A team of continuous-action learning automata for noise-tolerant learning of half-spaces, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 40 (1) (2010) 19–28.
- [16] B.J. Oommen, M. Agache, Continuous and discretized pursuit learning schemes: various algorithms and their comparison, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 31 (3) (2001) 277–287.
- [17] M. Thathachar, P.S. Sastry, Varieties of learning automata: an overview, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 32 (6) (2002) 711–722.
- [18] B.J. Oommen, J.K. Lancôt, Discretized pursuit learning automata, *IEEE Trans. Syst. Man Cybern.* 20 (4) (1990) 931–938.
- [19] M. Agache, B.J. Oommen, Generalized pursuit learning schemes: new families of continuous and discretized learning automata, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 32 (6) (2002) 738–749.
- [20] J. Zhang, C. Wang, M. Zhou, Last-position elimination-based learning automata, *IEEE Trans. Cybern.* 44 (12) (2014) 2484–2492.
- [21] H. Ge, W. Jiang, S. Li, J. Li, Y. Wang, Y. Jing, A novel estimator based learning automata algorithm, *Appl. Intell.* 42 (2) (2015) 262–275.
- [22] G.I. Papadimitriou, M. Sklira, A.S. Pomportsis, A new class of  $\epsilon$ -optimal learning automata, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 34 (1) (2004) 246–254.
- [23] Y. Wang, W. Jiang, Y. Ma, H. Ge, Y. Jing, *Learning Automata Based Cooperative Student-Team in Tutorial-Like System*, Springer International Publishing, Cham, 2014, pp. 154–161. [http://dx.doi.org/10.1007/978-3-319-09339-0\\_16](http://dx.doi.org/10.1007/978-3-319-09339-0_16).
- [24] R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*, 2nd Edition, Unpublished results.
- [25] P. Sastry, *Systems of learning automata: Estimator algorithms applications*, (Ph.D. thesis), Dept of Electrical Engineering, Indian Institute of Science, Bangalore, India, 1985.
- [26] M.A.L. Thathachar, P.S. Sastry, A new approach to the design of reinforcement schemes for learning automata, *IEEE Trans. Syst. Man Cybern.* SMC 15 (1) (1985) 168–175. <http://dx.doi.org/10.1109/TSMC.1985.6313407>.
- [27] S. Karlin, H. Taylor, *A First Course in Stochastic Processes*, Elsevier Science, 2012.
- [28] B.J. Oommen, Stochastic searching on the line and its applications to parameter learning in nonlinear optimization, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 27 (4) (1997) 733–739.
- [29] B.J. Oommen, G. Raghunath, Automata learning and intelligent tertiary searching for stochastic point location, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 28 (6) (1998) 947–954.
- [30] B.J. Oommen, G. Raghunath, B. Kuipers, Parameter learning from stochastic teachers and stochastic compulsive liars, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 36 (4) (2006) 820–834.
- [31] B.J. Oommen, S.-W. Kim, M.T. Samuel, O.-C. Granmo, A solution to the stochastic point location problem in metalevel nonstationary environments, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 38 (2) (2008) 466–476.
- [32] D.-S. Huang, W. Jiang, A general cpl-ads methodology for fixing dynamic parameters in dual environments, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 42 (5) (2012) 1489–1500.
- [33] W. Jiang, D.S. Huang, S. Li, Random walk-based solution to triple level stochastic point location problem, *IEEE Trans. Cybern.* 46 (6) (2016) 1438–1451. <http://dx.doi.org/10.1109/TCYB.2015.2446198>.



**Hao Ge** received B.E. degree in School of Information Science and Engineering from Southeast University, Nanjing, China in 2010. He is currently working toward the Ph.D. degree with School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include learning automata and their applications, artificial neural network, computer communication network and data mining.



**Jianhua Li**, a professor and doctoral supervisor, executive sub decanal of School of Information Security Engineering at Shanghai Jiao Tong University, is the director of Shanghai Key Laboratory of Integrated Administration Technology for Information Security and Network Information Security Management and Service Engineering Research Center of the State Ministry of Education. He received the B.S., the M.S. and the Ph.D. degrees in electronic engineering from Shanghai Jiaotong University, China, in 1986, 1991 and 1998, respectively. Since 2000, he has become chief expert and management expert of Information Security Technology Panel in National Project 863 during the 10th Five-Year Plan Period, member of National E-government Pilot Demonstration Project Total Panel, expert of China E-government Standardization Total Group, member of National Information Security Standardization Expert Committee, consultant of the Administration for the Protection of State Secrets, leader and chief expert of Total Group in National Information Security Application Demonstration Project(S219) Stage II. He was selected as one of the 1st Shanghai IT Top 10 New-Sharp Youth and offered special allowance for outstanding young experts by the State Council in 2002. He was also selected to be member of the 1st group of New Century National Hundred, Thousand and Ten Thousand Talent Project Level I & \$2 II, Shanghai Excellent Academic leader and Shanghai Youth Science and Technology Elite in 2004. Besides, he was awarded the honor of Shanghai Pace-setter in the New Long March in 1995.



**Shenghong Li** received the B.S. and the M.S. degrees in electrical engineering from Jilin University of Technology, China, in 1993 and 1996 respectively, and received the Ph.D. degree in radio engineering from Beijing University of Posts and Telecommunications, China, in 1999. Since Sept.1999, he has been working in Shanghai Jiaotong University, China, as research fellow, associate professor and professor, successively. In 2010, he worked as visiting scholar in Nanyang Technological University, Singapore. His research interests include information security, signal and information processing, artificial intelligence. He published more than 80 papers, co-authored four books, and holds ten granted patents. In 2003, he received the 1st Prize of Shanghai Science and Technology Progress in China. In 2006 and 2007, he was elected for New century talent of Chinese Education Ministry and Shanghai dawn scholar.



**Wen Jiang** received the B.Sc. and M.S. degree in automation from Henan University, Kaifeng, China, in 2009 and in pattern recognition and intelligent system at the University of Science and Technology of China, Hefei, China, in 2012, respectively. He was selected as a straight “A” student of Henan province and won the National Inspiration Scholarship, the National Scholarship, and three times the First Class Scholarship of Henan University. He is currently pursuing his Ph. D degree with Department of Electronic Engineering, Shanghai Jiao Tong University. His research interests include learning automata and their applications, time series analysis, pattern recognition, data mining (big data), text and Web mining, machine learning, and hybrid intelligent systems.



**Yifan Wang** received the B.E. degree in automation from Nanjing University of Posts and Telecommunications, Jiangsu, China, in 2012. She is currently working toward the M.E. Degree in information and communication engineering at the Shanghai Jiao Tong University, Shanghai, China. She was selected as outstanding student leader of Jiangsu province and won the National Scholarship and twice the First Class Scholarship of Nanjing University of Posts and Telecommunications. She is also in Information Security Institute of Shanghai Jiao Tong University, Shanghai, China. Her research interest include networks of learning automata and its applications, recommendation system.