A Conclusive Analysis of the Finite-Time Behavior of the Discretized Pursuit Learning Automaton

Xuan Zhang, Lei Jiao, *Senior Member, IEEE*, B. John Oommen, *Life Fellow, IEEE*, and Ole-Christoffer Granmo

***Abstract*—This paper[1] deals with the *Finite-Time* Analysis (FTA) of Learning Automata (LA), which is a topic for which very little work has been reported in the literature. This is as opposed to the asymptotic steady-state analysis for which there are, probably, scores of papers. As clarified later, unarguably, the FTA of Markov Chains, in general, and of LA, in particular, is far more complex than the asymptotic steady-state analysis. Such a FTA provides rigid bounds for the time required for the LA to attain to a given convergence accuracy. We concentrate on the FTA of the Discretized Pursuit Automaton (DPA), which is probably one of the fastest and most accurate reported LA. Although such an analysis was carried out many years ago, we record that the previous work is flawed. More specifically, in all brevity, the flaw lies in the wrongly "derived" monotonic behavior of the LA after a certain number of iterations. Rather, we claim that the property that should be invoked is the submartingale property. This renders the proof to be much more involved and deep. In this paper, we rectify the flaw and re-establish the FTA based on such a submartingale phenomenon. More importantly, from the derived analysis, we are able to discover and clarify, for the first time, the underlying dilemma between the DPA's exploitation and exploration properties. We also non-trivially confirm the existence of the optimal learning rate, which yields a better comprehension of the DPA itself.**

Keywords *Learning automaton, Pursuit algorithms, DPA, Finite-time analysis*

I. Introduction

The field of Learning Automata (LA) [14] has been studied as a typical model of reinforcement learning for decades. The LA operates in conjunction with an Environment, with the goal of learning the optimal action from among a set of actions offered by the Environment. LA have been applied in many dozens of fields[2]. Conceptually, the LA works with an Environment in an iterative manner. In each iteration, the LA selects one action, which triggers either a *stochastic* reward or a penalty from the Environment. Then, based on the responses received from the Environment, the LA adjusts its action selection strategy in order to make a "wiser" decision in the next iteration. In such a way, the LA learns the optimal decision via repeated interactions with the Environment.

Non-Estimator LA: The development of LA has gone through three periods: the Fixed Structure Stochastic Automata (FSSA), the Variable Structure Stochastic Automata (VSSA) and the Estimator Algorithms (EAs). The FSSA are LA whose state update function and decision function are time invariant. The Tsetlin, Krylov and Krinsky automata [14] are the most notable examples of this type. As opposed to FSSA, in VSSA, either the state update function or the decision function (or both) vary with time. Interestingly, VSSA can be characterized by functions that update the probability of selecting the various actions. Typical examples of VSSA include the Linear Reward-Penalty (L_{R-P}) scheme, the Linear Reward-Inaction (L_{R-I}) scheme, the Linear Inaction-Penalty (L_{I-P}) scheme and the Linear Reward-εPenalty ($L_{R-\varepsilon P}$) scheme [14].

Estimator-based LA: Among the families of LA, Estimator Algorithms (EAs) are the fastest and the most accurate type of LA. They augment an action probability updating scheme with the use of estimates of the respective actions' reward probabilities. Typically, EAs maintain running Maximum Likelihood (ML) reward probability estimates to determine whether any specific action is "better" than another. Within this family, the set of *Pursuit* Algorithms (PAs) were the pioneering schemes, whose design and analysis were initiated by Thathachar and Sastry [28]. The first PA was designed to operate by updating the action probabilities based on the L_{R-I} paradigm. In each iteration, the current "Best" action is pursued by linearly increasing *its* action probability. As the PA considers both the *short-term* responses of the Environment and the *long-term* reward probability estimates in formulating the action probability updating rules, it outperforms traditional VSSA schemes in terms of its accuracy and its rate of convergence.

This author can be contacted at: Centre for Artificial Intelligence Research, University of Agder, Grimstad, Norway. E-mail: xuan.z.jiao@gmail.com. This author is also a Data Analyst in Confirmit AS, Norway.

Senior member: IEEE. This author can be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: lei.jiao@uia.no.

Chancellor's Professor; *Life Fellow: IEEE* and *Fellow: IAPR*. This author can be contacted at: School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. E-mail address: oommen@scs.carleton.ca.

This author can be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: ole.granmo@uia.no.

[1] The work of the third author was partially supported by NSERC, the Natural Sciences and Engineering Council of Canada. We are very grateful for the feedback from the anonymous Referees of the *original* submission. Their input significantly improved the quality of this final version.

[2] With regard to applications, the entire field of LA and stochastic learning has had a myriad of applications [8], [13], [14], [23], [29], which (apart from the many applications listed in these books) include solutions for problems in network and communications [12], [17], [22], network call admission, traffic control, quality of service routing [1], [2], [31], distributed scheduling [26], training hidden Markov models [7], neural network adaptation [11], intelligent vehicle control [30] service selection [32] and even fairly theoretical problems such as graph partitioning [20] and string taxonomy [18]. Besides these fairly generic applications, LA have been used to assist in solving numerous other optimization problems in stochastic domains, for example, in telecommunications [4], [5], [38] and in the energy sector [25], [27].

On Using Bayesian estimates: Another family of EAs, called the family of Bayesian Pursuit Algorithms (BPAs) has been designed within the Pursuit learning paradigm. It enhances the rate of convergence by substituting ML estimates with more optimistic Bayesian estimates.

Discretized EAs: To enhance the convergence, Oommen and Lanctot [21] presented the Discretized Pursuit Algorithm (DPA) by discretizing[3] the action probability space. The DPA was shown to be superior to its continuous counterpart in terms of its rate of convergence. Besides, the DPA has the potential to be processed in a batch mode, and a fast version of the DPA was proposed in [6] to achieve a lower computational complexity. Along the same vein, the continuous BPA [33] has also been discretized to yield the Discretized BPA (DBPA) [34], with the latter being shown to be superior to its continuous counterpart in terms of its speed of convergence.

II. Proof Methodologies for LA

Like other randomized learning algorithms, in the development of LA, one of the most important aspects to validate the design of an LA is to mathematically analyze its convergence, i.e., to investigate if the LA is able to converge to the optimal action with an arbitrarily large probability. If the answer is positive, the LA is considered "ε-optimal". The mathematical techniques used for the various formermentioned families of LA are quite distinct. The methodology for the family of FSSA involves formulating the Markov chain for the LA, computing its equilibrium probabilities, and then computing the asymptotic action selection probabilities. The proofs of convergence for VSSA involve the theory of small-step Markov processes, distance diminishing operators, and the theory of regular functions. The proofs for discretized LA involve the asymptotic analysis of the Markov chain that represents the LA in the discretized space, whence the *total* probability of convergence to the various actions is evaluated.

Understandably, the most difficult proofs involve the family of EAs. This is because the convergence involves two intertwined phenomena, i.e., the convergence of the reward estimates *and* the convergence of the action probabilities themselves. Ironically, the *combination* of these in the updating rule is what renders the EA fast. However, if the accuracy of the estimates are poor because of inadequate estimation (i.e., the sub-optimal actions are not sampled "enough number of times"), the convergence accuracy can be diminished, which is really a dilemma! The original proofs of convergence of PAs erroneously invoked the monotonicity property. These have since been rectified to invoke the submartingale property in [19], [33], [37].

A. Prior Flawed "Proofs" for EAs

As PAs hold the pioneering status in the development of EAs, a lot of work has been done on the analysis of the PAs [9], [10], [19], [21] and [24]. A thorough investigation on the convergence *and* finite-time behavior of PAs were done in [24], where the probability of selecting the optimal action was considered monotonically increasing after some time instant "t_0", and where the number of iterations that guarantees this t_0 and the number of iterations that allows the LA to converge to the optimal action after t_0, were bounded. Though the work in [24] was innovative and compass both the CPA and the DPA, there was a flaw in its reasoning, which rendered the analysis incorrect. To correct this flaw, new proofs for the CPA and DPA's convergence were proposed in [36] and [35] respectively, where the authors investigated the submartingale property of the probability of selecting the optimal action, and invoked the theory of regular functions to prove that the PAs converge to the optimal action in probability. They also claimed that the new proof methodology could be extended to prove the convergence of other Pursuit-based LAs. Though the proof for the convergence of PAs has been rectified by [36] and [35], the corresponding finite-time analyses is still open.

B. Complexity of Finite-time Analysis

Most of the analysis of Markov chains and LA deal with the scenario *after* the transient phase of the chain has elapsed. This is because the *asymptotic* analysis of Markov chains is well established. In the case of ergodic Markov chains it involves determining the eigenvector associated with the eigenvalue $\lambda = 1$, and this eigenvector can be computed by an explicit eigenvalue computation or by repeatedly multiplying the underlying Markov matrix. It can also be computed by simulation and recording the ensemble average of the time after a reasonable number of observations have passed. Alternatively, in the case of absorbing Markov chains, the analysis is obtained by computing the absorption probabilities determined by the solution of the first passage probabilities (from the Kolmogorov-Chapman equations). Both of these have formed the basis for analysis even when the number of actions are large and when they are arranged hierarchically.

The *Finite-time* analysis of Markov chains and LA is far more complex. The reason for this is that these depend on the non-unity eigenvalues of the Markov matrix and the way by which they lead to the convergence to the asymptotic value in a geometric matter. Unlike the asymptotic case, which only involves the case when $\lambda = 1$, the *Finite-time* analysis involves all the eigenvalues. Due to the complexity of the situation, understandably there is very little work done on the Finite-time analysis of any family of LA, including the families of FSSA and/or VSSA.

C. Goal and Contributions of this Paper

With the above as a backdrop, we state that this is precisely the contribution of this present paper, i.e., to achieve a *Finite-time* analysis for a specific PA. Since we have argued on the complexity of such an analysis, we consider the process by which the specific LA can converge to the optimal action. Thereafter, by invoking the submartingale property of the underlying random process, we succeed in achieving this Finite-time analysis.

[3] In order to highlight the distinct characteristics of the DPA and the original PA, for the rest of the paper, the latter is referred to as the Continuous Pursuit Algorithm (CPA), and PAs refers to the general family of Pursuit algorithms, including both the CPA and the DPA.

More specifically, in this paper, we aim at re-analyzing the Finite-time behavior of the DPA. That being said, we state that we are not providing a Finite-time analysis for the CPA based on submartingale property because the state space of the CPA is open and varies with time. As far as we know, this renders the analysis impossible, and the previous "proof" inaccurate. This is because one cannot prove this result by alluding to the monotonic property of the random process because the process is, quite simply, *not monotonic*. Our proof for the DPA is based on the new methodology used in [35] and [36], using which we succeed in obtaining bounds on the Finite-time behavior.

Thus, we can summarize the main contributions of this paper to be as follows:

- We rectify the error in the previous Finite-time analysis of the DPA, which has existed as a benchmark for over 20 years.
- We illustrate, for the first time, the dilemma between the DPA's exploitation and exploration. Using these, and based on our theoretical study, the existence of the DPA's optimal learning rate is justified via numerical analysis.

To pictorially clarify the main contribution of the paper, we include, in Figure 1, a schematic of how the DPA's finite time analysis is achieved[4].

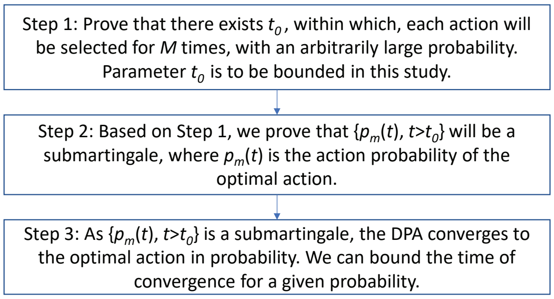

Step 1: Prove that there exists t_0, within which, each action will be selected for M times, with an arbitrarily large probability. Parameter t_0 is to be bounded in this study.
Step 2: Based on Step 1, we prove that $\{p_m(t), t>t_0\}$ will be a submartingale, where $p_m(t)$ is the action probability of the optimal action.
Step 3: As $\{p_m(t), t>t_0\}$ is a submartingale, the DPA converges to the optimal action in probability. We can bound the time of convergence for a given probability.

Fig. 1: A schematic of how the proof of the DPA's convergence is achieved.

III. Notations and a Brief Review of the DPA

Before we analyze the Finite-time behavior of the DPA, we present the notations used in the DPA and the algorithm (and the proofs), as follows:

- r: The number of actions.
- α_i: The i^{th} action that can be selected by the LA, $\alpha_i \in \{\alpha_1, \dots \alpha_r\}$.
- t: Time index of the learning process and t belongs the natural numbers.
- $\mathbf{P}(t)$: The action probability vector $\mathbf{P}(t) = [p_1(t), p_2(t), ..., p_r(t)]$ at time step t.
- $p_i(t)$: The i^{th} element of the action probability vector, $\mathbf{P}(t)$, at time step t.
- $u_i(t)$: The number of times that action α_i has been rewarded by time step t.
- $v_i(t)$: The number of times that action α_i has been selected by time step t.
- $\mathbf{D} = [d_1, d_2, \dots, d_r]$: The true reward probability vector of the Environment.
- d_i: The i^{th} element of the true reward probability vector $\mathbf{D}$.
- $\hat{\mathbf{D}}(t) = [\hat{d_1}(t), \hat{d_2}(t), \dots, \hat{d_r}(t)]$: The reward probability estimates vector of the Environment at time step t.
- $\hat{d_i}(t)$: The i^{th} element of the reward probability estimates vector $\hat{\mathbf{D}}(t)$, $\hat{d_i}(t) = \frac{u_i(t)}{v_i(t)}$.
- m: The index of the optimal action.
- h: The index of the greatest element in $\hat{\mathbf{D}}(t)$.
- $R(t)$: The response from the Environment at time step t, where $R(t) = 0$ corresponds to a Reward, and $R(t) = 1$ to a Penalty.
- Δ: The discretized step size, where $\Delta = \frac{1}{rN}$, with N being a positive integer.

The DPA follows a "pursuit" paradigm of learning, which consists of three steps. Firstly, it maintains an action probability vector $\mathbf{P}(t) = [p_1(t), p_2(t), ..., p_r(t)]$ to determine the action to be selected, where $\sum_{i=1}^{r} p_i(t) = 1$, $\forall t$. More specifically, at time t, an action is chosen according to the probability distribution $\mathbf{P}(t)$. Secondly, it maintains running ML reward probability estimates to determine which action can be reckoned to be the "best" in the current iteration. Thus, it updates $\hat{d_i}(t)$ based on the Environment's response as:

$$u_i(t) = u_i(t-1) + (1 - R(t)),$$
$$v_i(t) = v_i(t-1) + 1,$$
$$\hat{d_i}(t) = \frac{u_i(t)}{v_i(t)}.$$

Thirdly, based on the response of the Environment and the knowledge of the current best action, the DPA increases the probability of selecting *this* action as per the Discretized L_{R-I} rules. So if $\hat{d_h}(t)$ is the greatest element in $\hat{\mathbf{D}}(t)$, we update $p(t)$ as:

```
If $R(t) = 0$ Then
    $p_i(t+1) = \max\{p_i(t) - \Delta, 0\}$, $\forall\ i \neq h$,
    $p_h(t+1) = 1 - \sum_{\forall i,\ i \neq h} p_i(t+1)$.
Else
    $\mathbf{P}(t+1) = \mathbf{P}(t)$.
EndIf
```

We now consider the various methods by which the Finite-time analysis for the DPA has been tackled.

IV. Previous Finite-time Analysis on the DPA

To place our current result in the right perspective, it is prudent to consider the previous Finite-time analysis on the DPA submitted in [24]. In that work, t_0 is defined as the time instant after which the probability of selecting the optimal action, i.e., $p_m(t)$, $\forall t > t_0$, is assumed to be monotonically increasing. The Finite-time analysis can thus be divided into two periods, namely, the one before the time instant t_0 and the one after t_0. For ease of expression, we call the learning process before time t_0 as the Learning Period (LP), and the learning process after t_0 as the Converging Period (CP). We will outline how the previous paper [24] analyzed the DPA's behavior during both the Learning Period and the Converging Period, and then point out where the flaw of the analysis lies.

[4] We are grateful to the anonymous Referee who requested this figure.

We first summarize the analysis on the LP [24], where the aim is to calculate the bound of t_0. Since $p_m(t)$ is assumed to be monotonically increasing $\forall t > t_0$, a certain accuracy for the reward probability estimates is required at t_0 for all actions. To achieve the required accuracy for these estimates, each action needs to be selected at least a certain number of times. We denote $M_i(t_0), 1 \leq i \leq r$, as the number of times that each action is *required* to be selected in order to achieve the accuracy of the estimate, and then define $M = \max\{M_i(t_0), 1 \leq i \leq r\}$. Let $n_i(t)$ be the number of times that α_i has been selected by time t, and θ be a positive number close to 0. The next step is to determine the time instant t_0, after which, $Pr(n_i(t) \geq M) \geq 1-\theta$ holds[5]. Note that we have to describe the event "$n_i(t) \geq M$" from the perspective of the corresponding probability for the given parameter θ, due to the randomness of the behavior of the DPA. The method to achieve this is briefly explained below, while the reader is referred to [24] for additional details.

First of all, we have

$$\begin{aligned} & Pr(n_i(t) \geq M) \geq 1-\theta \\ \Longleftrightarrow\; & Pr(n_i(t) < M) < \theta \\ \Longleftrightarrow\; & \sum_{j=0}^{M-1} Pr(n_i(t) = j) < \theta \\ \Longleftarrow\; & Pr(n_i(t) = j) < \frac{\theta}{M}, \forall j, \end{aligned} \tag{1}$$

where the symbol $\Longleftrightarrow$ is meant to state "is equivalent to", while the symbol $\Longleftarrow$ indicates "can be deduced by". Obviously, if we can find the upper bound of $Pr(n_i(t) = j)$ as a function of t_0, we can derive the requirement of t_0 as a function of M and θ.

Clearly, $p_i(t) \geq p_i(0) - t\Delta$ holds. Therefore, we have

$$Pr(\alpha(t) \neq \alpha_i) \leq 1 - (p_i(0) - t\Delta) = 1 - p_i(0) + t\Delta,$$

and accordingly,

$$\begin{aligned} Pr(n_i(t) = j) &< \binom{t}{j} (1)^j [1 - p_i(0) + t\Delta]^{t-j} \\ &< t^j [1 - p_i(0) + t\Delta]^{t-j} \end{aligned} \tag{2}$$

holds. Hence, considering Eq. (1) and Eq. (2), if

$$t^j [1 - (p_i(0) - t\Delta)]^{t-j} < \frac{\theta}{M} \tag{3}$$

holds, Eq. (1) follows.

Moreover, the authors of [24] determined that if

$$\Delta = \frac{1}{r \times 2t} \tag{4}$$

is satisfied, Eq. (3) is implied.

Given the value of Δ as shown in Eq. (4), t_0 can be calculated as a function of M and θ. This is done by denoting t_0 as $t_0 = f(M, \theta)$ as shown in Eq. (5):

$$f(M,\theta) = \left\lceil \frac{2M}{\ln(1/\sigma)} \ln \left[\frac{M}{\ln(1/\sigma)} \frac{1}{\sigma} \left(\frac{M}{\theta} \right)^{\frac{1}{M}} \right] \right\rceil, \tag{5}$$

where $\sigma = \frac{2r-1}{2r}$.

The deduction of $f(M, \theta)$ involves quite a lot of algebraic manipulations, omitted here due to space limitations. But it can be seen as Lemma 4.1 and the corresponding proof in [24]. The final consequential result is that if $\Delta = \frac{1}{r \times 2t}$, then till the time instant $t_0 = f(M, \theta)$, the probability that each action is selected at least M times is greater than or equal to $1-\theta$. Note that the derivation of Eq. (5) does not require any monotonic property. It just returns the required time span by considering the required number of times, M, and the required probability resolution, θ, as inputs, both of which are entirely based on the nature of DPA. Therefore, Eq. (5) can still be reused in our new submartingale-based analysis, detailed in Section V.

A. "Analysis" on the Converging Period

We now revisit the "analysis" on the CP, where by this juncture, a proper learning parameter, Δ, has been determined, as shown in Eq. (4). Given Δ, and according to the updating rule of $p_i(t)$, it is ensured[6] that $p_i(t_0) \geq \frac{p_i(0)}{2}$, which implies that before t_0, each action gets no less than half of the initial probability of being selected, and whence each action can been selected a sufficiently large number of times (for example, M times) before the time instant t_0. However, it should be noted that Eq. (4) is only a theoretical assumption for Δ, which is very conservative. In practice, the learning parameter can be much greater than the theoretical assumption.

Let t_1 be the number of iterations that the LA takes to converge after t_0. The goal is to determine t_1 such that $p_m(t_0 + t_1) > 1 - \varepsilon$. As in [24], $p_m(t)$, $\forall t > t_0$ is considered to be monotonically increasing[7], and thus:

$$\begin{aligned} & p_m(t_0 + t_1) > 1 - \varepsilon \\ \Longleftrightarrow\; & \sum_{\forall j,\ j \neq m} p_j(t_0 + t_1) < \varepsilon, \\ \Longleftrightarrow\; & \sum_{\forall j,\ j \neq m} (p_j(t_0) - t_1\Delta) < \varepsilon, \quad \textit{Monotonicity is used.} \\ \Longleftarrow\; & p_j(t_0) - t_1\Delta < \frac{\varepsilon}{r-1}, \\ \Longleftarrow\; & 1 - t_1\Delta < \frac{\varepsilon}{r-1}, \\ \Longleftrightarrow\; & t_1 > \frac{1 - \frac{\varepsilon}{r-1}}{\Delta}. \end{aligned} \tag{6}$$

Hence, if after time t_0, the LA goes on running for more than $\left\lceil \frac{1 - \frac{\varepsilon}{r-1}}{\Delta} \right\rceil$ iterations, then $p_m(t) = p_m(t_0 + t_1) > 1 - \varepsilon$. Therefore, if we denote $T_0 = t_0 + \left\lceil \frac{1 - \frac{\varepsilon}{r-1}}{\Delta} \right\rceil = f(M, \theta) + \left\lceil \frac{1 - \frac{\varepsilon}{r-1}}{\Delta} \right\rceil$, then when $t > T_0$, the LA converges. This leads to the finite-time instant required for the LA to converge. Again, one should note that T_0 is very conservative because of the small theoretical learning parameter Δ.

Flaw in the previous "Proof": In the above analysis of the DPA's finite-time behavior, the second part that focuses on the CP, has an inherent infirmity. It is based on the flawed "assertion" that attempts to prove the DPA's ε-optimality, namely the one that claims that $p_m(t)$ is monotonically increasing after

[5] The expression $Pr(A)$ refers to the probability of the event A.

[6] $p_i(0)$ is most commonly set to be $\frac{1}{r}$.

[7] This is, precisely, the juncture where the argument is flawed.

t_0. This is a very strong claim that actually is unfounded. Rather, instead of the strong phenomenon of monotonicity, $p_m(t),\ t > t_0$, the property that does, indeed, hold true is its martingale-related property. After a certain time instant, $p_m(t)$ is, in fact, a submartingale [35], [36], which is a weaker phenomenon than monotonicity. Thus, to render the proof accurate, we need to re-define t_0 as the time instant after which, $p_m(t),\ t > t_0$ is *a submartingale*, and analyze the converging period based on such a submartingale phenomenon.

Having clarified this, it is also prudent to crystallize the differences and similarities between [35] and this current paper. With regard to the similarity: Both [35] and this current paper abandon the futile attempt to use the monotonicity property of the random process. The difference between the two papers is that whereas [36] deals with the asymptotic properties of the process, this current paper deals with the process' *finite-time* behavior, which as explained above, is far more cumbersome and difficult to formalize.

V. Our Proposed Finite-time Analysis of the DPA

Before we proceed, we first mention that our presently-proposed Finite-time Analysis does not depend on an eigenvalue-based strategy but rather by invoking the theory of martingales. First of all, we split the convergence into two phases. The first involves the period when the estimates converge accurately enough so that the Pursuit phenomenon can take over, and the second involves the LA converging to the best action. To achieve this, we first calculate the required number of times that each action has to be selected, say M. This is to guarantee that the stochastic process becomes a submartingale, which is required so as to attain the final convergence. Thereafter, even though the number M is known, we do not yet know the *time instant* at which all the actions will have been selected at least M times. To estimate *this* time, we devise a scheme to compute the quantity t_0, after which each action will have been selected at least M times with the high probability $1-\theta$, where due to the randomness of the learning process, t_0 can only be bounded probabilistically with a parameter θ. Once the quantity t_0 is calculated, we can guarantee that after t_0, the process becomes a submartingale with a high probability. Thereafter, we show how we can calculate the average time required in the convergence stage. These steps lead to a very interesting proof, and amazingly fascinating results that are quite distinct from the state-of-the-art.

As the submartingale property is weaker than the property of monotonicity, we are not able to bound t_1 but rather, provide a conservative bound for its expected value. The new finite-time analysis has two parts. The first phase (Section V-A) is to calculate t_0, where $p_m(t)$ becomes a submartingale. The second phase (Section V-B) is to derive $E[t_1]$. Therefore, if we denote $T_0 = t_0 + E[t_1]$, we can assert that on the average, (i.e., in the Expected sense), the LA converges when $t > T_0$.

A. Analysis of the Submartingale $p_m(t),\ \forall t > t_0$, and t_0

In this subsection, we briefly introduce the submartingale property of $p_m(t),\ \forall t > t_0$, and then proceed to derive the time span required for the random process to become a submartingale, i.e., t_0.

We define $q(t)$ as the probability that the reward probability estimate of the optimal action is the greatest among all the reward probability estimates at time t, i.e.,

$$q(t) = Pr(\hat{d}_m(t) > \hat{d}_i(t)), \forall i \neq m. \tag{7}$$

According to the description of the DPA algorithm, we see that the quantity $q(t)$ increases as $p_m(t)$ grows. Therefore, based on the action probability updating rules of the DPA where we have worked within the Reward-Inaction paradigm[8]:

$$\begin{aligned}
&E[p_m(t+1)|\mathbf{P}(t)] \\
&= p_m\left(d_m\left(q(p_m + c_t\Delta) + (1-q)(p_m - \Delta)\right) + (1-d_m)p_m\right) + \\
&\quad \sum_{\forall j,\ j\neq m} p_j\left(d_j\left(q(p_m + c_t\Delta) + (1-q)(p_m - \Delta)\right) + (1-d_j)p_m\right) \\
&= \sum_{j=1}^{r} p_j\left(d_j\left(q(p_m + c_t\Delta) + (1-q)(p_m - \Delta)\right) + (1-d_j)p_m\right) \\
&= \sum_{j=1}^{r}(p_j d_j q c_t\Delta) - \sum_{j=1}^{r} p_j d_j \Delta + \sum_{j=1}^{r} p_j d_j q\Delta + \sum_{j=1}^{r} p_j p_m \\
&= p_m + \sum_{j=1}^{r} p_j d_j\left(q(c_t\Delta + \Delta) - \Delta\right),
\end{aligned}$$

where, c_t represents the number of actions' probabilities, other than p_m, that have not converged to zero at time t. Therefore, c_t is bounded by 0 and $r-1$, and it is non-increasing as t grows. Note that the condition $c_t = 0$ implies that $p_m = 1$ holds, and this further implies that the entire random process has converged to a unit vector. However, in this phase of our analysis, our intention is to study the time span when c_t is in between 1 and $r-1$. The difference between $E[p_m(t+1)]$ and $p_m(t)$ can be expressed as:

$$\begin{aligned}
Diff_{p_m(t)} &= E[p_m(t+1)|\mathbf{P}(t)] - p_m(t) \\
&= \sum_{j=1}^{r} p_j(t) d_j\left(q(t)(c_t\Delta + \Delta) - \Delta\right).
\end{aligned} \tag{8}$$

Based on Eq. (8), $q(t)(c_t\Delta+\Delta)-\Delta > 0$ implies that $q(t) > \frac{\Delta}{c_t\Delta+\Delta}$. Thus, we conclude that the sequence $p_m(t),\ \forall t > t_0$, is a submartingale if there exist a time instant t_0 such that for every time instant $t > t_0$, $q(t) > \max\{\frac{\Delta}{c_t\Delta+\Delta}\} = \frac{1}{2}$. Therefore, if we can prove that for $\delta \in (0, \frac{1}{2})$, $\forall t > t_0$, $q(t) > 1-\delta$, it implies that for the specific value of $\delta = \frac{1}{2}$, $\forall t > t_0$, $q(t) > 1-\delta$, guaranteeing the submartingale property.

Consider a two-action Environment. Without loss of generality, let α_1 be the optimal action and α_2 the inferior one. We are to prove that $\forall t > t_0$,

$$Pr(\hat{d}_1(t) - \hat{d}_2(t) > 0) > 1-\delta.$$

If we define

$$H = d_1 - d_2,\ \hat{H}(t) = \hat{d}_1(t) - \hat{d}_2(t),$$

then

$$Pr(\hat{d}_1(t) - \hat{d}_2(t) > 0) \Leftrightarrow 1 - Pr(\hat{H}(t) - H \leq -H), \text{ and}$$

[8] In the interest of conciseness, $p_m(t)$ and $q(t)$ are respectively written as p_m and q whenever there is no confusion.

$$Pr(\hat{d_1}(t)-\hat{d_2}(t)>0)>1-\delta \Leftrightarrow Pr(\hat{H}(t)-H\leq -H)\leq \delta.$$

Hence, we can equivalently prove that

$$Pr(\hat{H}(t)-H\leq -H)\leq \delta.$$

To achieve this, we denote $n_1(t)$ as the number of times that α_1 has been selected up to time t. Then, by invoking the "two-action" version of Hoeffding's Inequality [3] (Page 16), we have:

$$Pr(\hat{H}(t)-H\leq -H|n_1(t)=n)\leq e^{-\frac{2H^2}{n^{-1}+(t-n)^{-1}}}.$$

We thus are to find a proper value of n such that

$$e^{-\frac{2H^2}{n^{-1}+(t-n)^{-1}}}\leq \delta, \tag{9}$$

which guarantees that $Pr(\hat{H}(t)-H\leq -H)\leq \delta$.

Considering n as the variable, we solve Eq. (9) and get

$$e^{-\frac{2H^2}{n^{-1}+(t-n)^{-1}}}>\delta, \text{ when } t<\frac{-2\ln\delta}{H^2},$$

and when $t\geq \frac{-2\ln\delta}{H^2}$,

$$e^{-\frac{2H^2}{n^{-1}+(t-n)^{-1}}}\begin{cases}\leq \delta, & \text{when } n_{r1}\leq n\leq n_{r2},\\ >\delta, & \text{otherwise.}\end{cases},$$

where n_{r1} and n_{r2} are the two real roots of Eq. (9):

$$n_{r1}=\frac{t}{2}-\frac{\sqrt{H^2t^2+2t\ln\delta}}{2H}, \text{ and}$$
$$n_{r2}=\frac{t}{2}+\frac{\sqrt{H^2t^2+2t\ln\delta}}{2H}. \tag{10}$$

Therefore, to make sure that Eq. (9) holds, we need to ensure that $\forall t>t_0$, $n_{r1}\leq n\leq n_{r2}$. Omitting the algebraic manipulations here[9], we present the conclusion that $t_0\geq \frac{-2\ln\delta}{H^2}$ and $n\geq \frac{-\ln\delta}{H^2}$.

As the above analysis is also applicable to α_2, which is, indeed, symmetric to α_1 in this two action Environment considered, the consequence is the following: Let us suppose that we define the time instant t_0 such that within the time defined by t_0, α_1 and α_2 have each been selected more than $\left\lceil\frac{-\ln\delta}{H^2}\right\rceil$ times. As a result of this, in such a case:

$$e^{-\frac{2H^2}{n^{-1}+(t-n)^{-1}}}\leq \delta,$$

whence we can conclude that for the given $\delta\in(0,\frac{1}{2})$, $\forall t>t_0$,

$$q(t)=Pr(\hat{d_1}(t)-\hat{d_2}(t)>0)>1-\delta.$$

The result can be easily extended to the r-action Environment, where if we define:

$$H_j=d_m-d_j, j\neq m,$$
$$\hat{H_j}(t)=\hat{d_m}(t)-\hat{d_j}(t), j\neq m.$$

Then, given any $\delta\in(0,\frac{1}{2})$, if we denote $\delta^\star=1-\sqrt[r-1]{1-\delta}$, we can show that there exists a time instant t_0, such that within the time defined by t_0, α_m has been selected more than $\left\lceil\frac{-\ln\delta^\star}{(\min\{H_j\})^2}\right\rceil$ times, and α_j, $\forall j\neq m$, has been selected more than $\left\lceil\frac{-\ln\delta^\star}{H_j^2}\right\rceil$ times. Consequently, for $\forall t>t_0$, $q_j(t)>1-\delta^\star$ holds and $q(t)\geq \prod_{j,\ j\neq m} q_j(t)>1-\delta$.

As there exists a t_0 such that if $\forall t>t_0$, $q(t)>\frac{1}{2}$, holds, $p_m(t)$, $\forall t>t_0$, is indeed a submartingale. Thus, according to the submartingale convergence theory [14] (Page 440),

$$p_m(\infty)=0 \text{ or } 1.$$

If we denote $\mathbf{e}_j$ as the unit vector with the j^{th} element being 1 and 0 otherwise, then $p_m(\infty)=1 \Longleftrightarrow \mathbf{P}(\infty)=\mathbf{e}_m$. If we define the convergence probability

$$\Gamma_m(\mathbf{P})=Pr(\mathbf{P}(\infty)=\mathbf{e}_m|\mathbf{P}(0)),$$

where $\mathbf{P}(0)$ is the initial action probability vector, then by utilizing the theory of Regular functions, we can prove that $\Gamma_m(\mathbf{P})\to 1$ [35]. Thus the DPA being ε-optimal is proven based on the submartingale property of $p_m(t)$, $\forall t>t_0$. Besides, the number of times each action is required to be selected to achieve the reward probability estimates accuracy, $q(t)>1-\delta$, can be provided by

$$M=\left\lceil\frac{-\ln\delta^\star}{(\min\{H_j\})^2}\right\rceil, \text{ where } \delta^\star=1-\sqrt[r-1]{1-\delta}. \tag{11}$$

Given the Δ and M as in Eq. (4) and Eq. (11), respectively, by the same deduction as in [24], t_0 can be calculated as $t_0=f(M,\theta)$.

B. The Average Converging Period of the Submartingale

We now present the new analysis on the behavior of the DPA in the Converging Period. Note that in the new analysis, the fundamental difference is the definition of t_0, which is, after t_0, the probability sequence of $p_m(t)$, $t>t_0$, does not monotonically increase, but is a submartingale.

We shall utilize Wald's Equality to achieve this analysis. In Wald's Equality [15], we have X_i, $i=1,2,...$, as a sequence of Independent and Identically Distributed (IID) random variables, and I as a random variable that is a stopping time for the sequence of X_i. Let $S_I=X_1+X_2+...+X_I$, if $E(I)$ and $E(X)$ are finite, then $E(S_I)=E(X)E(I)$.

Firstly, we set a new time system $t'=t-t_0$, which differs from the old time system t, by t_0. As we are dealing with the LA's behavior after time t_0, by working within the new time system, we are able to get rid of the notation that involves t_0, and to affirm that for the sequence after t_0, the time index starts from the time instant 0.

Secondly, we define

$$X(t')=\begin{cases}(r-1)\Delta, \text{ with probability } q(t')\\ -\Delta, \text{ with probability } 1-q(t')\end{cases}. \tag{12}$$

For ease of analysis, if we further freeze the update of $q(t')$ and adopt a fixed value of $q(t')$ when $t'=0$, denoted by q_0, then $X(t')$ becomes a sequence of IID random variables, whose expectation can be calculated as

$$\begin{aligned}E(X(t'))&=q_0(r-1)\Delta+(1-q_0)\times(-\Delta)\\ &=(q_0r-1)\Delta.\end{aligned} \tag{13}$$

[9] The reader is referred to [35] for additional details.

Thirdly, we consider $p_m(t')$ as a random point walking between the interval of $[0,1]$, with its initial position being at $p_m(t=t_0)$[10]. According to the action probability updating rules of the DPA,

$$p_m(t') = p_m(t_0) + X(1) + X(2) + \dots + X(t'),$$

and so, if we define $S_{t'} = p_m(t') - p_m(t_0)$, then

$$S_{t'} = p_m(t') - p_m(t_0) = X(1) + X(2) + \dots + X(t').$$

Obviously, $p_m(t')$ stops walking when it reaches the point of 0 or 1. Accordingly, $S_{t'}$ stops at the point of $-p_m(t_0)$ or $1-p_m(t_0)$. The probability of $p_m(t')$ stopping at point 1 (i.e., when $S_{t'} = 1-p_m(t_0)$) is $\Gamma_m(\mathbf{P})$, and the probability[11] of $p_m(t')$ stopping at point 0 (i.e, when $S_{t'} = -p_m(t_0)$) is $1-\Gamma_m(\mathbf{P})$. Therefore, we calculate the expectation of $S_{t'}$ as:

$$\begin{aligned} E(S_{t'}) &= \Gamma_m(\mathbf{P})(1-p_m(t_0)) + (1-\Gamma_m(\mathbf{P}))(-p_m(t_0)) \\ &= \Gamma_m(\mathbf{P}) - p_m(t_0). \end{aligned} \tag{14}$$

The number of times it takes for $p_m(t')$ to reach 0 or 1 (or equivalently, for $S_{t'}$ to reach $-p_m(t_0)$ or $1-p_m(t_0)$) is called the Stopping Time, which, we also denote as t_1 in the interest of keeping the notation consist with what has been defined in the flawed analysis. According to Wald's Equality, the expectation of the Stopping Time t_1 is:

$$E(t_1) = \frac{E(S_{t_1})}{E(X(t'))} = \frac{\Gamma_m(\mathbf{P}) - p_m(t_0)}{(q_0 r - 1)\Delta}. \tag{15}$$

We thus have given an explicit expression for $E(t_1)$.

The consequence of the above is the following: We see from Eq. (15) that if $\Gamma_m(\mathbf{P})$, $p_m(t_0)$, q_0 and Δ can be determined, $E(t_1)$ can be estimated yielding a rough idea about the number of time instances it takes for the LA to converge.

In summary, the main hurdles that we encountered when arriving at these results are the following[12]:

1) To calculate M, i.e., the required number of times that each action has to be selected for the given probability parameter θ. This is to guarantee that the stochastic process becomes a *submartingale*. With M and θ available, we can bound the time that the stochastic process requires to become a submartingale. The key to this involves invoking Hoeffding's Inequality.
2) To estimate the mean time of the convergence period for the given probability parameter $\Gamma_m(\mathbf{P})$. When the process becomes a submartingale, it is necessary to bound, in a statistical sense, the time horizon that the process requires to converge. The key to obtain this estimate is to invoke Wald's Equality.

C. Details of the Terms Involved

Although the various terms have been derived above, we believe that it will be prudent to explain their respective significances. Thus, in this section, we shall examine these four items one by one.

1) We first explain the meaning of the term $\Gamma_m(\mathbf{P})$. $\Gamma_m(\mathbf{P})$ is the probability of the LA converging to the correct unit vector, and so its value must be as close to unity as desired to guarantee ε-optimality. Generally speaking, one uses a value of $\Gamma_m(\mathbf{P})$ which is smaller than $1-\theta$. The reason for this is that $1-\theta$ is the lower bound of the probability that each action has been selected a sufficient number of times in order to make the process a submartingale. Once the process exhibits the property of a submartingale, it is possible for the LA to become ε-optimal with accuracy $\Gamma_m(\mathbf{P})$.
2) We now consider the learning parameter Δ and how it is related to Eq. (4). To reach a given converging probability, the parameter that one can adjust is Δ. This is because for every Δ, there exists a corresponding $\Gamma_m(\mathbf{P})$. The smaller the value that Δ is, the closer $\Gamma_m(\mathbf{P})$ is to unity. From a pragmatic perspective, a small Δ ensures that each action is selected enough number of times before the LA converges, and this, in turn, ensures that that the estimation of each reward probability is accurate enough to rank the actions properly. Once the ranking of the actions becomes accurate enough, the LA will be more likely to converge to the best action, thus enhancing the converging probability, $\Gamma_m(\mathbf{P})$. This rationale is precisely what Eq. (4) is based on. Indeed, if Δ is defined as in Eq. (4), the given converging probability can be guaranteed. Therefore, we set $\Delta = \frac{1}{r \times 2t}$. Since t is required to be greater than t_0, we can configure t as t_0+1.
3) The next question is that of setting $p_m(t_0)$. Observe that as mentioned in Section IV, $p_i(0)$ is configured as $\frac{1}{r}$, as is customarily done. If the step size is $\Delta = \frac{1}{r \times 2(t_0+1)}$, at time instant t_0, in the worst case, this action probability will become:
$$p_m(t_0) = \frac{1}{r} - \Delta t_0 = \frac{1}{r} - \frac{t_0}{2r(t_0+1)} = \frac{1}{2r} + \frac{1}{2r(t_0+1)}. \tag{16}$$
This means that in the worst case, the probability $p_m(t_0)$ is decreasing for all those t_0 iterations. In other words, $p_m(t_0) \geq \frac{1}{2r} + \frac{1}{2r(t_0+1)}$ holds. We can thus set $p_m(t_0) = \frac{1}{2r} + \frac{1}{2r(t_0+1)}$.
4) The last quantity to be determined is $q_0 < 1$. The value of $q_0 < 1$ needs only to be greater than $\frac{1}{2}$ to keep $p_m(t)$, $\forall t > t_0$ to be a submartingale. Note also that $q_0 \geq 1-\delta$ holds. Thus, q_0 can be set to be any value in the interval of $\left(\max\{1-\delta, \frac{1}{2}\}, 1\right)$. The equality, where it is set to $1-\delta$, is when the process is neither a submartingale or a supermartingale, but a "pure" martingale.

In summary, for the required $\Gamma_m(\mathbf{P})$, if we set:

$\Delta = \frac{1}{r \times 2(t_0+1)}$,
$p_m(t_0) = \frac{1}{2r} + \frac{1}{2r(t_0+1)}$, and

[10] $p_m(t=t_0)$ is $p_m(t_0)$ in the old time system. Without causing any confusion, we consider $p_m(t'=0) = p_m(t=t_0)$ a constant, and will write them simply as $p_m(t_0)$ in the rest of the paper.

[11] $\Gamma_m(\mathbf{P})$ is precisely the probability of the absorbing LA converging to the unit vector $\mathbf{e}_m$. This is exactly the notation used in the literature [14].

[12] We are grateful to the Anonymous Referee who requested this and the following Sub-section.

q_0 to be any value from the interval $\left(\max\{1-\delta, \frac{1}{2}\}, 1\right)$, the quantity $E(t_1)$ can be calculated. This would thus yield a rough estimate as to how much time the LA would spend after time t_0 to converge.

Note that as $\Delta = \frac{1}{r\times 2(t_0+1)}$ is a very conservative assumption and $p_m(t_0) = \frac{1}{2r} + \frac{1}{2r(t_0+1)}$ is the lowest value that $p_m(t_0)$ can be. In addition, $q(t')$ will increase as the iterations proceed while the value utilized in the calculation, i.e., q_0 is when $t' = 0$, which is also very conservative. Thus the results stated for $E(t_1)$ and $t_0 + E(t_1)$ are very conservative.

VI. Numerical Results

In the previous sections, we derived closed-form expressions for the number of iterations needed for the DPA (with its specified resolution parameters) to converge. We now formally summarize the sequence of tasks that this entails, and then discuss the result of the bound itself by providing certain concrete values for the parameters. The three resolution parameters utilized in our calculation are $\Gamma_m(\mathbf{P})$, θ and δ, where:

- $\Gamma_m(\mathbf{P})$ is the final accuracy of the LA after convergence;
- θ is the resolution of the probability that each action has been selected at least M times till the time instant t_0;
- As per Item (4) above, δ can be used to derive the required number of times that an action is selected, i.e., M. It is also utilized for calculating t_0.

Based on the results of Sections IV and V, we now formally outline the procedure to calculate the bound when one uses concrete values for the parameters. To calculate the total bound, one can follow the following steps:

1) We first calculate the required number of times that each action has to be selected, i.e., M, by using Eq. (11). This is to guarantee that the stochastic process becomes a submartingale, which is required so as to attain the final convergence. Although M is calculated by invoking Eq. (11), the required time span to fulfill this condition is not yet bounded. Thus, even though the number M is known, we do not yet know the *time instant* at which all the actions will have been selected at least M times.
2) To estimate this time, we adopt Eq. (5). This yields the quantity t_0, after which each action will have been selected at least M times with the high probability $1-\theta$. Note that due to the randomness of the learning process, the time instant t_0 can only be bounded probabilistically with a parameter θ. In other words, when t_0 approaches infinity, all actions will surely be selected at least M times, which is also indicated from Eq. (5). However, once the quantity t_0 is calculated, we know that after t_0, the process becomes a submartingale with a high probability. Thereafter, one can use Eq. (15) to calculate the average time required in the convergence stage.

Figure 2 illustrates how the time bound for the DPA's convergence has been calculated.

A. Numerical Examples for the Bounds

To clarify the above, we have numerically calculated the value of the bound for a few specific configurations.

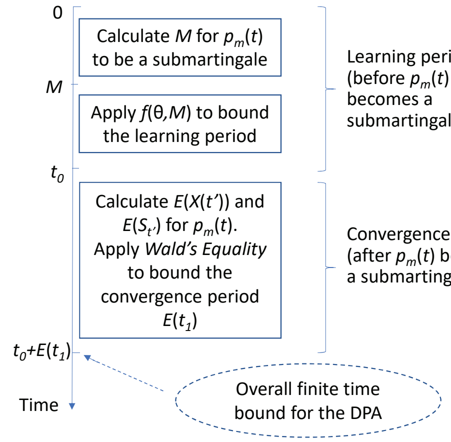

0
Calculate M for $p_m(t)$ to be a submartingale
M
Apply $f(\theta,M)$ to bound the learning period
t_0
Learning period (before $p_m(t)$ becomes a submartingale)
Calculate $E(X(t'))$ and $E(S_{t'})$ for $p_m(t)$. Apply *Wald's Equality* to bound the convergence period $E(t_1)$
Convergence period (after $p_m(t)$ becomes a submartingale)
t_0+$E(t_1)$
Time
Overall finite time bound for the DPA

Fig. 2: A schematic of how the time bound of the DPA's convergence is calculated.

1) In the first set of experiments (in Table I), we have fixed the final accuracy of the LA, ($\Gamma_m(\mathbf{P})$), and the resolution of the probability that the optimal action has to converge to, for it to be considered as being optimal at time t_0, (δ). We have then varied the resolution of the probability that each action has been selected at least M times, (θ).
2) In the second set of experiments (in Tables II and III), we have fixed the final accuracy of the LA ($\Gamma_m(\mathbf{P})$, and resolution of the probability that each action has been selected at least M times, (θ). We have then varied the resolution of the probability that the optimal action has to converge to for it to be considered as being optimal at time t_0, (δ).
3) In the final set of experiments (in Table IV), we have fixed the resolution of the probability that each action has been selected at least M times, (θ), and the resolution of the probability that the optimal action has to converge to for it to be considered as being optimal at time t_0, (δ). We have then varied the final accuracy of the LA, ($\Gamma_m(\mathbf{P})$).

The results obtained have been recorded in Tables I, II, III and IV respectively. To generate those results, we have set the number of actions, r, to be 2, and the difference between the true reward probabilities, H, to be 0.2. For the results in Table I, we configured $\Gamma_m(\mathbf{P})$ as $1-\theta-0.001$, δ as 0.2, and q_0 as 0.96. The reason to configure $\Gamma_m(\mathbf{P})$ as $1-\theta-0.001$ was to make sure that $\Gamma_m(\mathbf{P}) < 1-\theta$ holds.

TABLE I: The times required for the DPA to converge when θ varies for the setting where $\Gamma_m(\mathbf{P})$ as $1-\theta-0.001$, δ as 0.2, and q_0 as 0.96.

θ	M	t_0	$E(t_1)$	$t_0+E(t_1)$
0.1	41	1,538	4,312	5,880
0.05	41	1,543	4,691	6,234
0.04	41	1,544	4,762	6,306
0.03	41	1,546	4,834	6,380
0.02	41	1,549	4,912	6,461
0.01	41	1,554	4,995	6,549

For the results in Table II, we configured $\Gamma_m(\mathbf{P})$ as 0.98, θ as 0.015, and q_0 as $1-\delta+0.001$. To further illustrate these convergence phenomena, we also illustrated the numerical results when we configured $\Gamma_m(\mathbf{P})$ as 0.99, θ as 0.005, and q_0 as $1-\delta+0.0005$. These results are given in Table III.

TABLE II: The times required for the DPA to converge when δ varies for the settings where $\Gamma_m(\mathbf{P})$ as 0.98, θ as 0.015, and q_0 as $1-\delta+0.001$.

δ	M	t_0	$E(t_1)$	$t_0+E(t_1)$
0.5	18	603	881,330	881,940
0.4	23	798	11,545	12,343
0.3	31	1,124	8,169	9,293
0.2	41	1,151	7,526	9,077
0.1	58	2,314	8,427	10,741
0.08	64	2,591	8,988	11,579
0.06	71	2,920	9,669	12,589
0.04	81	3,398	10,764	14,162
0.02	98	4,230	12,841	17,071

TABLE III: The times required for the DPA to converge when δ varies for the settings where $\Gamma_m(\mathbf{P})$ as 0.99, θ as 0.005, and q_0 as $1-\delta+0.0005$.

δ	M	t_0	$E(t_1)$	$t_0+E(t_1)$
0.3	31	1,132	8,361	9,493
0.275	33	1,216	7,985	9,201
0.25	35	1,300	7,685	8,985
0.23	37	1,386	7,587	8,973
0.215	39	1,472	7,634	9,106
0.2	41	1,559	7,682	9,241
0.15	48	1,868	7,891	9,759
0.12	54	2,138	8,319	10,457
0.1	58	2,321	8,579	10,900

For the results in Table IV, we configured θ as $1-\Gamma_m(\mathbf{P})-0.001$, δ as 0.2, and q_0 as 0.98.

TABLE IV: The times required for the DPA to converge when $\Gamma_m(\mathbf{P})$ varies.

$\Gamma_m(\mathbf{P})$	M	t_0	$E(t_1)$	$t_0+E(t_1)$
0.7	75	1,530	2,870	4,400
0.8	75	1,533	3,514	5,047
0.9	75	1,538	4,167	5,751
0.95	75	1,543	4,502	6,045
0.96	75	1,545	4,573	6,118
0.97	75	1,547	4,643	6,190
0.98	75	1,550	4,717	6,267
0.99	75	1,555	4,767	6,352

VII. Discussions on the Numerical Bounds

From Tables I and IV, we can observe that both t_0 and $E(t_1)$ monotonically increase as θ decreases and as $\Gamma_m(\mathbf{P})$ grows. In fact, $1-\theta$ and $\Gamma_m(\mathbf{P})$ describe the same trend. As mentioned before, $\Gamma_m(\mathbf{P}) \leq 1-\theta$ holds because $1-\theta$ describes the probability of the learning process being a submartingale from t_0 onward, which is the preliminary phase of the final convergence. Since we make one parameter as a function of the other parameter in the calculation, i.e., $\theta = 1-\Gamma_m(\mathbf{P})-0.001$, the trends of the results in those tables are the same. We also calculate the results when we decouple those two parameters, whence we observe the same trend. The reason of having this trend is, in general, because of the fact that the greater the desired convergence accuracy, the larger is the number of iterations that is needed.

As opposed to the results in Tables I and IV, the results in Tables II and III are quite different. As shown in Tables II and III, when δ decreases, M and t_0 increase monotonically while $E(t_1)$ has a decreasing trend in the beginning and increases again when δ is greater than a certain value. This is a very interesting trend which reveals the underlying characteristic nature of the DPA. When δ is large, i.e., $\delta = 0.5$, q_0 is small. A smaller value of q_0 indicates a smaller t_0 because less samples are required for the underlying estimation. However, a smaller value of q_0 also provides *less accurate* reward probability estimates and a consequent less proper ranking of the actions till t_0. In such a case, the LA is less certain about which action is the best, and thus spends a significant amount of time after t_0, in converging to the correct action for the given required accuracy. This, in turn, results in a larger $E(t_1)$. As δ becomes smaller, q_0 increases. A larger value of q_0 requires a larger number of samples for the estimation, and this results in a larger t_0. But after the time instant t_0, as the LA is more certain about the best action, it converges faster, resulting in a smaller $E(t_1)$. This explains the falling trend of the required number of iterations from 881,940 for $\delta = 0.5$ to 7,526 for $\delta = 0.2$, which reveals the process of balancing between the exploration and exploitation phases.

As δ becomes even smaller, i.e., when δ is 0.1 and smaller, $E(t_1)$ increases again. This is due to the fact that the high resolution of the initial accuracy at t_0 results in a small learning rate, Δ. The small value of Δ leads to a very slow convergence before and after t_0, which makes $E(t_1)$ large again, i.e., from 7,526 for $\delta = 0.2$ to 12,841 for $\delta = 0.02$.

In general, as δ decreases, t_0 increases and therefore Δ decreases. From the results of Table II and Table III, we realize that the total number of steps has a convex trend with respect to δ, and thus Δ. To illustrate this trend more clearly, we have also plotted the time bounds of the DPA based on the configurations of the environment given of Tables II and III. This plot is given in Figure 3. Obviously, the function is not a convex function, but has a convex envelop. Therefore, we can also understand, theoretically, that there exists an optimal[13] Δ, which can deliver a required overall convergence accuracy, i.e., $\Gamma_m(\mathbf{P})$, with a minimum number of steps.

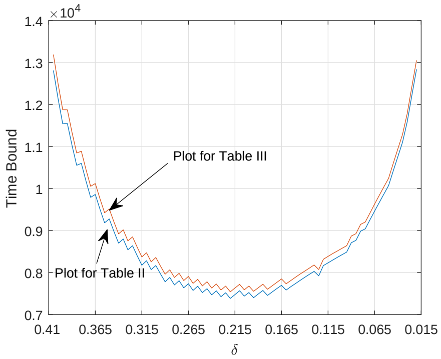


Fig. 3: The time bounds of the DPA's convergence in different configurations as a function of δ.

As the total time required for the LA to converge consists of both t_0 and t_1, i.e., $t_0+E(t_1)$, Δ needs to be carefully chosen to

[13] It is worth mentioning that the result of this study indicates that although large δ (and thus large Δ) can reduce t_0, there is no way to reduce the total number of steps in order to achieve the same given accuracy of the convergence $\Gamma_m(\mathbf{P})$. In reality, when Δ is large, the total number of steps is indeed reduced, but the convergence accuracy of the LA is also compromised, which does not conflict with our theoretic analysis.

minimize this sum. However, this is never easy, because there are many parameters for a given Environment, including the number of actions, r, the parameter, H, indicating the hardness of the Environment etc., and these will collectively influence the relationship between Δ and $\Gamma_m(\mathbf{P})$. That is why the task of deciding the optimal learning rate for a specific Environment is still open. This is precisely what research in reinforcement learning has been trying to achieve for decades, namely to balance between the dilemmas of exploration and exploitation in an "optimal" way!

VIII. Conclusions

In this paper, we have concentrated on the the Finite-time Analysis of the Discrete Pursuit Automaton (DPA), which is probably one of the fastest and most accurate reported LA. This analysis does not invoke an eigenvalue-eigenvector strategy. Rather, we work with the time required for the LA to attain a certain level of confidence and then proceed to its absorption state. Although such an analysis was carried out many years ago, we have shown how that the previous monotonicity-based analysis was flawed. In this paper, we have shown that the property that should be invoked is the submartingale property, forcing the proof to be much more intricate. We have rectified the flaw and demonstrated the derivation of convergence probability based on such a submartingale phenomenon. From the derived analysis, we are able to discover and explicitly clarify, for the first time, the underlying dilemma between the DPA's exploitation and exploration properties. We also confirm, in a non-trivial manner, the existence of the optimal learning rate, which yields a better comprehension for the nature of the DPA.

References

[1] A. F. Atlassis, N. H. Loukas, and A. V. Vasilakos. The use of learning algorithms in ATM networks call admission control problem: A methodology. *Computer Networks*, 34:341–353, 2000.

[2] A. F. Atlassis and A. V. Vasilakos. The use of reinforcement learning algorithms in traffic control of high speed networks. *Advances in Computational Intelligence and Learning*, pages 353–369, 2002.

[3] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, pp. 13–30, 1963.

[4] L. Jiao, X. Zhang, O.-C. Granmo, and B. J. Oommen, "A Bayesian learning automata-based distributed channel selection scheme for cognitive radio networks," in *Proceedings of IEAAIE2014*. Cham: Springer International Publishing, 2014, pp. 48–57.

[5] L. Jiao, X. Zhang, B. J. Oommen, and O.-C. Granmo, "Optimizing channel selection for cognitive radio networks using a distributed Bayesian learning automata-based approach," *Applied Intelligence*, vol. 44, no. 2, pp. 307–321, Mar 2016.

[6] J. Zhang, C. Wang, and M. Zhou, "Fast and epsilon-optimal discretized pursuit learning automata," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2089 – 2099, 2014.

[7] J. Kabudian, M. R. Meybodi, and M. M. Homayounpour. Applying continuous action reinforcement learning automata (CARLA) to global training of hidden markov models. In *Proceedings of the International Conference on Information Technology: Coding and Computing, ITCC'04*, pages 638–642, Las Vegas, Nevada, 2004.

[8] S. Lakshmivarahan, *Learning Algorithms Theory and Applications*. New York Springer-Verlag, 1981.

[9] J. K. Lanctot and B. J. Oommen, "On discretizing estimator-based learning algorithms," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 2, pp. 1417–1422, 1991.

[10] J. K. Lanctot and B. J. Oommen, "Discretized estimator learning automata," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 22, no. 6, pp. 1473–1483, 1992.

[11] M. R. Meybodi and H. Beigy. New learning automata based algorithms for adaptation of backpropagation algorithm pararmeters. *International Journal of Neural Systems*, 12:45–67, 2002.

[12] S. Misra and B. J. Oommen. GPSPA: A new adaptive algorithm for maintaining shortest path routing trees in stochastic networks. *International Journal of Communication Systems*, 17:963–984, 2004.

[13] K. Najim and A. S. Poznyak. *Learning Automata: Theory and Applications*. Pergamon Press, Oxford, 1994.

[14] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*. Prentice Hall, 1989.

[15] R. Nelson, *Probability, Stochastic Processes, and Queueing Theory*. Springer, 1995.

[16] M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis. Learning automata: Theory, paradigms, and applications. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 32(6):706–709, December 2002.

[17] M. S. Obaidat, G. I. Papadimitriou, A. S. Pomportsis, and H. S. Laskaridis. Learning automata-based bus arbitration for shared-edium ATM switches. *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, 32:815–820, 2002.

[18] B. J. Oommen, "Stochastic searching on the line and its applications to parameter learning in nonlinear optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 4, pp. 733–739, 1997.

[19] B. J. Oommen and M. Agache, "Continuous and discretized pursuit learning schemes: various algorithms and their comparison," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 31, no. 3, pp. 277–287, 2001.

[20] B. J. Oommen and T. de St. Croix, "Graph partitioning using learning automata," *IEEE Transactions on computers*, vol. 45, pp. 195–208, 1996.

[21] B. J. Oommen and J. K. Lanctot, "Discretized pursuit learning automata," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, pp. 931–938, 1990.

[22] G. I. Papadimitriou and A. S. Pomportsis. Learning-automata-based TDMA protocols for broadcast communication systems with bursty traffic. *IEEE Communication Letters*, pages 107–109, 2000.

[23] A. S. Poznyak and K. Najim. *Learning Automata and Stochastic Optimization*. Springer-Verlag, Berlin, 1997.

[24] K. Rajaraman and P. S. Sastry, "Finite time analysis of the pursuit algorithm for learning automata," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, pp. 590–598, 1996.

[25] R. Thapa, L. Jiao, B. J. Oommen, and A. Yazidi, "Scheduling domestic shiftable loads in smart grids: A learning automata-based scheme," in *Smart Grid Inspired Future Technologies*. Cham: Springer International Publishing, 2017, pp. 58–68.

[26] F. Seredynski. Distributed scheduling using simple learning machines. *European Journal of Operational Research*, 107:401–413, 1998.

[27] R. Thapa, L. Jiao, B. J. Oommen, and A. Yazidi, "A learning automaton-based scheme for scheduling domestic shiftable loads in smart grids," *IEEE Access*, vol. 6, pp. 5348–5361, 2018.

[28] M. A. L. Thathachar and P. S. Sastry, "Estimator algorithms for learning automata," in *Proceedings of the Platinum Jubilee Conference on Systems and Signal Processing*, Bangalore, India, Dec. 1986, pp. 29–32.

[29] M. A. L. Thathacha and P. S. Sastry, *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic Publishers, 2004.

[30] C. Unsal, P. Kachroo, and J. S. Bay, "Multiple stochastic learning automata for vehicle path control in an automated highway system," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 29, pp. 120–128, 1999.

[31] A. V. Vasilakos, M. P. Saltouros, A. F. Atlassis, and W. Pedrycz. Optimizing QoS routing in hierarchical ATM networks using computational intelligence techniques. *IEEE Transactions on Systems, Man and Cybernetics: Part C*, 33:297–312, 2003.

[32] A. Yazidi, O.-C. Granmo, and B. J. Oommen. Service selection in stochastic environments: A learning-automaton based solution. *Applied Intelligence*, 36:617–637, 2012.

[33] X. Zhang, O.-C. Granmo, and B. J. Oommen, "Discretized Bayesian pursuit - a new scheme for reinforcement learning," in *Proceedings of IEAAIE2012*, Dalian, China, Jun. 2012, pp. 784–793.

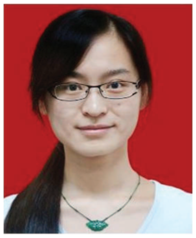
[34] X. Zhang, O.-C. Granmo, and B. J. Oommen, "On incorporating the paradigms of discretization and Bayesian estimation to create a new family of pursuit learning automata," *Applied Intelligence*, vol. 39, no. 4, pp. 782–792, 2013.

[35] X. Zhang, O.-C. Granmo, B. J. Oommen, and L. Jiao, “A formal proof of the ε-optimality of discretized pursuit algorithms,” *Applied Intelligence*, vol. 44, no. 2, pp. 282–294, 2016.

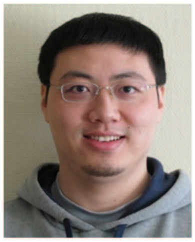
[36] X. Zhang, O.-C. Granmo, B. J. Oommen, and L. Jiao, “A formal proof of the ε-optimality of absorbing continuous pursuit algorithms using the theory of regular functions,” *Applied Intelligence*, vol. 41, pp. 974–985, 2014.

[37] X. Zhang, O.-C. Granmo, and B. J. Oommen. On incorporating the paradigms of discretization and Bayesian estimation to create a new family of pursuit learning automata. *Applied Intelligence*, 39:782–792, 2013.

[38] X. Zhang, L. Jiao, O. C. Granmo, and B. J. Oommen, “Channel selection in cognitive radio networks: A switchable Bayesian learning automata approach,” in *IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2013, pp. 2362–2367.



Xuan Zhang Xuan Zhang received her Ph. D. degree on Artificial Intelligence from the University of Agder in 2015. She obtained her Master’s degree on Signal and Information Processing from Shandong University, China, in 2008 and Bachelor’s degree on Electronics and Information Engineering from Hunan University, China, in 2005. She is now working for Confirmit as a Data Analyst, at the same time, she is a researcher at the Centre of Artificial Intelligence Research (CAIR) in the University of Agder. Her research interests include: Machine Learning, Mathematical Analysis on Learning Algorithms, Language Processing, Text Analytics, and Pattern Recognition and Classification.



Lei Jiao received his B.E. and M.E. degrees in Telecommunication Engineering, and Communication and Information System from Hunan University and Shandong University, China respectively in 2005 and 2008. He received his Ph.D. degree in Information and Communication Technology from University of Agder (UiA), Norway in 2012. He is currently an Associate Professor with the Department of Information and Communication Technology at University of Agder. His research interests include resource allocation in wireless communications, smart grid, and reinforcement learning.

John Oommen was born in Coonoor, India on September 9, 1953. He obtained his B.Tech. degree from the Indian Institute of Technology, Madras, India in 1975. He obtained his M.E. from the Indian Institute of Science in Bangalore, India in 1977. He then went on for his M.S. and Ph. D. which he obtained from Purdue University, in West Lafayettte, Indiana in 1979 and 1982 respectively. He joined the School of Computer Science at Carleton University in Ottawa, Canada, in the 1981-82 academic year. He is still at Carleton and holds the rank of a Full Professor. Since July 2006, he has been awarded the honorary rank of Chancellor’s Professor, which is a lifetime award from Carleton University. His research interests include Automata Learning, Adaptive Data Structures, Statistical and Syntactic Pattern Recognition, Stochastic Algorithms and Partitioning Algorithms. He is the author of more than 465 refereed journal and conference publications, and is a *Life Fellow* of the IEEE and a *Fellow* of the IAPR. Dr. Oommen has also served on the Editorial Board of the *IEEE Transactions on Systems, Man and Cybernetics*, and *Pattern Recognition*.

Ole-Christoffer Granmo is director and founder of the Centre for Artificial Intelligence Research (CAIR) at the University of Agder, Norway. He obtained his master’s degree in 1999 and the PhD degree in 2004, both from the University of Oslo, Norway. Granmo develops theory and algorithms for systems that explore, experiment and learn in complex real-world environments. His research interests include artificial intelligence, machine learning, learning automata, bandit algorithms, deep reinforcement learning, Bayesian reasoning, and computational linguistics. Within these areas of research, Dr. Granmo has written more than 125 refereed journal and conference publications. He is currently board member of the Norwegian Artificial Intelligence Consortium (NORA). Apart from his academic endeavors, Granmo is also co-founder of the company Anzyz Technologies AS.