An Analytical Model of a New Type of Learning Automata

Prateek S. Aggarwal

Boston University

prateekji@gmail.com

Chenhui Liu

Boston University

Dr.chenhui.liu@gmail.com

Lev Levitin

Boston University

levitin@bu.edu

**Abstract**

We present the novel concept of Probabilistically-Switch-Action-on-Failure learning automaton (PSAFA). It is a fixed structure stochastic automaton (FSSA), with a fan-shaped state transition diagram where each branch of the state space is a chain of states, associated with a particular action. The first states of all chains form a circle of initial states. The PSAFA can switch from a present state in any chain to the initial state of the next chain in the circle, on each failure, with some fixed non-zero probability. This action-switching probability is a function of the distance (in the number of states) of the present state from the initial state of its branch. The learning behavior of PSAFA is determined by the dependence of the action switching probability on the distance from the initial state. The probabilistic action-switching capability distinguishes PSAFA from conventional FSSA where only some states transit to states with a different action. This action-switching capability at any time is also typical for conventional variable structure stochastic automata (VSSA) but it comes with added computational complexity. VSSA are more adaptive than traditional FSSA in non-stationary environments because of this action-switching capability. We believe that the addition of this capability should also make the PSAFA more adaptive in non-stationary environments than classical FSSA while preserving the simplified computational complexity of FSSA. We further identified different learning automata within this class, that differ in their response to penalties from the environment, and we named them as ambivalent-PSAFA, optimistic-PSAFA and pessimistic-PSAFA. We found that the three automata have very different $ε$-optimality properties. The effectiveness of the proposed framework is demonstrated through the theoretical analysis of optimality of the PSAFA in stationary environments. We also believe that the model will be relevant for multiple fields like reinforcement learning, mathematical psychology, neuroscience, behavioral science, and mathematical finance.

# Introduction

Over the past few decades, the study of learning automata (LA) (Narendra & Thatachar, 2012; Rezvanian, Saghiri, Vahidipour, Esnaashari & Meybodi 2018) has taken a center-stage in the field of machine learning and computational intelligence. Early work on learning developed in the context of mathematical psychology (Bush & Mosteller, 1955; Tsetlin, 1962; Atkinson & Bower, 1965; Norman, 1972). Learning is the ability to improve performance using past experience in an unknown environment. The theory of LA provides a framework for such a learning ability.

An LA is a simple entity comprised of multiple states, with at least one of the states being described as current. At each time instant, an automaton selects one of several available actions, according to action probabilities determined by the current state or states. The environment provides a random response to the action selected; the response is simple and can be either a reward indicating success, or a penalty indicating failure (Narendra & Thatachar, 2012). Depending on the environment response, the automaton makes a transition into a new current state. Provided that the reward/penalty from the environment is only weakly related to the action selected, an LA represents a suitable strategy to maximize the probability of reward over multiple attempts at selecting the best possible action. For the rest of the paper, we refer to each attempt as a step in a sequence of attempts. Thus, an LA is an adaptive decision-making device that operates in an unknown stochastic environment and progressively improves its performance via a learning process. Such an LA can form the nucleus of a learning system with much more elaborate logic, with the architecture of the LA handling the random nature of the environment.

Since action selection scenarios are prevalent in various machine learning and real life situations, such as in training deep neural networks (Guo, Li, Qi, Guo & Xu, 2020) and clustering (Hasanzadeh-Mofrad & Rezvanian, 2018), intelligent cloud computing and resource allocations (Oommen & Roberts, 1998), adaptive recommender system and social networks (Ghavipour & Meyboudi, 2016), network and filter design (Misra, Chatterjee & Guizani, 2015), optimization of cooperative tasks (Zhang, Wang & Gao 2021) and queuing systems (Vahidipour & Esnaashari 2018), theoretical research in this field has acquired further significance in recent years (Economides & Kehagias, 2002). In finance, accurate prediction of bankruptcy is important to mitigate economic loss (Mazhari & Monsef 2012) and LA can be used for selecting components during financial portfolio optimization (Sbruzzi, Leles & Nascimento, 2018). LA can be applied to a broad range of control problems, which are characterized by nonlinearity and a high degree of uncertainty (Ghaleb & Oommen, 2019; Abeyrathna, Granmo, Zhang, Jiao & Morten Goodwin, 2020). Modeling human learning has also been pursued using LA (Oommen & Hashem, 2009). A key feature of LA which makes them applicable to a broad range of applications is their ability to combine rapid and accurate convergence with a low computational complexity and better interpretability.

A variety of frameworks have been set up for the design of such LA. When the action probabilities of each state remain time-invariant, we have a Fixed Structure Stochastic Automaton (FSSA). When the action probabilities change in time, we have a variable-structure stochastic automaton (VSSA). Tsetlin (Tsetlin, 1962) started the work on LA by considering the problem of finding an optimal action out of a set of allowable actions and attempted to address this problem using fixed structure stochastic automata. A detailed review presented in (Varshavskii 7 Vorontsova, 1964) indicates that later, interest shifted to the study of VSSA which appeared to be more adaptable. On the other hand, FSSAs are easier to implement and require less computation per time step. This motivated us, just as Economides and Kehagias (Economides & Kehagias, 2002), to return to the FSSA idea and search for FSSA designs which perform as well or even better than corresponding VSSAs.

In this paper, we introduce the Probabilistically-Switch-Action-on-Failure Automaton (PSAFA), an FSSA, and compare its behavior to that of several “classical” FSSAs. For conciseness, the PSAFA is also referred by the symbol Ж. We aim to identify key learning properties of the PSAFA that cannot be achieved using classical LA. The fan-shaped structure of the transition diagram is displayed in Figure 1.



**Figure 1**. The PSAF(*r*,*D*) automaton with *r* = 4.

Each branch of the fan consists of several states, which comprise a chain of length *D* and are “committed” to one of the actions available to the automaton. The length *D*, of branches is one parameter of the automaton called “depth”, and the number of branches, which is equal to the number of possible actions, *r*, is the other parameter of the automaton. Hence we speak, in general, of Ж(*r,D*). The PSAFA can switch from its active state in any chain to the initial state of the next chain in the circle, on each failure, with some finite probability. Because this state-switching probability in case of penalty, and hence the action-switching probability can be selected in multiple ways, the PSAFA becomes a framework for developing different FSSA.

The primary motivation for studying the PSAFA is the non-zero action-switching probability for all states, reflected in its fan shaped structure. In general, the action-switching probability is zero for most states of FSSA with deterministic action selection in each state, simulated in previous studies. Automata designed under the VSSA framework continuously maintain an action selection probability vector, and therefore can select any of the available actions at any given instant. In essence, there always exists a finite probability for a VSSA based LA to switch from any action to any of the available actions in the future. This characteristic of VSSA makes them more adaptable. By incorporating the action-switching-on-penalty feature in an FSSA, we aim to design an FSSA that is more adaptable than previously proposed FSSAs, and with its adaptability comparable to the classical VSSAs. The presence of non-zero action switching probability in every state in PSAFA increases the probability flow from one branch to the next in the LA and thus makes it more adaptable than classical FSSA.

An essential feature of the PSAFA is that we designed the state transitions in response to penalty to be always probabilistic. Analytical and simulation results by previous researchers [28] indicate that the performance of LA possessing deterministic state transitions in response to reward, and probabilistic state transitions in response to penalty can asymptotically approach optimality in any environment. Therefore, we have chosen to explore the different action-switching configurations within this reward-deterministic/penalty-probabilistic state transition framework, in this paper.

Another motivation for studying LA is the apparently stochastic, randomized behavior of biological learning systems (Tsetlin, 1962). It has been shown that stochastic LA can perform better than their deterministic counterparts (Economides & Kehagias, 2002; Oommen & Christensen 1988). However, a direct answer to the necessity of random behavior in learning systems has defied researchers for decades, primarily because of existence of deterministic counterparts of stochastic LA simulated in previous studies (Narendra & Thatachar, 2012, pp 59-101). The PSAFA proposed in this paper does not have a non-trivial deterministic counterpart. For example, if the action-switching is made deterministic in the 2-action PSAFA, the automaton degrades into a trivial two-state deterministic Tsetlin automaton, irrespective of the depth of the PSAFA. It is this characteristic of the PSAFA that motivates us to compare its behavior with classical FSSA, that, unlike the PSAFA, have both deterministic and stochastic non-trivial versions (Narendra & Thatachar, 2012, pp 59-101).

The rest of the paper is organized as follows. Section II contains a review of the fundamental concepts of stochastic LA. Section III describes the PSAFA with depth *D* and examines its optimality properties of its different action-switching configurations. In Section IV, we present our conclusions, and propose some directions for future research.

# Fundamental Concepts of Stochastic LA

In this Section, we summarize the standard mathematical description of the LA model. This involves the definition of the automaton itself, the environment with which it interacts, the objective of this interaction and the learning method. For the purpose of the current description, time is represented as a series of discrete instants separated by regular intervals, with one such instant recognized as the first instant. A more detailed description is available in [28]. **Environment** is defined by a triple {*A*, *B*, *C*}, where

* *A* ≡ {*αi*} , *i* = 1, 2,…, *r* is the set of actions (input to the environment); the action at any instant *n* is represented as *α*(*n*)
* *B* ≡ {*βj*} *j* = 0, 1 is the set of responses (output of environment), where 0 indicates a reward and 1 indicates a penalty; the actual response at any instant *n* is represented as *β*(*n*), and
* *C* ≡{*ci*} i, *i* = 1, 2,…, *r* is a penalty probability set, which is unknown to the automaton, with *ci* corresponding to action *i*, such that Pr[*β*(*n*) = 1 | *α*(*n*) = *αi*] = *ci* .

Note that the the above definition holds for a stationary environment, where the penalty probability corresponding to each action is independent of time. For a non-stationary environment, however, the penalty probabilities change with time, and a more accurate representation of the relationship between action and penalty is Pr[*β*(*n*) = 1 | *α*(*n*) = *αi*] = *ci*(*n*).

**Automaton** is defined by a quintuple {Ф, *A*, *B*, *G*(•**)**, F(•|•,•,•**)**}, where

* Ф ≡ {φ*k*} , *k* = 1, 2,…, *s* is the set of the internal states,
* *A* ≡ {*αi* } , *i* = 1, 2,…, *r* is the set of actions (output of the automaton),
* *B* ≡ {*βj*} *j* = 0, 1 is the set of responses (input to the automaton),
* *G*(•**)**:Ф → *A* is the action selection function for choosing the action corresponding to the present state. Each state is associated with one specific action, so that the automaton takes the same action each time it is in a given state
* *F*(*l*| *k*,*i*,*j*)= Pr[Ф(*n*+1) = φ*l* | Ф(*n*) = φ*k*, *α*(*n*) = *αi*, *β*(*n*) = *βj*] is the conditional state transition probability function for choosing the next state depending on the present state, the action selected, and the environment response. Note that

 *F*(*l*| *k*,*i*,*j*) = 1, *k*,*i*,*j*



**Figure 2**. Key state transitions in the PSAFA.

The above description of the class of automata considered in this paper can be contrasted with the usual characterization of VSSAs in terms of a time-varying action selection probability vector *p*(*n*)≡[*p*1(*n*) *p*2(*n*) … *pr*(*n*)]’ and a learning algorithm **Γ**: (*n*+1) ≡ **Γ** (*p*(*n*), *α*(*n*), *β*(*n*)), that updates the action probability vector, at any instant *n*+1, based on the action selection probability at instant *n*, *p*(*n*), actual action selected, *α*(*n*), and the environment response *β*(*n*). Note that the automata we present in this paper belong to the FSSA rather than the VSSA class.

For a stationary environment, it is desirable that the automaton selects the action *α*\* associated with the minimum penalty probability *c*\* ≡ {*ci*}. Automaton performance is usually evaluated by the average cost for a given *p*(*n*): *M*(*n*) ≡ *E*[*β*(*n*)|*p*(*n*)] =*cipi*(*n*). A *Pure-Chance automaton*, selects actions with equal probabilities, and the average cost is the mean of the penalty probabilities *M*0(*n*) = (1/*r*)*ci*. An LA learns about the environment over multiple action-selection and reward/penalty instances, and attempts to reduce *M*(*n*). An LA that asymptotically behaves better than a pure chance automaton and will in the limit have average cost lim*n*→∞ *E*[*M*(*n*)] < *M0*, is termed as *expedient*. Similarly, an LA is said to be *optimal* if lim*n*→∞ *E*[*M*(*n*)] = *c*\*. Previous studies (Narendra & Thatachar, 2012; Economides & Kehagias 2002) have established that different LA can be optimal provided the action penalty probabilities are constrained (*c*\*<0.5, for example). If an LA is optimal irrespective of the penalty probabilities of the actions, we refer to such an LA as universally optimal.

For non-stationary environments where limiting probabilities do not exist in general, the desirable qualities of an LA are still open to debate. For example, while optimality is desirable in *stationary* environments, a suboptimal performance may be preferable in *nonstationary* environments, where the penalty probabilities *c* vary with time. In non-stationary environments, where the penalty probabilities change in time in an ergodic process, limiting probabilities still exist, and the automaton must not only learn the characteristics of the environment, but also “forget” old characteristics in favor of new ones, in response to the time-varying situation. An optimal automaton may be too rigid to accommodate such requirements (Narendra & Thatachar, 2012, Chapter 7, pp. 227–279), because it may get locked in an action which is originally optimal but later becomes pessimal. In such cases, *ε*-optimalautomata, which satisfy lim*n*→∞*E*[*M*(*n*)] < *c*\* + *ε*, with *ε* > 0, are more capable at responding to a changing environment. Please note that *c*\* could be equal to *ci*  where *i* corresponds to different actions in time for a non-stationary environment of ergodic Markov process as the penalty probabilities fluctuate with time, and the LA is supposed to recognize the action corresponding to this least penalty probability in such a non-stationary environment.

**The PSAF Learning Automaton**

In this Section, we examine the *r*-action PSAFA with depth *D*, which we denote by **Ж**(*r*,*D*). As mentioned earlier, the action set is A ≡ {*α1*, *α2*, ..., *αr*}, the environment response set is B ≡ {0,1} (reward and penalty), and the PSAFA has a fan-shaped structure consisting of *r* branches, each consisting of *D* states, arranged as a chain. The first states of all chains form a circle of initial states. When the automaton is in one of the states of the *i*th branch, it performs action *αi* with probability one. In other words, each state is “committed” to a corresponding action. A state that is committed to action *αi* and whose position along its branch’s state-chain is *d*-1 states from the initial state, is denoted as *k* = (*i*,*d*), and is said to have a depth of *d*.The fan-shaped structure, the state transition and action selection mechanisms are illustrated in Figure 1. The structure of the PSAFA consists of three different types of states i.e. intermediate, terminal, and initial, depicted in Figure 2(a-c), respectively. The different types of state transitions possible from each state, for a PSAFA, are also shown in Figure 2. The action selection mechanism specifies the function *G* as *G*(*i*,*d*) = *αi* . For the purpose of conducting mathematical operations using probability matrices, the function *G* can also be represented as a probability matrix ***G****,* where

*G*(*i’|*(*i*,*d*)) ≡ Pr[*α*(*n*) = *i’*|Ф(*n*) = (*i*,*d*)] = δ*ii*’, ∀ *i*, *i’* = 1,2,...,*r*; *d* = 1,2,...,*D* (1)

In other words, all probabilities above where *i’* ≠ *i* are equal to zero. To evaluate the expediency and optimality of the automaton, we need to know the action probabilities *pi*(*n*) written in vector form as *p*(*n*) = [*p1*(*n*) *p2*(*n*).... *pr*(*n*)]. We also define the probability of being at state *k* = (*i*,*d*), at instant *n*: *πk*(*n*); written in vector form as π(*n*) = [π(*1,1*)(*n*) π(*1,2*)(*n*) ... π(*1****,****D*)(*n*) ...π(*r,1*)(*n*) π(*r,2*)(*n*) ... π(*r,D*)(*n*)]. We have the following relationship between action probabilities and state probabilities:

 *pi*(*n*) = π(*i’*,*d*)(*n*) . *G*(*i|*(*i’*,*d*)) = π(*i*,*d*)(*n*) (2)

Hence, both the learning behavior and optimality properties depend on the state probabilities π(*n*), which in turn depend on the state transition mechanism defined by the probabilities , as follows:

*F*(*k*’|*k*,*i*,*j*)= Pr[Ф(*n*+1) = *k*’| Ф(*n*) = *k*, *α*(*n*) = *αi*, *β*(*n*) = *βj*], (3)

where *k*’= (*i’*,*d*’). These probabilities depend on the current state, action, and response. We will present several possible choices of ***F*** for all of which the state process Ф(*n*) is an ergodic Markov chain with state transition matrix ***P***, where ***P****k*’,*k*= Pr[Ф(*n*+1) = (*i’*,*d*’)| Ф(*n*) = (*i*,*d*)] in the presence of a environment penalty probability set {*ci*}. Hence, lim*n*→∞ π(*n*) = π. Here π is the uniquelimiting (equilibrium) probability vector π = π*i*,*d*$π\_{i,d}),$ where *i=*0,1,2, … *r*, *d=*1,2*,…D.* Later in this paper, when we propose the different FSSA, we will investigate the limiting behavior of the automata by deriving formulae for the limiting probability of the *i*th action as n→∞, which is lim*n*→∞ *pi*(*n*) = *pi*, for a givent penalty probability set {*ci*}.

The generic PSAF structure is presented at Figure1 (for *r=*4). The complete set of state transitions available in response to success and failure are shown at Figure 2. We denote the conditional probability of changing the action in the case of failure, i.e. *β*=1*,* while in dth state by *θd*

 *F*((*i‘,*1)|(*i,d*), *i,*1) = *θd* *,* ∀ *i =* 0,1,… *r -*1; *i‘=* (*i+*1)(mod *r*) ; *d =* 1,2,… *D* (4)

In the state of equilibrium, the total probability to change the branch from *i* to (*i* +1)(mod *r*) at each step of the process is

ξ*i* = $\sum\_{d=1}^{D}c\_{i }θ\_{d } π\_{i,d}$*ci* θ*d* π*i,d* (5)

An important property of the equilibrium state of PSAFA is given by the following proposition.

**Proposition.** ξ*i* **=** ξfor all*i =* 1, 2, … *r -*1. (6)

**Proof.**  ξ*i* is equal to the total probability flow from branch *i* to branch (*i +*1)(mod *r*) in the limiting case as *n*→∞. At the equilibrium, the incoming flow of probabilities to every branch must be equal to the probability flow leaving the branch. Therefore,

ξ0 = $ξ\_{0 }=$ ξ1*,*  ξ1 = $ξ\_{0 }=$ ξ2 , ξ1 = $ξ\_{0 }=$ ξ2, ... ξ*r*-1 = $ξ\_{0 }=$ ξ0 (7)

Thus, all ξ*i* are equal, ξ*i* = $ξ\_{0 }=$ ξ0 = ξ ;  *i =* 0,1,… *r -*1

We now proceed to define *F*, for three different schemes of the generic PSAFA structure presented in Figure 1, each restricting the state transitions available in different ways. The response to success is deterministic and identical in all three schemes, with the response to failure being probabilistic and different from each other in all three schemes.

**Figure 3**. The canonical state-transition configuration (Ambivalent PSAFA).

**Ambivalent Ж(r,D)**

In this scheme, success and failure cause state transitions according to the following rules. The rules are also illustrated in Figure 3(a-c).

1) When in an initial state (*i*,1), if rewarded (Figure 3c, *β* = 0) go to state (*i*,2) with probability one

*F*((*i*,2)| (*i*,1),*i*,0) = 1, ∀ *i* = {0,1,2, ... *r*-1}. (8)

If punished (Figure 3c, *β* = 1), transit to the initial state of the next branch, i.e. ((*i* +1)(mod *r*),1) in the *r*-action PSAFA, with probability θ1, and stay in state (*i*,1) with probability 1- θ1

*F*((*i’*,1)| (*i*,1),*i*,1) = θ1, *F*((*i*,1)| (*i*,1),*i*,1) = 1- θ1 ∀ *i* = {0,1,2, ..., *r*-1}, *i*’= (*i*+1)(mod *r*) (9)

2) When in an intermediate state (*i*,*d*), if rewarded (Figure 3a, *β* = 0), go to state (*i*,*d*+1) with probability one

*F*((*i*,*d*+1)| (*i*,*d*),*i*,0) = 1 ∀ *i* = {0,1,2, ... *r-*1}, 1 < *d* < *D*. (10)

If punished (Figure 3a, *β* = 1), transit to the initial state of the next branch, i.e. ((*i+*1)(mod *r*),1),in the *r*-action PSAFA, with probability θ*d*, and stay in state (*i*,*d*) with probability 1- θ*d*

*F*((*i’*,1)| (*i*,*d*),*i*,1) = θ*d* , *F*((*i*,*d*)| (*i*,*d*),*i*,1) = 1- θ*d*

∀ *i* = {0,1,2, ... *r*-1}, 1 < *d* < *D*, *i*’= (*i*+1)(mod *r*) (11)

3) When in a terminal state (*i*,*D*)(Figure 3b, *β* = 0), if rewarded stay in same state with probability one

*F*((*i*,*D*)| (*i*,*D*),*i*,0) = 1, ∀ *i* = {0,1,2, ..., *r*-1} (12)

If punished (Figure 3b, *β* = 1), transit to the initial state of the next branch, i.e. ((*i* +1)(mod *r*),1), in the *r*-action PSAFA, with probability θ*D*, and stay in state (*i*, *D*) with probability 1- θ*D*

*F*((*i’*,1)| (*i*,*D*),*i*,1) = θ*D*, *F*((*i*,*D*)| (*i*,*D*),*i*,1) = 1- θ*D*, ∀ *i* = {0,1,2, ..., *r*-1}, *i*’= (*i*+1)(mod *r*) (13)

Equations (8), (10) and (12) persist for the other 2 models presented in Section IIIB and IIIC too and will not be repeated for the reward case for the two other automata.

From *F*(•|•,•,•**)** we can compute the limiting state probabilities and action probabilities and evaluate the expediency and optimatility of the automaton. Here, we only present the results of our analysis; detailed derivations are given in (Aggarwal, Liu & Levitin, 2022). The nonzero elements of ***P*** turn out to be

 P(*i*,*d*),(*i*,*d*+1) = 1-*ci*,P(*i*,*d*),(*i*,*d*) = *ci* (1-θ*d*),

P(*i*,*D*-1),(*i*,*D*) = 1-*ci*,P(*i*,*D*),(*i*,*D*) = (1-*ci*)+*ci*(1-θ*D*) = 1- *ci*θ*D*,

P(*i*,*d*),( *i*’,1) = *ci* θ*d*, ∀ *i* = {0,1,2, ... *r*-1}, *i*’= (*i* +1)(mod *r*) (14)

and all the other elements of ***P*** are zero. It is obvious from (14) that for all the diagonal elements of ***P***, *P*(*i*,*d*),(*i*,*d*)>0 . Furthermore, it is easy to check that ***P****D+r*-1>0. Intuitively, this corresponds to the fact that in *r* steps we can get from any state to the initial state of any branch including the initial state of the current branch, and in another *D*-1 steps, we can get from initial state of a branch to any other state in the branch with positive probability. In other words in *D*+*r*-1 steps, we can get from any state to any other state in the automaton with positive probability. Hence, the state process Ф(*n*) coresponding to the Ambivalent Ж(*r*,*D*) automaton is irreducible, aperiodic and, as a consequence, ergodic (Narendra & Thatachar 2012). Using (14), we obtain the following equations for the limiting state probabilities π*i*,*d* :

ξ*i*  = ξ = *ci* θ*d*π *i,d*,π*i*,1 = , π*i*,*d* = , π*i*,*D* = (15)

We derived the formulae in (15) using the assumption that the probability flow into any state at equilibrium will be equal to the probability flow out of the state. Therefore,

π*i*,*d* = *,* π*i*,*D*= (16)

Here ξ is the joint probability of three events: (1) the last active state was in the *i*th branch of the automaton, (2) the action selected is punished, (3) the automaton switches its current state to the next branch. It is indicative of the relative frequency with which action-switching is occuring between any two actions corresponding to adjacent branches in the automaton, or the flow rate of the current state from one branch to the next branch. The derivation of above equations is based on the observation that, at equilibrium, the probability ξ is the same for all branches of the automaton (see the Proposition in (6)). For an *r*-action automaton, the action probabilities for the *u*th and *v*th actions *pu* and *pv*, respectively, is given by

 *pu* = +

*pv* = + *n* (17)

Taking the ratio of the two equations in (17) eliminates ξ i.e.

Ω*D*= = (18)

Based on the motivation for the machine learning algorithm derived from the extended discrete Kalman filter algorithm in [49], the parameters in the above equations, θ*d*, are chosen as

 θ*d = ,* θ*D =* (19)

Using (19), the ratio of probabilities in Eq. 18 can be rewritten as

Ω*D*≈ [ ]  *bu*< *bv* <1 (20)

where *b* = *c*/(1-*c*), (Aggarwal, Liu & Levitin, 2022, Appendix A, (A1.9)). Thus, when *bu*< *bv* <1, Ω*D* → ∞ as *D* → ∞ and the automaton is *ε*-optimal when all the penalty probabilities are <0.5. Also, if *bu*< 1 and *bv* > 1, then

Ω*D*≈  *bu*< 1 and *bv* > 1 (21)

as shown in (Aggarwal, Liu & Levitin, 2022, Appendix A, (A1.10)). Thus, Ω*D* is proportional to and Ω*D* → ∞ as *D* → ∞ and the automaton is *ε*-optimal when atleast one of the penalty probabilities is <0.5 . However, if *b*1 and *b*2 are both greater than one, then Ω*D* → (2*cv*-1)/(2*cu*-1) as *D* → ∞ (Aggarwal, Liu & Levitin, 2022, Appendix A, (A1.11)) and the automaton is not *ε*-optimal in such an environment, when all the penalty probabilities are >0.5.

Thus, for the Ambivalent **Ж** to be *ε*-optimal, at least one of the penalty probabilities should be less than 0.5. As suggested previously by other researchers (Economides & Kehagias, 2002; Oommen & Christensen 1988), this condition can be enforced by treating 50% of randomly selected failures as successes, and thus reducing all effective penalty probabilities by 50%. However, under this approach, the Ambivalent **Ж** requires a larger set of input data to get trained, and this adversely affects its convergence rate and adaptability in non-stationary enviroments [50]. The Optimistic PSAFA presented in the following Section is universally *ε*-optimal, and does not suffer from this deficit.



**Figure 4**. Two alternative configurations of the PSAF automaton i.e. the Optimistic (Top) and Pessimistic (Bottom) PSAF automata.

**Optimistic Ж(r,D)**

In this scheme, state transitions in response to success are identical to the Ambivalent **Ж** (Equations (8), (10), (12)). However, state transitions in response to failure differ from the Ambivalent **Ж**, and are shown for the intermediate, terminal, and initial states in Figure 4. These differences in state transitions in response to failure are described in detail below.

1) When in an initial state (*i*,1), if punished (Figure 4c, *β* = 1, Optimistic automaton), transit to the initial state of the next branch of the *r* branches, i.e. ((*i* +1)(mod *r*),1) in the *r*-action PSAFA, with probability θ1, and transit to state (*i*,2) with probability 1- θ1

*F*((*i’*,1)| (*i*, 1),*i*,1) = θ1, *F*((*i*, 2)| (*i*, 1),*i*,1) = 1- θ1  *i* = {0,1,2, ..., *r*-1}, *i*’= (*i* +1)(mod *r*) (22)

2) When in an intermediate state (*i*,*d*), if punished (Figure 4a, *β* = 1, Optimistic automaton), transit to the initial state of the next branch of the *r* branches, i.e. ((*i* +1)(mod *r*),1) in the *r*-action PSAFA, with probability θ*d*, and transit to state (*i*,*d*+1) with probability 1- θ*d*

*F*((*i’*,1)| (*i*, *d*),*i*,1) = θ*d* , *F*((*i*,*d*+1)| (*i*,*d*),*i*,1) = 1- θ*d*

  *i* = {0,1,2, ..., *r*-1}, 1 < *d* < *D*, *i*’= (*i* +1)(mod *r*) (23)

3) When in a terminal state (*i*,*D*), if punished (Figure 4b, *β* = 1, Optimistic automaton), transit to the initial state of the next branch of the *r* branches, i.e. ((*i* +1)(mod *r*),1) in the *r*-action PSAFA, with probability θ*D*, and stay in state (*i*,*D*) with probability 1- θ*D*.

*F*((*i’*,1)| (*i*, *D*),*i*,1) = θ*D* , *F*((*i*, *D*)| (*i*, *D*),*i*,1) = 1- θ*D*,  *i* = {0,1,2, ..., *r*-1}, *i*’= (*i* +1)(mod *r*) (24)

Using (8), (10), (12), (22), (23), (24) we can compute the limiting state probabilities and action probabilities and evaluate the expediency and optimality of the automaton. Here, we only present the results of our analysis; detailed derivations are given in (Aggarwal, Liu & Levitin, 2022, Appendix B). The nonzero elements of ***P*** turn out to be

*P*(*i*,*d*),(*i*,*d*+1) = (1-*ci*)+*ci*(1-θ*d*) = 1- *ci* θ*d*, *P*(*i*,*D*),(*i*,*D*) = (1-*ci*)+*ci*(1-θ*D*) = 1- *ci* θ*D*,

*P*(*i*,*d*),( *i*’,1) = *ci* θ*d*,  *i* = {0,1,2, ... *r*-1}, *i*’= (*i* +1)(mod *r*) (25)

and all the other elements of ***P*** are zero. It is obvious from (25) that for all the diagonal elements of ***P***, *P*(*i*,*d*),(*i*,*d*)>0 . Furthermore, it is easy to check that ***P****D+r*-1>0. Therefore, using the same logic as in section 3A it can be inferred that the state process Ф(*n*) coresponding to the Optimistic Ж(*r*,*D*) automaton is irreducible, aperiodic and, as a consequence, ergodic [1]. From ***P*** in (25) we can compute the limiting state probabilities π*i*,*d*, which turn out to be given by the following equations:

ξ = c*i* θ*d* π *i,d*,π*i*,1 = ξ *,*π*i*,*d* = , π*i*,*D* = (26)

Therefore,

π*i*,*d*= ,π*i*,*D*=  (27)

For a *r*-action automaton, the action probabilities for any two actions, say action *u* and *v* without loss of generality, i.e. *pu* and *pv*, respectively, will be given by

*pu* = + ξ +

*pv* = + ξ +  (28)

Taking the ratio of the two equations in (28) eliminates ξ i.e.

 Ω*D* = = (29)

*θk* is selected as in (19). Thus, as shown in (Aggarwal, Liu & Levitin, 2022, Appendix B, (A2.6)) gives

 Ω*D*≈  *D*>>1 (30)

Thus, following analysis similar to that for the Ambivalent **Ж**, we note that the ratio of probabilites is proportional to *D* for large values of D. Therefore, the Optimistic **Ж** is *ε*-optimal for all penalty probabilities since Ω*D* → ∞ as *D* → ∞ when *cv* > *cu*.

**Pessimistic Ж(r,D)**

In this scheme, state transitions in response to success are identical to the Ambivalent **Ж** (Equations (8), (10), (12)). However, state transitions in response to failure differ from the Ambivalent **Ж**, and are shown for the intermediate, terminal, and initial states in Figure 4. These differences in state transitions in response to failure are described in detail below.

1) When in an initial state (*i*,1), if punished (Figure 4c, *β* = 1, Pessimistic automaton), transit to the initial state of the next branch of the *r* branches, i.e. ((*i* +1)(mod *r*),1) in the *r*-action PSAFA, with probability 1.

*F*((*i*’,1)| (*i*, 1),*i*,1) = 1,  *i* = {0,1,2, ... *r*-1}, *i*’= (*i* +1)(mod *r*) (31)

2) When in an intermediate state (*i*,*d*), if punished (Figure 4a, *β* = 1, Pessimistic automaton), transit to the initial state of the next branch of the *r* branches, i.e. ((*i* +1)(mod *r*),1) in the *r*-action PSAFA, with probability θ*d*, and transit to state (*i*,*d*-1) with probability 1- θ*d*.

*F*((*i*’,1)| (*i*, *d*),*i*,1) = θ*d*, *F*((*i*,*d*-1)| (*i*,*d*),*i*,1) = 1- θ*d*

 *i* = {0,1,2, ... *r*-1}, 1 < *d* < *D*, *i*’= (*i* +1)(mod *r*) (32)

3) When in a terminal state (*i*,*D*), if punished (Figure 4b, *β* = 1, Pessimistic automaton), transit to the initial state of the next branch of the *r* branches, i.e. ((*i* +1)(mod *r*),1) in the *r*-action PSAFA, with probability θ*D*, and transit to (*i*,*D-*1) with probability 1- θ*D*.

*F*((*i*’,1)| (*i*, *D*),*i*,1) = θ*D*, *F*((*i*, *D*-1)| (*i*, *D*),*i*,1) = 1- θ*D*,  *i* = {0,1,2, ... *r*-1}, *i*’= (*i* +1)(mod *r*) (33)

Using (8), (10), (12), (31-33) we can compute the limiting state probabilities and action probabilities and prove the expediency and optimality of the automaton. Here, we only present the results of our analysis; detailed derivations are presented in (Aggarwal, Liu & Levitin 2022) since the derivations are rather long for the Pessimistic PSAFA model. The nonzero elements of ***P*** turn out to be

*P*(*i*,*d*),(*i*,*d*+1) = 1-*ci*, *P*(*i*,*d*),(*i*,*d-*1) = *ci* (1-θ*d*),

 *P*(*i*,*d*),( *i’*,1) = *ci* θ*d*, *P*(*i*,*D*),(*i*,*D*) = 1- *ci* , *P*(*i*,1),(*i’*,1) = *ci*

 1< *d*  ≤ *D* *and i* = {0,1,2, ... *r*-1}, *i*’= (*i* +1)(mod *r*) (34)

all the other elements of ***P*** are zero. From (34), it is obvious that for a given state (*i*,*d*) the diagonal element in ***P***2 corresponding to that state, i.e. ***P***2(*i*,*d*),(*i*,*d*) is larger than zero. Intuitively, this also follows from the observation that, although like the two other automata, every state in the Pessimistic **Ж**, has its probability outflow equal to its state probability after every time step, it also has positive probability inflow after 2 time steps. This is because, unlike the other two automata, the probability outflow following punishment is also inward within a branch, rather than just outward in the case of the other two automata. Furthermore, it is easy to check that all the elements of ***P****D+r*-1 are larger than zero. Hence, the state process Ф(*n*) coresponding to the Pessimistic Ж(*r*,*D*) automaton is irreducible, aperiodic and, as a consequence, ergodic [1]. From ***P*** in (34) we can compute the limiting state probabilities , which turn out to be be solutions of the following equations

 ξ = *ci* θ*d* π *i,d* , π*i*,1= (1- θ*2*)*ci*π*i*,2 + ξ ,

π*i*,*d*= π*i*,*d*+1 (1- θ*d+*1) *ci+*π*i*,*d-*1 (1- *ci* ) , π*i*, *D*= π*i*,*D-*1  (35)

For a *r*-action automaton, the action probabilities for any two actions, say action *u* and *v* without loss of generality, i.e. *pu* and *pv*, respectively, will be given by

*pu* = π*u*,*d , pv* = π*v*,*d*

Thus, the ratio of the probabilities of the two actions *u* and *v*, becomes

Ω*D* =  = = where π*i*,*d* **=**  *i*,*d*

*i*,*D***=** ; *i*,*D*-1**=** π*i*,*D*; *i*,*d* = *ad*(*ci*) + γ*d*(*ci*) *i*,*d***+1** ;

*Kd* = *Kd* -1 - *ci*(1- *ci*)(1- )\**Kd* – 2 ; *K*0 = *K*1 = 1

*ad*(*ci*) = ; γ*d*(*ci*) = [ 1 – ] (36)

θ*d* is selected as in (19). Thus, as shown in (Aggarwal, Liu & Levitin 2022), the ratio of probabilites in (36), Ω*D* , is proportional to (*D*-1)(b’*v*- b’*u*), where b’ = , if *cu* and *cv* are both less than 1/3. If c*v*>1/3 and c*u*<1/3, Ω*D* is proportional to (*D*-1)(1- b’*u*). If *cv* , *cu* > 1/3, Ω*D* converges to a finite value as *D* → ∞. Hence, the Pessimistic PSAFA is *ε*-optimal only if at least one of the penalty probabilities c*i*, is less than (1/3) since Ω*D* → ∞ as *D* → ∞ only for this case.

In summary, based on the results in sections III0, 0, and 0, the Optimistic **Ж** is *ε*-optimal for all possible penalty probabilities. This result is significant since an LA model that is universally *ε*-optimal without manipulating the penalty probabilites (as in Economides & Kehagias, 2002; Oommen & Christensen 1988), has defied researchers for a long time. Furthermore, the PSAF automata are also easier to implement, requiring no floating point multiplications, unlike VSSA frameworks. Finally, PSAF automata are mathematically more tractable, since they can be analyzed by the theory of finite Markov chains; the analysis of LR-εP behavior requires the use of stochastic difference equations and an approximation argument (Narendra & Thatachar, 2012, pp. 166–168).

**Conclusions and Future Work**

In this paper, we designed the PSAFA framework for FSSA. We presented an analytical model of the asymptotic behavior of three of the many possible FSSA within that framework. It is worth noting that, of these three, only the Optimistic PSAFA is universally ε-optimal. This observation implies that universal ε-optimality results from very specific characteristics in any given FSSA. In general, previous researchers have claimed, implicitly or explicitly, that to achieve ε-optimality, 50% of penalties from the environment need to be randomly ignored, and treated as rewards (e.g. Tsetlin, 1962). No such strategy has to be applied in the case of Optimistic PSAFA. In contrast, the Pessimistic PSAFA would require that two-thirds of the failures be ignored to become universally ε-optimal. This could mean that there exists an entire spectrum of FSSA that would require ignoring different fractions of the failures, besides 1/2 and 2/3.

We believe that the architecture of the different PSAFA should be able to shed some light on what strategies need to be present in an FSSA for it to be ε-optimal. For example, the states in the three automata presented in this paper have two broad classes of state transitions, inter-branch state transitions and intra-branch state transitions. While the inter-branch state transition probabilities are identical in all the three automata, their intra-branch state transitions are distinctly different. So, the difference in their ε-optimality properties is a result of their intra-branch state transitions. However, the inter-branch state transitions also play an important role. Indeed, if θ becomes a constant, i.e. independent of the depth of a state *d*, then the behavior of PSAFA will be independent of *D*, and will just degenerate into the behavior of LN,N automaton, where there is only one state per action i.e. there are N states corresponding to N actions in total and all state transitions are deterministic.

The FSSA framework presented in this paper has no non-trivial deterministic counterpart. FSSA presented in previous studies have achieved *ε*-optimality by randomizing the response to a reward/penalty by a deterministic FSSA. In fact, this approach has also been used in the VSSA framework, where different techniques are used for estimating penalty probabilities, and then action selection is randomized based on these estimated values. It has been noted in previous research that action selection schemes can be designed using multiple FSSAs, rather than a single FSSA, that will outperform the single FSSA. A corollary to this observation is that there exists a VSSA design corresponding to each FSSA design, based on estimating the probability of occupation of every state in the FSSA, and then using the state occupation probabilities, to calculate action selection probabilities (Oommen & Agache 2001). However, the computational load in this case increases with increasing number of steps, and hence makes these VSSA designs incapable of simulating optimality in real-world problems. Thus, the requirement for randomized action switching and finite computational load with increasing number of possible actions and attempts/steps, makes ε-optimal FSSA designs more likely for solving real-world problems.

The primary differentiating feature of PSAFA is highlighted by non-zero values of the state action switching probability parameter θ. Non-zero values of this parameter ensure a non-zero action switching probability in every state. If θ is set to zero, the 2-action Pessimistic PSAFA transforms into the Tsetlin automaton L2N,2. Therefore, comparison of the dynamic behavior of the Pessimistic PSAFA and Tsetlin automaton, for example, can be used to clarify the impact of non-zero values of θ on the PSAFA performance. Another FSSA with similar response to penalty but different response to reward is the Krinsky automaton. An FSSA with similar response to reward but different response to penalty is the Krylov automaton . An alternate strategy was proposed by Ponamerov where a horizontal row of action switching states with alternate actions taken by adjacent states, separate two vertical rows of non-action switching states that have the same response to reward and penalty as L2N,2 in a 2-action selection environment. All these automata are described in (Narendra & Thatachar, 2012, pp 92-96). A similar and more popular automaton was also proposed in (Cover & Hellman, 1970). The comparison of dynamic behavior of these automata with the Pessimistic PSAFA as well as other PSAFA will be pursued in future work.

Previous researchers have investigated *ε*-optimal automaton designs in the past because designing a strictly optimal automaton would require an infinite number of states in an automaton. Ideally, though, the real goal of LA is to identify designs that can solve the action-selection problem optimally. An *ε*-optimal FSSA can approach such optimality if we take into consideration the fact that, in real-world, the number of steps is always finite, the computational load for an FSSA remains constant even as the number of steps increase, and the maximum depth of the active state at any instant increments at a lesser rate than the number of steps.

For given values of *r* and *D*, the PSAFA achieves an order of success probability different from that of the corresponding Tsetlin, or STAR automata since the PSAFA achieve ε-optimality according to a power law vs. depth, rather than exponentially, unlike most LA. There is a possibility that the larger number of states used by PSAFA to achieve the same level of success probability results in better dynamic properties. We will explore this possibility further when we present detailed simulation results in part 2 of this 2 part paper to compare the performance of PSAFA and other FSSA, in simple non-stationary environments.

Note that the framework for LA dictates that the environment has only two responses, either a reward or a penalty, with no quantitative value. That was the framework we worked with in this paper. In case a larger number of responses is considered, the reinforcement learning framework starts resembling a supervised learning framework since the response starts serving as a label that can be used to segment actions.

The PSAFA has a non-zero action-switching probability in all its states, irrespective of the previous history. The time evolution of the automaton can also be described by a discrete-time finite-state Markov decision process. In future research, we aim to explore applications of the PSAFA that leverage on these two properties, besides the ε-optimality characteristics.

**References**

Abeyrathna, K. D., Granmo, O., Zhang, X., Jiao, L. & Goodwin, M. (2020) The regression Tsetlin machine: a novel approach to interpretable nonlinear regression, *Philos Trans A Math Phys Eng Sc.; vol. 378*, issue 2164, Feb 7 2020.

Aggarwal, P. S., & Kumar, A. (1997) An unsupervised stochastic learning model for syntactic pattern recognition using the discrete Kalman filter scheme, *IEEE International Conference on Systems, Man, and Cybernetics, 1997, 'Computational Cybernetics and Simulation', vol. 4*, pp. 3046-3052, 12-15 Oct. 1997.

Aggarwal, P. S., Liu, C., Levitin, L., (2022) A New Class of Learning Automata: The Probabilistically-Switch-Action-on-Failure Automaton: Part 1 – Analytical Models. Can be retrieved from <https://www.techrxiv.org/articles/preprint/A_New_Class_of_Learning_Automata_The_Probabilistically-Switch-Action-on-Failure_Automaton_Part_1_Analytical_Models/19425761/1>

Atkinson, R. C. & Bower, G. H. (1965) *An Introduction to Mathematical Learning Theory*. *New York: Wiley*, 1965.

Bush, R. R. & Mosteller, F. (1955) *Stochastic Models and Learning*. *New York: Wiley*, 1955.

Cover, T. M., & Hellman, M. E. (1970) The Two–Armed Bandit Problem with Time Invariant Finite Memory, *IEEE Trans, on Information Theory*, Vol. 16(2), pp. 185–95, 1970.

Economides, A. A., & Kehagias, A. (2002) **The STAR automaton: expediency and optimality properties**, *IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 32,* pp.723 – 737, Issue: 6, Dec. 2002.

Ghaleb, O., & Oommen, B. J. (2019) Learning automata-based solutions to the single elevator problem, *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innov. Cham*, *Switzerland*: Springer, pp. 439–450, 2019.

Ghavipour, M., & Meybodi, M. R. (2016) An adaptive fuzzy recommender system based on learning automata, *Electron. Commerce Res. Appl., vol. 20*, pp. 105–115, Nov. 2016.

Guo, H., Li, S., Qi, K., Guo, Y., & Xu, Z. (2020) Learning Automata Based Competition Scheme to Train Deep Neural Networks, *IEEE Transactions on Emerging Topics in Computational Intelligence*, *vol. 4*(2). pp. 151-158, April 2020.

Hasanzadeh-Mofrad, M. & Rezvanian, A. (2018) Learning automata clustering*, J. Comput. Sci., vol. 24,* pp. 379–388, Jan. 2018.

Liu, C. (2006) *Design and analysis of novel fixed structure stochastic automata in non-stationary environments*, Ph.D. Dissertation, College of Engineering, Boston University, 2006.

Mazhari, S. M., & Monsef, H. (2012) A generalized model for bankruptcy prediction of the electricity industrial firms: Empirical evidence for the restructured Iranian distribution companies, 2012 Proceedings of 17th Conference on Electrical Power Distribution, 2012, pp. 1-7.

Misra, S., Chatterjee, S.S., & Guizani, M. (2015) Stochastic learning automata-based channel selection in cognitive radio/dynamic spectrum access for WiMAX networks, *Int. J. Commun. Syst., vol. 28*(5), pp. 801–817, 2015.

Narendra, K.S., & Thathatchar, M. A. L. (2012). *Learning Automata: An introduction*. *Englewood Cliffs, NJ: Dover Publications,* 2012.

Norman, M. F. (1972) Markov Processes and Learning Models. New York: Academic, 1972.

Oommen, B. J., & Roberts, T. D. (1998) A fast and efficient solution to the capacity assignment problem using discretized learning automata, *Proc. 11th Int. Conf. Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, vol. II*, pp. 56–65. 1998.

Oommen, B, J., & Hashem, M. K. (2010) Modeling a Student's Behavior in a Tutorial-Like System Using Learning Automata, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 40*(2), pp. 481-492, April 2010, doi: 10.1109/TSMCB.2009.2027220.

Oommen, B, J., and Agache, M. A. (1999) A comparison of continuous and discretized pursuit learning schemes, *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics, vol. IV,* pp. 1061–1067, 1999.

Oommen, B, J., and Agache, M. A. (2001) Continuous and discretized pursuit learning schemes: Various algorithms and their comparison, IEEE Trans. Syst., Man, Cybern. B, vol. 31, pp. 277–287, Apr. 2001.

Rezvanian, A., Saghiri, A. M., Vahidipour, S. M., Esnaashari, M., Meybodi, M. R. (2018) *Recent Advances in Learning Automata*. *Springer 2018, Volume 754* ISBN: 978-3-319-72427-0

Sbruzzi, E. F., Leles M. C. R., & Nascimento, C. L. (2018) Introducing learning automata to financial portfolio components selection, *2018 Annual IEEE International Systems Conference (SysCon), Vancouver, BC, Canada*, pp. 1-6, doi: 10.1109/SYSCON.2018.8369522.

Tsetlin, M. L. (1962) On the behavior of finite automata in random media, *Autom. Remote Control, vol. 22,* pp. 1210–1219, 1962.

Vahidipour, M., & Esnaashari, M. (2018) Priority assignment in queuing systems with unknown characteristics using learning automata and adaptive stochastic Petri nets, *J. Comput. Sci., vol. 24*, pp. 343–357, Jan. 2018.

Varshavskii, V. I., & Vorontsova, I. P. (1964) On the behavior of stochastic automata with a variable structure, *Autom. Remote Control, vol. 24*, pp. 327–333, 1964.

Zhang, Z., Wang, D., & Gao, J. (2021) Learning Automata-Based Multiagent Reinforcement Learning for Optimization of Cooperative Tasks, *IEEE Transactions on Neural Networks and Learning Systems, vol. 32*(10), pp. 4639-4652, Oct. 2021, doi: 10.1109/TNNLS.2020.3025711.

**Biographies**

Prateek S. Aggarwal received his PhD in Biomedical Engineering from Boston University in 2003. He has been working as a data scientist in different companies since 2005 including Oracle Retail, Adobe and JP Morgan & Chase. He is currently employed as a senior data scientist with MasterCard. He is a member of IEEE.

Chenhui Liu is currently the Chief Scientist & Architect at Learning Equity, Inc. Her research interests include learning automaton design, efficient encoding and decoding of learning processes, AI and its applications to portfolio optimization and wealth management, default and bankruptcy prediction, and optimal design of learning paths with interactive cross-media systems. She received her Ph.D. in Computer Engineering from Boston University in 2006. She is a member of IEEE.

Lev Levitin is currently distinguished professor of Engineering Science with the Department of Electrical and Computer Engineering at Boston University, Boston, MA. He has published over 200 papers, presentations, and patents. His research areas include information theory; quantum communication systems; physics of computation; quantum computing; quantum theory of measurements, mathematical linguistics; theory of complex systems; coding theory; theory of computer hardware testing, reliable computer networks, and bioinformatics. He is a life fellow of IEEE, a member of the International Academy of Informatics and other professional societies. Prof. Levitin may be reached at levitin@bu.edu