Appl Intell (2014) 41:974–985
DOI 10.1007/s10489-014-0541-1

A formal proof of the ε-optimality of absorbing continuous pursuit algorithms using the theory of regular functions

Xuan Zhang · Ole-Christoffer Granmo ·
B. John Oommen · Lei Jiao

Published online: 31 May 2014

Abstract The most difficult part in the design and analysis of Learning Automata (LA) consists of the formal proofs of their convergence accuracies. The mathematical techniques used for the different families (Fixed Structure, Variable Structure, Discretized etc.) are quite distinct. Among the families of LA, Estimator Algorithms (EAs) are certainly the fastest, and within this family, the set of *Pursuit* algorithms have been considered to be the pioneering schemes. Informally, if the environment is stationary, their ε-optimality is defined as their ability to converge to the optimal action with an arbitrarily large probability, if the learning parameter is sufficiently small/large. The existing proofs of all the reported EAs follow the same fundamental principles, and to clarify this, in the interest of simplicity, we shall concentrate on the family of *Pursuit* algorithms. Recently, it has been reported Ryan and Omkar (J Appl Probab 49(3):795–805, 2012) that the previous proofs for ε-optimality of *all* the reported EAs have a common flaw.The flaw lies in the condition which apparently supports the so-called "monotonicity" property of the probability of selecting the optimal action, which states that after some time instant t_0, the reward probability estimates will be ordered correctly *forever*. The authors of the various proofs have rather offered a proof for the fact that the reward probability estimates are ordered correctly *at a single point of time* after t_0, which, in turn, does not guarantee the ordering *forever*, rendering the previous proofs incorrect. While in Ryan and Omkar (J Appl Probab 49(3):795–805, 2012), a rectified proof was presented to prove the ε-optimality of the Continuous Pursuit Algorithm (CPA), which was the pioneering EA, in this paper, a new proof is provided for the Absorbing CPA (ACPA), i.e., an algorithm which follows the CPA paradigm but which artificially has absorbing states whenever any action probability is arbitrarily close to unity. Unlike the previous flawed proofs, instead of examining the monotonicity property of the action probabilities, it rather examines their *submartingale* property, and then, unlike the traditional approach, invokes the theory of *Regular* functions to prove that the probability of converging to the optimal action can be made arbitrarily close to unity. We believe that the proof is both unique and pioneering, and adds insights into the convergence of different EAs. It can also form the basis for formally demonstrating the ε-optimality of other Estimator algorithms which are artificially rendered absorbing.

Keywords Pursuit algorithms · CPA · Absorbing CPA · ε-optimality

1 Introduction

Learning automata (LA) have been studied as a typical model of reinforcement learning for decades. A LA is an

X. Zhang (✉) · O.-C. Granmo · B. J. Oommen · L. Jiao
Department of ICT, University of Agder, Grimstad, Norway
e-mail: xuan.z.jiao@gmail.com

O.-C. Granmo
e-mail: ole.granmo@uia.no

L. Jiao
e-mail: lei.jiao@uia.no

B. J. Oommen
School of Computer Science, Carleton University,
Ottawa, K1S 5B6, Canada
e-mail: oommen@scs.carleton.ca

adaptive decision-making unit that learns the optimal action from among a set of actions offered by the Environment it operates in. At each iteration, the LA selects one action, which triggers either a reward or a penalty as a response from the Environment. Based on the response and the knowledge acquired in the past iterations, the LA adjusts its action selection strategy in order to make a "wiser" decision in the next iteration. In such a way, the LA, even though it lacks a complete knowledge about the Environment, is able to learn through repeated interactions with the Environment, and adapts itself to the optimal decision.

LA have found applications in a variety of fields, including game playing [3, 4], parameter optimization [5], solving knapsack-like problems and utilizing the solution in web polling and sampling [6], vehicle path control [7], assigning capacities in prioritized networks [8], and stochastically optimally allocating limited resources [6, 9–11]. LA have also been used in natural language processing, string taxonomy [12], graph patitioning [13], map learning [14], service selection in stochastic environments [15], numerical optimization [16], web crawling [17], microassembly path planning [18], multiagent learning [19], and in batch sequencing and sizing in just-in-time manufacturing systems [20].

Initial LA were designed to be Fixed Structure Stochastic Automata (FSSA), whose state update and decision functions are time invariant. The most notable examples of this type include the Tsetlin, Krylov and Krinsky automata [21]. Later, Variable Structure Stochastic Automata (VSSA) were developed, which are characterized by functions that update the probability of selecting the various actions. Representatives 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 [21]. As one observes, the L_{R-I} and $L_{R-\varepsilon P}$ schemes assign more importance to reward responses than to penalties; they are also ε-optimal in all stationary environments. This is also the case with FSSA, where, for example, the Krinsky automaton, which treats rewards significantly "more seriously" than penalties, is ε-optimal in all stationary environments, while the Tsetlin automaton, which treats rewards and penalties equally, is only ε-optimal when the largest reward probability is greater than 0.5 [21].

Among the families of LA, Estimator Algorithms (EAs) work with a noticeably different paradigm, and are certainly the fastest and most accurate. Within this family, the set of *Pursuit* Algorithms (PAs) were the pioneering schemes, whose design and analysis were initiated by Thathachar and Sastry [22]. EAs augment an action probability updating scheme with the use of estimates of the reward probabilities of the respective actions. The first Pursuit Algorithm (PA) was designed to operate by updating the action probabilities based on the L_{R-I} paradigm. By the same token, being an EA in its own right, the PA maintains running Maximum Likelihood (ML) reward probability estimates, which further determines the current "Best" action for the present iteration. The PA then pursues the current best action 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.

The most difficult part in the design and analysis of LA consists of the formal proofs of their convergence accuracies. The mathematical techniques used for the various families (FSSA, VSSA, Discretized etc.) are quite distinct. The proof methodology for the family of FSSA is the simplest: it quite simply involves formulating the Markov chain for the LA, computing its equilibrium (or steady state) probabilities, and then computing the asymptotic action selection probabilities. The proofs of convergence for VSSA are more complex and 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. However, understandably, the most difficult proofs involve the family of EAs. This is because the convergence involves two intertwined phenomena, namely the convergence of the reward estimates *and* the convergence of the action probabilities themselves. Ironically, the combination of these vectors 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., if the sub-optimal actions are not sampled "enough number of times"), the convergence accuracy can be diminished. Hence the dilemma!

The ε-optimality of the EAs, more specifically, the family of Pursuit Algorithms including the Continuous Pursuit Algorithm (CPA) and the Discretized Pursuit Algorithm (DPA), have been studied and presented in [23–27]. The basic result stated in these papers is that by utilizing a sufficiently small/large value for the learning parameter, both the CPA and the DPA will converge to the optimal action with an arbitrarily large probability. However, these proofs have a common flaw, which involves a very fine argument. In fact, the proofs reported in these papers "deduced" the ε-optimality based on the conclusion that after a sufficiently large time instant, t_0, the probability of selecting the optimal action is monotonically increasing, which, in turn, is based on the condition that the reward probability estimates are ordered properly *forever* after t_0. This ordering is, indeed, true by the law of large numbers only if all the actions are chosen infinitely often, which renders the time instant, t_0, to be infinite also. If such an "infinite" selection does not occur, the ordering cannot be guaranteed for *all* time instants after t_0. In other words, the authors of these papers misinterpreted the concept of ordering "forever" with the ordering "most of the time" after t_0.

As a consequence of this misinterpretation, the condition supporting the monotonicity property is false, which further leads to an incorrect proof for both the CPA and the DPA being ε-optimal. Even though this has been the accepted argument for almost three decades (even by the third author of this present paper who was the principal author of many of the above-mentioned papers), we credit the authors of [2] for discovering this flaw.[1] Further, in [2], a rectified proof was provided to prove the ε-optimality of the CPA. The rectified proof is based on the monotonicity property of the probability of selecting the optimal action, and further requires an external condition that the learning parameter, λ, is decreasing with time. We respectfully grant credit to these authors for this proof which, to the best of our knowledge, seems to be the only way to prove the ε-optimality of the CPA.

One of the main messages in the paper [2] is that they argue that probability calculations using a fixed t are not enough to show ε-optimality. The reason for this is that so little is known about the dynamics of the process $\{P(t)|t \geq 0\}$,[2] due to the complex dependencies, namely, because $\{P(t)|t \geq 0\}$ depends on $\{\hat{D}(t)|t \geq 0\}$ which, in turn, dictates how $\{P(t')|t' \geq t\}$ is determined. The arguments that the authors of [2] invoke show that by utilizing changing values of λ to control its behavior, they can ensure that the *companion* process $\{\hat{d}_j(t)|\forall j, t \geq 0\}$ is *eventually* properly ordered, i.e., *monotonically*, forever. Indeed, their arguments demonstrate that any single-t probability calculations cannot capture the "forever" behavior of the *monotonicity* of these processes. From this perspective, we believe that it is not trivial to extend their arguments even for the submartinagale property if the value of λ is kept fixed.[3] However, the arguments that we present are distinct from the previously-used flawed arguments, because we have not required that the process $\{\hat{d}_j(t)|t \geq 0\}$ satisfies the *monotonicity* property. Rather, by constraining the process to jump to an absorbing barrier in a single step when any $p_j(t) \geq T$, where T is a user-defined threshold close to unity, we have attempted to demonstrate ε-optimality because of a *weaker* property, i.e., the submartinagale property of $p_m(t)$, where t is greater than a finite time index, t_0. The latter absorbing version of the CPA will be explained presently.

With this as a background, we now move to the primary intent of this work, i.e., to correct the above-mentioned flaw. However, we shall introduce a new method which can, hopefully, be used to prove the ε-optimality of all EAs which are specifically enhanced with artificially-enforced absorbing states [28], in particular, the ACPA which is the CPA with artificially-enforced absorbing states. Though the method used in [2] is also applicable for absorbing EAs, we will show that while the monotonicity property is sufficient for convergence, it is not really *necessary* for proving that the ACPA is ε-optimal. Rather, we will present a completely new proof methodology which is based on the convergence theory of submartingales and the theory of Regular functions [21]. This current proof is, thus, distinct in principle and an argument from the proof reported in [2]. Further, we believe that our proof adds insights into the convergence of different EAs, and that it can be easily extended to formally demonstrate the ε-optimality of all the known EAs which are artificially augmented with absorbing states.

2 Overview of the ACPA

The ACPA is the CPA with absorbing states that are created artificially as explained below. Just as in the case of the CPA, the ACPA follows a "pursuit" paradigm of learning, which consists of three steps. Firstly, at time t, it maintains an action probability vector $P = [p_1(t), p_2(t), ..., p_r(t)]$ to determine the issue of which action is to be selected, where $\sum_{j=1...r} p_j(t) = 1$, and where r is the number of actions. Secondly, it maintains running ML reward probability estimates to determine which action can be reckoned to be the "best" in the current iteration. Thirdly, based on the response of the Environment and the knowledge of the current best action, the ACPA increases the probability of selecting the current best action as per the continuous L_{R-I} rules. The only difference between the ACPA and the CPA is that if any one of the action probabilities, $p_j(t+1)$, surpasses the terminating Threshold, T, which is a user-defined quantity set to be very close to *unity*, $p_j(t+1)$ will jump directly to *unity* and the learning process is terminated. At this juncture, we say that the LA has been "absorbed" into one of the absorbing barriers, where the r unit vectors are the absorbing states.

When a LA is operating in a stationary environment, i.e., where the reward probability for each action does not change with time, the difference between the CPA and the ACPA is trivial. This is because, in practice, the CPA does not need to run for an *infinite* number of iterations. Rather, the learning is terminated when one of the action probabilities is greater than or equal to a value that is close to unity, which is equivalent to the concept of the threshold

[1] While a detailed explanation of this is found in [2], in the interest of completeness, a brief explanation of this issue is also included in this paper, in Section 3.

[2] Please refer to Algorithm ACPA for the notations.

[3] We are grateful to an anonymous Referee of a previous version of this paper for shedding light on this fine point.

introduced in the ACPA. However, on the other hand, the difference between ACPA and the CPA is of fundamental importance as it provides a very convenient condition using which one is able to analyze the convergence of the pursuit algorithms, and in that sense, the ACPA is more accurate when it concerns defining the learning process which terminates within a finite number of times.

However, when it concerns dynamic environments, i.e., where the reward probabilities change from time to time, the ACPA, due to its absorbing property, is no longer applicable. This is because the CPA, without absorbing barriers, is able (though limitedly) to draw back from a previously-learned optimal action and to adjust itself to a new optimal action. However, the proof of the CPA's convergence in *dynamic* environments remains unsolved.

We present below the ACPA's notations and description, after which we visit its proof of convergence.

3 Previous proofs for CPA's ε-optimality

Since the ACPA has all the fundamental properties of the CPA except near the absorbing boundary states, in this section, all the descriptions and statements are made based on the CPA, but are also applicable to the ACPA.

The formal assertions of the ε-optimality of the CPA [27] are stated in Theorem 1, where 't' is measured in terms of the number of iterations.

Theorem 1 *Given any $\varepsilon > 0$ and $\delta > 0$, there exist a $\lambda^\star > 0$ and a $t_0 < \infty$ such that for all time $t \geq t_0$ and for any positive learning parameter $\lambda < \lambda^\star$,*

$$Pr\{p_m(t) > 1 - \varepsilon\} > 1 - \delta.$$

The earlier reported proofs for the ε-optimality of the CPA follow the same strategy, which consists of four steps. Firstly, given a sufficiently small value for the learning parameter λ, all actions will be selected enough number of times before a finite time instant, t_0. Secondly, for all $t > t_0$, $\hat{d}_m$ will remain to be the maximum element of the reward probability estimates vector, $\hat{D}$. Thirdly, suppose $\hat{d}_m$ has been ranked as the largest element in $\hat{D}$ since t_0, the action probability sequence of $\{p_m(t)\}$, with $t > t_0$, will be monotonically increasing, whence one concludes that $p_m(t)$ converges to 1 with probability 1. Finally, given that the probability of $\hat{d}_m$ being the largest element in $\hat{D}$ is arbitrarily close to unity, and that $p_m(t) \to 1$ w.p. 1, ε-optimality is proven based on the axiom of total probability.

The formal assertions of these steps are catalogued below.

1. The first step of the proof can be described mathematically by Theorem 2 for the CPA.

Algorithm ACPA

Parameters:

α_j: The j^{th} action that can be selected by the LA, and is an element from the set $\{\alpha_1, \ldots \alpha_r\}$.

p_j: The j^{th} element of the action probability vector P.

λ: The learning parameter, where $0 < \lambda < 1$.

u_j: The number of times α_j has been rewarded when it has been selected.

v_j: The number of times α_j has been selected.

$\hat{d}_j$: The j^{th} element of the reward probability estimates vector $\hat{D}$, $\hat{d}_j = \frac{u_j}{v_j}$.

m: The index of the optimal action.

h: The index of the greatest element of $\hat{D}$.

R: The response from the Environment, where $R = 0$ corresponds to a Reward, and $R = 1$ to a Penalty.

T: A Threshold, where $T \geq 1 - \varepsilon$.

Initialization:

1. $p_j(0) = 1/r$, where r is the number of actions.
2. Initialize $\hat{d}_j(0) = \frac{u_j}{v_j}$ by selecting each action a small number of times.
3. t:=1.

Method:

```
Loop
1. Select an action, $\alpha(t)$, by randomly sampling as per the
   action probability vector $P(t)$. Suppose $\alpha(t) = \alpha_i$.
2. Update $\hat{d}_i(t)$ based on the response from the Environment:
       $u_i(t) = u_i(t-1) + (1 - R(t))$
       $v_i(t) = v_i(t-1) + 1$
       $\hat{d}_i(t) = \frac{u_i(t)}{v_i(t)}$.
3. If $\hat{d}_h(t)$ is the largest element of $\hat{D}(t)$, update $P(t)$ as:
   If $R(t) = 0$ Then
       $p_j(t+1) = (1-\lambda)p_j(t),\ j \neq h$
       $p_h(t+1) = 1 - \sum_{j \neq h} p_j(t+1)$.
   Else
       $P(t+1) = P(t)$
   EndIf
4. If any $p_j(t+1) \geq T$, make $p_j(t+1)$ jump to 1 and
   break the loop:
   If $p_j(t+1) \geq T, \forall j \in (1, 2, ..., r)$
       $p_j(t+1) = 1$
       $Break$
   EndIf
   $t = t + 1$
```

Theorem 2 *For any given constants $\hat{\delta} > 0$ and $M < \infty$, there exist a $\lambda^\star > 0$ and $t_0 < \infty$ such that under the CPA algorithm, for all positive $\lambda < \lambda^\star$,*

$Pr\{$All actions are selected at least M times each before time $t_0\} > 1-\hat{\delta}$, for all $t > t_0$.

The detailed proof for this result can be found in [26].

2. The sequence of probabilities, $\{p_m(t)_{(t>t_0)}\}$, is stated to be *monotonically* increasing. The previous proofs attempted to do this by showing that:

$$\begin{aligned} |p_m(t)| &\leq 1, \text{ and} \\ \Delta p_m(t) &= E[p_m(t+1) - p_m(t)|\bar{A}(t_0)] \\ &= d_m\lambda(1-p_m(t)) \geq 0, \ t > t_0, \end{aligned} \tag{1}$$

where $\bar{A}(t_0)$ is the condition that after time t_0, for any $j \in (1,2,...,r)$, $\hat{d}_j$ remains within a small enough neighborhood of d_j so that $\hat{d}_m$ remains the greatest element in $\hat{D}$. If this step of the "proof" was flawless[4], $p_m(t)$ can be shown to converge to 1 w.p. 1.

3. Since $p_m(t) \to 1$ w.p. 1, if it can, indeed, be proven that $Pr\{\bar{A}(t_0)\} > 1-\delta$, by the axiom of total probability, one can then see that:

$$Pr\{p_m(t) > 1-\varepsilon\} \geq Pr\{p_m(t) \to 1\}Pr\{\bar{A}(t_0)\} > 1-\delta,$$

and ε-optimality is proven.

According to the sketch of the proof above, the key is to prove $Pr\{\bar{A}(t_0)\} > 1-\delta$, i.e.,

$$Pr\{\bar{A}(t_0)\} = Pr\{\bigcap_{t>t_0}\{\hat{d}_j(t)_{\forall j} \text{ is within a } \frac{w}{2} \text{ neighborhood of } d_j \text{ at time } t\}\} > 1-\delta. \tag{2}$$

In (2), w is defined as the difference between the two *highest* reward probabilities.

In the proofs reported in the literature, (2) is considered to be true according to the weak law of large numbers, i.e., if each α_j has been selected enough number of times, then for $\forall j$,

$$Pr\{\hat{d}_j(t) \text{ is within a } \frac{w}{2} \text{ neighborhood of } d_j \text{ at time } t\} > 1-\bar{\delta}', \text{ with } \bar{\delta}' = 1-\sqrt[r]{1-\delta}, \tag{3}$$

so that[5]

$$\prod_{j=1,2,...,r} Pr\{\hat{d}_j(t) \text{ is within a } \frac{w}{2} \text{ neighborhood of } d_j \text{ at time } t\} > 1-\delta. \tag{4}$$

However, there is a flaw in the above argument. In fact, if we define

$$A(t) = \{\hat{d}_j(t)_{\forall j} \text{ is within a } \frac{w}{2} \text{ neighborhood of } \mathrm{d_j} \text{ at time } t\},$$

then the result that can be deduced from the weak law of large numbers when $t > t_0$ is that

$$Pr\{A(t)\} = \prod_{j=1,2,...,r} Pr\{\hat{d}_j(t) \text{ is within a } \frac{w}{2} \text{ neighborhood of } \mathrm{d_j} \text{ at time t}\} > 1-\delta.$$

But, indeed, the condition which (1) is based on is:

$$\bar{A}(t_0) = \bigcap_{t>t_0} A(t),$$

which means that for every single time instant in the future, i.e., $t > t_0$, $\hat{d}_j(t)_{(\forall j)}$ needs to be within the $\frac{w}{2}$ neighborhood of d_j. The flaw in the previous proofs reported in the literature is that they made a mistake by reckoning that $A(t)$ is equivalent to $\bar{A}(t_0)$. This renders the existing proofs for the CPA being ε-optimal, to be incorrect.

The flaw is documented in [2], which further provided a way of correcting the flaw, i.e., by proving $Pr\{\bar{A}(t_0)\} > 1-\delta$ instead of proving $Pr\{A(t)\} > 1-\delta$. Although their proof requires a sequence of *decreasing* values for the learning parameter λ, to the best of our knowledge, it currently stands as the only correct way to prove the ε-optimality of the CPA. We applaud the authors of [2] for discovering this flaw, and for submitting an accurate proof for the CPA for the scenario when the λ's are changing with time.

However, the proof methodology that we have used here for the ACPA is quite distinct (and uses completely different techniques) than the proof reported in [2]. The reasons why we have sought an alternate proof are the following:

The monotonicity property which all the previous flawed proofs and the proof in [2] were based on, is indeed, a very strong condition. The condition requires that $\hat{d}_m(t)$ is ranked as the largest element in $\hat{D}(t)$ *at every single point of time* for all $t > t_0$, which, in turn, requires that for the CPA to achieve its ε-optimality, one must rely on an additional external assumption that the learning parameter, λ, is gradually decreasing during the learning process. Though there is, currently, no way to circumvent this external constraint so as to prove the CPA's ε-optimality, the fine difference introduced here in creating the ACPA, i.e., enhancing the CPA by incorporating into it artificially-enforced absorbing states, makes it possible for us to prove the ACPA's ε-optimality without including the constraint of decreasing the learning parameter over time.

In the next section, we shall correct the above-mentioned flaw that exists in the previous proofs of EAs. As mentioned, our new proof strategy for the ε-optimality of the ACPA

[4] The error in the proofs lies precisely at this juncture, as we shall show presently. One can also refer to [2] for the description of the error.

[5] In the interest of simplicity, at this juncture we have assumed that $\hat{d}_j$ are independent of each other. We believe that this assumption can be easily relaxed by considering only the individual d_j's as in (3), and not all of them together, as in (4).

does not require that the learning parameter λ is gradually decreasing.

The new proof is based on the convergence theory of submartingales, and on the theory of Regular functions.

4 The new proof for the ACPA's ε-optimality

4.1 The moderation property of ACPA

The property of moderation can be described precisely by Theorem 2, which have been proven in [26]. This implies that under the ACPA, by utilizing a sufficiently small value for the learning parameter, λ, each action will be selected an arbitrarily large number of times.

4.2 The key condition $\bar{B}(t_0)$ for $\{p_m(t)_{t>t_0}\}$ being a submartingale

In our proof strategy, instead of examining the condition for $\{p_m(t)_{t>t_0}\}$ being *monotonically increasing*, we will investigate the condition for $\{p_m(t)_{t>t_0}\}$ being a *submartingale*. By doing this, the previous strong condition required by the authors of [2], i.e., of $\bar{A}(t_0)$, which asserts that $\{p_m(t)_{t>t_0}\}$ possesses the property of *monotonicity*, will not be necessary any longer. Instead, we base our arguments on the weaker *submartingale* phenomena, $\bar{B}(t_0)$, defined as follows:

$$\begin{aligned} q_j(t) &= Pr\{|\hat{d}_j(t) - d_j| < \frac{w}{2}\}, \\ q(t) &= Pr\{|\hat{d}_j(t) - d_j| < \frac{w}{2}, \forall j \in (1, 2, ..., r)\} \\ &= \prod_{j=1,2,...,r} q_j(t), \qquad (5) \\ B(t) &= \{q(t) > 1 - \bar{\delta}\}, \bar{\delta} \in (0, 1), \\ \bar{B}(t_0) &= \{\bigcap_{t>t_0} \{q(t) > 1 - \bar{\delta}\}\}. \qquad (6) \end{aligned}$$

Note that $\bar{A}(t_0)$ is stronger than $\bar{B}(t_0)$ in the sense that the former requires that when $t > t_0$, $\hat{d}_j(t)$ is absolutely within a $\frac{w}{2}$ neighborhood of d_j, while the latter requires the $\hat{d}_j(t)$ to be within a $\frac{w}{2}$ neighborhood of d_j, with an arbitrarily large probability.

Our goal in this step is to prove the following result, formulated in Theorem 3.

Theorem 3 *Given a $\bar{\delta} \in (0, 1)$, there exists a time instant $t_0 < \infty$, such that the condition $\bar{B}(t_0)$ holds. In other words, for this given $\bar{\delta}$, there exists a $t_0 < \infty$, such that $\forall t > t_0$:*

$$q(t) > 1 - \bar{\delta}.$$

Proof First of all, we set $\bar{\delta}' = 1 - \sqrt[r]{1 - \bar{\delta}}$. We observe that $\forall t > t_0$, if

$$\text{for } \forall j, \quad q_j(t) > 1 - \bar{\delta}',$$

then

$$q(t) = \prod_{j=1,2,...,r} q_j(t) > \prod_{j=1,2,...,r} (1 - \bar{\delta}') = 1 - \bar{\delta}.$$

We thus need to prove that for $\forall t > t_0$ and $\forall j$,

$$q_j(t) = Pr\{|\hat{d}_j(t) - d_j| < \frac{w}{2}\} > 1 - \bar{\delta}'.$$

If we define $n_j(t)$ as the number of times α_j has been selected up to the time instant t, then by applying the Hoeffding's inequality [29], we have:

$$Pr\{|\hat{d}_j(t) - d_j| \geq \frac{w}{2}|n_j(t) = k\} \leq 2e^{-\frac{2k^2(\frac{w}{2})^2}{k}} = 2e^{-\frac{kw^2}{2}},$$

and hence

$$q_j(t) = Pr\{|\hat{d}_j(t) - d_j| < \frac{w}{2}|n_j(t) = k\} > 1 - 2e^{-\frac{kw^2}{2}}.$$

Then, we only need to set

$$\bar{\delta}' \geq 2e^{-\frac{kw^2}{2}}, \tag{7}$$

to certainly have

$$q_j(t) = Pr\{|\hat{d}_j(t) - d_j| < \frac{w}{2}|n_j(t) = k\} > 1 - 2e^{-\frac{kw^2}{2}} \geq 1 - \bar{\delta}'.$$

Besides, from (7), we also have

$$k \geq \frac{-2\ln\frac{\bar{\delta}'}{2}}{w^2}, \tag{8}$$

which means that for a given specific value of $\bar{\delta}'$, α_j needs to be selected for at least $\frac{-2\ln\frac{\bar{\delta}'}{2}}{w^2}$ times to guarantee that $q_j(t) > 1 - \bar{\delta}'$.

As the above arguments apply to $\forall j \in (1, 2, ..., r)$, we can draw the following conclusion: for the given $\bar{\delta} \in (0, 1)$ and $\bar{\delta}' = 1 - \sqrt[r]{1 - \bar{\delta}}$, we can define t_0 as a time instant such that within t_0, each action has been selected for more than $\lceil \frac{-2\ln\frac{\bar{\delta}'}{2}}{w^2} \rceil$ times. If that is the case, then $\bar{B}(t_0)$ holds, i.e., for $\forall t > t_0$, $q(t) > 1 - \bar{\delta}$, thus proving Theorem 3. □

4.3 $\{p_m(t)_{t>t_0}\}$ is a submartingale under the ACPA

We now prove the submartingale properties of $\{p_m(t)_{t>t_0}\}$ for the ACPA[6].

Theorem 4 *Under the ACPA, the quantity $\{p_m(t)_{t>t_0}\}$ is a submartingale.*

[6] Because we are maintaining the parameter λ to be a constant, we cannot currently prove that the corresponding quantity for the CPA, in and of itself, is a submartingale. It appears as if we have to currently enforce the artificial absorbing barrier.

Table 1 The various possibilities for updating p_m for the next iteration of the ACPA whenever any $p_j(t) < T$, where T is the user defined threshold close to unity

	Selected action	Responses	The greatest element in $\hat{D}$	Updating p_m
$p_m(t+1)$	$\alpha_m, (p_m)$	Reward, (d_m)	$\hat{d}_m, (q)$	$(1-\lambda)p_m(t)+\lambda$
			$\hat{d}_j, j \neq m, (1-q)$	$(1-\lambda)p_m(t)$
		Penalty, $(1-d_m)$	$\hat{d}_j, j = 1...r, (1)$	$p_m(t)$
	$\alpha_j, j \neq m, (p_j)$	Reward, (d_j)	$\hat{d}_m, (q)$	$(1-\lambda)p_m(t)+\lambda$
			$\hat{d}_j, j \neq m, (1-q)$	$(1-\lambda)p_m(t)$
		Penalty, $(1-d_j)$	$\hat{d}_j, j = 1...r, (1)$	$p_m(t)$

Proof Firstly, since $p_m(t)$ is a probability, we have $E[p_m(t)] \leq 1 < \infty$.

Secondly, we proceed to explicitly calculate $E[p_m(t)]$. Using the ACPA's updating rule, we can describe the update of $p_m(t)$ as per Table 1. Thus, we have:

$$\begin{aligned} E[p_m(t+1)|P(t)] = {} & p_m \left(d_m \left(q[(1-\lambda)p_m+\lambda]\right.\right. \\ & + \left.(1-q)[(1-\lambda)p_m]\right) + (1-d_m)p_m\big) \\ & + \sum_{j\neq m} p_j \left(d_j \left(q[(1-\lambda)p_m+\lambda]\right.\right. \\ & + \left.(1-q)[(1-\lambda)p_m]\right) + (1-d_j)p_m\big) \\ = {} & p_m d_m q\lambda - d_m\lambda p_m^2 + p_m \\ & + \lambda(q-p_m)\sum_{j\neq m} p_j d_j \\ = {} & p_m + \lambda(q-p_m)\sum_{j=1...r} p_j d_j, \end{aligned}$$

where $p_m(t)$ and $q(t)$ are respectively written as p_m and q in the interest of conciseness. Thus,

$$\begin{aligned} Diff_{p_m(t)} &= E[p_m(t+1)|P(t)] - p_m(t) \\ &= \lambda(q(t)-p_m(t))\sum_{j=1...r} p_j(t)d_j. \end{aligned}$$

Invoking the definition of a submartingale, we know that if for all $t > t_0$, we have $Diff_{p_m(t)} > 0$, i.e., $q(t) - p_m(t) > 0$, then $\{p_m(t)_{t>t_0}\}$ is a submartingale. We now invoke the terminating condition for the ACPA, in which we force the learning process to jump to the absorbing state and attain convergence if $p_j(t) > T = 1-\varepsilon$, $(j = 1, 2, ..., r)$. Therefore, if we set the quantity $(1-\bar{\delta})$ defined in Theorem 3 to be greater than the threshold T, then as per Theorem 3, there exists a time instant $t_0 < \infty$, such that for every single time instant subsequent to $t > t_0$, $q(t) > 1-\bar{\delta} > T > p_m(t)$, which, in turn, guarantees that $\{p_m(t)_{t>t_0}\}$ is a submartingale. Hence the result!

One may also notice that for the original CPA, i.e., which does not possess artificially created absorbing barriers, the quantity $Diff_{p_m(t)}$ will never be certainly greater than 0. This is because there is no such time instant that after which, $q(t) > p_m(t)$ is guaranteed. Consequently, as far as we can see, we do not believe that we can prove the CPA's ε-optimality by invoking the submartingale property alone. It appears as if the method utilized in [2] is currently the only way to achieve this proof.[7] □

4.4 $Pr\{p_m(\infty) = 1\} \rightarrow 1$ under the ACPA

We can now finally prove the ε-optimality of the ACPA.

Theorem 5 *The ACPA is ε-optimal in all random stationary Environments. More formally, let $T = 1-\varepsilon$ be a value arbitrarily close to 1, with ε being arbitrarily small. Then, given any $1-\bar{\delta} > T$, there exists a positive integer $\lambda^\star < 1$ and a time instant $t_0 < \infty$, such that for all learning parameters $\lambda < \lambda^\star$ and for all $t > t_0$, $q(t) > 1-\bar{\delta}$, the quantity*[8] *$Pr\{p_m(\infty) = 1\} \rightarrow 1$.*

Proof Since we are dealing with the ACPA, as per the submartingale convergence theory [21],

$p_m(\infty) = 0$ or 1.

If we denote e_j as the unit vector with the j^{th} element being 1, then $p_m(\infty) = 1$ is equivalent to the assertion that $P(\infty) = e_m$. If we define the convergence probability

$$\Gamma_m(P) = Pr\{P(\infty) = e_m | P(0) = P\},$$

our task is to now prove:

$$\Gamma_m(P) \rightarrow 1. \tag{9}$$

□

To prove (9), we shall use the theory of Regular functions, and the arguments used follow the lines of the arguments found in [21] for the convergence proofs of Absolutely Expedient schemes.

[7] This does not, however, mean that the result, in and of itself, is false. Indeed, it has been rigorously demonstrated with numerous experiments and by many authors that the CPA with a fixed parameter converges to an accuracy that is arbitrarily close to unity. The discouraging point is that no one has succeeded in proving *why* this is true!

[8] Since the ACPA has absorbing barriers, whenever $p_m(t) > T = 1-\varepsilon$, $p_m(t+1)$ will jump to *unity*. This is why we can assert that $Pr\{p_m(\infty) = 1\} \rightarrow 1$ instead of $Pr\{p_m(\infty) > 1-\varepsilon\} \rightarrow 1$ here.

Let $\Phi(P)$ as a function of P. We define an operator U as

$$U\Phi(P) = E[\Phi(P(t+1))|P(t) = P].$$

If we now repeatedly apply U, we get the result of the t-step invocation of U as:

$$U^t\Phi(P) = E[\Phi(P(t))|P(0) = P].$$

We refer to the function $\Phi(P)$ as being:

- Superregular: If $U\Phi(P) \leq \Phi(P)$. Then applying U repeatedly yields:

  $$\Phi(P) \geq U\Phi(P) \geq U^2\Phi(P) \geq \ldots \geq U^\infty\Phi(P). \quad (10)$$

- Subregular: If $U\Phi(P) \geq \Phi(P)$. In this case, if we apply U repeatedly, we have

  $$\Phi(P) \leq U\Phi(P) \leq U^2\Phi(P) \leq \ldots \leq U^\infty\Phi(P). \quad (11)$$

- Regular: If $U\Phi(P) = \Phi(P)$. In such a case, it follows that:

  $$\Phi(P) = U\Phi(P) = U^2\Phi(P) = \ldots = U^\infty\Phi(P). \quad (12)$$

Moreover, if $\Phi(P)$ satisfies the boundary conditions

$$\Phi(e_m) = 1 \text{ and } \Phi(e_j) = 0, \text{ (for } j \neq m), \quad (13)$$

then, as per the definition of Regular functions and the submartingale convergence theory, we have

$$\begin{aligned} U^\infty\Phi(P) &= E[\Phi(P(\infty))|P(0) = P] \\ &= \sum_{j=1}^{r} \Phi(e_m)Pr\{P(\infty) = e_j|P(0) = P\} \\ &= Pr\{P(\infty) = e_m|P(0) = P\} \\ &= \Gamma_m(P). \end{aligned} \quad (14)$$

Comparing (14) with (12), we see that $\Gamma_m(P)$ is exactly the function $\Phi(P)$ upon which if U is applied an infinite number of times, the sequence of operations will lead to a function that equals the function $\Phi(P)$ itself, because it would then be a *Regular* function. This observation readily leads us to the conclusion that $\Gamma_m(P)$ can be indirectly obtained by investigating a Regular function of P. However, as in the case of Absolutely Expedient LA, a Regular function is not easily found, although its *existence* is guaranteed. Fortunately, (10) and (11) tell us that $\Gamma_m(P)$, i.e., the Regular function of P, can be bounded from above (below) by the superregular (subregular) function of P. Furthermore, as we are most interested in the lower bound of $\Gamma_m(P)$, our goal is to find a proper *Subregular* function of P, which also satisfies the boundary conditions given by (13), which will then guarantee to bound $\Gamma_m(P)$ from below.

Consider a specific instantiation of Φ to be the function Φ_m, defined below as:

$$\Phi_m(P) = e^{-x_m p_m},$$

where x_m is a positive constant. Then, under the ACPA,

$$\begin{aligned} U(\Phi_m(P)) - \Phi_m(P) &= E[\Phi_m(P(n+1))|P(n) = P] - \Phi_m(P) \\ &= E[e^{-x_m p_m(n+1)}|P(n) = P] - e^{-x_m p_m} \\ &= \sum_{j=1\ldots r} e^{-x_m(p_m(1-\lambda)+\lambda)} p_j d_j q \\ &\quad + \sum_{j=1\ldots r} e^{-x_m(p_m(1-\lambda))} p_j d_j (1-q) \\ &\quad + \sum_{j=1\ldots r} e^{-x_m p_m} p_j(1-d_j) - e^{-x_m p_m} \\ &= \sum_{j=1\ldots r} p_j d_j e^{-x_m p_m} \left(q e^{-x_m(1-p_m)\lambda} \right. \\ &\quad \left. + (1-q)e^{x_m p_m \lambda} - 1\right). \end{aligned}$$

Our task is to determine a proper value for x_m such that $\Phi_m(P)$ is superregular, i.e.,

$$U(\Phi_m(P)) - \Phi_m(P) \leq 0.$$

This is equivalent to solving the following inequality:

$$q e^{-x_m(1-p_m)\lambda} + (1-q)e^{x_m p_m \lambda} - 1 \leq 0. \quad (15)$$

We know that when $b > 0$ and $x \to 0$,

$$b^x \doteq 1 + (\ln b)x + \frac{(\ln b)^2}{2}x^2.$$

If we set $b = e^{-x_m}$, when $\lambda \to 0$, (15) can be re-written as

$$q\left(1 + (\ln b)(1-p_m)\lambda + \frac{(\ln b)^2}{2}(1-p_m)^2\lambda^2\right)$$
$$+(1-q)\left(1 + (\ln b)p_m\lambda + \frac{\ln b^2}{2}p_m^2\lambda^2\right) - 1 \leq 0.$$

Substituting b with e^{-x_m}, we see that

$$x_m\left(x_m - \frac{2(q(1-p_m) + p_m(1-q))}{\lambda(q - 2qp_m + p_m^2)}\right) \leq 0.$$

As x_m is defined as a positive constant, we have

$$0 < x_m \leq \frac{2(q(1-p_m) + p_m(1-q))}{\lambda(q - 2qp_m + p_m^2)}. \quad (16)$$

If we denote

$$x_{m_0} = \frac{2(q(1-p_m) + p_m(1-q))}{\lambda(q - 2qp_m + p_m^2)},$$

we have $x_{m_0} > 0$, implying that when $\lambda \to 0$, $x_{m_0} \to \infty$.

We now introduce another function

$$\phi_m(P) = \frac{1 - e^{-x_m p_m}}{1 - e^{-x_m}},$$

where x_m is the same as defined in $\Phi_m(P)$. Moreover, we observe the property that if $\Phi_m(P) = e^{-x_m p_m}$ is a superregular (subregular), then $\phi_m(P) = \frac{1-e^{-x_m p_m}}{1-e^{-x_m}}$ is a subregular (superregular) [21]. Therefore, the quantity x_m, as defined in (16), which renders $\Phi_m(P)$ to be superregular, makes the $\phi_m(P)$ to be subregular.

Table 2 Bernoulli distributed reward probabilities used in the benchmark configurations

Config./Actions	1	2	3	4	5	6	7	8	9	10
1	0.70	0.50	0.30	0.20	-	-	-	-	-	-
2	0.10	0.45	0.84	0.76	-	-	-	-	-	-
3	0.70	0.50	0.30	0.20	0.40	0.50	0.40	0.30	0.50	0.20
4	0.10	0.45	0.84	0.76	0.20	0.40	0.60	0.70	0.50	0.30

Obviously, $\phi_m(P)$ meets the boundary conditions, i.e.,

$$\phi_m(P) = \frac{1-e^{-x_m p_m}}{1-e^{-x_m}} = \begin{cases} 1, & \text{when } P = e_m, \\ 0, & \text{when } P = e_j. \end{cases}$$

Therefore, according to (11),

$$\Gamma_m(P) \ge \phi_m(P) = \frac{1-e^{-x_m p_m}}{1-e^{-x_m}}. \tag{17}$$

As (17) holds for every x_m bounded by (16), we take the greatest value x_{m_0}. Moreover, as $\lambda \to 0$, $x_{m_0} \to \infty$, whence $\Gamma_m(P) \to 1$. We thus shown that $Pr\{p_m(\infty) = 1\} \to 1$, proving the claim of the theorem!

Remark 1 Note that the statement that when λ is maintained as a constant satisfying $\lambda \to 0$, the conclusion that $Pr\{p_m(\infty) = 1\} \to 1$ confirms that we do not have to continuously decrease the value λ over time as the proof in [2] requires. This result can be summarized to see that there exists a sufficiently small (*but not continuously decreasing value*) $\lambda^\star \in (0, 1)$ such that x_{m_0} will be sufficiently large so as to make $\Gamma_m(P)$ to be sufficiently close to *unity*.

Remark 2 Having proven the ACPA's ε-optimality, we can also use the above arguments to provide expressions for t_0 and the number of times each action needs to be selected. Let $\bar\delta' = 1 - \sqrt[r-1]{1-\bar\delta}$, where $\bar\delta$ is the quantity specified in the statement of Theorem 5. Then, the theorem confirms the *existence* of a time instant, t_0, where:

$$\sum_{j=1}^{r}\left\lceil \frac{-2\ln\frac{\bar\delta'}{2}}{w^2}\right\rceil \le t_0 < \infty, \tag{18}$$

and where for *this* time instant t_0, we can guarantee that each action will have been selected more than $\left\lceil \frac{-2\ln\frac{\bar\delta'}{2}}{w^2}\right\rceil$ times.

It should be noted that the quantity t_0 defined in (18) is very conservative. In other words, it has been rendered to be very large to ensure that $\{p_m(t)_{t>t_0}\}$ is a submartingale. This is consistent with the fact that the learning parameter λ has to be very small to ensure the ACPA's ε-optimality. This is because it is only when λ is sufficiently small that each action will have been "sampled" enough number of times by the time instant t_0, in which case, the estimates of the reward probabilities can be ordered correctly with an arbitrarily large probability, i.e., greater than $1-\bar\delta$, to ensure the submartingale property of $\{p_m(t)_{t>t_0}\}$. From this perspective, the rather conservative theoretical assumption for λ leads to an analogous very conservative value for t_0 that is *much larger* than the actual, practically-obtained value. We will justify this with the following experimental results that are based on the benchmark environments shown in Table 2.

Table 3 shows the comparison between the t_0 calculated from (18) and the average number of iterations needed for the LA to converge, in practice. As the CPA and ACPA are well-established algorithms, we know that numerous experiments have been conducted to confirm their validity, and so we merely use the figures from [30] to record the practical results.

As the reader will observe from Table 3, the quantity t_0, though it is not equivalent to the theoretical number of iterations required for the LA to converge, is used as a rough theoretical metric for us to compare the analytic results with the practical results. The reason why we have done this is twofold: Firstly, due to the fact that the state space of the CPA/ACPA is open and varies with time, we are not able to analyze the learning process *after* t_0, which makes it impossible for us to calculate the theoretical convergence time and to compare it in a meaningful way to the *practical* convergence time. Secondly, as mentioned earlier, t_0 is very conservative. In all the experiments we conducted, this index is much larger than the practical convergence time as one can see from Table 3. For example, in Table 3 and in Conf. 1, when δ is set to be 0.001, each action needs to

Table 3 The comparison between t_0 and the average number of iterations for the ACPA to converge in benchmark configurations

Conf.	t_0 (when $\delta = 0.001$)	Average No. of times to converge [30]
Conf. 1	$\ge 435 \times 4$	654.220
Conf. 2	$\ge 2719 \times 4$	3155.000
Conf. 3	$\ge 490 \times 10$	1876.370
Conf. 4	$\ge 3062 \times 10$	7645.190

be selected 435 times by t_0, which implies that t_0 must be greater than or equal to $1,740$ (i.e., 435×4). As opposed to this, the time to converge *in practice*, obtained by averaging over all the 750 experiments in which the LA converged to the optimal action, was only 654.220, and was much less than t_0.

We can thus confidently affirm that t_0 is, indeed, *a very conservative quantity* due to the choice of a very small learning parameter. However, we emphasize that t_0 is *finite*. This implies that within this time, the submartingale property of $\{p_m(t)_{t>t_0}\}$ can be guaranteed, whence the ε-optimality of the ACPA can be further proven.

5 Conclusions

Estimator algorithms are acclaimed to be the fastest Learning Automata (LA), and within this family, the set of *Pursuit* algorithms have been considered to be the pioneering schemes. But, as is well known, the most difficult part in the design and analysis of LA consists of the formal proofs of their convergence accuracies. The ε-optimality of Pursuit algorithms is of fundamental importance and has been studied for years. In almost all the existing papers, the proofs involved in demonstrating the ε-optimality of the Pursuit algorithms have a common flaw. The flaw was discovered by the authors of [2], whom we applaud for this. While a correct proof has been provided for the CPA in [2], it requires the scheme's parameter to be constantly decreasing. This paper aims at removing the latter stringent requirement by defining the CPA's modified version, the ACPA, whose boundary states are artificially absorbing. The paper provides a formal proof for the ACPA's ε-optimality.

Rather than examining the monotonicity property of the $\{p_m(t)_{(t>t_0)}\}$ sequence as done in the previous papers and in [2], our current proof studies the *submartingale* property of $\{p_m(t)_{(t>t_0)}\}$. Accordingly, instead of constraining the reward probability to be ordered correctly *forever* after a certain time instant, t_0, we merely require a weaker condition, i.e., one that only requires that the reward probability of the optimal action is ranked as the largest with a sufficiently large probability. Thereafter, by virtue of the submartingale property and the weaker condition, the new proof invokes the theory of Regular functions, and does not require the learning parameter to decrease gradually.

Our current proof is distinct in principle and argument from the proof reported in [2]. We believe that our proof can be easily extended to formally demonstrate the ε-optimality of other Absorbing Estimator Algorithms, without changing their respective learning parameters.

Finally, we suggest that the open unsolved problem lies in understanding the true dynamics of the process $\{P(t)_{(t\geq 0)}\}$ in terms of the complex dependencies between it and $\{\hat{D}(t)_{(t\geq 0)}\}$. We believe that a different proof methodology is not possible unless the research community obtains a deeper understanding of the inter-dependencies between both these processes.

Acknowledgments This work was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada. A preliminary version of some of the results of this paper was presented at IEAAIE-2013, the 26th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Amsterdam, Holland, in June 2013 [1]. We also record our gratitude to the Associate Editor and anonymous Referees of the original version of this paper for their painstaking reviews. The changes that they requested certainly improved the quality of this paper.

References

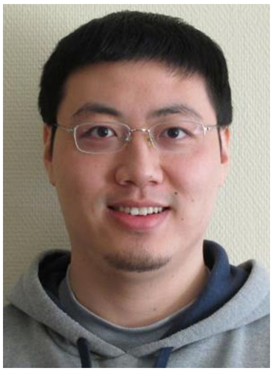
1. Zhang X, Granmo O-C, Oommen BJ, Jiao L (2013) On using the theory of regular functions to prove the ε-optimality of the continuous pursuit learning automaton. In: Proceedings of IEA-AIE 2013. Springer, Amsterdan, pp 262–271
2. Ryan M, Omkar T (2012) On ε-optimality of the pursuit learning algorithm. J Appl Probab 49(3):795–805
3. Oommen BJ, Granmo O-C, Pedersen A (2007) Using stochastic AI techniques to achieve unbounded resolution in finite player Goore Games and its applications. In: Proceedings of IEEE symposium on computational intelligence and games. Honolulu, pp 161–167
4. Granmo O-C, Glimsdal S (2013) Accelerated Bayesian learning for decentralized two-armed bandit based decision making with applications to the goore game. Appl Intell 38:479–488
5. Beigy H, Meybodi MR (2000) Adaptation of parameters of BP algorithm using learning automata. In: Proceedings of 6th Brazilian symposium on neural networks. JR, Brazil, pp 24–31
6. Granmo O-C, Oommen BJ, Myrer S-A, Olsen MG (2007) Learning automata-based solutions to the nonlinear fractional knapsack problem with applications to optimal resource allocation. IEEE Trans Syst Man Cybern B 37(1):166–175
7. Unsal C, Kachroo P, Bay JS (1999) Multiple stochastic learning automata for vehicle path control in an automated highway system. IEEE Trans Syst Man Cybern 29:120–128
8. Oommen BJ, Roberts TD (2000) Continuous learning automata solutions to the capacity assignment problem. IEEE Trans Comput 49:608–620
9. Granmo O-C, Oommen BJ (2006) On allocating limited sampling resources using a learning automata-based solution to the fractional knapsack problem. In: Proceedings of the 2006 international intelligent information processing and web mining conference, Advances in Soft Computing, vol. 35. Ustron, Poland, pp 263–272
10. Granmo O-C, Oommen BJ (2010) Optimal sampling for estimation with constrained resources using a learning automaton-based solution for the nonlinear fractional knapsack problem. Appl Intell 33(1):3–20
11. Granmo O-C (2010) Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata. IEEE Trans Comput 59(4):545–560
12. Oommen BJ, Croix TDS (Apr. 1997) String taxonomy using learning automata. IEEE Trans Syst Man Cybern 27:354–365
13. Oommen BJ, Croix TDS (1996) Graph partitioning using learning automata. IEEE Trans Comput 45:195–208

14. Dean T, Angluin D, Basye K, Engelson S, Aelbling L, Maron O (1995) Inferring finite automata with stochastic output functions and an application to map learning. Mach Learn 18:81–108
15. Yazidi A, Granmo O-C, Oommen BJ (2012) Service selection in stochastic environments: A learning-automaton based solution. Appl Intell 36:617–637
16. Vafashoar R, Meybodi MR, Momeni AAH (2012) Cla-de: a hybrid model based on cellular learning automata for numerical optimization. Appl Intell 36:735–748
17. Torkestani JA (2012) An adaptive focused web crawling algorithm based on learning automata. Appl Intell 37:586–601
18. Li J, Li Z, Chen J (2011) Microassembly path planning using reinforcement learning for improving positioning accuracy of a $1cm^3$ omni-directional mobile microrobot. Appl Intell 34:211–225
19. Erus G, Polat F (2007) A layered approach to learning coordination knowledge in multiagent environments. Appl Intell 27:249–267
20. Hong J, Prabhu VV (2004) Distributed reinforcement learning control for batch sequencing and sizing in just-in-time manufacturing systems. Appl Intell 20:71–87
21. Narendra KS, Thathachar MAL (1989) Learning automata: an introduction. Prentice Hall
22. Thathachar MAL, Sastry PS (1986) Estimator algorithms for learning automata. In: Proceedings of the platinum jubilee conference on systems and signal processing. Bangalore, India, pp 29–32
23. Oommen BJ, Lanctot JK (1990) Discretized pursuit learning automata. IEEE Trans Syst Man Cybern 20:931–938
24. Lanctot JK, Oommen BJ (1991) On discretizing estimator-based learning algorithms. IEEE Trans Syst Man Cybern B Cybern 2:1417–1422
25. Lanctot JK, Oommen BJ (1992) Discretized estimator learning automata. IEEE Trans Syst Man Cybern B Cybern 22(6):1473–1483
26. Rajaraman K, Sastry PS (1996) Finite time analysis of the pursuit algorithm for learning automata. IEEE Trans Syst Man Cybern B Cybern 26:590–598
27. Oommen BJ, Agache M (2001) Continuous and discretized pursuit learning schemes: various algorithms and their comparison," IEEE Trans Syst Man Cybern B Cybern 31(3):277–287
28. Oommen BJ (1986) Absorbing and ergodic discretized two-action learning automata. IEEE Trans Syst Man Cybern 16:282–296
29. Hoeffding W (1963) Probability inequalities for sums of bounded random variables. J Am Stat Assoc 58:13–30
30. Zhang X, Granmo O-C, Oommen BJ (2013) On incorporating the paradigms of discretization and Bayesian estimation to create a new family of pursuit learning automata. Appl Intell 39: 782–792

Xuan Zhang obtained her M.E. degree from Shandong University, China, in 2008. She obtained her B.E. degree from Hunan University, China, in 2005. She is now working as a Ph.D. research fellow in the University of Agder, Norway. Her research interests include Machine Learning, Learning Automata, Stochastic Modeling and Optimization, and Data Mining.

Prof. Ole-Christoffer Granmo is Director of the Centre for Integrated Emergency Management (CIEM) and heads the Artificial Intelligence and Its Industrial Applications Group at University of Agder, Norway. He obtained his M.Sc. in 1999 and the Ph.D. degree in 2004, both from the University of Oslo, Norway. His research interests include Intelligent Systems, Stochastic Modelling and Inference, Machine Learning, Pattern Recognition, Reinforcement Learning, Distributed Computing, Computational Linguistics, and Surveillance and Monitoring. Within these areas of research, Dr. Granmo has written more than 80 refereed journal and conference publications. He also serves on the Editorial Board of Crisis Communications, specializing within artificial intelligence support for crisis preparedness and management.

Dr. B. 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 400 refereed journal and conference publications, and is a *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*.



Lei Jiao received his BE and ME 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 now working at the Department of Information and Communication Technology, University of Agder, as a postdoc researcher. His research interests include cognitive radio networks, wireless sensor networks, and artificial intelligence.