Bayesian Inference Based Learning Automaton Scheme in Fuzzy Environments

Chong Di[1], Fang-Qi Li[1], and Sheng-Hong Li[1*]

[1] *School of Cyber Security, School of Electronic Information and Electrical Engineering, Artificial Intelligence Institute, Shanghai Jiao Tong University, Shanghai 200240, China.*
[*] *Corresponding author: Sheng-Hong Li (shli@sjtu.edu.cn).*

Abstract

Learning automaton (LA), a powerful tool in reinforcement learning, is of crucial importance for its adaptivity in uncertain environments and its applicability in various engineering fields. In particular, the LA is a decision maker that adaptively explores the optimal action that maximizes the rewards among all possible choices by interacting with the environment. Great efforts have been made to improve the performance of LA in the P-model environments which provide only two types of feedback: reward and penalty. However, in many practical scenarios, the feedback from the environment is fuzzy rather than two-folded. This paper studies the LA in fuzzy environments, where the environment can conduct mediocre signals apart from crisp positive or negative ones. A novel Bayesian Inference based LA that is capable of functioning in fuzzy environments is proposed, which we name after BIFLA to emphasize its utility in fuzzy environments.

The Bayesian inference is utilized to model the environments response behavior corresponding to each action, collaborating with the Kullback-Leibler divergence in achieving the adaptive decision-making and learning of LA. The BIFLA scheme is proved to be ϵ-optimal and is evaluated to be superior to established LA frameworks by comprehensive experiments.

Keywords:
Learning automaton, Bayesian inference, Fuzzy system

1. Introduction

Learning automaton (LA), one of the most powerful tools in reinforcement learning (RL) [1], is a self-adaptive decision maker that can adaptively explore the optimal action that maximizes the rewards among all possible choices by interacting with the environment [2]. Early studies of LA date back to the 1960s [3], but it has surged much-renewed interest owing to its modern applications in a broad range of engineering contexts such as cloud computing [4]-[6], signal processing [7]-[9], optimization [10]-[14], and data mining [15]-[18]. The applications of LA have been comprehensively reviewed in [19].

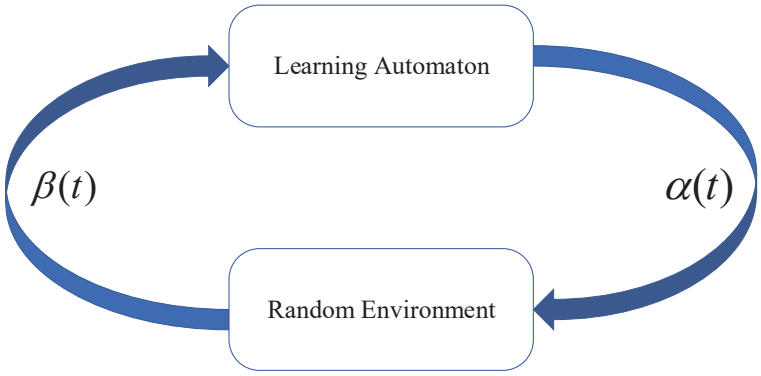


Figure 1: LA interacts with the random environment.

As shown in Figure 1 [2], during the t-th interaction with the random environment, an LA selects an action $\alpha(t)$, sends it to the environment and gets feedback $\beta(t)$, which is then utilized to update the internal state of the LA. The LA converges to the final state after a number of interactions, i.e., it learns the optimal action to interact with the given environment. From the transition or updating functions between states, LA can be characterized as fixed structure stochastic automaton (FSSA) [20] or variable structure stochastic automaton (VSSA) [21]. As the names suggest, the transition between states of FSSA is deterministic, while that of VSSA is stochastic. FSSA is the prototype of LA, and VSSA improves FSSA by being more flexible and having broader application scenarios [21].

The feedback from a P-model environment given an action is either 0 or 1, i.e., an explicit reward or penalty. As the elementary situation, the

interaction between an LA and a P-model environment has been abundantly studied. The last decade has witnessed the emergence of fruitful learning schemes together with theoretical achievements for a VSSA interacting with a P-model environment [19].

However, the P-model or the duality of feedback is an over-simplified setting for many practical applications. For example, one can not easily claim whether the influence of a specific setting for a system is absolutely positive or negative. Instead, the performance of the system is described more naturally and conveniently with non-binary indicators. Then the evaluation of the current setting of the system is *fuzzy*.[1] For fuzzy scenarios, the environment can conduct mediocre signals apart from crisp positive or negative ones. Hitherto, researches about the fuzzy environment in the discipline of LA have been scanty. Jamalian et al. [22] introduced the concept of the triple level environment with three kinds of feedback: a reward, a small scale penalty, and a large scale penalty. This is an instantiation of the general fuzzy environment. For the studied triple level environment, Jamalian et al. proposed a corresponding FSSA algorithm named TILA. Inspired by Jamalian et al.'s studies, Jiang and Li proposed a general method that makes P-model FSSA schemes able to function in triple level environments [23].

To the best of our knowledge, studies about an LA interacting with a fuzzy environment have been few apart from the investigations of FSSA in the triple level environment. Hence a more general formulation of the fuzzy environment with the application of updated VSSA might yield more informative outcomes. In view of this, we aim to propose an effective VSSA scheme that can learn the optimal action in fuzzy environments. The contributions of this paper are as follows.

1. We present a novel LA scheme, BIFLA for the interaction between a VSSA and a fuzzy environment. This proposal expands the application of LA to more general scenarios.

2. Compared with traditional schemes designed for P-model environments, the proposed BIFLA scheme is brand-new, where the Bayesian inference and the Kullback-Leibler divergence are utilized to realize the action selection and state update.

[1]In some literature, the environments with more than two types of feedback used to be denoted by Q-model or S-model environments. In this paper, we adopt the concept of 'fuzzy environments' to characterize them.

3. A rigorous proof is provided to ensure the ϵ-optimality of BIFLA.

4. Comprehensive comparisons among the state-of-the-art schemes in P-model environments are given to validate the theoretical analyses and demonstrate the superiority of the proposed BIFLA scheme. Simulations in various fuzzy environments are also conducted to verify the effectiveness of BIFLA.

The rest of this paper is organized as follows. Section 2 is dedicated to the problem formulation and a review of the VSSA schemes in P-model environments. In Section 3, we present the Bayesian inference based scheme in fuzzy environments and give the proof of ϵ-optimality. Section 4 provides the experimentations that verify the advantages of the proposed scheme. Finally, Section 5 concludes this paper.

2. Preliminaries

2.1. Formulation of the Problem

The formulation of an automaton with finite actions interacting with a fuzzy environment is a triplet $< A, B, \mathbf{D} >$, where

- $A = \{\alpha_1, \alpha_2, \cdots, \alpha_R\}$ is the finite set of actions. The action selected at the t-th interaction with the environment is denoted by $\alpha(t) \in A$.

- $B = \{\beta_0, \beta_1, \cdots, \beta_Q\}, (Q \geq 1)$ is the set of feedback from the fuzzy environment to the automaton. The value of β_q is a fuzzy measure of penalty or reward. It is usually convenient to exert the assumption that $\beta_q = \frac{q}{Q}$. As the extreme cases, $\beta_0 = 0$ and $\beta_Q = 1$ denote a crisp penalty and a crisp reward respectively. The feedback at the t-th iteration is denoted by $\beta(t) \in B$.

- $\mathbf{D} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \cdots \\ \mathbf{d}_R \end{bmatrix}$ is the matrix of reward probabilities in the fuzzy environment, where $\mathbf{d}_r = [d_{r,0}, d_{r,2}, \cdots, d_{r,Q}]$ is the reward probability vector of action $\alpha_r \in A$ which is subject to normalization and non-negativity

(so there are Q independent components in $\mathbf{d}_r$), where $d_{r,q}$ is the probability of the environment giving feedback β_q given the action α_r during any iteration, i.e.,

$$d_{r,q} = \Pr\{\beta(t) = \beta_q | \alpha(t) = \alpha_r\}. \tag{1}$$

For $\alpha_r \in A$, it is the expectation of rewards in the environment that reflects the profit of choosing this action:

$$\mathrm{E}[\beta(t)|\alpha(t) = \alpha_r] = \sum_{q=0}^{Q} d_{r,q}\beta_q = \frac{1}{Q}\sum_{q=0}^{Q} d_{r,q} \cdot q. \tag{2}$$

The expectations of rewards for all actions are collected in the rewards vector $\mathbf{\Xi} = [\xi_1, \xi_2, \cdots, \xi_R]^{\mathrm{T}}$, where $\xi_r = \mathrm{E}[\beta(t)|\alpha(t) = \alpha_r], \forall \alpha_r \in A$.

By interacting with the fuzzy environment, the LA aims to identify the optimal action α_m with the highest expectation of rewards, i.e.,

$$m = \arg \max_{r=1,2,\cdots,R} \xi_r = \arg \max_{r=1,2,\cdots,R} \sum_{q=0}^{Q} d_{r,q} \cdot q. \tag{3}$$

When $\mathbf{D}$ is independent of time, the stochastic environment is stationary. Otherwise, it is a non-stationary environment with $\mathbf{D}$ a function of t. Since the performance of an LA in a stationary environment forms the basis of further generalization to non-stationary environments, the following discussions focus on the stationary environment.

2.2. VSSA in P-model Environments

In P-model environments, the above statement of the problem is still tenable by setting $Q = 1$, thus $B = \{0, 1\}$. The normalization condition reduces the reward probability matrix to a column vector

$$\mathbf{D} = [d_{1,1}, d_{2,1}, \cdots, d_{R,1}]^{\mathrm{T}}. \tag{4}$$

And accordingly

$$\begin{cases} \Pr\{\beta(t) = 1|\alpha(t) = \alpha_r\} = d_{r,1} \\ \Pr\{\beta(t) = 0|\alpha(t) = \alpha_r\} = d_{r,0} = 1 - d_{r,1} \end{cases}. \tag{5}$$

In this subsection, some representative VSSA schemes in P-model environments are reviewed in order to build a primary comprehension of their operating mechanisms and to estimate their possible applicability in fuzzy environments.

The framework for VSSA was first studied in [21], where a popular learning process based on the action probability vector $\mathbf{P}$ was proposed. For an action set with R actions, $\mathbf{P}(t)$ has R non-negative components $p_r(t)$, $r = 1, 2, \cdots, R$ that sum up to one, where $p_r(t)$ represents the probability that an LA chooses the r-th action at the t-th iteration.

This framework consists of three phases: i). selecting an action according to $\alpha(t) \sim \mathbf{P}(t)$; ii). interacting with the environment and getting the feedback $\beta(t)$; iii). updating the action probability vector into $\mathbf{P}(t+1)$. The VSSA gets converged when the maximum action probability in $\mathbf{P}$ is greater than a predefined threshold $\mathcal{V}$, i.e., $\max_r \{p_r(t)\} \geq \mathcal{V}$. Adhering to the framework, various schemes are proposed to speed up the convergence of VSSA. Particularly, the family of estimator algorithms [24]-[33] that exploits historical information about the environment to guide the update is the most prevalent one.

For estimator algorithms, there is an estimator vector $\mathbf{E} = [e_1, e_2, \cdots, e_R]$, where e_r is the estimate of the reward for α_r. From the perspective of the estimator, LA algorithms can be divided into various classes: the maximum likelihood estimate based class including the discretized pursuit scheme ($\mathrm{DP_{RI}}$) [24], the discretized generalized pursuit LA scheme (DGPA) [25], the stochastic estimator algorithm ($\mathrm{SE_{RI}}$) [26], the last-position elimination-based LA scheme (LELA) [27], and the optimal budget allocation based LA ($\mathrm{LA_{OCBA}}$) [28]; the confidence interval estimate based class such as discretized generalized confidence pursuit algorithm (DGCPA) [29], and the frequency-based algorithm with confidence interval (F-LA) [30]; the Bayesian estimate based class such as the discretized bayesian pursuit algorithm (DBPA) [31], the parameter-free LA (PFLA) [32], the loss function based parameterless LA (LFPLA) [33], the Bayesian-based algorithm with HPD interval (B-HPD-LA), and the Bayesian-based algorithm with ET interval [30]. The estimator-based LA schemes above are designed to function only when the random environment returns a binary signal to the automaton. Taking the maximum likelihood estimate based class for an example, the estimate e_r of the rewards of action α_r is defined as

$$e_r = \frac{W_r}{Z_r}, \tag{6}$$

where W_r and Z_r are the numbers of times α_r being rewarded and selected respectively. Apparently, the estimate cannot be calculated in a fuzzy environment as W_r is unavailable. For other classes of estimators, a similar predicament also exists. Besides, the estimator vector is used to update the action probability vector $\mathbf{P}$, and for the majority of estimator algorithms, $\mathbf{P}$ is updated only when the LA receives a reward, i.e., $\beta(t) = 1$. However, the variety of feedback in fuzzy environments makes the update of $\mathbf{P}$ intractable.

3. Bayesian Inference Based Learning Automaton Scheme in Fuzzy Environments

In this section, we propose the Bayesian inference based LA for fuzzy environments, named BIFLA. Bayesian inference is invoked to characterize the performances of actions in fuzzy environments. To measure the difference between actions, the Kullback-Leibler divergence is utilized. Intuitively, the action with the highest estimation of rewards as well as the distinguishability from all the other actions is the optimal choice. In this section, the preliminary knowledge of Bayesian inference for the LA in fuzzy environments is reviewed, followed by the proposal of BIFLA in fuzzy environments.

3.1. Bayesian Inference of LA in Fuzzy Environments

Suppose the r-th action is selected at the t-th iteration, i.e., $\alpha(t) = \alpha_r$, then the feedback $\beta(t)$ follows:

$$\Pr\{\beta(t) = \beta_q | \alpha(t) = \alpha_r\} = d_{r,q}. \tag{7}$$

Drop the assumption $\beta_q = \frac{q}{Q}$ and return to the general case, where different β_i and β_j represent different feedback. Then the interaction with a fuzzy environment forms a generalized Bernoulli trial and the feedback $\beta(t)$ is a random variable following the categorical distribution with parameter $\mathbf{d}_r$.

For an action α_r that has interacted with the environment for n times, the feedback at different interactions are independent, identically distributed (i.i.d.) random variables in a stationary environment. Therefore the feedback vector $\mathbf{f}_r(n) = [f_{r,0}(n), f_{r,1}(n), \cdots, f_{r,Q}(n)]$ follows a multinomial distribution with parameters n and $\mathbf{d}_r$:

$$\mathbf{f}_r(n) \sim M(n, \mathbf{d}_r) \tag{8}$$

where $f_{r,q}(n)$ is the number of times when the feedback is β_q provided that the action α_r has been selected for n times. In Bayesian inference, the Dirichlet

distribution is the conjugate prior of the categorical distribution and multinomial distribution[34]. Take the feedback vector $\mathbf{f}_r(n)$ as the parameter of the Dirichlet distribution, the probability density function (PDF) with respect to Lebesgue measure on the Euclidean space $\mathbb{R}^{Q+1}$ is

$$f\Big(\mathbf{x}_r|\mathbf{f}_r(n)\Big) = \frac{\Gamma(n)}{\prod_{q=0}^{Q}\Gamma\big(f_{r,q}(n)\big)}\prod_{q=0}^{Q} x_q^{f_{r,q}(n)-1}, \tag{9}$$

where $\mathbf{x}_r = [x_{r,0}, x_{r,1}, \cdots, x_{r,Q}]$ belongs to the standard $(Q+1)$ simplex and

$$\mathrm{E}[x_{r,q}] = \frac{f_{r,q}}{n}. \tag{10}$$

For different actions α_i and $\alpha_j \in A$, given the feedback vector $\mathbf{f}_i(n_i)$ and $\mathbf{f}_j(n_j)$, one can compute the Kullback-Leibler divergence, or the relative entropy between Dirichlet distributions $f\big(\mathbf{x}|\mathbf{f}_i(n_i)\big)$ and $f\big(\mathbf{x}|\mathbf{f}(n_j)\big)$ as follows:

$$\begin{aligned}
&D_{KL}\Big\{f\big(\mathbf{x}|\mathbf{f}_i(n_i)\big)\|f\big(\mathbf{x}|\mathbf{f}_j(n_j)\big)\Big\}\\
&= \int f\big(\mathbf{x}|\mathbf{f}_i(n_i)\big)\log\frac{f\big(\mathbf{x}|\mathbf{f}_i(n_i)\big)}{f\big(\mathbf{x}|\mathbf{f}_j(n_j)\big)}\mathrm{d}\mathbf{x}\\
&= \mathrm{E}_{f(\mathbf{x}|\mathbf{f}_i(n_i))}\left[\log f\big(\mathbf{x}|\mathbf{f}_i(n_i)\big) - \log f\big(\mathbf{x}|\mathbf{f}_j(n_j)\big)\right]\\
&= \mathrm{E}_{f(\mathbf{x}|\mathbf{f}_i(n_i))}\Bigg[\log\Gamma(n_i) - \sum_{q=0}^{Q}\log\Gamma\big(f_{i,q}(n_i)\big) + \sum_{q=0}^{Q}\big(f_{i,q}(n_i)-1\big)\log x_q\\
&\qquad\qquad - \log\Gamma(n_j) + \sum_{q=0}^{Q}\log\Gamma\big(f_{j,q}(n_j)\big) - \sum_{q=0}^{Q}\big(f_{j,q}(n_j)-1\big)\log x_q\Bigg]\\
&= \log\Gamma(n_i) - \sum_{q=0}^{Q}\log\Gamma\big(f_{i,q}(n_i)\big) - \log\Gamma(n_j) + \sum_{q=0}^{Q}\log\Gamma\big(f_{j,q}(n_j)\big)\\
&\qquad\qquad + \sum_{q=0}^{Q}\big(f_{i,q}(n_i) - f_{j,q}(n_j)\big)\mathrm{E}_{f(\mathbf{x}|\mathbf{f}_i(n_i))}\left[\log x_q\right]\\
&= \log\Gamma(n_i) - \sum_{q=0}^{Q}\log\Gamma\big(f_{i,q}(n_i)\big) - \log\Gamma(n_j) + \sum_{q=0}^{Q}\log\Gamma\big(f_{j,q}(n_j)\big)\\
&\quad + \sum_{q=0}^{Q}\big(f_{i,q}(n_i) - f_{j,q}(n_j)\big)\left[\psi\big(f_{i,q}(n_i)\big) - \psi\big(\sum_{q=0}^{Q} f_{i,q}(n_i)\big)\right],
\end{aligned} \tag{11}$$

where $\psi(\cdot)$ is the digamma function.

Utilizing the Kullback-Leibler divergence, we can measure the difference between two Dirichlet distributions. That is, given the feedback vectors $\mathbf{f}_r(n_r), \alpha_r \in A$, the performance of each action in the fuzzy environment can be readily distinguished. However, the Kullback-Leibler divergence cannot reflect the difference in the rewards between actions that is what the LA concerns. In view of this, we use the estimation of rewards to evaluate the rewards of each action α_i in the fuzzy environment, which is defined as:

$$\hat{d}_r(n) = \frac{1}{n} \sum_{q=0}^{Q} f_{r,q}(n_r)\beta_q. \tag{12}$$

Furthermore, to simplify the computation of the Kullback-Leibler divergence, we propose to transform the fuzzy environment into an equivalent binary environment, i.e., re-define the set of feedback from the fuzzy environment as a two-tuple while the expectation of the rewards is not changed. Recall the assumption $\beta_q = \frac{q}{Q}$, $\forall \beta_q \in B$. Re-define the set of feedback as $\tilde{B} = \{0, 1\}$ which is the same as a P-model environment. The feedback vector is transformed from $\mathbf{f}_r(n) = [f_{r,0}(n), f_{r,1}(n), \cdots, f_{r,Q}(n)]$ into $\tilde{\mathbf{f}}_r(n) = [\tilde{f}_{r,0}(n), \tilde{f}_{r,1}(n)]$, where

$$\tilde{f}_{r,0}(n) = \sum_{q=0}^{Q} f_{r,q}(n)(1 - \beta_q) \tag{13}$$

and

$$\tilde{f}_{r,1} = \sum_{q=0}^{Q} f_{r,q}(n)\beta_q. \tag{14}$$

Then the transformed estimation of rewards is:

$$\tilde{\hat{d}}_r(n) = \frac{\tilde{f}_{r,1}}{n} \tag{15}$$

The following theorem examines the invariance of the expectation of rewards with respect to the transformation.

Theorem 1. *The fuzzy environment with* $B = \{\beta_0, \beta_1, \cdots, \beta_Q\}, (Q \geq 1, \beta_q = \frac{q}{Q})$ *and the transformed environment with* $\tilde{B} = \{0, 1\}$ *possess the same optimal action with the transformation rule of feedback vectors introduced in (13) and (14).*

Proof. For action α_r with the reward probability vector $\mathbf{d}_r = [d_{r,0}, d_{r,2}, \cdots, d_{r,Q}]$, by (2), the expectation of rewards of action α_r is $\sum_{q=0}^{Q} d_{r,q}\beta_q$. In the transformed environment, the transformation of the feedback vector implies the transformation of the reward probability vector. That is, the transformed reward probability vector $\tilde{\mathbf{d}}_r = [\tilde{d}_{r,0}, \tilde{d}_{r,1}]$ is given by

$$\tilde{d}_{r,0} = \sum_{q=0}^{Q} d_{r,q}(1 - \beta_q) \tag{16}$$

and

$$\tilde{d}_{r,1} = \sum_{q=0}^{Q} d_{r,q}\beta_q. \tag{17}$$

Then, the expectation of rewards for action α_r in the transformed environment is still $\sum_{q=0}^{Q} d_{r,q}\beta_q$, which is equivalent to that in the fuzzy environment. Thus, the transformation does not change the expectations of rewards of actions, i.e., two environments share the same optimal action. □

By Theorem 1, the multinomial distribution with parameters n and $\mathbf{d}_r$ is transformed into a binomial distribution with parameters n and $\tilde{\mathbf{d}}_r$, whose conjugate prior is a binary Dirichlet distribution, i.e., a Beta distribution [34].

3.2. Bayesian Inference Based Learning Automaton Scheme in Fuzzy Environments

Base on the discussion above, we propose the BIFLA scheme for fuzzy environments, which consists of four parts: the initialization, the action selection, the state update, and the convergence judgment. To help illustration, we first present BIFLA in two-action fuzzy environments and then extend it into general fuzzy environments.

3.2.1. Two-Action Fuzzy Environments

Consider an LA with two available actions interacting with a fuzzy environment with $(Q + 1)$ types of feedback, i.e.,

$$A = \{\alpha_1, \alpha_2\}, \tag{18}$$

$$B = \{\beta_0, \beta_1, \cdots, \beta_Q\}, \tag{19}$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}, \tag{20}$$

where $\mathbf{d}_1 = [d_{1,0}, d_{1,1}, \cdots, d_{1,Q}]$ and $\mathbf{d}_2 = [d_{2,0}, d_{2,1}, \cdots, d_{2,Q}]$.

1) Initialization

In the beginning, the feedback vector for each action is $\mathbf{0}$. However, for the majority of LA schemes, there is an initialization process that chooses each action for some times to boost the convergence. In our scheme, a useful trick known as the technique of optimistic initial values is adopted [1]. That is, the initial feedback vector $\mathbf{f}_r$ for each action α_r is indiscriminately set to $[2, 2, \cdots, 2]$.

2) Action Selection

As stated in [35], in two-action environments, selecting the action with less number of interactions with the environment benefits the convergence. Thus, at each iteration t, BIFLA adopts the following action selection strategy:

$$\alpha(t) = \begin{cases} \alpha_1 & \text{if} \quad n_1(t) < n_2(t) \\ \alpha_2 & \text{if} \quad n_1(t) > n_2(t) \\ \text{random selection} & \text{if} \quad n_1(t) = n_2(t) \end{cases}, \tag{21}$$

where $n_r(t)$ is the number that α_r being selected until the t-th iteration.

3) State Update

Suppose action α_1 is selected at the t-th iteration, the fuzzy environment responds feedback $\beta(t)$ to the LA by (1). Then, the feedback vector $\mathbf{f}_1$ is updated as follows:

$$f_{1,q}(n_1 + 1) = \begin{cases} f_{1,q}(n_1) + 1 & \text{if} \quad \beta_q = \beta(t) \\ f_{1,q}(n_1) & \text{if} \quad \beta_q \neq \beta(t) \end{cases}. \tag{22}$$

Vice versa for $\mathbf{f}_2$.

Using (13) and (14), the feedback vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ are transformed into $\tilde{\mathbf{f}}_1$ and $\tilde{\mathbf{f}}_2$, respectively. Then, the estimations of rewards $\tilde{\hat{d}}_1$ and $\tilde{\hat{d}}_2$ are calculated by (15). Given $\tilde{\mathbf{f}}_1$ and $\tilde{\mathbf{f}}_2$ as parameters of the Dirichlet distributions $Dir(\tilde{\mathbf{f}}_1)$ and $Dir(\tilde{\mathbf{f}}_1)$, the Kullback-Leibler divergence is computed from (11). Define the Kullback-Leibler divergence factor $\tilde{D}_{KL}^{1,2}$ as a symmetric measure where

$$\tilde{D}_{KL}^{1,2} = \min\left\{ D_{KL}\left\{ f(x|\tilde{\mathbf{f}}_1) \| f(x|\tilde{\mathbf{f}}_2) \right\}, D_{KL}\left\{ f(x|\tilde{\mathbf{f}}_2) \| f(x|\tilde{\mathbf{f}}_1) \right\} \right\}. \tag{23}$$

4) Convergence Judgment

The estimations $\tilde{\hat{d}}_1$ and $\tilde{\hat{d}}_2$ reflect the rewards of actions in the fuzzy environment, and the Kullback-Leibler divergence factor measures the difference of the performance between actions. Set a threshold $\mathcal{T}$, if $\tilde{D}_{KL}^{1,2}$ is greater than $\mathcal{T}$, we claim the distinguishability between actions. Thence a discrimination can be made. In the two-action environment, the convergence judgment is straightforward, i.e., the optimal action α_m is obtained by

$$\text{if} \quad \tilde{D}_{KL}^{1,2} \geq \mathcal{T} \quad \text{then} \quad \alpha_m = \begin{cases} \alpha_1 & \text{if} \quad \tilde{\hat{d}}_1 > \tilde{\hat{d}}_2 \\ \alpha_2 & \text{if} \quad \tilde{\hat{d}}_1 < \tilde{\hat{d}}_2 \end{cases}. \tag{24}$$

3.2.2. Multi-Action Fuzzy Environments

It is straightforward to generalize BIFLA into multi-action fuzzy environments with the three components as follows:

1) Initialization

The technique of optimistic initial values is kept to initialize the feedback vector $\mathbf{f}_r$ for each action α_r, where $\mathbf{f}_r$ is set to $[2, 2, \cdots, 2]$.

2) Action Selection Strategy and State Update

Being different from two-action environments, in multi-action cases, the trade-off between exploration and exploitation is an inevitable dilemma in RL [1]. To maximize the cumulative rewards, an agent must examine the actions that it has tried and found to be superior in producing rewards, i.e., *exploit* what it has already learned. Meanwhile, the agent has to try actions that have been chosen for few times in order to obtain sufficient information for every action that is possibly optimal, i.e., to *explore* all possibilities. Thus, it is necessary to balance exploration and exploitation as each action must be tried for plentiful times in a stochastic environment. In view of this, we propose an action selection as well as state update strategy for BIFLA in multi-action environments based on the Kullback-Leibler divergence factor named the KLD-greedy strategy. The components of KLD-greedy strategy are as follows.

- *Optimal action estimation:*

  Suppose action α_r is selected at the t-th iteration for the $(n_r + 1)$-th time, after receiving the feedback $\beta(t)$ from the fuzzy environment, the

feedback vector $\mathbf{f}_r$ is updated as follows:

$$f_{r,q}(n_r+1) = \begin{cases} f_{r,q}(n_r)+1 & \text{if} \quad \beta_q = \beta(t) \\ f_{r,q}(n_r) & \text{if} \quad \beta_q \neq \beta(t) \end{cases}. \tag{25}$$

Transform the feedback vector $\mathbf{f}_r$ into $\tilde{\mathbf{f}}_r$ by (13) and (14). Then, the estimation of rewards for each action is calculated by (15). Let $\alpha_{\hat{m}}$ be the action with the highest estimation of rewards as the estimated optimal action, i.e.,

$$\alpha_{\hat{m}} = \arg\max_{\alpha_r \in A} \hat{\tilde{d}}_r. \tag{26}$$

- *Difference computation:*

  Compute the Kullback-Leibler divergence factor $\tilde{D}_{KL}^{\hat{m},j}$ between action $\alpha_{\hat{m}}$ and action $\alpha_j, j \neq \hat{m}$ by (11), where

$$\tilde{D}_{KL}^{\hat{m},j} = \min\left\{ D_{KL}\left\{ f(x|\tilde{\mathbf{f}}_{\hat{m}}) \| f(x|\tilde{\mathbf{f}}_j) \right\}, D_{KL}\left\{ f(x|\tilde{\mathbf{f}}_j) \| f(x|\tilde{\mathbf{f}}_{\hat{m}}) \right\} \right\}. \tag{27}$$

- *Action probability vector update*

  At the t-iteration, the action selection is guided by a probability vector $\mathbf{P} = [p_1, p_2, \cdots, p_R]$, where

$$\Pr\{\alpha(t) = \alpha_r\} = p_r, \forall \alpha_r \in A. \tag{28}$$

  Suppose action $\alpha_{\hat{m}}$ is the estimated optimal action in the $(t-1)$-th iteration, given the Kullback-Leibler divergence factor $\tilde{D}_{KL}^{\hat{m},j}$ between action $\alpha_{\hat{m}}$ and action $\alpha_j, j \neq \hat{m}$, the action probability vector $\mathbf{P}$ is updated using

$$\begin{cases} p_{\hat{m}} = \frac{1}{2} \\ p_j = \frac{1}{2} \frac{1/\tilde{D}_{KL}^{\hat{m},j}}{\sum_j 1/\tilde{D}_{KL}^{\hat{m},j}}, \forall j \neq \hat{m} \end{cases}. \tag{29}$$

Note that if action $\alpha_{\hat{m}}$ and action $\alpha_j, j \neq \hat{m}$ have the same feedback vectors, i.e.,

$$\tilde{\mathbf{f}}_{\hat{m}} = \tilde{\mathbf{f}}_j, \tag{30}$$

which usually occurs at the beginning of the learning process, then the factor $\tilde{D}_{KL}^{\hat{m},j}$ is zero, thus the update of the action probability vector $\mathbf{P}$ following (29)

is unrealizable. In this situation, BIFLA randomly selects an action with the same feedback vector as the action $\alpha_{\hat{m}}$.

3) Convergence Judgment

In multi-action environments, an action is considered as the optimal one whenever it is superior to all other actions in rewards and is distinguishable from others by the Kullback-Leibler divergence factor. Thus, for a given convergence threshold $\mathcal{T}$, BIFLA gets converged when

$$\tilde{D}_{KL}^{\hat{m},j} \geq \mathcal{T}, \forall j \neq \hat{m}. \tag{31}$$

And the action $\alpha_{\hat{m}}$ is the optimal one chosen by BIFLA.

The proposed procedure is summarized in Algorithm 1.

3.3. Theoretical Analyses

One of the desirable properties of VSSA is the ϵ-optimality which ensures that an LA can converge to the optimal action with probability one as the number of interactions with the environment approaches infinity [2]. It is crucial that this optimality is justified or the LA could not perform as a reliable and tunable decision maker. In this section, the ϵ-optimality of BIFLA is proven.

Lemma 1. *BIFLA can converge by probability. Formally, for any given $\mathcal{T}$ and ϵ_1, there exists $\hat{N}_r$ such that if α_r has been selected for no less than $\hat{N}_r$ times, then BILFA converges with probability no less than $(1-\epsilon_1)$.*

Proof. Note that the convergence fails only if for some $\mathcal{T}$, a pair of actions α_1 and α_2 (we use the index 1 and 2 without loss of generality) and any possible value of their corresponding chosen times N_1 and N_2, $\tilde{D}_{KL}^{1,2}(\mathbf{f}_1(N_1), \mathbf{f}_2(N_2)) < \mathcal{T}$ holds. So given that $\tilde{D}_{KL}^{1,2}(\mathbf{f}_1(N_1), \mathbf{f}_2(N_2))$ grows approximately with N_1 or N_2 will prove that BIFLA can converge.

Without of generality, we prove for $D_{KL}(f(x|\mathbf{f}_1)\|f(x|\mathbf{f}_2))$ only, and $D_{KL}(f(x|\mathbf{f}_2)\|f(x|\mathbf{f}_1))$ can be formulated analogously. After transforming into the equivalent P-model environment, the relative entropy reads:

$$\log\frac{\Gamma(a_1+b_1)\Gamma(a_2)\Gamma(b_2)}{\Gamma(a_2+b_2)\Gamma(a_1)\Gamma(b_1)}+(a_1-a_2)(\psi(a_1)-\psi(a_1+b_1))+(b_1-b_2)(\psi(b_1)-\psi(a_1+b_1)), \tag{32}$$

where a_i and b_i denote the parameters for the beta distribution introduced by $\alpha_i, i=1,2$. As a continuous function of a_i, b_i, the value of (32) converges with probability one to:

Algorithm 1 The Bayesian Inference Based LA Scheme in Fuzzy Environments

Require: The convergence threshold: $\mathcal{T}$.

Initialize Convergence Flag: $\mathcal{F} = 0$.
Initialize Iteration: $t = 0$.
Initialize Feedback vector: $\mathbf{f}_r = [2, 2, \cdots, 2], \forall \alpha_r \in A$.
Initialize Estimation of rewards: $\hat{\tilde{d}}_r = \frac{1}{2}, \forall \alpha_r \in A$.
Initialize Action selection probability vector: $\mathbf{P} = [\frac{1}{R}, \frac{1}{R}, \cdots, \frac{1}{R}]$
repeat
  Select an action $\alpha(t) = \alpha_s$ according to $\mathbf{P}$.
  Receive the feedback $\beta(t)$ from the fuzzy environment.
  Update the feedback vector by (25).
  Transform the feedback vector $\mathbf{f}_s$ into $\tilde{\mathbf{f}}_s$ by (13) and (14).
  Update the estimation of rewards $\hat{\tilde{d}}_s$ by (15).
  Get the estimated optimal action $\alpha_{\hat{m}}$ by (26).
  for $\forall \alpha_j \in A, j \neq \hat{m}$ **do**
    Compute the Kullback-Leibler divergence factor $\tilde{D}_{KL}^{\hat{m},j}$ between action $\alpha_{\hat{m}}$ and action α_j using (27).
  end for
  if $\forall j \neq \hat{m}$, $\tilde{D}_{KL}^{\hat{m},j} \geq \mathcal{T}$ **then**
    $\mathcal{F} = 1$
  end if
  Update $\mathbf{P}$ according to (29).
  $t = t + 1$.
until $\mathcal{F} = 1$
Output: the optimal action $\alpha_{\hat{m}}$ which has the highest estimation of rewards.

$$\log \frac{\Gamma(N_1)\Gamma(N_2\tilde{d}_2)\Gamma(N_2(1-\tilde{d}_2))}{\Gamma(N_2)\Gamma(N_1\tilde{d}_1)\Gamma(N_1(1-\tilde{d}_1))} + (N_1\tilde{d}_1 - N_2\tilde{d}_2)(\psi(N_1\tilde{d}_1) - \psi(N_1)) \quad (33)$$
$$+ (N_1(1-\tilde{d}_1) - N_2(1-\tilde{d}_2))(\psi(N_1(1-\tilde{d}_1)) - \psi(N_1))$$

Where $\tilde{d}_i$ denotes the reward probability after transformation for action α_i. Formally, the value of (32) as a random variable of N_1 and N_2 will fall into the range centered at (33) with error δ_1 with probability $(1-\epsilon_2)$ when $N_2 > N_{2,1}(\delta_1, \epsilon_2)$ and N_1 larger than some tractable threshold. To further simplify (33), we resort to the following approximations:

$$N_i\tilde{d}_i \approx \lfloor N_i\tilde{d}_i \rfloor \text{ or } \lceil N_i\tilde{d}_i \rceil, N_i(1-\tilde{d}_i) \approx \lfloor N_i(1-\tilde{d}_i) \rfloor \text{ or } \lceil N_i(1-\tilde{d}_i) \rceil \quad (34)$$

$$\psi(n) = \sum_{i=1}^{n-1} \frac{1}{i} - \gamma \quad (35)$$

Where γ in (35) is the Euler-Mascheroni constant. The errors caused by these approximations are denoted by δ_2 and δ_3, where δ_2 grows as a logarithm in (32) by taking Stirling approximation and δ_3 is no larger than a constant.

With (34) and (35), we can replace the parameters of gamma and digamma functions in (33) with their integer approximations and the digamma functions with logarithms. Then (33) is reduced to:

$$\log \frac{N_1!(N_2\tilde{d}_2)!(N_2(1-\tilde{d}_2))!}{N_2!(N_1\tilde{d}_1)!(N_1(1-\tilde{d}_1))!} + (N_1\tilde{d}_1 - N_2\tilde{d}_2)\ln\tilde{d}_1 \quad (36)$$
$$+ (N_1(1-\tilde{d}_1) - N_2(1-\tilde{d}_2))\ln(1-\tilde{d}_1)$$

The asymptotic behavior of (36) is now tractable, expanding with respect

to N_2 and using Stirling approximation ends in:

$$\begin{aligned}
&\log \frac{N_1!(N_2\tilde{d}_2)!(N_2(1-\tilde{d}_2))!}{N_2!(N_1\tilde{d}_1)!(N_1(1-\tilde{d}_1))!} + (N_1\tilde{d}_1 - N_2\tilde{d}_2)\ln\tilde{d}_1 \\
&\qquad\qquad + (N_1(1-\tilde{d}_1) - N_2(1-\tilde{d}_2))\ln(1-\tilde{d}_1) \\
\doteq &-\log N_2! + \log(N_2\tilde{d}_2)! + \log(N_2(1-\tilde{d}_2))! - N_2(\tilde{d}_2\ln\tilde{d}_1 + (1-\tilde{d}_2)\ln(1-\tilde{d}_1)) \\
= &-N_2\ln N_2 + N_2 + N_2\tilde{d}_2\ln N_2\tilde{d}_2 - N_2\tilde{d}_2 + N_2(1-\tilde{d}_2)\ln N_2(1-\tilde{d}_2) - N_2(1-\tilde{d}_2) \\
&- N_2(\tilde{d}_2\ln\tilde{d}_1 + (1-\tilde{d}_2)\ln(1-\tilde{d}_1)) + o(\ln N_2) \\
= &N_2(\tilde{d}_2\ln\frac{\tilde{d}_2}{\tilde{d}_1} + (1-\tilde{d}_2)\ln\frac{(1-\tilde{d}_2)}{(1-\tilde{d}_1)}) + o(\ln N_2) \\
= &N_2\cdot D_{KL}(\tilde{d}_2||\tilde{d}_1) + o(\ln N_2) = O(N_2)
\end{aligned} \tag{37}$$

Where $D_{KL}(\tilde{d}_2||\tilde{d}_1)$ denotes the relative entropy between two Bernoulli distributions with parameter $\tilde{d}_2$ and $\tilde{d}_1$ and $\doteq$ denotes an equivalence relationship dropping terms independent of N_2. The fact that Kullback-Leibler divergence is non-negative together with (37) yields that when N_2 grows, the KL divergence between these two beta distributions can be arbitrarily large except for $\tilde{d}_1 = \tilde{d}_2$. To see how the approximations above fail to disturb the convergence, note that the value of (32) grows linearly with N_2 and the error terms δ_1, δ_3 are constants while δ_2 is a logarithm. In fact, applying Stirling' s approximation of gamma function yields the fact that the influence from N_1 as well as the logarithm term of N_2 is trivial. Therefore the dominating term is still the linear one. And the error caused by using Stirling approximation can be absorbed into δ_2 and δ_3.

Formally, the value of (32) falls into $(d_{2,1}\cdot N_2 - c_{2,1}\cdot\ln N_2 - e_{2,1}, d_{2,1}\cdot N_2 + c_{2,1}\cdot\ln N_2 + e_{2,1})$ with probability higher than $(1-\epsilon_2)$, where $d_{2,1}, c_{2,1}, e_{2,1}$ are constants dependent on the chosen pair of actions, let:

$$\hat{N}_2 > \max\left\{N_{2,1}(\delta_1,\epsilon_2), N_{2,1}^*\right\}, \tag{38}$$

where $N_{2,1}(\delta_1,\epsilon_2)$ exists as long as (32) is continuous with respect to N_2, and $N_{2,1}^*$ is any integer larger than $\frac{1}{d_{2,1}}$ and satisfies $d_{2,1}N_{2,1}^* - c_{2,1}\ln N_{2,1}^* - e_{2,1} > \mathcal{T}$. Such an $N_{2,1}^*$ always exists since $d_{2,1}N_{2,1}^* - c_{2,1}\ln N_{2,1}^* - e_{2,1}$ grows linearly asymptotically. For $N_2 > \hat{N}_2$, the convergence of relative entropy by probability holds and the converged value is necessarily larger than $\mathcal{T}$.

This proposition holds with probability no less than $(1-\epsilon_2)$, and the convergence of BIFLA is necessarily held if such proposition is to hold for any pair of actions. The probability for all $R(R-1)$ ordered pairs to hold this condition is: $(1-\epsilon_2)^{R(R-1)} \approx 1 - R(R-1)\epsilon_2$. Hence let $\epsilon_2 \leq \frac{\epsilon_1}{R(R-1)}$ ensures that BIFLA converge at length by probability no less than $(1-\epsilon_1)$. During which the selection times for α_2 has to be the maximum of all possible comparisons as (38), vice versa for an arbitrary α_r, i.e,

$$\hat{N}_r > \max_{s \neq r} \left\{ N_{r,s}(\delta_1, \epsilon_2), N^*_{r,s} \right\}. \tag{39}$$

□

Lemma 1 shows that increasing selection times N_r for α_r is capable of leading to convergence. And an increase in selection times before convergence is always possible since (29) does not set the selection probability of any action to zero. We nextly show that the increase in selection times is motivated by increasing the only configurable parameter $\mathcal{T}$.

Lemma 2. *In BIFLA each action can be selected for infinite times. Formally, given N and ϵ_3, there exists $\mathcal{T}\{N, \epsilon_3\}$ such that for any $\mathcal{T} > \mathcal{T}\{N, \epsilon_3\}$ as the parameter, each action $\alpha_r \in A$ has to be selected for no less than N times before convergence, this statement holds with probability no less than $(1-\epsilon_3)$.*

Proof. It has been demonstrated in the proof of lemma 1 that $D^{1,2}_{KL}$ grows asymptotically in a linear way. To address the case where N_2 could be too small to reach the asymptotic behavior. Assume that (32) converges to (33) with probability $(1-\epsilon_4)$ and error δ_1 when $N_2 \geq N_{2,1}(\delta_1, \epsilon_4)$. Compute the deviation between (32) and (33) for finite value of N_2 less than $N_{2,1}(\delta_1, \epsilon_4)$ and let the maximal deviation be $\delta'_{2,1}$. Then it is deduced that (32) converges to (33) with probability $(1-\epsilon_4)$ and error $\delta_1 + \delta'_{2,1}$ for all possible N_2.

Thus the difference measured falls in the range of $(d_{2,1}N_2 - c_{2,1}\ln N_2 - e'_{2,1}, d_{2,1}N_2 + c_{2,1}\ln N_2 + e'_{2,1})$ with probability $(1-\epsilon_4)$, where $d_{2,1} = D_{KL}(\tilde{d}_2 \| \tilde{d}_1)$, and $c_{2,1}, e'_{2,1}$ are constants dependent on the action pair. Note that $e'_{2,1}$ has been rectified to be distinguished from its asymptotic counterpart $e_{2,1}$ in Lemma 1.

Analogously, for any ordered pair of actions α_i, α_j, with probability no less than $(1-\epsilon_4)$ the difference $D^{j,i}_{KL}$ falls in $(d_{i,j}N_i - c_{i,j}\ln N_i - e'_{i,j}, d_{i,j}N_i + c_{i,j}\ln N_i + e'_{i,j})$.

Finally let

$$\mathcal{T}\{N,\epsilon_3\} \geq \max_{\alpha_i,\alpha_j \in A} \left\{ d_{i,j} N + c_{i,j} \ln N + e'_{i,j} \right\}. \tag{40}$$

For any α_r, the probability that $D_{KL}^{s,r} > \mathcal{T}\{N,\epsilon_3\}$ when $N_r < N$ is less than ϵ_4. Since with probability no less than $(1-\epsilon_4)$, the value of $D_{KL}^{r,s}$ is upper bounded by a value no larger than $\mathcal{T}\{N,\epsilon_3\}$. Therefore with probability $(1-\epsilon_4)$, N_r has to exceed N in order to make $D_{KL}^{s,r}$ larger than $\mathcal{T}\{N,\epsilon_3\}$.

Exert this reasoning for all pairs of actions and it is obvious that let $\epsilon_4 \leq \frac{\epsilon_3}{R(R-1)}$ completes the proof.

□

Moreover, the estimated optimal action $\alpha_{\hat{m}}$ is always the one that is most likely to be re-examined during the next iteration as exploitation. And the more similar one action α_j is to $\alpha_{\hat{m}}$, the more α_j is likely to be examined in the next iteration as trial discrimination between the optimal and sub-optimal choices. For any pair of actions, in order to meet the convergence judgment, both of them need to be selected for sufficient times or (27) will reject the convergence.

Theorem 2. *(ϵ-optimality) Given arbitrary ϵ, there exists a corresponding $\mathcal{T}$ such that BIFLA with $\mathcal{T}$ as the parameter converges to the optimal action with probability no less than $(1-\epsilon)$.*

Proof. It has already been indicated by Lemma 1 that BIFLA will converge with probability one, intuitively, it is sufficient to show that the probability of a correct convergence can approximate one as well. For comparison between the expected rewards, it is the convergence by probability for each action that guarantees the fact that the optimal action will win at length, i.e.,

$$\hat{\tilde{d}}_r \to \tilde{d}_r \tag{41}$$

by probability. This convergence in expectation holds with $(1-\epsilon_r)$ and with error δ_r when $N_r > N_r^E(\epsilon_r,\delta_r)$ and Lemma 2 suggests that the increasing of N_r is feasible by increasing $\mathcal{T}$.

Formally, a correct convergence requires that: i). BIFLA converges; ii). the estimation of reward for any action approximates its theoretical value; iii). the error for any pair of actions can not disturb the authentic difference between the optimal and sub-optimal actions.

Note that i) holds with probability $(1-\epsilon_1)$ by lemma 1. While ii) can be obtained by letting $\epsilon_r = \frac{\epsilon_5}{R}$, then the probability that ii) holds is:

$$\prod_{r=1}^{R}(1-\frac{\epsilon_r}{R}) = (1-\frac{\epsilon_5}{R})^R \approx 1-\epsilon_5. \tag{42}$$

And iii) holds trivially by let $\delta_r = \frac{\delta_o}{2} < \delta_o, \forall \alpha_r \in A$, where δ_o is the difference in reward expectation between the optimal action and the second optimal one, i.e:

$$\delta_o = \min_{\alpha_r \in A} \left\{ \tilde{d}_m - \tilde{d}_r \right\}. \tag{43}$$

Finally, let $N' = \max_{\alpha_r \in A} \left\{ N_r^E(\frac{\epsilon_5}{R}, \frac{\delta_o}{2}) \right\}$ and $\mathcal{T}_1 = \mathcal{T}\left\{N', \epsilon_3\right\}$ using the notation in Lemma 2. Then with probability no less than $(1-\epsilon_3)(1-\epsilon_5)$ BIFLA with $\mathcal{T}_1$ can find the optimal action. And the convergence is ensured by Lemma 1.

It is trivial to ensure $(1-\epsilon_3)(1-\epsilon_5) \geq (1-\epsilon)$ by let $\epsilon_3 = \epsilon_5 = \frac{\epsilon}{2}$. To conclude, we have:

$$\mathcal{T}_1 = \mathcal{T}\left\{ \max_{\alpha_r \in A} \left\{ N_r^E\Big(\frac{\epsilon}{2R}, \frac{\delta_o}{2}\Big) \right\}, \frac{\epsilon}{2} \right\} \tag{44}$$

as a possible choice to verify the theorem.

□

The reasoning above suggests that it is always possible to increase accuracy by increasing $\mathcal{T}$, which indicates that BIFLA is a reliable decision-maker.

4. Simulation Results

To illustrate the efficacy of the proposed LA, BIFLA is evaluated in both P-model environments and fuzzy environments. The P-model environment is a specialization of the general fuzzy environment by setting $Q = 1$ and is the scenario in most surveys. We first demonstrate the effect of the convergence threshold $\mathcal{T}$ and compare BIFLA with the state-of-the-art LA schemes in P-model environments, followed by the performance of BIFLA in various fuzzy environments. The performance of an LA scheme is usually evaluated by the convergence rate on the premise of a certain accuracy [24], where

- The accuracy is defined as the probability of correct convergence in a series of repeated experiments, i.e., the probability for an LA to find the action with the highest expected rewards in the environment.
- The convergence rate is the average times of interactions with the environment for an LA to converge correctly.

4.1. Performance of BIFLA in P-model Environments

4.1.1. Setting of the Environments

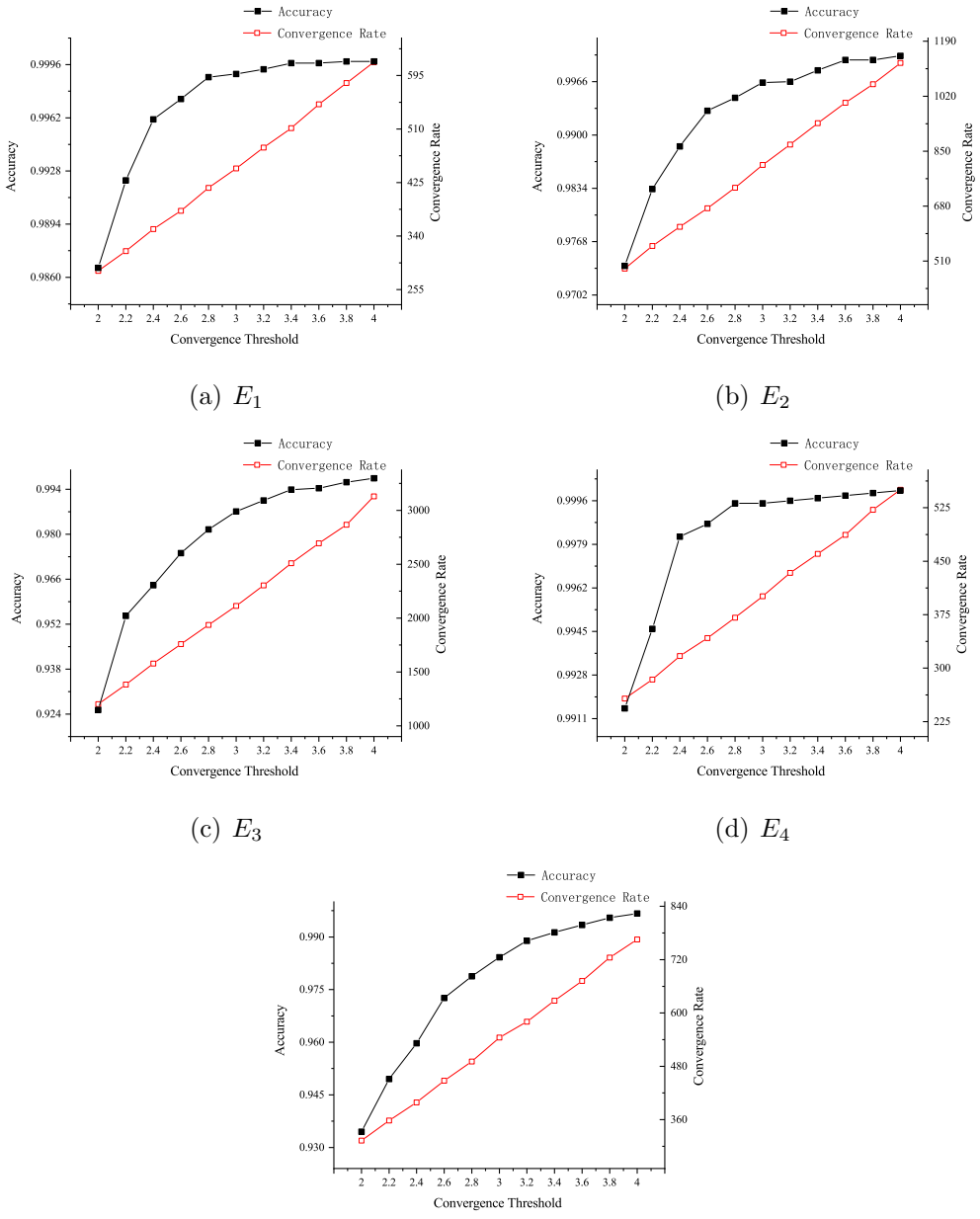
In the studies of LA in P-model environments, there are five benchmark environments, where the reward probability vectors defined in (4) for each environment are as follows.

- E_1: $\mathbf{D} = [\mathbf{0.65}, 0.50, 0.45, 0.40, 0.35, 0.30, 0.25, 0.20, 0.15, 0.10]$.
- E_2: $\mathbf{D} = [\mathbf{0.60}, 0.50, 0.45, 0.40, 0.35, 0.30, 0.25, 0.20, 0.15, 0.10]$.
- E_3: $\mathbf{D} = [\mathbf{0.55}, 0.50, 0.45, 0.40, 0.35, 0.30, 0.25, 0.20, 0.15, 0.10]$.
- E_4: $\mathbf{D} = [\mathbf{0.70}, 0.50, 0.30, 0.20, 0.40, 0.50, 0.40, 0.30, 0.50, 0.20]$.
- E_5: $\mathbf{D} = [0.10, 0.45, \mathbf{0.84}, 0.76, 0.20, 0.40, 0.60, 0.70, 0.50, 0.30]$.

For the benchmarks, the complexity of an environment is reflected in the difference of the reward probabilities between the optimal and the suboptimal actions denoted by δ. From this perspective, the environment E_3 is the most complicated one among $E_1 - E_5$ with $\delta = d_1 - d_2 = 0.05$, and E_4 is the simplest one with $\delta = 0.2$. The simulations in P-model environments are carried on the five benchmark environments above, where the accuracy and the convergence rate of schemes in each environment are evaluated in terms of $250,000$ repeated trials in each environment.

4.1.2. Effect of the Convergence Threshold $\mathcal{T}$

As the only configurable parameter of BIFLA, the convergence threshold $\mathcal{T}$ is used to realize the trade-off between the accuracy and the efficiency (convergence rate). Figure 2 presents the accuracy and the convergence rate of BIFLA with various convergence threshold $\mathcal{T}$ in environments $E_1 - E_5$. It can be concluded that a large value of $\mathcal{T}$ leads to high accuracy and low efficiency, while a small value of $\mathcal{T}$ leads to relatively a low accuracy and high efficiency. This is an experimental support of the $\epsilon-$optimality.



(a) E_1

(b) E_2

(c) E_3

(d) E_4

(e) E_5

Figure 2: Curves of the accuracy and the convergence rate of BIFLA versus the convergence threshold $\mathcal{T}$.

The intuitive reason after this results is that when $\mathcal{T}$ is large the BIFLA requires more interactions with the environment to get a higher value of the Kullback-Leibler divergence between action as it increases with the increasing number of selected times of actions, which results in the higher accuracy as well as more iterations. When $\mathcal{T}$ is small, the converse reasoning holds. Besides, as shown in Figure 2, the linear increasing of the convergence rate with respect to $\mathcal{T}$ demonstrates the validity of Lemma 1 experimentally.

4.1.3. Comparisons with State-of-The-Art Schemes

To show the superiority of the proposed scheme, BIFLA is compared with the state-of-the-art LA schemes in P-model environments. The compared schemes are proposed in [30], including the F-LA with confidence level 90%, the B-HPD-LA with HPD credible level 90% , and the B-ET-LA with ET credible level 90%.

Table 1: Accuracy and convergence rate of BIFLA with various convergence threshold $\mathcal{T}$ in P-model environments.

	$\mathcal{T}=2$		$\mathcal{T}=2.4$		$\mathcal{T}=2.8$	
	Acc.	Con. Rate	Acc.	Con. Rate	Acc.	Con. Rate
E_1	0.9866	284	0.9961	351	0.9988	416
E_2	0.9738	486	0.9886	615	0.9946	736
E_3	0.9253	1201	0.9641	1578	0.9815	1936
E_4	0.9915	257	0.9982	317	0.9995	370
E_5	0.9345	312	0.9597	398	0.9788	490
	$\mathcal{T}=3.2$		$\mathcal{T}=3.6$		$\mathcal{T}=4$	
	Acc.	Con. Rate	Acc.	Con. Rate	Acc.	Con. Rate
E_1	0.9993	480	0.9997	549	0.9998	616
E_2	0.9966	870	0.9993	999	0.9998	1123
E_3	0.9905	2303	0.9943	2694	0.9974	3129
E_4	0.9996	433	0.9999	486	1	550
E_5	0.9889	580	0.9934	672	0.9967	765

Table 1 and 2 list the simulation results of the schemes above, from which we can construe the following conclusions:

1. All accuracies are quite high, yielding the ϵ-optimality of LA.

Table 2: The accuracy and the convergence rate of F-LA, B-HPD-LA, and B-ET-LA in benchmark environments.

	F-LA		B-HPD-LA		B-ET-LA	
	Acc.	Con. Rate	Acc.	Con. Rate	Acc.	Con. Rate
E_1	0.9996	665	0.9980	581	0.9992	607
E_2	0.9983	1238	0.9955	1133	0.9975	1164
E_3	0.9354	3189	0.9341	3022	0.9372	3071
E_4	0.9999	577	0.9993	508	0.9998	527
E_5	0.9929	1051	0.9895	1033	0.9907	909

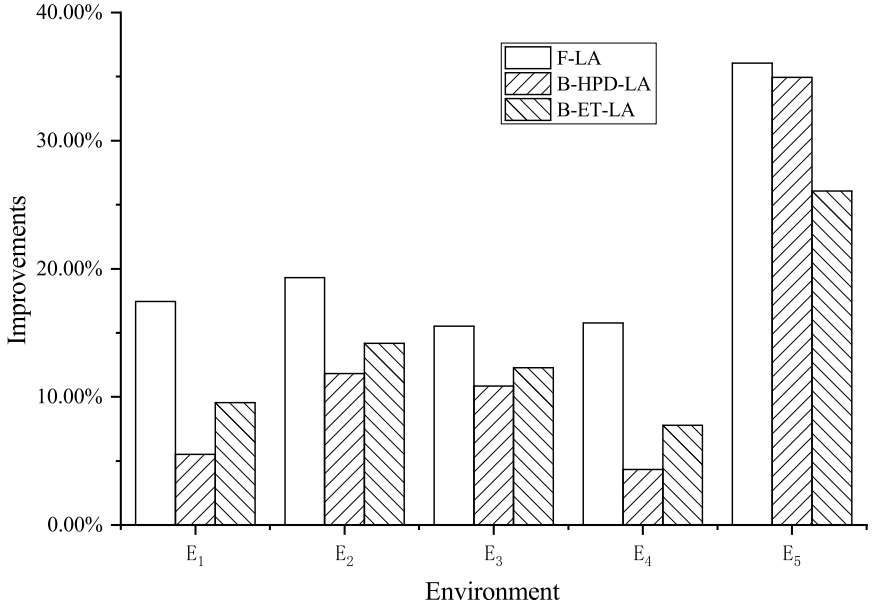


Figure 3: The improvements of the compared schemes relative to BIFLA with $\mathcal{T} = 3.6$ in benchmark environments.

2. BIFLA has comparative performances compared with all the other schemes. Specifically, BIFLA can converge faster with higher accuracy in each benchmark environment when setting the convergence threshold $\mathcal{T} = 3.6$. The improvement σ which is defined as

$$\sigma = \frac{\text{Convergence Rate}_{\text{Compared Scheme}} - \text{Convergence Rate}_{\text{BIFLA}}}{\text{Convergence Rate}_{\text{Compared Scheme}}} \quad (45)$$

prompted by BIFLA with $\mathcal{T} = 3.6$ in $E_1 - E_5$ is illustrated in Figure 3, from which we can draw that the superiority of BIFLA is clear.

4.2. Performance of BIFLA in Fuzzy Environments

As the topic studied in this paper, VSSA in fuzzy environments, there has not been established benchmark environments for evaluation. In view of this, we coin various fuzzy environments with different parameter settings as follows. This is done by setting Q and R beforehand and generating a stochastic matrix of $R \cdot (Q+1)$. The optimal action is the one with the highest expectation of rewards. The expectations of rewards for all actions are collected in the reward vector $\mathbf{\Xi}$ with ξ_r denotes the expectation of rewards for α_r.

- $E_6 : \mathbf{D} = \begin{bmatrix} 0.1 & 0.2 & 0.7 \\ 0.4 & 0.4 & 0.3 \end{bmatrix}$, $Q = 2$, $R = 2$, $\mathbf{\Xi} = \begin{bmatrix} 0.8 \\ 0.5 \end{bmatrix}$.
- $E_7 : \mathbf{D} = \begin{bmatrix} 0.1 & 0.8 & 0.1 \\ 0.6 & 0.2 & 0.2 \end{bmatrix}$, $Q = 2$, $R = 2$, $\mathbf{\Xi} = \begin{bmatrix} 0.5 \\ 0.3 \end{bmatrix}$.
- $E_8 - E_{12}$: $\mathbf{D} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \cdots \\ \mathbf{d}_{Q+1} \end{bmatrix}$, for $Q = 2, 4, 6, 8, 10$, respectively, $R = Q+1$.

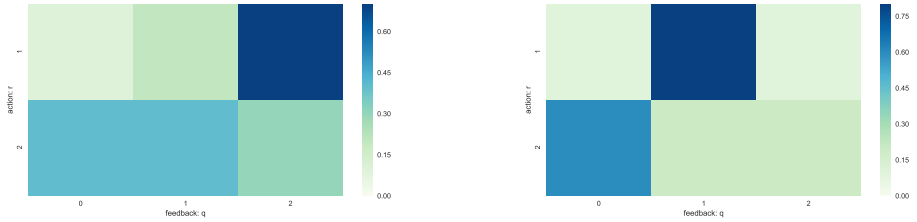
  For each $\mathbf{d}_r$ in each environment,

$$\begin{cases} d_{r,q} = \frac{1}{2Q}, r \neq q \\ d_{r,q} = \frac{1}{2}, r = q \end{cases} . \tag{46}$$

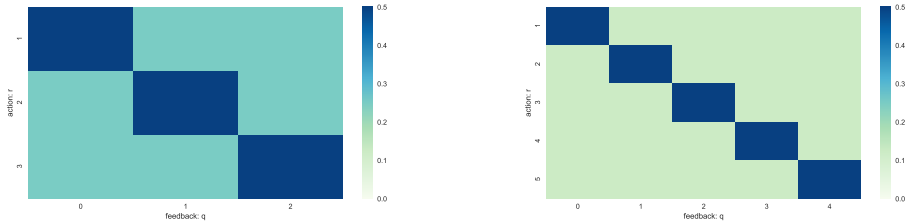
Fuzzy environments E_6 and E_7 are the simplest environments with two actions and triple-level feedback. Environments $E_8 - E_{12}$ are incremental environments where the difference of rewards between actions α_r and $\alpha_{r+1} (r = 1, 2, \cdots, R-1)$ is the same. $E_8 - E_{12}$ have different numbers of actions as well as different types of feedback to represent varying degrees of complexity. Figure 4 illustrates the reward probability matrices for environments $E_6 - E_{12}$ with heat maps.

For $E_8 - E_{12}$, the rewards vectors $\mathbf{\Xi}$ are as follows.

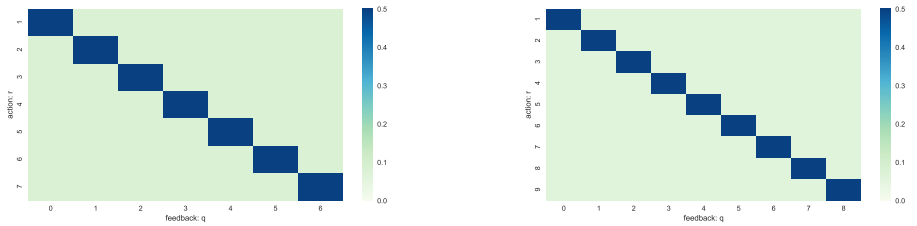
- $E_8 : \mathbf{\Xi} = [0.37500 \quad 0.50000 \quad \mathbf{0.62500}]^{\mathrm{T}}$.
- $E_9 : \mathbf{\Xi} = [0.31250 \quad 0.40625 \quad 0.50000 \quad 0.59375 \quad \mathbf{0.68750}]^{\mathrm{T}}$.



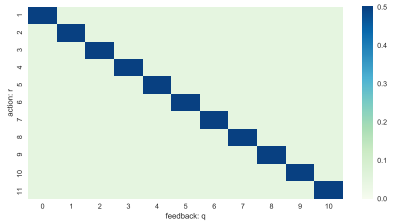
(a) E_6 (b) E_7



(c) E_8 (d) E_9



(e) E_{10} (f) E_{11}



(g) E_{12}

Figure 4: The heat maps of the reward probability matrices in environments E_6, E_7, and incremental environments E_8-E_{12}.

- E_{10}: $\boldsymbol{\Xi} = [0.29167 \quad 0.36111 \quad 0.43055 \quad 0.50000 \quad 0.56944 \quad 0.63889 \quad \mathbf{0.70833}]^{\mathrm{T}}$.
- E_{11}: $\boldsymbol{\Xi} = [0.28125 \quad 0.33594 \quad 0.39063 \quad 0.44531 \quad 0.50000 \quad 0.55469 \quad 0.60938 \quad 0.66406 \quad \mathbf{0.71875}]^{\mathrm{T}}$.
- E_{12}: $\boldsymbol{\Xi} = [0.27500 \quad 0.32000 \quad 0.36500 \quad 0.41000 \quad 0.45500 \quad 0.50000 \quad 0.54500 \quad 0.59000 \quad 0.63500 \quad 0.68000 \quad \mathbf{0.72500}]^{\mathrm{T}}$.

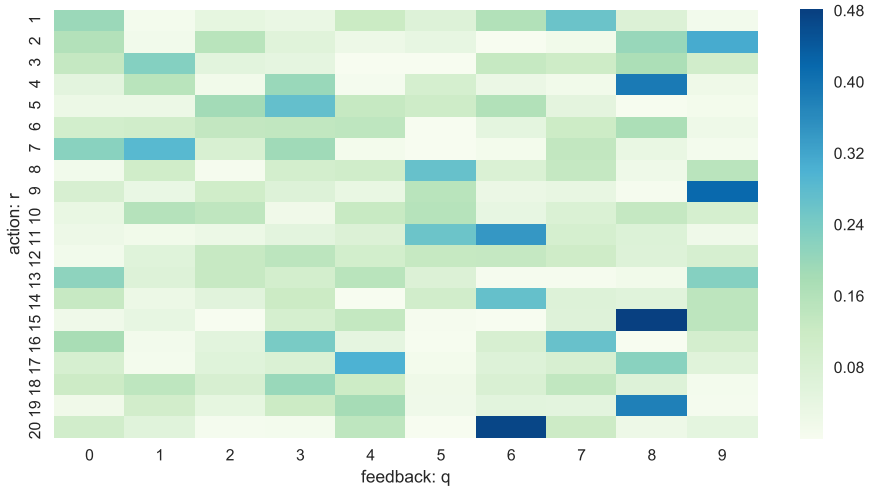
E_{13} and E_{14} are randomly generated, characterizing the general fuzzy environments. With hyperparameters:

- E_{13}: $Q = 4$, $R = 10$ with:
  $\boldsymbol{\Xi} = [0.62541 \quad 0.56868 \quad 0.61410 \quad 0.53287 \quad 0.36910$
  $\mathbf{0.74921} \quad 0.52934 \quad 0.64272 \quad 0.51611 \quad 0.30873]^{\mathrm{T}}$.
- E_{14}: $Q = 9$, $R = 20$ with:
  $\boldsymbol{\Xi} = [0.51138 \quad 0.60027 \quad 0.49653 \quad 0.55380 \quad 0.42432$
  $0.45927 \quad 0.28334 \quad 0.57628 \quad 0.63195 \quad 0.50610$
  $0.59570 \quad 0.53346 \quad 0.43134 \quad 0.55480 \quad \mathbf{0.72827}$
  $0.48437 \quad 0.55624 \quad 0.40563 \quad 0.58385 \quad 0.55926]^{\mathrm{T}}$.

Figure 5 illustrates the heat maps of the reward probability matrix for E_{13} and E_{14}. The corresponding numerical details are saved for Appendix A.

Simulations are carried on the above fuzzy environments, and the results including the accuracy and the convergence rate in each environment are listed in Table 3. The high accuracies demonstrate the effectiveness of the proposed BIFLA scheme in fuzzy environments and reveal that BIFLA can successfully learn the optimal action in fuzzy environments, hence experimentally verify the theoretical analyses.

(a) E_{13}



(b) E_{14}

Figure 5: The heat maps of the reward probability vectors in incremental environments in randomly generated environments E_{13} and E_{14}.

Table 3: Accuracy and convergence rate of BIFLA with various convergence threshold $\mathcal{T}$ in fuzzy environments.

	$\mathcal{T}=2$		$\mathcal{T}=2.2$		$\mathcal{T}=2.4$	
	Acc.	Con. Rate	Acc.	Con. Rate	Acc.	Con. Rate
E_6	1	21	1	22	1	23
E_7	0.9997	59	0.9997	64	0.9999	69
E_8	0.9939	160	0.9955	180	0.9974	195
E_9	0.9983	313	0.9989	345	0.9993	385
E_{10}	0.9932	544	0.9977	593	0.9979	670
E_{11}	0.9856	799	0.9915	901	0.9947	1009
E_{12}	0.9689	1064	0.9796	1242	0.9887	1414
E_{13}	0.9986	476	0.9995	537	0.9999	595
E_{14}	0.9984	731	0.9995	819	0.9998	925
	$\mathcal{T}=2.6$		$\mathcal{T}=2.8$		$\mathcal{T}=3$	
	Acc.	Con. Rate	Acc.	Con. Rate	Acc.	Con. Rate
E_6	1	25	1	26	1	27
E_7	0.9999	73	0.9999	79	0.9999	83
E_8	0.9980	211	0.9985	226	0.9993	244
E_9	0.9998	416	0.9998	459	1	492
E_{10}	0.9996	744	0.9999	822	1	885
E_{11}	0.9979	1134	0.9982	1251	0.9991	1398
E_{12}	0.9916	1569	0.9963	1782	0.9969	1940
E_{13}	1	661	1	719	1	790
E_{14}	0.9999	1045	1	1165	1	1297

5. Conclusion

In this paper, we propose a novel LA scheme BIFLA which is capable of learning from a fuzzy environment by interactions. The proposed model provides a bridge between LA theories and practical applications. The mechanism of BIFLA is brand-new, where the Bayesian inference and the relative entropy are utilized to realize the action selection and state update. Besides, we prove its ϵ-optimality rigorously. Simulations conducted in P-model benchmark environments demonstrate the superiority of BIFLA compared with the state-of-the-art schemes. Furthermore, the experimental results in various fuzzy environments verify the efficacy of BIFLA. Our further work is to extend our scheme into non-stationary random environments.

Acknowledgements

This work was supported by the National Key Research and Development Project of China under Grant 2016YFB0801003, and in part by the National Nature Science Foundation of China under Grants 61771342 and 61731006.

References

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

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

[3] Tsetlin, Michael L. "On behaviour of finite automata in random medium." Avtom I Telemekhanika 22 (1961): 1345-1354.

[4] Hasanzadeh, Mohammad, and Mohammad Reza Meybodi. "Grid resource discovery based on distributed learning automata." Computing 96.9 (2014): 909-922.

[5] Akaki Jobava, Anis Yazidi, B. John Oommen, and Kyrre Begnum. "On achieving intelligent traffic-aware consolidation of virtual machines in a data center using Learning Automata." Journal of computational science 24 (2018): 290-312.

[6] Rahmanian, Ali Asghar, Mostafa Ghobaei-Arani, and Sajjad Tofighy. "A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment." Future Generation Computer Systems 79 (2018): 54-71.

[7] Mohammad Hasanzadeh Mofrad, Sana Sadeghi, Alireza Rezvanian, and Mohammad Reza Meybodi. "Cellular edge detection: Combining cellular automata and cellular learning automata." AEU-International Journal of Electronics and Communications 69.9 (2015): 1282-1290.

[8] Kumar, Neeraj, Jong-Hyouk Lee, and Joel JPC Rodrigues. "Intelligent mobile video surveillance system as a Bayesian coalition game in vehicular sensor networks: Learning automata approach." IEEE Transactions on Intelligent Transportation Systems 16.3 (2015): 1148-1161.

[9] Adinehvand Karima, Sardari Dariusha, Hosntalab Mohammada, and Pouladian Majidb. "An efficient multistage segmentation method for accurate hard exudates and lesion detection in digital retinal images." Journal of Intelligent & Fuzzy Systems 33.3 (2017): 1639-1649.

[10] Vafashoar, Reza, and Mohammad Reza Meybodi. "Multi swarm bare bones particle swarm optimization with distribution adaption." Applied Soft Computing 47 (2016): 534-552.

[11] Kordestani, Javidan Kazemi, Hossein Abedi Firouzjaee, and Mohammad Reza Meybodi. "An adaptive bi-flight cuckoo search with variable nests for continuous dynamic optimization problems." Applied Intelligence 48.1 (2018): 97-117.

[12] Rezvanian, Alireza, and Mohammad Reza Meybodi. "Sampling algorithms for stochastic graphs: a learning automata approach." Knowledge-Based Systems 127 (2017): 126-144.

[13] 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.

[14] Rezapoor Mirsaleh, Mehdi, and Mohammad Reza Meybodi. "Balancing exploration and exploitation in memetic algorithms: a learning automata approach." Computational Intelligence 34.1 (2018): 282-309.

[15] Ahangaran, Meysam, Nasrin Taghizadeh, and Hamid Beigy. "Associative cellular learning automata and its applications." Applied Soft Computing 53 (2017): 1-18.

[16] Sohrabi, Mohammad Karim, and Reza Roshani. "Frequent itemset mining using cellular learning automata." Computers in human behavior 68 (2017): 244-253.

[17] Ghavipour, Mina, and Mohammad Reza Meybodi. "Trust propagation algorithm based on learning automata for inferring local trust in online social networks." Knowledge-Based Systems 143 (2018): 307-316.

[18] Hasanzadeh-Mofrad, Mohammad, and Alireza Rezvanian. "Learning automata clustering." Journal of computational science 24 (2018): 379-388.

[19] Alireza Rezvanian,Behnaz Moradabadi, Mina Ghavipour, Mohammad Mehdi Daliri Khomami, and Mohammad Reza Meybodi. "Introduction to Learning Automata Models." Learning Automata Approach for Social Networks. Springer, Cham, 2019. 1-49.

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

[21] Varshavskii, V. I., and I. P. Vorontsova. "On the behavior of stochastic automata with a variable structure." Avtomatika i Telemekhanika 24.3 (1963): 353-360.

[22] A. H. Jamalian, R. Rezvani, H. Shams, and SH. Mehrabil. "A new learning automaton for interaction with triple level environments." 2012 IEEE 11th International Conference on Cognitive Informatics and Cognitive Computing. IEEE, 2012.

[23] Jiang, Wen, and Sheng-Hong Li. "A general method for P-model FSSA learning in triple level environment." Neurocomputing 137 (2014): 150-156.

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

[25] 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.

[26] 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.

[27] Zhang, Junqi, Cheng Wang, and MengChu Zhou. "Last-position elimination-based learning automata." IEEE Trans. Cybernetics 44.12 (2014): 2484-2492.

[28] Junqi Zhang, Cheng Wang, Di Zang, and Mengchu Zhou. "Incorporation of optimal computing budget allocation for ordinal optimization into learning automata." IEEE Transactions on Automation Science and Engineering 13.2 (2016): 1008-1017.

[29] Hao Ge, Wen Jiang, Shenghong Li, Jianhua Li, Yifan Wang, and Yuchun Jing. "A novel estimator based learning automata algorithm." Applied Intelligence 42.2 (2015): 262-275.

[30] Guo, Ying, and Shenghong Li. "A Non-Monte-Carlo Parameter-Free Learning Automata Scheme Based on Two Categories of Statistics." IEEE transactions on cybernetics 99 (2018): 1-14.

[31] 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.

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

[33] Guo, Ying, Hao Ge, and Shenghong Li. "A loss function based parameterless learning automaton scheme." Neurocomputing 260 (2017): 331-340.

[34] Casella, George, and Roger L. Berger. Statistical inference. Vol. 2. Pacific Grove, CA: Duxbury, 2002.

[35] Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem." Machine learning 47.2-3 (2002): 235-256.

Appendix A. Reward Probability Vectors of Fuzzy Environments in Simulations

The reward probability matrix for E_{13} and E_{14} is listed as follows:

- E_{13}:

$$\mathbf{D} = \begin{bmatrix} 0.00849 & 0.24682 & 0.06541 & 0.59314 & 0.08615 \\ 0.05115 & 0.18911 & 0.37028 & 0.21280 & 0.17666 \\ 0.12176 & 0.29542 & 0.06441 & 0.04150 & 0.47692 \\ 0.19803 & 0.30670 & 0.03477 & 0.08675 & 0.37375 \\ 0.20619 & 0.18478 & 0.56559 & 0.01331 & 0.03013 \\ 0.07418 & 0.11670 & 0.05023 & 0.25587 & 0.50302 \\ 0.12722 & 0.26640 & 0.19611 & 0.18236 & 0.22791 \\ 0.14136 & 0.05378 & 0.28127 & 0.13980 & 0.38379 \\ 0.05403 & 0.18352 & 0.56493 & 0.03904 & 0.15849 \\ 0.20165 & 0.59167 & 0.02714 & 0.12915 & 0.05038 \end{bmatrix}.$$

- E_{14}:

$$\mathbf{D} = \begin{bmatrix} 0.19911 & 0.01002 & 0.04660 & 0.03600 & 0.12327 & 0.06812 & 0.16487 & 0.26214 & 0.07555 & 0.01433 \\ 0.16196 & 0.01653 & 0.15372 & 0.06291 & 0.02729 & 0.04331 & 0.00186 & 0.01864 & 0.20170 & 0.31209 \\ 0.13299 & 0.22850 & 0.05592 & 0.04895 & 0.00196 & 0.00239 & 0.13124 & 0.11661 & 0.17484 & 0.10659 \\ 0.05397 & 0.15408 & 0.01910 & 0.20118 & 0.00893 & 0.09585 & 0.03394 & 0.01882 & 0.38921 & 0.02493 \\ 0.03087 & 0.03194 & 0.18678 & 0.27169 & 0.13028 & 0.11474 & 0.16377 & 0.05169 & 0.00501 & 0.01322 \\ 0.10195 & 0.10864 & 0.13559 & 0.13989 & 0.14416 & 0.00262 & 0.05057 & 0.11715 & 0.17417 & 0.02527 \\ 0.22043 & 0.28685 & 0.08656 & 0.19067 & 0.01232 & 0.00113 & 0.01273 & 0.13907 & 0.03886 & 0.01138 \\ 0.01575 & 0.10966 & 0.00798 & 0.09817 & 0.10803 & 0.26863 & 0.08063 & 0.13311 & 0.02682 & 0.15122 \\ 0.08999 & 0.03773 & 0.11074 & 0.07147 & 0.03840 & 0.15395 & 0.03394 & 0.04024 & 0.00660 & 0.41694 \\ 0.03964 & 0.15965 & 0.14296 & 0.01989 & 0.12666 & 0.15805 & 0.04370 & 0.08048 & 0.13347 & 0.09548 \\ 0.02857 & 0.01816 & 0.02906 & 0.05346 & 0.07474 & 0.26073 & 0.34348 & 0.09570 & 0.07371 & 0.02239 \\ 0.01678 & 0.06569 & 0.12973 & 0.14799 & 0.10203 & 0.13075 & 0.13242 & 0.11205 & 0.07176 & 0.09079 \\ 0.21501 & 0.06958 & 0.12878 & 0.10111 & 0.15384 & 0.07427 & 0.00634 & 0.00755 & 0.01768 & 0.22584 \\ 0.12956 & 0.03002 & 0.05908 & 0.12095 & 0.00380 & 0.10688 & 0.26970 & 0.07178 & 0.06214 & 0.14609 \\ 0.02159 & 0.04192 & 0.00312 & 0.09588 & 0.13436 & 0.00629 & 0.00197 & 0.06775 & 0.48051 & 0.14662 \\ 0.17942 & 0.01472 & 0.05574 & 0.24288 & 0.04758 & 0.00503 & 0.09042 & 0.26514 & 0.00112 & 0.09796 \\ 0.09097 & 0.01089 & 0.06439 & 0.08017 & 0.30096 & 0.01214 & 0.06815 & 0.08919 & 0.21997 & 0.06316 \\ 0.11929 & 0.14433 & 0.09035 & 0.20010 & 0.11819 & 0.02429 & 0.08344 & 0.14035 & 0.06886 & 0.01081 \\ 0.01951 & 0.10503 & 0.04734 & 0.12184 & 0.18367 & 0.02308 & 0.05605 & 0.05444 & 0.37936 & 0.00967 \\ 0.10629 & 0.06066 & 0.00821 & 0.00999 & 0.14491 & 0.00061 & 0.47107 & 0.11970 & 0.02771 & 0.05084 \end{bmatrix}.$$