# An Efficient Parameter-Free Learning Automaton Scheme

Chong Di, Qilian Liang, *Fellow, IEEE,* Fangqi Li, Shenghong Li, *Senior Member, IEEE,* Kefeng Fan, and Hao Ge

***Abstract*—The learning automaton (LA) that simulates the interaction between an intelligent agent and a stochastic environment to learn the optimal action is an important tool in reinforcement learning. Being confronted with an unknown environment, most learning automata have more than one parameters to be tuned during a pre-training process in which the LA interacts with the environment. Only after the parameters are tuned properly can an LA act most properly during the training procedure to obtain the optimal behavior. The cost of parameter tuning can be enormous, e.g. possibly millions of interactions are required to seek the best parameter configuration. Therefore the parameter-free LA that uses identical parameters for every environment and saves further tuning has become the hot spot of current research. This paper proposes an efficient parameter-free learning automaton (EPFLA) that depends on a separating function. Taking advantage of both frequentist inference and Bayesian inference, the separating function plays a dual role in the proposed scheme: (1) evaluating the difference in performance between actions in the environment; (2) exploring actions by coining an action selection strategy. A proof is provided to ensure the $\epsilon$-optimality of EPFLA. Comprehensive comparisons verify the privileges of EPFLA over both parameter-based schemes and existing parameter-free schemes.**

***Index Terms*—Reinforcement learning, Learning automaton, Parameter-free, Bayesian inference.**

## I. Introduction

REINFORCEMENT learning maps the situations to actions so as to maximize the reward by interacting with an unknown environment [1]. It is an important branch of machine learning and has been extensively studied for decades. Meanwhile, deep learning that uses computational models composed of multiple processing layers to learn representations of data with multiple levels of abstraction is leading the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains [2]. The deep reinforcement learning [3] that combines the philosophy of both reinforcement learning and deep learning has aroused tremendous attention from both academia and industry, and has been widely applied to intelligent robots, driverless vehicles, etc. The rise of deep reinforcement learning has brought new requirements and standards to reinforcement learning algorithms and renewed their development.

Reinforcement learning can be divided into associative tasks and non-associative ones respecting the relationship between the intelligent agent and the environment [4]. In associative reinforcement learning problems, the actions will affect the environment. For example, each choice in a video game will change the state of the game, and the best choice will vary with the changing state [5]. Thus, an associative reinforcement learning algorithm aims to learn the optimal policy that maps the state of the environment to the optimal action. However, when the environment is not influenced by the actions and the agent aims to explore knowledge of the unknown environment by interactions, the problem is non-associative. For example, in the stochastic shortest path routing problem, the nodes selected at each interaction do not affect the structure of the network [6]. A reinforcement learning algorithm in this non-associative task aims to find the nodes in the shortest path with feedback information from the unknown environment through interactions.

As an elementary part of non-associative reinforcement learning, the learning automaton (LA) can adaptively learn the optimal action among possible choices in a stochastic environment. The mechanism of LA can be summarized as a loop in which the LA selects an action to interact with the environment and receives the corresponding feedback [7]. The LA then utilizes the feedback corresponding to the selected action to update its internal state according to some learning strategy. After sufficient iterations, the LA learns the optimal action that maximizes the expected reward in the environment by configuring its internal state to a destination stage. Since this adaptivity is desirable in both theoretical and practical scenarios, LA has found wide applications such as cellular automata [8], optimization [9]- [11], game theory [12], pattern recognition [13], information security [14], computer vision [15] [16], data mining [17], social networks [18], cloud computing [19], and allocation problems [20] [21]. A broad range of LA and their applications are reviewed in [22]–[25].

The performance of an LA is primarily measured by the *accuracy* and the *efficiency*, the later is reflected by the *convergence rate* and the *convergence time*.

- The *accuracy* is defined as the probability for an LA to obtain the optimal action in a stochastic environment.
- The *convergence rate* is the average number of interactions with the environment for an LA to converge to the optimal action. When interaction itself is expensive (e.g.

This research work is funded by the National Nature Science Foundation of China under Grant 61971283 and 2019 Industrial Internet Innovation Development Project of Ministry of Industry and Information Technology of P.R. China “Comprehensive Security Defense Platform Project for Industrial/Enterprise Networks” under Grant SGFJ0000HLHZ2000034.(*Corresponding author: Shenghong Li*)

Chong Di (dichong95@sjtu.edu.cn), Shenghong Li (shli@sjtu.edu.cn), Fangqi Li (solour_lfq@sjtu.edu.cn) and Hao Ge (sjtu_gehao@sjtu.edu.cn) are with School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China.

Qilian Liang (liang@uta.edu) is with Department of Electrical Engineering, University of Texas at Arlington, Arlington, TX 76019.

Kefeng Fan (kefengfan@163.com) is with China Electronics Standardization Institute, Beijing 100007, China.

medical trial,) the convergence rate becomes a crucial metric.

- The *convergence time* is defined as the average running time for convergence and is involved with not only the feedback delay from the environment but also the acting delay of the agent. If the mechanism of the LA is too complicated so it can not yield a decision quickly during an interaction then the corresponding convergence time could be prohibitive and deny time-sensitive applications.

Recent years have observed significant achievements [26]- [37] that improve the performance of LA. Particularly, the family of estimator algorithms that takes advantage of various estimators to improve learning strategies has been acknowledged as the state-of-the-art. The estimator algorithms are divided into two categories respecting the type of estimators: (1) the deterministic estimator based LA, including the discretized pursuit reward-inaction algorithm ($\mathrm{DP_{RI}}$) [26], the discretized generalized pursuit algorithm (DGPA) [27], the discretized Bayesian pursuit algorithm (DBPA) [28], the last-position elimination-based learning automata (LELA) [29], and the optimal computing budget allocation based scheme ($\mathrm{LA_{OCBA}}$) [30]; (2) the stochastic estimator based LA, including the stochastic estimator algorithm ($\mathrm{SE_{RI}}$) [31], the generalized Bayesian stochastic estimator learning automata (GBSE) [32], and the discretized generalized confidence pursuit algorithm (DGCPA) [33]. Generally speaking, the learning process of an estimator algorithm consists of three-folded loops: (1) selecting an action according to the action probability vector; (2) updating the estimator on the basis of the selected action and the feedback from the random environment; (3) updating the action probability vector according to the state updating strategy with assistance of the estimator. The process terminates until some threshold condition is satisfied. Specifically, the internal state of an estimator based LA consists of the action probability vector and the estimator.

There are multiple parameters to be tuned in estimator algorithms. Apart from the general convergence threshold condition, the deterministic estimator based LA scheme has a resolution parameter $n$, which is used to determine the step by which the action probability vector is updated. For stochastic estimator based LA schemes, a perturbation factor $\gamma$ is further required. These parameters, i.e. $n$ and $\gamma$, need to be tuned for each individual random environment to stabilize the performance of LA. Unfortunately, the parameters can only be tuned by **extra interactions** with the environments. Therefore the LA has to undertake interactions with the environment even before the training process. This is unadorable since the environment should be kept unknown to the LA until the training process. Moreover, the times of interactions required for parameter tuning can be enormous, which might be unbearable in practical applications, especially when the cost of interaction is prohibitive.

In view of this dilemma, an LA that can function properly in all random environments without environment-dependent parameter tuning is of crucial significance. In [34], Ge *et al.* firstly introduced the idea of **parameter-free** LA (PFLA). The term *parameter-free* does not mean no configurable parameters are involved, instead it indicates that involved parameters are independent of the environment. Specifically, the tradeoff between accuracy and efficiency is still guided by some tunable environment-independent parameters. The PFLA proposed in [34] can only be applied to two-action environments, in [35], Ge *et al.* further extended the scheme into multi-action environments, in which the Monte-Carlo simulation [36] is adopted to solve a complex computational problem. Guo *et al.* [37] proposed another parameter-free LA scheme named loss function based parameterless learning automaton (LFPLA). The LFPLA utilizes a loss function [38] as the measure of the capability of actions, and the loss function is further used as the judge of convergence. Similarly, in the extension from two-action environments to multi-action environments, the Monte-Carlo method is utilized to solve an analytically intractable integral. Though PFLA or LFPLA can be applied to both two-action and multi-action environments without parameter tuning for each environment, the Monte-Carlo simulation used in both schemes results in an enormous consumption of time and computational resources, which might be fatal for applications where the environments are resource-limited or time-sensitive. Thus, the efficiency, especially the time efficiency of parameter-free LA needs further improvement.

In this paper, we propose an efficient parameter-free learning automaton (EPFLA) scheme for stationary stochastic environments. By being stationary, the environment' s feedback pattern with respect to each action does not change with time or the behavior of the LA. That is to say, the proposed model belongs to non-associative RL. The studies on stationary environments form the basis of more general research such as generalizations to non-stationary environments, which is an associative RL task. Based on Bayesian and frequentist inference, we define a separating function to measure the difference in expected reward between actions. Taking advantage of the separating function, the proposed LA can efficiently converge according to $\epsilon$-optimality. Moreover, the proposed scheme can be readily extended into multi-action environments while the desirable efficiency is kept intact.

The contributions of this paper are as follows:

1) We propose an efficient parameter-free learning automaton scheme. The parameter-free property indicates that identical parameters can be universally applied to all stationary stochastic environments, and the high-efficiency is reflected by both convergence rate and convergence time.
2) Utilizing both Bayesian inference and frequentist inference, we define a separating function as the measure of confidence when asserting the superiority among actions. Moreover, we design an action probability vector based on the separating function to further improve the exploration strategy.
3) A proof is provided to ensure that EPFLA is $\epsilon$-optimal.
4) Comprehensive comparisons with both parameter-based schemes and parameter-free schemes are conducted to verify the superiority of EPFLA.

The rest of the paper is organized as follows. In section II, we introduce some background knowledge, including the formulation of learning automata and the Bayesian inference.

Section III introduces the separating function based EPFLA and the $\epsilon$-optimality is proved in Section IV. Experimental results are shown in Section V to show the advantages of EPFLA over both parameter-based schemes and existing parameter-free schemes. The applications of LA in real-world are discussed in VI. Finally, Section VII concludes this paper.

## II. Background

### A. Learning automata

The mathematical model of an LA interacting with a stationary stochastic environment can be formulated as a triplet $\langle A, B, C \rangle$ [7], where

- $A = \{a_1, a_2, \cdots, a_r\}$ is the set of actions, and $r$ is cardinality of $A$, i.e. the number of possible actions. The action selected at the $t$-th interaction with the environment is denoted by $a(t)$.
- $B$ is the set of possible response from the random environment. In this paper we focus on P-model environments, in which the response is binary, i.e. $B = \{0, 1\}$ where zero denotes a reward and one denotes a penalty. At the $t$-th iteration, the feedback from the environment receiving $a(t)$ is $b(t)$.
- $C = \{c_1, c_2, \cdots, c_r\}$ is the set of reward probabilities corresponding to action set $A$. It is $C$ that identifies the environment by
$$c_i = \Pr\{b(t) = 0 | a(t) = a_i\}. \tag{1}$$
The optimal action in the environment is the action with the maximal expected reward, i.e. the highest reward probability. For a stationary environment, $C$ is independent of $t$.

The aim of LA is to identify the optimal action by interacting with the environment. The philosophy behind is to collect feedback from the environment and uses the information to extract evidence that supports an optimal assertion [35].

An estimator-based LA is further featured by three other components denoted by $\mathbf{P}$, $\mathbf{E}$, and $\mathcal{T}$, in which $\mathbf{P}$ and $\mathbf{E}$ vary in the process of iterations.

- $\mathbf{P}(t) = [p_1(t), p_2(t), \cdots, p_r(t)]$ is the action probability vector subject to $\sum\limits_{i=1}^{r} p_i(t) = 1$ with
$$p_i(t) = \Pr\{a(t) = a_i\}. \tag{2}$$
An LA gets converged after the $t_C$-th iteration when
$$\max_{i=1,2,\cdots,r} \{p_i(t_C)\} \geq \tau,$$
where $\tau$ is a predefined threshold, and the optimal action learned by the LA is the one with the maximal action probability whose index is given by
$$\arg \max_{i=1,2,\cdots,r} \{p_i(t_C)\}.$$
- $\mathbf{E}(t) = [e_1(t), e_2(t), \cdots, e_r(t)]$ is the estimator vector corresponding to the action set $A$. It records the estimation of the reward probability of each action. In majorities of the deterministic estimator based LA schemes [26], the estimate $e_i$ of action $a_i$ is the maximum likelihood (ML) estimate according to the frequentist inference [39]:
$$e_i(t) = \frac{W_i(t)}{Z_i(t)}, \tag{3}$$
where $W_i(t)$ and $Z_i(t)$ are the number of times that action $a_i$ has been rewarded and selected up to the $t$-th iteration. The estimate $e_i(t)$ for a stochastic estimator is defined as [31]
$$e_i(t) = \frac{W_i(t)}{Z_i(t)} + R(\gamma, t),$$
where $R(\gamma, t)$ provides a stochastic perturbation.
- $\mathcal{T}$ is the update strategy that characterizes an estimator-based LA, formally
$$\mathbf{P}(t+1) = \mathcal{T}(\mathbf{P}(t), \mathbf{E}(t)).$$
The action probability vector is usually updated in a discretized fashion, hence a step size $\Delta$ is usually required, it is defined as
$$\Delta = \frac{1}{rn},$$
where $n$ is the resolution parameter, a parameter that has to be tuned respecting different environments. After each iteration, the components of $\mathbf{P}$ are modified with granularity $\Delta$. The role of $n$ is similar to that of the learning rate in other learning systems. If $n$ is too large then LA might fail to converge to the optimum, if $n$ is too small then convergence might be too slow.

### B. Bayesian Inference

Bayesian inference [40], an important technique in mathematical statistics, uses the Bayesian theorem to update the probability distribution for a hypothesis. In Bayesian inference, the conjugate prior distribution of the Bernoulli distribution is a Beta distribution Beta$(\alpha, \beta)$. The probability density function (PDF) and the cumulative distribution function (CDF) of the Beta distribution are
$$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}, \tag{4}$$
$$\begin{aligned} F(x; \alpha, \beta) &= \int_0^x f(x; \alpha, \beta) dx \\ &= \frac{B(x, \alpha, \beta)}{B(\alpha, \beta)} = I_x(\alpha, \beta), \end{aligned} \tag{5}$$
where $B(\alpha, \beta)$ is the beta function, $B(x, \alpha, \beta)$ is the incomplete beta function, and $I_x(\alpha, \beta)$ is the regularized incomplete beta function [40].

Given $s$ success in $n$ conditionally independent Bernoulli trials with probability $p$, the posterior distribution of $p$ is Beta$(s+1, n-s+1)$ [41]. With regard to LA in P-model environments, the process of an action interacting with an environment is essentially a Bernoulli trial. Thus, the Bayesian estimate up to the $t$-th interaction of the reward probability $c_i(t)$ with respect to observed information including $W_i(t)$ and $Z_i(t)$ is a Beta distribution Beta$(W_i(t)+1, Z_i(t)-W_i(t)+1)$. Fig.1 shows the Bayesian estimates of the reward probability $c_i(t)$ with given $W_i(t)$ and $Z_i(t)$.

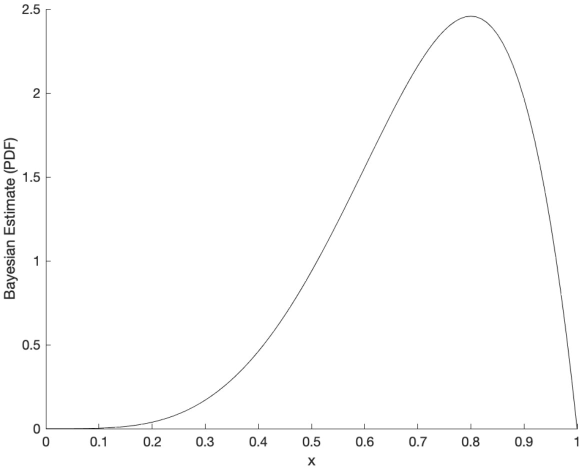


Fig. 1. The Bayesian estimates (PDF) of the reward probability $c_i(t)$ with $W_i(t) = 4$ and $Z_i(t) = 5$.

## III. An Efficient parameter-free Learning Automaton

In this section, an efficient parameter-free learning automaton scheme is proposed. A separating function is utilized to evaluate the difference in reward probabilities between two actions in stationary stochastic environments.

### A. The separating function

Consider an LA in a P-model environment formulated by $\langle A, B, C\rangle$. As introduced in Section II, the posterior distribution of an action's reward probability is a beta distribution in Bayesian statistics. The Bayesian estimates of actions' reward probabilities after the $t$-th iteration form a vector of distributions $\hat{\mathbf{E}}(t) = [\hat{e}_1(t), \hat{e}_2(t), \cdots, \hat{e}_r(t)]$, where $\hat{e}_i(t) = \text{Beta}(\alpha_i(t), \beta_i(t))$, the parameters $\alpha_i(t)$ and $\beta_i(t)$ are the natural paramater of a Beta distribution:

$$\alpha_i(t) = W_i(t) + 1,$$

$$\beta_i(t) = Z_i(t) - W_i(t) + 1.$$

While the ML estimates of actions' reward probabilities are denoted as $\bar{\mathbf{E}}(t) = [\bar{e}_1(t), \bar{e}_2(t), \cdots, \bar{e}_r(t)]$ according to (3)

$$\bar{e}_i(t) = \frac{W_i(t)}{Z_i(t)}. \tag{6}$$

Intuitively, the action with the highest ML estimate and is distinguishable from other actions is the optimal one. So a metric between the Bayesian estimates of two actions $a_i$ and $a_j$, specifically two Beta distributions are necessary for the proposed LA. Such a metric is expected to converge to some value (e.g. zero) when the times of interactions with the environment approach infinity. Under this intuition we propose the following separating function as the metric, the philosophy behind is to integrate the tail of one distribution over the mean of another. The tails are illustrated by the shadowed areas in Fig.2. When the learning process continues, both distributions are expected to converge to a degenerated distribution that has a peak at its mean, so the separating function will degenerate to zero.

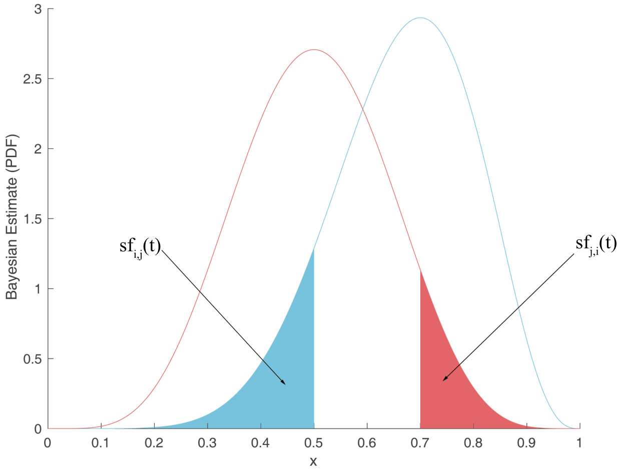


Fig. 2. The graphical representation of the separating function. The Bayesian estimates of action $a_i$ (in blue curve) and action $a_j$ (in red curve) are $\hat{e}_i = \text{Beta}(8, 4)$ and $\hat{e}_j = \text{Beta}(6, 6)$, and the ML estimates are $\bar{e}_i = 0.7$ and $\bar{e}_j = 0.5$.

**Definition 1** Given the Bayesian estimates and the ML estimates of actions $a_i$ and $a_j$ with $\bar{e}_i(t) > \bar{e}_j(t)$. The separating function (SF) is defined as follows using (4) and (5):

$$\begin{aligned}\text{sf}_{i,j}(t) &= F(\bar{e}_j(t); \alpha_i(t), \beta_i(t))\\ &= \int_0^{\bar{e}_j(t)} f(x; \alpha_i(t), \beta_i(t))\mathrm{d}x\\ &= I_{\bar{e}_j(t)}(\alpha_i(t), \beta_i(t)),\end{aligned}$$

$$\begin{aligned}\text{sf}_{j,i}(t) &= 1 - F(\bar{e}_i(t); \alpha_j(t), \beta_j(t))\\ &= 1 - \int_0^{\bar{e}_i(t)} f(x; \alpha_j(t), \beta_j(t))\mathrm{d}x\\ &= 1 - I_{\bar{e}_i(t)}(\alpha_j(t), \beta_j(t)),\end{aligned}$$

$$\text{SF}_{i,j}(t) = \text{SF}_{j,i}(t) = \text{sf}_{i,j}(t) + \text{sf}_{j,i}(t). \tag{7}$$

**Remark 1.** *Another benchmark metric between distributions is the relative entropy. Compared with the relative entropy, the proposed separating function is symmetric and converges to zero rather than infinity. Moreover, it enjoys some monotonic properties when being applied to Beta distributions.*

### B. Properties of the separating function

For a given Beta distribution $\text{Beta}(\alpha, \beta)$, we have [42]

$$I_x(\alpha + 1, \beta) = I_x(\alpha, \beta) - \frac{x^\alpha(1-x)^\beta}{\alpha B(\alpha, \beta)}, \tag{8}$$

$$I_x(\alpha, \beta + 1) = I_x(\alpha, \beta) + \frac{x^\alpha(1-x)^\beta}{\alpha B(\alpha, \beta)}. \tag{9}$$

Assume that $a_i$ is selected to interact with the environment at the $(t + 1)$-th iteration and $a_j$ is another action with $\bar{e}_i(t) > \bar{e}_j(t)$.

- If feedback $b(t+1)=0$ (reward) then $\alpha_i(t+1)=\alpha_i(t)+1$ and $\beta_i,\alpha_j,\beta_j$ remain invariant, using (8)

$$I_{\bar{e}_j(t)}(\alpha_i(t)+1,\beta_i(t))<I_{\bar{e}_j(t)}(\alpha_i(t),\beta_i(t)) \\ \Rightarrow \\ \mathrm{sf}_{i,j}(t+1)<\mathrm{sf}_{i,j}(t),$$

$$\bar{e}_i(t+1)>\bar{e}_i(t)\Rightarrow \mathrm{sf}_{j,i}(t+1)<\mathrm{sf}_{j,i}(t).$$

Thus,

$$\mathrm{SF}_{i,j}(t+1)=\mathrm{sf}_{i,j}(t+1)+\mathrm{sf}_{j,i}(t+1)<\mathrm{SF}_{i,j}(t).$$

- If feedback $b(t+1)=1$ (penalty) then $\beta_i(t+1)=\beta_i(t)+1$ and $\alpha_i,\alpha_j,\beta_j$ remain invariant, an analogous deduction using (9) yields

$$\mathrm{SF}_{i,j}(t+1)>\mathrm{SF}_{i,j}(t).$$

Similarly, when $a_j$ is selected as the $(t+1)$-th action,

- If feedback $b(t+1)=0$ (reward):

$$\mathrm{SF}_{i,j}(t+1)>\mathrm{SF}_{i,j}(t).$$

- If feedback $b(t+1)=1$ (penalty):

$$\mathrm{SF}_{i,j}(t+1)<\mathrm{SF}_{i,j}(t).$$

If neither $a_i$ nor $a_j$ is selected at the $(t+1)$-th iteration, then

$$\mathrm{SF}_{i,j}(t+1)=\mathrm{SF}_{i,j}(t).$$

An intuitive corollary from the previous analysis is that if $a_i$ enjoys a higher reward probability than $a_j$, then $a_i$ is more likely to be rewarded and $a_j$ is not. Therefore the value of $\mathrm{SF}_{i,j}$ will approximately monotonically decrease to zero.

Thus we can readily suggest that action $a_i$ is better than $a_j$ after the $t-$th iteration if the value of $\mathrm{SF}_{i,j}(t)$ is less than a predefined threshold $\tau$. The pairwise comparisons can thence be adopted to form a complete learning strategy. We demonstrate this in two-action as well as general environments.

*1) Two-action Environments:* In two-action cases, $A=\{a_1,a_2\}$. Each action is selected alternatively until the separating function takes a value less than a threshold

$$\mathrm{SF}_{1,2}(t)<\tau.$$

Then it can be concluded that the superior action has the index:

$$\arg\max_{i=1,2}\{\bar{e}_i(t)\}.$$

*2) Multi-action Environments:* Analogously, action $a_i\in A$ is the optimal one if it is superior to any other action. Hence $a_i$ is taken as the optimal action after the $t$-th iteration if the condition below holds for all $a_j\in A, j\neq i$:

$$\begin{cases}\bar{e}_i(t)>\bar{e}_j(t),\\ \mathrm{SF}_{i,j}(t)<\tau.\end{cases}$$

### C. Initialization and Exploration strategy

*1) Initialization:* In our scheme, Bayesian initialization [32] is utilized to encourage exploration. The rewarded time and penalized time of each action are initialized as one. Thus the initial Bayesian estimate $\hat{e}_i(0)$ of action $a_i$ is Beta$(2,2)$, and the ML estimate $\bar{e}_i(0)$ is 1/2.

*2) Exploration strategy:* Different from the majority of LA, the convergence of EPFLA is determined by the separating function rather than the action probability vector. However, the action probability vector is still kept to select the action interacting with the environment at each iteration. In our scheme, we construct the action probability vector from the separating function.

At the $t$-th iteration, let action $a_{m(t)}$ be the temporary optimal action with the highest ML estimate of expected reward probability,

$$m(t)=\arg\max_{i=1,2,\cdots,r}\{\bar{e}_i(t)\}. \tag{10}$$

Given the value of separating function $\mathrm{SF}_{m(t),i}(t)$ between action $a_{m(t)}$ and $a_i$ ($\forall a_i\in A, i\neq m(t)$), the action probability vector $\mathbf{P}(t+1)$ that is adopted to select the next action is defined as follows:

$$\begin{cases}p_{m(t)}(t+1)=\frac{1}{2},\\ p_i(t+1)=\frac{1}{2}\frac{\mathrm{SF}_{m(t),i}(t)}{\sum\limits_{i\neq m(t)}\mathrm{SF}_{m(t),i}(t)}, \forall i\neq m(t).\end{cases} \tag{11}$$

The proposed exploration strategy is similar to the $\epsilon$-greedy strategy [4]. The discrepancy between them is that the $\epsilon$-greedy strategy is entirely random when selecting an action among suboptimal ones, but the proposed one can take the difference in the estimated reward probabilities among actions into consideration. To summarize, in our exploration strategy, the action with the highest reward estimation is most likely to be selected again, which is beneficial for exploring the optimal action. And the action that is not selected for sufficient times (so its posterior distribution does not concentrate) or has a mean close to that of $a_{m(t)}$ is more likely to be chosen again. Let $a_j$ be such an action, either of the two reasons can make its Bayesian estimate having much have probability mass on the other side of $\bar{e}_{m(t)}$ and therefore a large tail $\mathrm{sf}_{j,m(t)}(t)$. Thus $\mathrm{SF}_{m(t),i}(t)$ would be relatively large, so $a_j$ is more likely to be selected at the next iteration. This is in accordance with the intuition since such $a_j$ is potentially the optimal action.

The process of EPFLA is summarized in Algorithm 1, where $\mathbf{1}_r$ denotes a column vector of length $r$ with all of its components taking value one.

## IV. Proof of $\epsilon$-optimality

The $\epsilon$-optimality of an LA scheme is of enormous importance since it ensures the theoretical convergence. Generally, the $\epsilon$-optimality implies that the LA will converge to the optimal action as long as each action can be selected for infinite times. In this section, we present three lemmas, followed by the proof of the $\epsilon$-optimality of EPFLA. When the response pattern of one specific action is analyzed, we drop $t$ without loss of generality.

**Lemma 1.** *Suppose the number of selected times $Z_i$ for action $a_i$ approaches infinity, the beta distribution $Beta(\alpha_i,\beta_i)$ converges to one-point degenerate distribution with a Dirac delta function spike at $c_i$ by probability. That is, for any $\epsilon\in(0,1)$ and $\delta_1,\delta_2\in(0,1)$, there exists $Z_i^{l_1}(\epsilon,\delta_1,\delta_2)$ such that:*

$$Pr\left\{\int_{|x-c_i|\leq\delta_1} f(x;\alpha_i,\beta_i)dx\geq(1-\delta_2)\right\}\geq(1-\epsilon), \tag{12}$$

**Algorithm 1** EPFLA

```
Require: A convergence threshold: $\tau$;
Initialize: Iteration: $t = 0$;
Initialize: Rewarded times: $\forall i, W_i(t) = 1$;
Initialize: Selected times: $\forall i, Z_i(t) = 2$;
Initialize: Bayesian estimate: $\forall i, \hat{e}_i(t) = \text{Beta}(W_i(t)+1, Z_i(t) - W_i(t) + 1) = \text{Beta}(2, 2)$;
Initialize ML estimate: $\forall i, \bar{e}_i(t) = \frac{W_i(t)}{Z_i(t)} = \frac{1}{2}$;
Initialize Action probability vector $\mathbf{P}(t)$, $\forall i, p_i(t) = \frac{1}{r}$;
repeat
    $\mathbf{F} = \mathbf{1}_r$;
    Select action $a(t+1) = a_j$ according to the action probability vector $\mathbf{P}(t)$ from (2).
    Receive the feedback $b(t+1)$ from the environment by (1) and update the parameters:
        $\begin{cases} W_j(t+1) = W_j(t) + 1 & \text{if feedback = 0 (reward)} \\ Z_j(t+1) = Z_j(t) + 1 \end{cases}$
    Compute the ML estimates of actions' reward probabilities by (6) and get the action $a_{m(t+1)}$ by (10).
    for $l = 1$ to $r, l \neq m(t+1)$ do
        Compute the value of separating function $\text{SF}_{m(t+1),l}(t)$ between action $a_{m(t+1)}$ and $a_l$ by (7).
        if $\text{SF}_{m(t+1),j}(t) < \tau$ then
            $\mathbf{F}_j = 0$
        end if
    end for
    Update $\mathbf{P}(t+1)$ according to (11).
    $t = t + 1$;
until $\mathbf{1}_r^{\mathrm{T}}\mathbf{F} = 1$.
Output: the optimal action $a_{m(t)}$ with highest ML estimate.
```

$$Pr\left\{\int_{|x-c_i|>\delta_1} f(x;\alpha_i,\beta_i)dx \leq \delta_2\right\} \geq (1-\epsilon), \tag{13}$$

*where $\alpha_i$ and $\beta_i$ are obtained feedback from the environment after $Z_i^{l_1}(\epsilon,\delta_1,\delta_2)$ rounds of interactions.*

*Proof.* According to the weak law of large number, the value of $\bar{e}_i$ should converge to its expectation $c_i$ by probability. Recall that $f(x;\alpha_i,\beta_i)$ or its integral can be transcribed as a continuous function of $\bar{e}_i$, therefore for given $\epsilon$, $\delta_3$ and $\delta_4$, there exists $Z_i^f(\epsilon,\delta_3,\delta_4)$ such that $Z_i \geq Z_i^f(\epsilon,\delta_3,\delta_4)$ implies

$$\text{Pr}\left\{\begin{array}{l} \Big|\displaystyle\int_{|x-c_i|>\delta_4} f(x;\alpha_i,\beta_i) - \\ \displaystyle\int_{|x-c_i|>\delta_4} \frac{[x^{c_i}(1-x)^{1-c_i}]^{Z_i}}{B(c_iZ_i+1,(1-c_i)Z_i+1)}\Big| \geq \delta_3 \end{array}\right\} \leq \epsilon.$$

To simplify the notation, let:

$$g(x) = x^{c_i}(1-x)^{1-c_i},$$

$$C_i = \frac{1}{B(c_iZ_i+1,(1-c_i)Z_i+1)} = \frac{1}{\|g\|_{Z_i}^{Z_i}}.$$

Now consider

$$\begin{aligned} &\int_{|x-c_i|>\delta_4} f(x;\alpha_i,\beta_i)\mathrm{d}x \\ &\leq \int_{|x-c_i|>\delta_4} \left[f\left(x;c_iZ_i+1,(1-c_i)Z_i+1\right)+\delta_3\right]\mathrm{d}x \\ &< \delta_3 + \int_{|x-c_i|>\delta_4} C_i g(x)\mathrm{d}x = \delta_3 + \left(\frac{\|g\|_{Z_i,|x-c_i|>\delta_4}}{\|g\|_{Z_i}}\right)^{Z_i}. \end{aligned} \tag{14}$$

Where $\|g\|_{Z_i,|x-c_i|>\delta_4}$ denotes the $Z_i$-norm of $g$ restricted to $|x-c_i| > \delta_4$, the second inequality in (14) holds with probability no less than $(1-\epsilon)$.

But the single maximum of $g(x)$ is on $g(c_i)$, which is a straightforward result of derivation. Therefore it is necessarily true that

$$\frac{\|g\|_{Z_i,|x-c_i|>\delta_4}}{\|g\|_{Z_i}} < 1.$$

Formally, the value $\frac{\|g\|_{Z_i,|x-c_i|>\delta_4}}{\|g\|_{Z_i}}$ should converge to

$$\begin{aligned} \frac{\|g\|_{\infty,|x-c_i|>\delta_4}}{\|g\|_\infty} &= \frac{\sup_{|x-c_i|>\delta_4} g(x)}{\sup_{x\in(0,1)} g(x)} \\ &= \frac{\max\{g(c_i-\delta_4), g(c_i+\delta_4)\}}{g(c_i)}. \end{aligned}$$

That is to say, for any $\delta_4$ and $\delta_5$, there exists $Z_i^N(\delta_4,\delta_5)$ such $Z_i > Z_i^{\text{Norm}}(\delta_4,\delta_5)$ implies

$$\left|\frac{\|g\|_{Z_i,|x-c_i|>\delta_4}}{\|g\|_{Z_i}} - \frac{\max\{g(c_i-\delta_4), g(c_i+\delta_4)\}}{g(c_i)}\right| < \delta_5.$$

Then $Z_i > Z_i^{\text{Norm}}(\delta_4, \frac{g(c_i)-\max\{g(c_i-\delta_4),g(c_i+\delta_4)\}}{2g(c_i)})$ implies

$$\frac{\|g\|_{Z_i,|x-c_i|>\delta_4}}{\|g\|_{Z_i}} \leq \frac{g(c_i)+\max\{g(c_i-\delta_4), g(c_i+\delta_4)\}}{2g(c_i)} < 1. \tag{15}$$

Which further indicates that given any $\delta_6$, there exists $Z_i^{\text{Expon}}(\delta_4,\delta_6)$ such that $Z_i > Z_i^{\text{Expon}}(\delta_4,\delta_6)$ together with (15) implies

$$\left(\frac{\|g\|_{Z_i,|x-c_i|>\delta_4}}{\|g\|_{Z_i}}\right)^{Z_i} < \delta_6.$$

Finally, if we let

$$\begin{aligned} &Z_i^{l_1}(\epsilon,\delta_1,\delta_2) = \\ &\qquad \max(Z_i^f(\epsilon,\frac{\delta_2}{2},\delta_1), \\ &\qquad\qquad Z_i^{\text{Norm}}\left(\delta_1, \frac{g(c_i)-\max\{g(c_i-\delta_1),g(c_i+\delta_1)\}}{2}\right), \\ &\qquad\qquad Z_i^{\text{Expon}}(\delta_1,\frac{\delta_2}{2})), \end{aligned}$$

then the following inequality will hold with probability no less than $(1-\epsilon)$

$$\int_{|x-c_i|>\delta_1} f(x;\alpha_i,\beta_i)\mathrm{d}x < \delta_2.$$

This completes the proof of Lemma 1. □

**Remark 2.** *Lemma 1 can be empirically verified by Figure 3.*

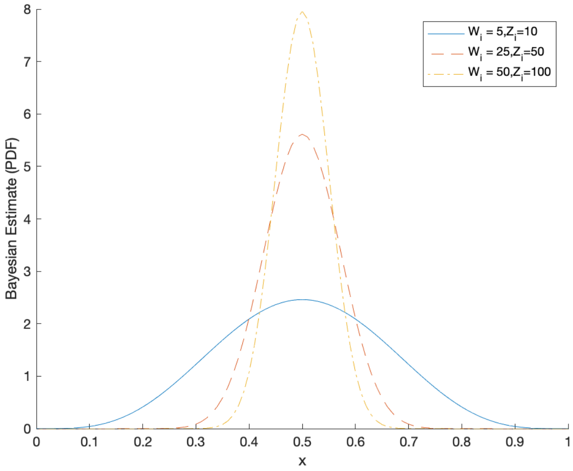


Fig. 3. The Bayesian estimates (PDF) of the reward probability $c_i = 0.5$ with $W_i = 5, 25, 50$ and $Z_i = 10, 50, 100$.

**Lemma 2.** *Assume $m$ is the index of action with the maximum ML estimate of reward probability, then the value of separating function between action $a_m$ and $a_i$ $(i = 1, 2, \cdots, r, i \neq m)$ approaches zero only when the number of selected times for each action approaches infinity by probability. Formally, for any $\epsilon, \delta$, there exists $Z_m^{l_2}(\epsilon, \delta)$ and $Z_i^{l_2}(\epsilon, \delta)$ such that $SF_{m,i} < \delta$ iff $a_m$ and $a_i$ has been selected for no less than $Z_m^{l_2}(\epsilon, \delta)$ and $Z_i^{l_2}(\epsilon, \delta)$ times. This implication holds with probability no less than $(1 - \epsilon)$.*

*Proof.* According to the definition of the separating function, we have

$$\begin{aligned}\mathrm{SF}_{m,i} &= \mathrm{sf}_{m.i} + \mathrm{sf}_{i,m} \\ &= F(\bar{e}_i; \alpha_m, \beta_m) + (1 - F(\bar{e}_m; \alpha_i, \beta_i)) \\ &= \int_0^{\bar{e}_i} f(x; \alpha_m, \beta_m)\mathrm{d}x + \int_{\bar{e}_m}^1 f(x; \alpha_i, \beta_i)\mathrm{d}x.\end{aligned}$$

Since $\mathrm{sf}_{m,i}$ and $\mathrm{sf}_{i,m}$ are nonnegative, $\mathrm{SF}_{m,i} < \delta$ implies $\int_0^{\bar{e}_i} f(x; \alpha_m, \beta_m)\mathrm{d}x < \delta$ and $\int_{\bar{e}_m}^1 f(x; \alpha_i, \beta_i)\mathrm{d}x < \delta$.

When $\bar{e}_i \neq \bar{e}_m$, let $\delta_1 = \frac{\bar{e}_m - \bar{e}_i}{2} > 0$ then (12) and (13) in Lemma 1 indicate that given

$$Z_i^{l_2}(\epsilon, \delta) = Z_i^{l_1}(\frac{\epsilon}{2}, \delta_1, \delta),$$

$$Z_m^{l_2}(\epsilon, \delta) = Z_m^{l_1}(\frac{\epsilon}{2}, \delta_1, \delta),$$

then with probability no less than $(1 - \frac{\epsilon}{2})^2 > (1 - \epsilon)$:

$$\int_0^{\bar{e}_i} f(x; \alpha_m, \beta_m)\mathrm{d}x < \int_0^{\bar{e}_m - \delta_1} f(x; \alpha_m, \beta_m)\mathrm{d}x < \delta,$$

$$\int_{\bar{e}_m}^1 f(x; \alpha_i, \beta_i)\mathrm{d}x < \int_{\bar{e}_i + \delta_1}^1 f(x; \alpha_i, \beta_i)\mathrm{d}x < \delta.$$

Conversely, for arbitrary $Z_i$ or $Z_m$, the probability that condition $\mathrm{SF}_{m,i} < \tau$ holds for any $\tau$ can not be arbitrarily close to unity. Since for $Z_i$ or $Z_m$ not sufficiently large, the approximation to the Dirac function by probability fails to become effective, which is obvious from the previous observation in Figure 3. Therefore the tails of the two Beta distributions remain large. This fact rejects the convergence condition for given $\tau$. Hence the lemma is proved. □

**Lemma 3.** *Suppose the number of iterations $t \to \infty$, the number of selected times for each action will tend to infinity,* i.e., *$Z_i \to \infty$, for $i = 1, 2, \cdots, r$.*

*Proof.* At each iteration, assume action $a_m$ is the 'optimal' action with the highest ML estimate, under the definition of action probability vector $P$, we have

$$p_m = \frac{1}{2} > 0.$$

When $Z_j < \infty$, for any action $a_j \in A, j \neq m$,

$$\mathrm{sf}_{j,m} = \int_{\bar{e}_m}^1 f(x; \alpha_j, \beta_j)\mathrm{d}x > 0.$$

Then, we have

$$\mathrm{SF}_{m,j} = \mathrm{sf}_{m,j} + \mathrm{sf}_{j,m} > 0.$$

Therefore,

$$p_j = \frac{1}{2} \frac{\mathrm{SF}_{m,j}}{\sum\limits_{j \neq m} \mathrm{SF}_{m,j}} > 0.$$

Thus, it is possible for each action to be selected at each iteration before $t \to \infty$. Consequently, when $t \to \infty$,

$$Z_i \to \infty, i = 1, 2, \cdots, r.$$

□

**Remark 3.** *The first half of Lemma 2 together with Lemma 3 readily yields the corollary that EPFLA will converge by probability. Moreover, one can intuitively observe that EPFLA will correctly converge with probability one. Since the ML estimation will yield the optimal action when each action has been selected for sufficient times (according to the law of large numbers), and such selection is ensured by choosing a $\tau$ sufficiently small from previous lemmas.*

**Theorem 1.** *EPFLA is $\epsilon$-optimal in every stationary random environment. That is to say, suppose $m$ is the index of the action that has the maximum reward probability $c_m$, $c_m = \max_{a_i \in A} \{c_i\}$. Given any $\epsilon > 0$, there exists a $\tau_0(\epsilon) > 0$ such that for all $\tau \leq \tau_0(\epsilon)$:*

$$Pr\{\textit{EPFLA converges to action } a_m\} > (1 - \epsilon).$$

*Proof.* The law of large number indicates that the ML estimation $\bar{e}_i$ for action $a_i$ converges into the interval $(c_i - \delta_1, c_i + \delta_1)$ with probability no less than $(1 - \epsilon_1)$ when $Z_i$ is larger than a threshold $Z_i^{\mathrm{ML}}(\epsilon_1, \delta_1)$, let

$$\delta_1 \leq \frac{1}{2} \min_{a_i \in A, i \neq m} \{c_m - c_i\},$$

thus if any action $a_i$ has been selected for at least $Z_i^{\mathrm{ML}}(\epsilon_1, \delta_1)$ times then EPFLA can correctly address the optimal action. What left to be proved is that such constraint can be exerted by picking a corresponding $\tau_0(\epsilon)$. This is a straightforward result of Lemma 2, let

$$\tau_0(\epsilon) = \inf\left\{\tau : \forall a_i \in A, Z_i^{l_2}(\frac{\epsilon}{2r}, \tau) > Z_i^{\mathrm{ML}}(\frac{\epsilon}{2r}, \delta_1)\right\},$$

then for any $\tau < \tau_0(\epsilon)$, with probability no less than $(1 - \frac{\epsilon}{2r})^{2r} > (1-\epsilon)$ all action $a_i$ has to be selected for no less than $Z_i^{\mathrm{ML}}(\frac{\epsilon}{2r}, \delta_1)$ times (the $2r$ in exponents comes from the $r$ML convergences for all actions and all $r$ pairwise convergences,) which necessarily implies the correct convergence of EPFLA to the optimal action. □

## V. Simulation Results

To verify the effectiveness of the proposed scheme, we compare EPFLA with both parameter-based schemes and parameter-free schemes. The parameter-based contenders consist of classic estimator algorithms, including $\mathrm{DP_{RI}}$ [26], DGPA [27] and $\mathrm{SE_{RI}}$ [31] which have been widely used as the baseline in LA literature, together with recently proposed algorithms including DBPA [28], DGCPA [33], GBSE [32], $\mathrm{LELA_R}$ [29], and $\mathrm{LA_{OCBA}}$ [30]. Specifically, as discussed in Section I, the $\mathrm{DP_{RI}}$, DGPA, DBPA, $\mathrm{LELA_R}$, and $\mathrm{LA_{OCBA}}$ are deterministic estimator based LA with one environment-dependent parameter $n$, while $\mathrm{SE_{RI}}$, DGCPA, and GBSE are stochastic estimator based LA with two environment-dependent parameters $(\gamma, n)$. For parameter-free contenders, the most representative algorithms, PFLA [35] and LFPLA [37] are involved.

All LA are evaluated in five benchmark environments denoted by $E_1$-$E_5$ [31] whose sets of reward probability are shown in Table I. These five environments have been widely used concerning LA [28], [31]–[35], [37]. The complexity of a benchmark environment is reflected by the difference of the reward probabilities between the optimal and the suboptimal actions denoted by $\sigma$. If $\sigma$ is smaller then it is naturally harder to distinguish the optimal action from the suboptimal ones. From this perspective, the environment $E_3$ is the most complicated one among $E_1 - E_5$ with $\sigma = c_1 - c_2 = 0.05$, and $E_4$ is the simplest one with $\sigma = 0.2$. The simulations are carried on the five benchmark environments above, where the accuracy, the convergence rate, and the convergence time of schemes in each environment are evaluated in terms of repeated trials in each environment [1].

In this section, the cost of parameter tuning (from which parameter-free LA is free) is first presented to show the advantage of parameter-free LA, followed by the environment-independent parameter tuning of EPFLA. Then we compare the accuracy, convergence rate and convergence time among various proposal and discuss potential applications.

### A. Cost of Parameter Tuning

Before comparing the performance among algorithms, the cost of parameter tuning for parameter-based algorithms is illustrated to show the advantages of parameter-free schemes. Some literature [35] has pointed out that, without parameter tuning, the performance of parameter-based algorithms is unstable and non-competitive. Thus, manual parameter tuning is indispensable for parameter-based algorithms.

As the common sense, the parameters are tuned by **extra** interactions with the environment during a pre-training process. The standard tuning process is spearheaded by [43]. For example, deterministic estimator based algorithms [26]–[30] need the resolution parameter $n$ to adjust the step size when updating the action probability vector $\mathbf{P}$. The smallest value of $n$ that yields the fastest convergence rate while makes no error in a batch of NE experiments is taken as the most appropriate value. So each integer starting from one is tested for $n$ until the no error condition is met. The NE no error experiments forms one category of pre-training. Besides, the same procedure will be performed for twenty times to ensure the reliability, and the best resolution parameter $n^*$ is set to the average of twenty optimal values. The number of extra iterations required for the tuning of resolution parameter $n$ can be estimated by:

$$\mathrm{cost}(n) = 20 \times \mathrm{NE} \times n^* \times \text{average convergence rate}, \quad (16)$$

where the number of no error experiments is usually set to $\mathrm{NE} = 750$ [43].

Meanwhile, the environment-dependent parameters of a stochastic estimator based LA [31]–[33] are $(\gamma, n)$. Since $\gamma$ and $n$ are not independent, for each assigned value of $\gamma$, there exists a corresponding optimal resolution parameter $n^*(\gamma)$, which is obtained by pre-training with a fixed $\gamma$. The optimal couple $(\gamma^*, n^*(\gamma^*))$ is the one with the fastest convergence rate. The cost of parameter tuning for stochastic estimator based algorithms is roughly estimated as:

$$\mathrm{cost}(\gamma, n) = \mathrm{range}(\gamma) \times 20 \times \mathrm{NE} \times n^* \times \text{convergence rate}, \quad (17)$$

where the search range of $\gamma$ is from 1 to 30, i.e., $\mathrm{range}(\gamma) = 30$ [35].

Table II lists the best parameters and the number of extra interactions for parameter tuning in parameter-based algorithms. The results verify (16)(17) and indicate that the cost of parameter tuning is enormous, since billions of extra interactions with the environment are required for searching the best parameters. The results also reveal the superiority of parameter-free schemes, which do not rely on any pre-training. In the following comparisons, the cost of parameter tuning is not going to be taken into account further. The proposed EPFLA will be compared with well-tuned algorithms using the best parameter configurations.

### B. Effect of the Convergence Threshold $\tau$

Recall that parameter-free property indicates a set of parameters for the scheme can be universally applicable for all environments. In EPFLA, there are only three configurable parameters, including the initial value of $W_i$ and $Z_i$ for each action, and the convergence threshold $\tau$. The $W_i$ and $Z_i$ are initialized using Bayesian initialization to encourage exploration. Meanwhile, as it is inevitable for an $\epsilon$-optimal scheme to make a trade-off between the accuracy and efficiency, a well-designed threshold $\tau$ is crucial to the performance of EPFLA. Table III shows the accuracy and convergence rate under different $\tau$. The results indicate that the performance of EPFLA is steady when using the same convergence threshold in different environments, which verifies the parameter-free

[1] Due to the enormous cost of time, $1,000$ repeated experiments were performed for PFLA and LFPLA in each environment, and $250,000$ repeated experiments were performed for other schemes in each environment.

TABLE I
THE REWARD PROBABILITY VECTORS OF THE BENCHMARK ENVIRONMENTS

| Env. | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ | $c_9$ | $c_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $E_1$ | **0.65** | 0.50 | 0.45 | 0.40 | 0.35 | 0.30 | 0.25 | 0.20 | 0.15 | 0.10 |
| $E_2$ | **0.60** | 0.50 | 0.45 | 0.40 | 0.35 | 0.30 | 0.25 | 0.20 | 0.15 | 0.10 |
| $E_3$ | **0.55** | 0.50 | 0.45 | 0.40 | 0.35 | 0.30 | 0.25 | 0.20 | 0.15 | 0.10 |
| $E_4$ | **0.70** | 0.50 | 0.30 | 0.20 | 0.40 | 0.50 | 0.40 | 0.30 | 0.50 | 0.20 |
| $E_5$ | 0.10 | 0.45 | **0.84** | 0.76 | 0.20 | 0.40 | 0.60 | 0.70 | 0.50 | 0.30 |

TABLE II
THE OPTIMAL PARAMETERS OF PARAMETER-BASED ALGORITHMS AND THE NUMBER OF EXTRA INTERACTIONS FOR PARAMETER TUNING

| Env. | $\text{DP}_{\text{RI}}$ | | DGPA | | $\text{SE}_{\text{RI}}$ | | DBPA | |
|---|---|---|---|---|---|---|---|---|
| | Para. | Cost. | Para. | Cost. | Para. | Cost. | Para. | Cost. |
| $E_1$ | n=298 | 4.85E+09 | n=33 | 4.36E+08 | n=16,$\gamma$=8 | 3.07E+09 | n=102 | 9.88E+08 |
| $E_2$ | n=653 | 2.45E+10 | n=65 | 1.64E+09 | n=32,$\gamma$=12 | 1.20E+10 | n=216 | 4.60E+09 |
| $E_3$ | n=2356 | 3.40E+11 | n=204 | 1.59E+10 | n=105,$\gamma$=25 | 1.20E+11 | n=820 | 6.67E+10 |
| $E_4$ | n=216 | 2.54E+09 | n=28 | 3.17E+08 | n=13,$\gamma$=6 | 1.90E+09 | n=57 | 3.69E+08 |
| $E_5$ | n=881 | 3.12E+10 | n=55 | 1.19E+09 | n=33,$\gamma$=12 | 1.08E+10 | n=326 | 6.77E+09 |
| Env. | GBSE | | DGCPA | | $\text{LELA}_{\text{R}}$ | | $\text{LA}_{\text{OCBA}}$ | |
| | Para. | Cost. | Para. | Cost. | Para. | Cost. | Para. | Cost. |
| $E_1$ | n=1,$\gamma$=7 | 1.80E+08 | n=3,$\gamma$=5 | 4.74E+08 | n=9 | 8.49E+07 | n=11 | 1.45E+08 |
| $E_2$ | n=3,$\gamma$=9 | 1.04E+09 | n=6,$\gamma$=9 | 1.83E+09 | n=17 | 2.88E+08 | n=21 | 4.10E+08 |
| $E_3$ | n=6,$\gamma$=17 | 6.11E+09 | n=17,$\gamma$=20 | 1.55E+10 | n=59 | 3.30E+09 | n=68 | 4.35E+09 |
| $E_4$ | n=1,$\gamma$=5 | 1.38E+08 | n=2,$\gamma$=4 | 2.68E+08 | n=9 | 7.91E+07 | n=12 | 1.35E+08 |
| $E_5$ | n=3,$\gamma$=8 | 8.26E+08 | n=5,$\gamma$=7 | 1.35E+09 | n=24 | 3.70E+08 | n=28 | 4.99E+08 |

property of EPFLA. It is obvious that a small $\tau$ leads to high accuracy and low efficiency, while a large $\tau$ leads to relatively low accuracy and high efficiency. The convergence threshold $\tau$ is set to 0.003 in the following comparisons to satisfy the demand for accuracy.

### C. Comparison of Accuracy

Table IV presents the accuracy of contenders. The $\epsilon$-optimality ensures that all LA schemes converge with pretty high accuracy. Specifically, each scheme can achieve the convergence accuracy no less than 99.3%. Compared with other schemes, EPFLA converges with relatively high accuracy, which verifies the effectiveness of the proposed scheme.

**Remark 4.** *Since the accuracy of LA is ensured to be relatively high by the $\epsilon$-optimality, it is customary to evaluate the performance of LA in some environment by comparing the convergence rate as well the convergence time given that the accuracy is higher than some threshold.*

### D. Comparison of Convergence Rate

Since the accuracy of all schemes is very close, comparisons of the convergence rate are of vital significance.

The convergence rates of different LA are presented in Table V, and Figure V-D illustrates the relative improvements. EPFLA outperforms all deterministic estimator based schemes in five benchmark environments, including $\text{DP}_{\text{RI}}$, DGPA, DBPA, $\text{LELA}_{\text{R}}$, and $\text{LA}_{\text{OCBA}}$. Compared with stochastic estimator based algorithms that contain two tunable parameters, EPFLA performs quite competitive with the classic $\text{SE}_{\text{RI}}$ scheme, and EPFLA converges slower than well-tuned DGCPA and DBPA in most environments. Take environment $E_2$ as an example, the convergence rate of EPFLA is improved by DGCPA and GBSE with $18.41\%$ and $7.10\%$, and is deteriorated by $\text{DP}_{\text{RI}}$, DGPA, DBPA, $\text{LELA}_{\text{R}}$, $\text{LA}_{\text{OCBA}}$, and $\text{SE}_{\text{RI}}$with $200.84\%$, $101.81\%$, $70.76\%$, $35.86\%$, $56.80\%$, and $0.36\%$, respectively. It is noteworthy that the accuracy of EPFLA in environment $E_2$ is $0.999$, which is no less than other algorithms. But the superiority of EPFLA is still obvious considering the cost of parameter tuning for parameter-based algorithms. EPFLA can achieve comparable or even better performance than the parameter-based schemes without relying on any extra interactions.

Meanwhile, compared with the parameter-free schemes PFLA and LFPLA, the improvements of EPFLA on convergence rate is also significant. EPFLA converges faster than both PFLA and LFPLA in all benchmark environments.

### E. Comparison of Convergence Time

The computational complexity in the field of LA is reflected by the convergence time. As presented in Table VI and Figure 5, almost all of the parameter based algorithms can converge in a very short period of time[2]. Except for DBPA where the Bayesian inference is utilized. The calculations of beta distribution based functions consume a certain amount of time, and the convergence time is slightly increased. However, the Monte-Carlo simulations used in PFLA and LFPLA demand enormous computational resources, which leads to a sharp increase in the required time for convergence. As a contrast, EPFLA can achieve parameter-free convergence without using

[2] All simulations are compiled in C++ programming language, under Visual Studio 2013, Windows 10, Intel(R) Core(TM) i7-6700 CPU @3.40GHz, 8.00GB RAM and Beta distributed functions are generated by Boost C++ Libraries.

TABLE III
ACCURACY AND CONVERGENCE RATE OF EPFLA USING DIFFERENT CONVERGENCE THRESHOLD $\tau$ IN BENCHMARK ENVIRONMENTS (250,000 EXPERIMENTS WERE CONDUCTED FOR EACH CONFIGURATION IN EACH ENVIRONMENT)

| Env. | $\tau = 0.005$ | | $\tau = 0.004$ | | $\tau = 0.003$ | | $\tau = 0.002$ | | $\tau = 0.001$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | Iteration | Accuracy | Iteration | Accuracy | Iteration | Accuracy | Iteration | Accuracy | Iteration |
| $E_1$ | 0.999 | 417 | 0.999 | 440 | 0.999 | 471 | 0.999 | 511 | 0.999 | 580 |
| $E_2$ | 0.998 | 739 | 0.999 | 779 | 0.999 | 831 | 0.999 | 906 | 0.999 | 1040 |
| $E_3$ | 0.991 | 1962 | 0.993 | 2079 | 0.996 | 2232 | 0.997 | 2454 | 0.999 | 2822 |
| $E_4$ | 0.999 | 379 | 0.999 | 390 | 0.999 | 426 | 0.999 | 462 | 0.999 | 527 |
| $E_5$ | 0.991 | 501 | 0.993 | 534 | 0.995 | 580 | 0.997 | 617 | 0.999 | 712 |

TABLE IV
ACCURACY OF THE COMPARED ALGORITHMS IN BENCHMARK ENVIRONMENTS (PARAMETER-BASED LA USE OPTIMAL PARAMETER CONFIGURATIONS)

| Env. | $\text{DP}_{\text{RI}}$ | DGPA | $\text{SE}_{\text{RI}}$ | DBPA | GBSE | DGCPA | $\text{LELA}_{\text{R}}$ | $\text{LA}_{\text{OCBA}}$ | PFLA | LFPLA | EPFLA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $E_1$ | 0.995 | 0.997 | 0.997 | 0.996 | 0.997 | 0.997 | 0.997 | 0.998 | 0.997 | 0.999 | 0.999 |
| $E_2$ | 0.994 | 0.996 | 0.996 | 0.994 | 0.996 | 0.996 | 0.996 | 0.998 | 0.999 | 0.998 | 0.999 |
| $E_3$ | 0.993 | 0.995 | 0.995 | 0.993 | 0.995 | 0.995 | 0.995 | 0.997 | 0.996 | 0.997 | 0.996 |
| $E_4$ | 0.996 | 0.997 | 0.998 | 0.996 | 0.997 | 0.998 | 0.997 | 0.998 | 0.999 | 0.999 | 0.999 |
| $E_5$ | 0.994 | 0.997 | 0.997 | 0.994 | 0.997 | 0.997 | 0.997 | 0.997 | 0.997 | 0.997 | 0.995 |

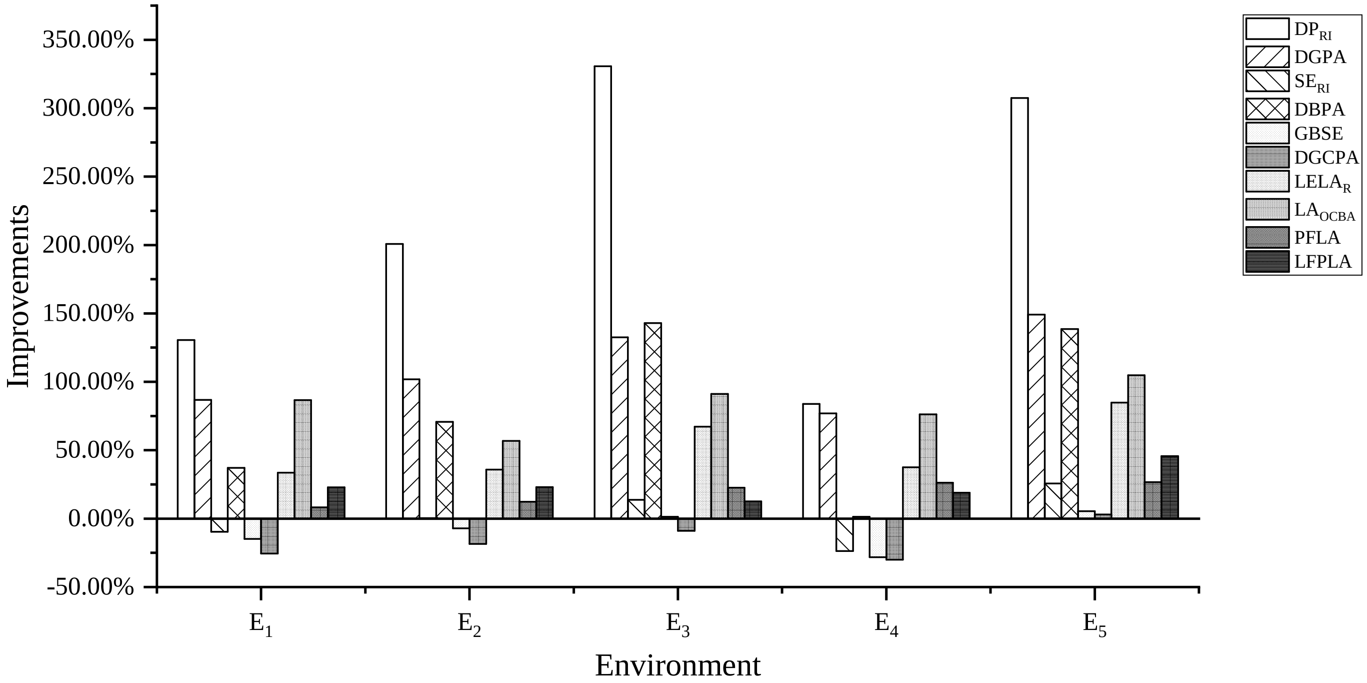


Fig. 4. Convergence rate improvements of the compared algorithms relate to EPFLA in environment $E_1$ to $E_5$, calculate by $\frac{\text{Iteration}_{\text{ComparedScheme}} - \text{Iteration}_{\text{EPFLA}}}{\text{Iteration}_{\text{EPFLA}}}$.

Monte-Carlo simulations or other inefficient techniques. The required convergence time for EPFLA is even less than the parameter-based DBPA scheme. For example, in environment $E_2$, the required time for convergence of PFLA and LFPLA are $1.08E+02$ seconds and $1.33E+02$ seconds, respectively. EPFLA only requires $3.59E-02$ seconds for a right convergence, which is more than three thousand times faster than PFLA and LFPLA.

### F. Discussions

The simulation result above implies the advantages of EPFLA in traditional applications over previous LA. In summary, EPFLA can select the optimal action with fewer interactions within less time. Specifically, the parameter-free property enlarges the usability of EPFLA into interaction-expensive environments. Furthermore, the significant improvements of EPFLA to existing parameter-free LA in convergence rate and convergence time verify the high-efficiency of EPFLA, promoting the potential applicability of EPFLA in both computation resource-limited environments and time-sensitive environments. For example, an efficient intrusion detection system with low energy consumption in wireless sensor networks [44] is desirable, because the wireless sensor networks consist of a large number of resource-constrained devices. Besides, the massive random access problem [45] in machine-to-machine communication networks is time-sensitive since the base sta-

TABLE V
CONVERGENCE RATE OF THE COMPARED ALGORITHMS IN BENCHMARK ENVIRONMENTS (PARAMETER-BASED LA USE OPTIMAL PARAMETER CONFIGURATIONS)

| Env. | $\mathrm{DP_{RI}}$ | DGPA | $\mathrm{SE_{RI}}$ | DBPA | GBSE | DGCPA | $\mathrm{LELA_R}$ | $\mathrm{LA_{OCBA}}$ | PFLA | LFPLA | EPFLA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $E_1$ | 1086 | 880 | 426 | 646 | 401 | 351 | 629 | 879 | 510 | 579 | 471 |
| $E_2$ | 2500 | 1677 | 834 | 1419 | 772 | 678 | 1129 | 1303 | 934 | 1022 | 831 |
| $E_3$ | 9613 | 5191 | 2540 | 5423 | 2262 | 2032 | 3733 | 4265 | 2737 | 2513 | 2232 |
| $E_4$ | 783 | 754 | 325 | 432 | 306 | 298 | 586 | 751 | 538 | 507 | 426 |
| $E_5$ | 2363 | 1445 | 729 | 1384 | 612 | 598 | 1027 | 1188 | 735 | 845 | 580 |

TABLE VI
CONVERGENCE TIME OF THE COMPARED ALGORITHMS IN BENCHMARK ENVIRONMENTS (PARAMETER-BASED LA USE OPTIMAL PARAMETER CONFIGURATIONS)

| Env. | $\mathrm{DP_{RI}}$ | DGPA | $\mathrm{SE_{RI}}$ | DBPA | GBSE | DGCPA | $\mathrm{LELA_R}$ | $\mathrm{LA_{OCBA}}$ | PFLA | LFPLA | EPFLA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $E_1$ | 2.62E-04 | 3.69E-03 | 2.39E-04 | 5.96E-02 | 2.25E-04 | 3.69E-03 | 1.42E-04 | 8.31E-05 | 5.88E+01 | 7.41E+01 | 1.54E-02 |
| $E_2$ | 6.02E-04 | 7.13E-03 | 4.67E-04 | 1.31E-01 | 4.32E-04 | 7.13E-03 | 2.56E-04 | 1.23E-04 | 1.08E+02 | 1.33E+02 | 3.59E-02 |
| $E_3$ | 2.32E-03 | 2.14E-02 | 1.42E-03 | 5.00E-01 | 1.27E-03 | 2.14E-02 | 8.45E-04 | 4.03E-04 | 3.16E+02 | 4.40E+02 | 1.31E-01 |
| $E_4$ | 1.89E-04 | 3.14E-03 | 1.82E-04 | 3.98E-02 | 1.71E-04 | 3.14E-03 | 1.33E-04 | 7.10E-05 | 6.21E+01 | 6.90E+01 | 1.27E-02 |
| $E_5$ | 5.69E-04 | 6.29E-03 | 4.08E-04 | 1.28E-01 | 3.43E-04 | 6.29E-03 | 2.43E-04 | 1.12E-04 | 8.48E+01 | 1.26E+02 | 1.72E-02 |

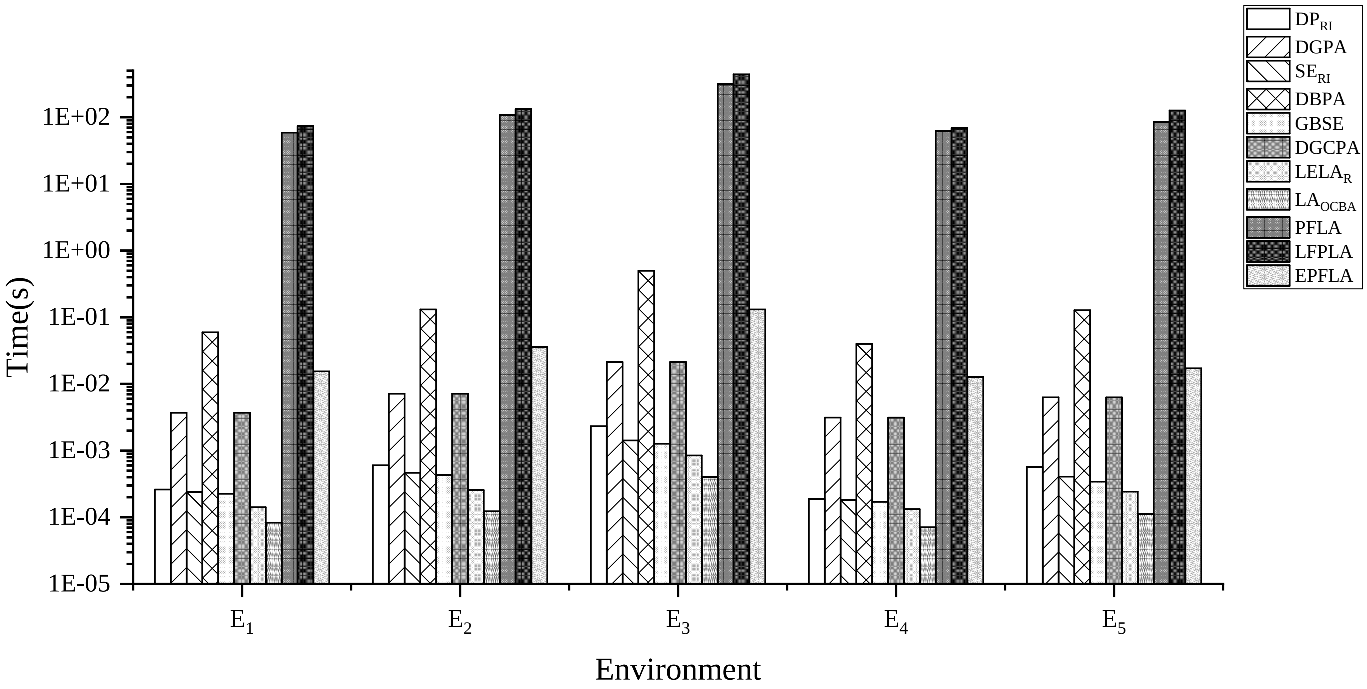


Fig. 5. Comparison of the average time required for convergence of the compared algorithms in benchmark environments

tion has to relieve the radio access network congestion timely, so as to provide quality services.

## VI. Discussion of Applications of LA

As has been discussed in Section I, LA works in non-associative tasks where the actions do not change the environments. In this section, we concretely discuss the scenarios that LA, especially EPFLA can be applied to and evaluate their performance.

To apply LA to a practical problem, one only has to define the action set, the feedback from the environment and the standard identifying the optimal action.

LA have been widely applied to optimization tasks [22]–[25]. In a typical optimization problem, there exists an objective function and a series of restrictions. In the language of LA, the setting of the problem equals the stochastic environment which is deterministic and non-associative. The restrictions are utilized to help to define the action set, i.e., the possible solutions to the optimization problem. The feedback from the environment is the objective function evaluation given some certain action as parameters. In this way, learning the optimal action for an LA is tantamount to solve the optimization problem, or find an approximate solution. Optimization is the basis of many scientific problems. So involving in optimization provides LA with infinite applicable scenarios. Among the applications of LA in various topics, the graph-based one, such as wireless sensor networks, peer-to-peer networks and social networks, is the most profitable field. Some examplary applications are:

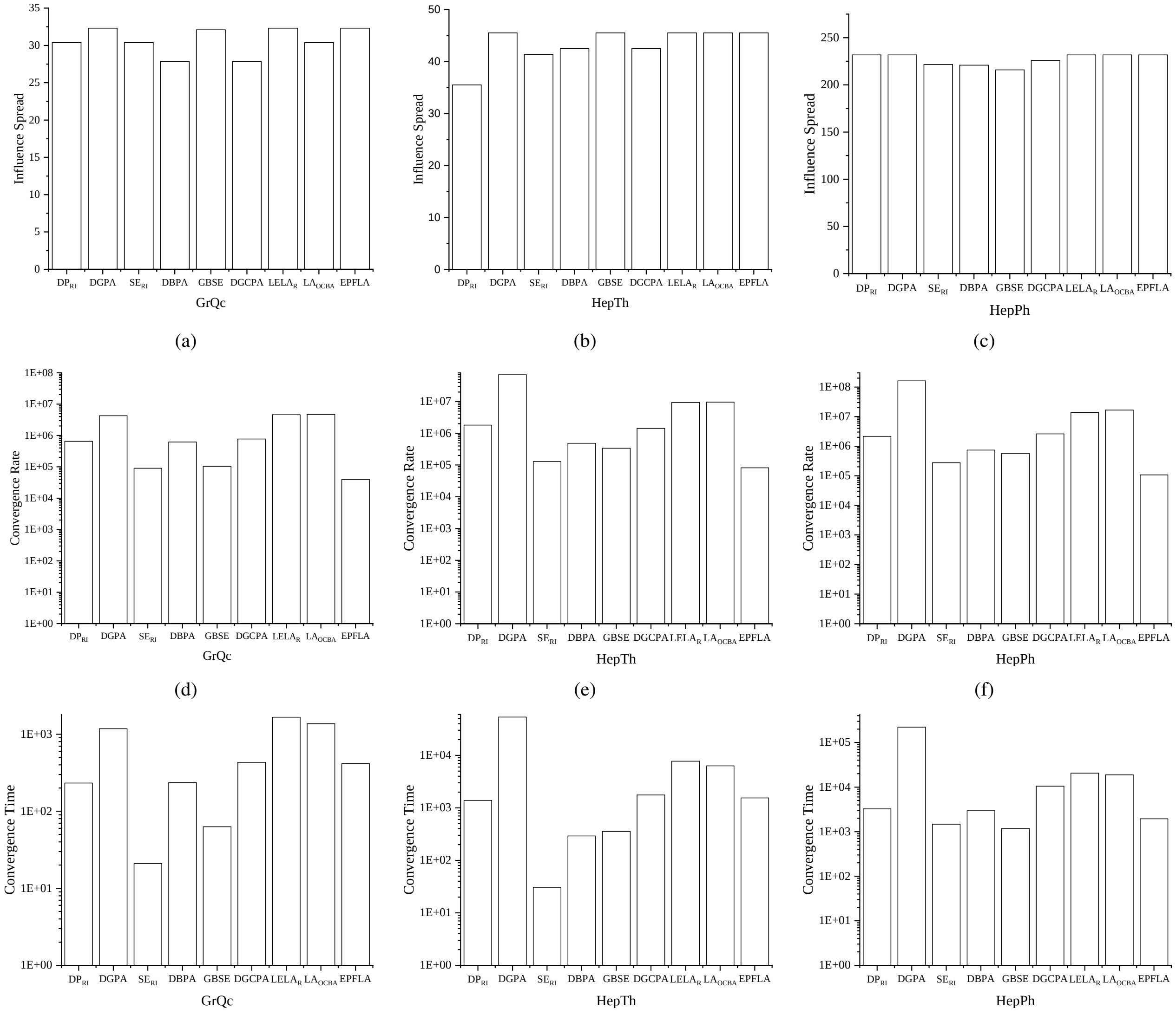
- *Stochastic shortest path routing* [46], [47]. Consider a graph whose nodes are connected by weighted-edges following some probability density functions. For a given source node and a destination node, the stochastic shortest path routing is the path from the source node to the destination node with the smallest expected cumulative weights. The stochastic shortest path routing problem is the abstract of many practical applications such as base station deployment and channel assignment in wireless communications. In the language of LA, a route is simulated from the stochastic graph at each iteration. The action set is defined as the possible next-hops for each node. The feedback from the environment is measured by the difference between the path selected by LA and the current shortest path, thus the path with the smallest expected cumulative weights is the optimal action for LA.
- *Influence maximization* [48], [49]. Influence maximization, promoted by marketing and advertising, aims at selecting a subset of participants in an online social network as ”seeds” to obtain the maximum influence spread. The ”seeds” propagate their information to other participants through connections represented by edges. Utilizing the sub-modular property of information diffusion models, the greedy-based paradigm that identifies the set of ”seeds” by including locally optimal nodes one at a time is a prescriptive solution. In this paradigm, when looking for a new node that maximizes the marginal increment of propagation range, the nodes in a social network are defined as actions for LA. The feedback from the environment is calculated by the influence spread of the node chosen by LA at each iteration. The optimal action is the node with the expected largest influence spread.

The graph-based problems have long been studied as combinatorial optimization. Taking the randomness of real-world tasks into consideration, LA is one of the most efficient and readily applicable solutions to such problems.

To evaluate the performance of EFPLA in real-world applications, we select the influence maximization problem as an example, because: (1) It is a recent hotspot in the research of social networks; (2) The number of actions is large (equals the number of nodes in a social network), by which we can test and verify the effect of EFPLA in complex environments. The detailed problem formulation for influence maximization follows [48], and the cardinal of ”seeds” is set to 1 for convenience. For influence maximization where the number of actions is intimidating and the environment is complex, it is unattainable to tune parameters for parameter-based LA schemes. Thus, the resolution parameter $n$ and the perturbation factor $\gamma$ are arbitrarily set to values averaged from the parameters in benchmarks $E_1 - E_5$. For comparison, the schemes are evaluated by the influence spread, the interaction cost and the required learning time, corresponding to the accuracy, the convergence rate and the convergence time for evaluating an LA scheme respectively. Three widely-used benchmarks in influence maximization, i.e., the GrQc, HepTh, TepPh collaboration networks, are adopted to evaluate the performance of LA schemes in large networks. The number of nodes in GrQc, HepTh, TepPh is $5242$, $9877$, and $12008$, respectively. The influence model is the classic weighted-cascade model where each node is influenced by its neighbors by a weighted probability. We conduct 10 independent simulations for each LA scheme in each benchmark. The conventional parameter-free LA schemes, PFLA and LFPLA, fail to converge due to unaffordable time cost. The averaged results of parameter-based schemes and EPFLA are presented in Figure 6, from which we can draw the following conclusions: (1) The proposed EPFLA scheme always converges to the best solution, i.e., the nodes with the largest influence spread, with the highest convergence rate. (2) When fixing the learning parameters, the parameter-based schemes converge either slowly or incorrectly. Taking DGPA and $\text{SE}_{\text{RI}}$ as examples, DGPA converges to the best node while costs hundreds of times of iterations compared with EPFLA, and $\text{SE}_{\text{RI}}$ converges to suboptimal nodes wrongly causing uncompetitive influence spread. Other schemes also suffer from a similar predicament. (3) Taking advantage of the fast convergence rate, the proposed EFPLA converges in comparable time compared with parameter-based schemes when obtaining the same influence spread. The above observations indicate that the proposed EPFLA scheme outperforms previous LA schemes in the influence maximization problem, revealing that EFPLA is an advisable choice in LA-based real-world applications.

## VII. Conclusion

In this paper, we propose an efficient parameter-free learning automaton scheme in stationary stochastic environments. It is free from the time-consuming pre-training process of traditional parameter-based algorithms. Moreover, it reduces the cost of time and computational resources of existing parameter-free schemes. The proof of $\epsilon$-optimality ensures the convergence of the proposed scheme in every stationary stochastic environment. Simulation results in five benchmark environments verify the superiority of EPFLA over previous works. Compared with parameter-based schemes, EPFLA realizes the comparable performance of the convergence rate without tuning parameters for each stochastic environment. Besides, compared with existing parameter-based schemes, EPFLA achieves great improvements on both convergence rate and convergence time. Our future work will concentrate on the extension of EPFLA into non-stationary stochastic environments and the extension of LA algorithms in associative reinforcement learning problems. Moreover, the parameter-free LA performs properly in unknown environments without pre-training, improving the applicability of LA in applications. Hence, the applications of parameter-free LA in interaction-expensive environments such as information security and medical science are valuable directions for future research.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Fig. 6. Performance of LA schemes in benchmarks for influence maximization. Subfigures (a)-(c) present the results of influence spread, subfigures (d)-(f) present the results of convergence rate, Subfigure (a)-(c), subfigures (g)-(i) present the results of convergence time.

## REFERENCES

[1] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. Vol. 1. No. 1. Cambridge: MIT press, 1998.

[2] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. ”Deep learning.” *Nature* 521.7553 (2015): 436.

[3] Mnih, Volodymyr, et al. ”Human-level control through deep reinforcement learning.” *Nature* 518.7540 (2015): 529.

[4] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[5] Silver, David, et al. ”Mastering the game of go without human knowledge.” Nature 550.7676 (2017): 354.

[6] Beigy, Hamid, and Mohammad Reza Meybodi. ”Utilizing distributed learning automata to solve stochastic shortest path problems.” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 14.05 (2006): 591-615.

[7] Narendra, Kumpati S., and Mandayam AL Thathachar. Learning automata: an introduction. Courier Corporation, 2012.

[8] Esnaashari, Mehdi, and Mohammad Reza Meybodi. ”Dynamic irregular cellular learning automata.” *Journal of computational science* 24 (2018): 358-370.

[9] Zeng, Xianyi, and Zeyi Liu. ”A learning automata based algorithm for optimization of continuous complex functions.” *Information Sciences* 174.3-4 (2005): 165-175.

[10] Vafashoar, Reza , and M. R. Meybodi . ”Cellular learning automata based bare bones PSO with maximum likelihood rotated mutations.” *Swarm and Evolutionary Computation* (2018).

[11] Guo, Haonan, et al. ”A new learning Automata-Based pruning method to train deep neural networks.” *IEEE Internet of Things Journal* 5.5 (2018): 3263-3269.

[12] Fu, King-Sun, and Timothy J. Li. ”Formulation of learning automata and automata games.” *Information Sciences* 1.3 (1969): 237-256.

[13] Thathachar, M. A. L., and P. S. Sastry. ”Learning Automata for Pattern Classification.” *Networks of Learning Automata*. Springer, Boston, MA, 2004. 139-176.

[14] Khomami, Mohammad Mehdi Daliri, et al. ”Minimum positive influence dominating set and its application in influence maximization: a learning automata approach.” *Applied Intelligence* 48.3 (2018): 570-593.

[15] Cuevas, Erik, et al. ”Fast algorithm for multiple-circle detection on images using learning automata.” *IET Image Processing* 6.8 (2012): 1124-1135.

[16] MLA Rezvanian, Alireza , and M. R. Meybodi . ”Sampling algorithms for stochastic graphs: A learning automata approach.” *Knowledge-Based Systems* (2017): 126-144.

[17] Sohrabi, Mohammad Karim , and R. Roshani . ”Frequent itemset mining using cellular learning automata.” *Computers in Human Behavior* 68(2017):244-253.

[18] Khomami, el al. ”A new cellular learning automata-based algorithm for

community detection in complex social networks.” *Journal of computational science* 24 (2018): 413-426.

[19] Morshedlou, Hossein , and M. R. Meybodi . ”A new learning automata based approach for increasing utility of service providers.” *International Journal of Communication Systems* (2017):e3459.

[20] Zhu, Junpeng, et al. ”Learning automata-based methodology for optimal allocation of renewable distributed generation considering network reconfiguration.” *IEEE Access* 5 (2017): 14275-14288.

[21] Saghiri, Ali Mohammad, and Mohammad Reza Meybodi. ”Open asynchronous dynamic cellular learning automata and its application to allocation hub location problem.” *Knowledge-Based Systems* 139 (2018): 149-169.

[22] Narendra, Kumpati S., and Mandayam AL Thathachar. ”Learning automata-a survey.” *IEEE Transactions on systems, man, and cybernetics* 4 (1974): 323-334.

[23] Thathachar, Mandayam AL, and P. Shanti Sastry. ”Varieties of learning automata: an overview.” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 32.6 (2002): 711-722.

[24] Najim, Kaddour, and Alexander S. Poznyak. Learning automata: theory and applications. Elsevier, 2014.

[25] Rezvanian, Alireza, et al. Recent advances in learning automata. Vol. 754. Springer, 2018.

[26] Oommen, B. John, and Joseph K. Lanctôt. ”Discretized pursuit learning automata.” *IEEE Transactions on systems, man, and cybernetics* 20.4 (1990): 931-938.

[27] Agache, Mariana, and B. John Oommen. ”Generalized pursuit learning schemes: New families of continuous and discretized learning automata.” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 32.6 (2002): 738-749.

[28] Zhang, Xuan, Ole-Christoffer Granmo, and B. John Oommen. ”On incorporating the paradigms of discretization and Bayesian estimation to create a new family of pursuit learning automata.” *Applied intelligence* 39.4 (2013): 782-792.

[29] Zhang, Junqi, Cheng Wang, and MengChu Zhou. ”Last-position elimination-based learning automata.” *IEEE Transactions on systems, man, and cybernetics* 44.12 (2014): 2484-2492.

[30] Zhang, Xuan, Ole-Christoffer Granmo, and B. John Oommen. ”On incorporating the paradigms of discretization and Bayesian estimation to create a new family of pursuit learning automata.” *Applied intelligence* 39.4 (2013): 782-792.

[31] Papadimitriou, Georgios I., Maria Sklira, and Andreas S. Pomportsis. ”A new class of/spl epsi/-optimal learning automata.” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34.1 (2004): 246-254.

[32] Jiang, Wen, et al. ”A new prospective for learning automata: a machine learning approach.” *Neurocomputing* 188 (2016): 319-325.

[33] Ge, Hao, et al. ”A novel estimator based learning automata algorithm.” *Applied Intelligence* 42.2 (2015): 262-275.

[34] Ge, Hao, et al. ”A parameter-free gradient Bayesian two-action learning automaton scheme.” *Proceedings of the 2015 International Conference on Communications, Signal Processing, and Systems*. Springer, Berlin, Heidelberg, 2016.

[35] Ge, Hao. ”A Parameter-Free Learning Automaton Scheme.” arXiv preprint arXiv:1711.10111 (2017).

[36] A. B. Owen, ”Monte carlo theory, methods and examples,” 2013, unpublished. [Online]. Available: http://statweb.stanford.edu/ $\sim$ owen/mc/.

[37] Guo, Ying, Hao Ge, and Shenghong Li. ”A loss function based parameterless learning automaton scheme.” *Neurocomputing* (2017).

[38] S. Chris, Bayesian A/B testing at VWO, 2015, [online; accessed 20 May 2016]. Available: http://cdn2.hubspot.net/hubfs/310840/VWO_SmartStats_technical_whitepaper.pdf.

[39] Neyman, Jerzy. ”X—Outline of a Theory of Statistical Estimation Based on the Classical Theory of Probability.” *Phil. Trans. R. Soc. Lond. A* 236.767 (1937): 333-380.

[40] Box, George EP, and George C. Tiao. Bayesian inference in statistical analysis. John Wiley & Sons, 2011.

[41] Pierre-Simon, Marquis De Laplace. A Philosophical Essay on Probabilities. Cosimo, Inc., 2007.

[42] Olver, Frank WJ, et al. ”NIST handbook of mathematical functions, US Department of Commerce, National Institute of Standards and Technology.” (2010).

[43] Oommen, B. John, and Mariana Agache. ”Continuous and discretized pursuit learning schemes: Various algorithms and their comparison.” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 31.3 (2001): 277-287.

[44] Misra, Sudip, P. Venkata Krishna, and Kiran Isaac Abraham. ”A simple learning automata-based solution for intrusion detection in wireless sensor networks.” *Wireless Communications and Mobile Computing* 11.3 (2011): 426-441.

[45] Di, Chong, et al. ”Learning Automata based Access Class Barring Scheme for Massive Random Access in Machine-to-Machine Communications.” *IEEE Internet of Things Journal* (2018).

[46] Misra, Sudip, and B. John Oommen. ”Dynamic algorithms for the shortest path routing problem: learning automata-based solutions.” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 35.6 (2005): 1179-1192.

[47] Guo, Ying, et al. ”Learning automata-based algorithms for solving the stochastic shortest path routing problems in 5G wireless communication.” *Physical Communication* 25 (2017): 376-385.

[48] Ge, Hao, et al. ”Learning automata based approach for influence maximization problem on social networks.” *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2017.

[49] Di, Chong, et al. ”Maximizing Influence on Social Networks with Conjugate Learning Automata.” *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019.