A Two-Timescale Learning Automata Solution to the Non-Linear Stochastic Proportional Polling Problem

Anis Yazidi, Hugo Hammer, and David S. Leslie

***Abstract*—In this paper, we introduce a novel Learning Automata (LA) solution to the Non-linear Stochastic Proportional Polling (NSPP) problem. The only available solution to this problem in the literature is that given by Papadimitriou et al. [1]–[3]. It was shown to solve a large set of adaptive resource allocation problems under noisy environments [1], [2], [4]–[10]. We make a threefold contribution. First, we take a two-timescale approach to the field of LA by estimating the reward probabilities on a faster timescale than the timescale for updating the polling probabilities. Second, by making a not-obvious choice of the objective function, we show that the NSPP problem is indeed an instantiation of the Stochastic Non-linear Fractional Equality Knapsack (NFEK) problem, which is a substantial resource allocation problem based on incomplete and noisy information [11], [12]. Third, in contrast to the legacy approach taken by Papadimitriou et al. [13], [14], we show through extensive experimental results that our solution is remarkably *robust* to the choice of tuning parameters and that it outperforms the state-of-the-art solution in terms of Bayesian expected loss.**

Keywords: *Learning Automata, Two-timescale Learning, Resource Allocation, Stochastic Non-linear Fractional Equality Knapsack* .

I. Introduction

A Learning Automaton (LA) is an adaptive learning mechanism that interacts with a stochastic environment in order to find an optimal action from among a set of offered actions [15], [16]. The field of LA was pioneered by Tsetlin more than five decades ago [17] through the introduction of the first LA scheme based on finite-state machines. Since then, the theory of LA has attracted extensive research interest [15], [17]–[20]. Most LA schemes in the literature aspire, as time elapses, to learn the optimal action among a usually finite set of actions offered by a stochastic environment, where the notion of optimality is often defined as converging to the action with the largest reward probability, or, equivalently, the action with the lowest penalty probability. An LA solution that is radically different from the latter stream of research is that proposed by Papadimitriou et al. [13], [14]. In a series of works, Papadimitriou et al. [13], [14] addressed the so-called Non-linear Stochastic Proportional Polling (NSPP) problem. In contrast to the main stream of LA algorithms that operate under the assumption that the reward probabilities of the actions are stationary over time, the NSPP problem instead works on the premise that reward probabilities vary over time and depend on the action probability vector [11], [19], [21].

The NSPP problem [1]–[3] is characterized by the following three main peculiarities:

- First, the reward probabilities are *stochastic* variables whose distributions are *unknown*.
- Second, the reward probability of an action may decrease as a function of the polling probability.
- Third, the optimal probability of an action should be proportional to its reward probability.

The only available solution to the NSPP problem is that proposed by Papadimitriou et al. [1]–[3], which was shown to have a large set of applications in computer science [1], [2], [4]–[10].

In this paper, we show that the NSPP problem can be formulated as an instantiation of the Stochastic Non-linear Fractional Equality Knapsack (NFEK) problem [22] thanks to a subtle choice of the objective function. We therefore bridge the gap between the Stochastic NFEK problem [11], [12], [22], [23] and the NSPP problem of Papadimitriou et al. [1]–[3]. It is worth mentioning that the first optimal solution to an instance of the Stochastic NFEK problem is that proposed by Granmo and Oommen [11], [12], which relies on two-action discretized LA. The solution was subsequently generalized by invoking a hierarchy of two-action discretized LA in order to resolve the generalized case of multi-materials. However, as we will elucidate later in the paper, our NSPP has a different objective function than the one found in [11], [12]. Therefore, the latter solution attributed to Granmo and Oommen [11], [12] cannot be used to solve the NSPP problem. In a recent work, Yazidi et al. [24] propose using the theory of a two-timescale approach to solve the latter problem. By virtue of introducing the two-timescale approach, Yazidi et al. [24] achieved a more stable convergence than in [11], [25]. The LA in [24] converges to a solution that equalizes the reward probabilities from each action. In informal terms, the decision maker is indifferent among

Anis Yazidi is with the Department of Computer Science, Oslo Metropolitan University, Oslo, Norway, NTNU: Norwegian University of Science and Technology, Norway and Oslo University Hospital, Oslo, Norway.

Hugo Hammer is with the Department of Computer Science, Oslo Metropolitan University, Oslo, Norway.

David S. Leslie is with the Department of Mathematics and Statistics, Lancaster University, UK.

the different actions at equilibrium as the reward probabilities from each action are equal. In this paper, we deal with the proportional polling problems where the action probabilities are proportional to the reward probabilities.

To the best of our knowledge, the latter work is the first work in the field of LA to invoke this concept.

In this paper, we introduce a novel strategy that allows the LA to obtain estimates of the reward probabilities, and a method for how these estimates can be utilized directly in the learning process. Our approach is distinct from classical LA schemes [26] that only require knowledge of the maximum estimates rather than the *exact* values of these estimates. To achieve synchronous updating and learning, we resort to carefully designed estimates of the reward probabilities when adjusting the action probabilities.

We present a threefold contribution that can be briefly outlined as follows:

1) First, we show that the NSPP problem [1]–[3] is an instantiation of the Stochastic NFEK.
2) Second, we report an application of the two-timescale approach to the field of LA by estimating the reward on a faster timescale than the update of the polling probabilities. This advance can pave the way for new LA schemes that build on the idea of connecting the updates of the action probabilities and reward probabilities in tandem using the concept of two-timescale updates.
3) Third, we show that our solution has high robustness to the choice of the tuning parameters while outperforming the state-of-the-art solution of Papadimitriou and colleagues [1]–[3].

The rest of this paper is organized in the following manner. First, in Section II, we review related work. Section IV presents the design of our solution together with sound theoretical results proving its convergence to an optimal fixed point. Section V provides some thorough experimental results investigating how our solution performs in different scenarios, and shows its superiority compared to the state-of-the-art solution. Finally, Section VI rounds off the paper with some concluding remarks.

II. Preliminaries

The vast majority of the LA schemes described in the literature are designed to converge on the action yielding the largest reward probability. Furthermore, in the field of LA, the reward probabilities are usually constant over time and independent of the action choice. However, in many real-life applications involving multiple agents that share some common resources, resorting to classical LA schemes to govern access to these resources among competing agents does not produce a fair result. To counter this problem, Papadimitriou, in collaboration with different researchers [1], [2], [4]–[10], has provided a novel LA design that yields a fair share of resources by ensuring that an entity can access or use a resource in proportion to its demand. This usually involves designing an LA that polls an action in a proportional manner to its reward probability. In the most generic case, the resource access can be modelled as a Non-linear Stochastic Proportional Polling (NSPP) problem [1]–[3] with the three aforementioned peculiarities, namely unknown distribution of the reward, monotonicity of the reward function, and an optimal fixed point characterized by a polling probability that is proportional to the corresponding reward probability. In this perspective, it is worth mentioning that the literature on LA includes two prominent studies that operate under the monotonicity assumption of the reward function, notably the work of Narendra and Thathachar on adaptive routing [19], [21], and the work of Granmo and Ommen on adaptive web crawling [11]. However, in contrast to the work of Papadimitriou et al. on the NSPP problem [1]–[3], the objective of the latter two works is to equalize the reward probabilities. The informed reader would remark that an ergodic LA is required in this case. However, work has been done on LA in non-stationary environments [27] where, even though the reward probabilities are time-varying, the optimal action is kept unchanged, which means that an absorbing LA scheme is required [28], [29]. Another type of non-stationarity in an LA environment appears in [30], where the source of the non-stationarity emanates from the fact that the joint local decisions of the neighboring cells affect the reward of the irregular cellular automata LA in the current cell.

In this section, we have tried to cover the most prominent proportional LA applications and schemes reported in the literature. For a complete list of proportional LA applications, we refer the reader to the following studies [31]–[39]. The latter articles do not present a new solution to the proportional polling problem originally devised in Papadimitriou et al. [13], [14]. These papers rather deal with different application domains of the originally devised algorithm in [13], [14]. The large amount of applications of the stochastic polling problem motivates a new efficient solution which is the objective of our paper.

The scheme developed in this paper can be deployed to model a large family of adaptive self-organizing systems that rely on Reinforcement Learning (RL) where there is a coupling between the reward probability and the action probability [40]–[45]. The main literature stream in RL deals with the assumption of ”no-effect of the chosen actions” on the future reward distribution. In those classical settings, the optimal policy is usually deterministic and consists in converging to the optimal action that yields the highest average reward. However, in many real-life scenarios, there is a coupling between reward outcomes and how often an action is chosen. This can be seen, for instance, in a resource allocation problem, where the more often a resource is chosen, the more it is exhausted, and thus, the less the resulting reward perceived by an RL agent [44]. In the latter work, we have demonstrated that classical RL fails dramatically due to exclusive action choice. Some recent literature in the field of RL have used the terminology the satiation effect to describe the phenomenon of diminishing reward [46], [47]. The satiation effect in RL describes the situation in which the perceived reward by an agent diminishes as it consumes more and more of the same resource. This is relevant for the case for instance of food recommendations where recommending the same food type might lead to boredom by the user. Wang et al. devised a per-load reward, where the reward depends on the number of players choosing the same arm [48], [49]. The main difference compared to our work is the fact that the latter two works consider decentralized settings, while ours consider centralized ones. Furthermore, in the latter two works, the reward depends on the number of players while in our settings it depends on the polling probabilities. In [46], the reward depends on the time since an arm was pulled in a consecutive manner. Other studies assume that the expectation of the reward from an action is a a function of the last time it was pulled [50], [51]. More precisely, the reward function is increasing concave function of the time since it was last played. In [52], [53], the expected reward of an arm increases since the last time it was pulled and decreases after pulling it. In this paper, we rather investigate a direct dependency between the reward and the action probabilities rather than a dependency with the last time it was pulled or the number of times it was pulled. The informed reader will observe that the polling problem considered in this paper is related to policy patrol optimization [54], which was solved using reinforcement learning in [55]. In [55], the aim is to choose a location to patrol at each time instant. The security state of a location degrades as the idling time between two consecutive patrols increases. In this sense, it is possible to model the reward probability of patrolling as a decreasing function of the patrol probability.

III. Problem formulation

In this section, we formulate the NSPP problem as an instance of the Stochastic NFEK problem [11]. The Stochastic NFEK involves finding an optimal allocation of n materials, denoted as $x^{opt} = [x_1^{opt}, \dots, x_n^{opt}]$, that can fit within a knapsack of capacity c. (Without loss of generality, we suppose that $c = 1$.) Granmo and Oommen formulate the problem as:

$$\begin{aligned} &\text{maximize } f(x) = \sum_{i=1}^{n} f_i(x_i) \\ &\text{where } f_i(x_i) = \int_0^{x_i} p_i(u)\,\mathrm{d}u \;\forall i \in \{1, \dots, n\}, \qquad (1) \\ &\text{subject to } \sum_{i=1}^{n} x_i = c \text{ and } x_i \geq 0 \;\forall i \in \{1, \dots, n\}. \end{aligned}$$

The functions p_i are monotonic, either all decreasing or all increasing, but otherwise unknown. Instead, the optimiser observes a random variable v_i with a Bernoulli$(p_i(x_i))$ distribution in response to selecting material i with probability x_i.

A. Formulation of NSPP as an instance of the Stochastic NFEK

In order to show that our NSPP problem is an instance of the Stochastic NFEK problem, we instead define $f_i(x_i) = \int_0^{x_i} u p_i(u)^{-1}\,\mathrm{d}u$. An alternative interpretation of this is to replace $p_i(u)$ with $\tilde{p}_i(u) := u/p_i(u)$ in (1), although the feedback available is still a $v_i \sim$ Bernoulli$(p_i(x_i))$ instead of the Bernoulli distribution with the parameter $\tilde{p}_i$. Without loss of generality, in the rest of the article, $f_i(x_i)$ will be defined as $f_i(x_i) = \int_0^{x_i} u p_i(u)^{-1} du$. Furthermore, we suppose that the functions p_i are strictly monotonically decreasing, Lipschitz and continuous.

$$\begin{aligned} &\text{maximize } f(x) = \sum_{i=1}^{n} f_i(x_i) \\ &\text{where } f_i(x_i) = \int_0^{x_i} u p_i(u)^{-1}\,\mathrm{d}u \;\forall i \in \{1, \dots, n\}, \qquad (2) \\ &\text{subject to } \sum_{i=1}^{n} x_i = 1 \text{ and } x_i \geq 0 \;\forall i \in \{1, \dots, n\}. \end{aligned}$$

We will show that, by resorting to this not-obvious choice of $f_i(x_i)$ given by $f_i(x_i) = \int_0^{x_i} u p_i(u)^{-1} du$, the solution to the above Stochastic NFEK problem will coincide with the solution to the NSPP problem.

B. Characterization of the Stochastic NFEK solution

We will first study the characteristics of the optimal solution to the above Stochastic NFEK problem described by the optimization problem (2).

Lemma 1. *The material mix $x^{opt} = [x_1^{opt}, \dots, x_n^{opt}]$ that solves the Stochastic NFEK Problem (2) is characterized by*

$$\frac{x_1^{opt}}{p_1(x_1^{opt})} = \frac{x_2^{opt}}{p_2(x_2^{opt})} = \dots = \frac{x_n^{opt}}{p_n(x_n^{opt})}$$

Furthermore, the problem (2) admits a unique solution.

Proof. We consider the following Lagrangian function $L(x, \eta)$ where $\eta \geq 0$ is the Lagrange multiplier corresponding to the optimization problem (2):

$$L(x, \eta) = \sum_{i=1}^{n} f_i(x_i) + \eta(\sum_{i=1}^{n} x_i - 1)$$

To solve the optimization, we set the partial derivatives for $1 \leq i \leq n$:

$$\frac{\partial L(x, \eta)}{\partial x_i} = 0$$

This gives for $1 \leq i \leq n$:

$$\frac{\partial f_i(x_i)}{\partial x_i} = \eta$$

Thus,

$$\frac{\partial f_1(x_1)}{\partial x_1} = \cdots = \frac{\partial f_n(x_n)}{\partial x_n}$$

It follows immediately that

$$\frac{x_1^{opt}}{p_1(x_1^{opt})} = \cdots = \frac{x_n^{opt}}{p_n(x_n^{opt})} \tag{3}$$

and simple algebra leads to

$$\frac{x_i^{opt}}{p_i(x_i^{opt})} = \frac{\sum_{j=1}^{n} x_j^{opt}}{\sum_{j=1}^{n} p_j(x_j^{opt})} \tag{4}$$

Using the fact that $\sum_{j=1}^{n} x_j^{opt} = 1$, we obtain

$$x_i^{opt} = \frac{p_i(x_i^{opt})}{\sum_{j=1}^{n} p_j(x_j^{opt})} \tag{5}$$

Now, we proceed to proving the uniqueness of the solution by contradiction. Let $x^{opt} = [x_1^{opt}, \ldots, x_n^{opt}]$ be an optimal solution. Let us suppose that another solution $y^{opt} = [y_1^{opt}, \ldots, y_n^{opt}]$, exists to the problem. Therefore, $y^{opt} = [y_1^{opt}, \ldots, y_n^{opt}]$ must verify equation (4).

Since x^{opt} and y^{opt} are two distinct probability vectors, we can affirm that they have two indexes i and j where on the one hand $x_i^{opt} > y_i^{opt}$ and on the other hand $x_j^{opt} < y_j^{opt}$. As a consequence of the fact that $p_i(\cdot)$ and $p_j(\cdot)$ are monotonically decreasing, we obtain $p_i(x_i^{opt}) < p_i(y_i^{opt})$ and $p_j(x_j^{opt}) > p_j(y_j^{opt})$. Thus $x_i^{opt}/p_i(x_i^{opt}) > y_i^{opt}/p_i(y_i^{opt})$ and $x_j^{opt}/p_j(x_j^{opt}) < y_j^{opt}/p_j(y_j^{opt})$. This contradicts (3), since we cannot have both $x_i^{opt}/p_i(x_i^{opt}) = x_j^{opt}/p_j(x_j^{opt})$ and $y_i^{opt}/p_i(y_i^{opt}) = y_j^{opt}/p_j(y_j^{opt})$.

Therefore, we deduce that the solution is unique.

□

We have shown that the Stochastic NFEK problem (2) has a unique solution

$$x_i^{opt} = \frac{p_i(x_i^{opt})}{\sum_1^n p_j(x_j^{opt})} \tag{6}$$

which is indeed the solution to the NSPP problem reported in the literature [13], [14]. Hence, a learning algorithm which solves the Stochastic NFEK problem will also solve the NSPP problem.

IV. A two-timescale LA solution to Resource Allocation

While we now have a Stochastic NFEK formulation of the problem, the p_i are considered unknown, so it is not possible to solve (2) directly. We therefore derive a learning algorithm to find optimal solutions, building on earlier linear automata approaches to NFEK [11], [12], [22]. LA is just an instance of RL algorithms. In the original paper introducing REINFORCE [56], LA was shown to be an instance of the REINFORCE framework. In this paper, what is original is the design of the updating scheme in the form of ODE in order to make the LA system converge to the desired equilibrium of Proportional Polling Problem where the optimal probability of an action should be proportional to its reward probability. LA can be seen as another variant or solution to the multi-armed bandit problem. LA is competitive to other RL algorithms such as for instance Q-learning. If one applies instead a classical RL algorithm such as the linear-reward inaction the RL algorithm will converge to equalizing the reward probabilities of the different actions which is not the desired equilibrium in the case of our proportional polling problem [11], [25]. The direct application of any bandit algorithm to this problem will not lead to the desired of equilibrium of the NSPP problem.

The general approach is to play repeatedly, evolving an action x in response to observations of the rewards received. Previous automata approaches are not effective for this problem, since the available feedback is a signal, $v_i \sim$ Bernoulli$(p_i(x_i))$, while our Stochastic NFEK formulation integrates $\tilde{p}_i(x) = x/p_i(x)$ instead of $p_i(x)$ in the objectives; the reason we care about this alternative Stochastic NFEK formulation is that the Bernoulli$(p_i(x_i))$ feedback signals are those we observe in an NSPP. Hence we develop a two-timescale approach in which we estimate the (current) response probabilities p_i, while evolving our action vector x.

A. Details of the two-timescale LA Solution

At each time instant t, the player:

- selects action $i(t)$ using strategy $x(t)$ (i.e., set $i(t) = i$ with probability $x_i(t)$),
- observes a response $v(t)$, which is a Bernoulli random variable with the parameter $p_{i(t)}(x_{i(t)}(t))$,
- updates the belief vector $\hat{p}(t)$ and the strategy $x(t)$.

In fact, we need to modify $x(t)$ slightly before playing (see below).

The interleaving updates of $\hat{p}$ and x are:

a) Updating $\hat{p}_i(t)$ for $1 \le i \le n$: The estimates of the reward probabilities are updated in the following manner:

$$\begin{aligned} \hat{p}_{i(t)}(t+1) &\leftarrow \hat{p}_{i(t)}(t) + \alpha_t \left[v(t) - \hat{p}_{i(t)}(t)\right] \\ \hat{p}_j(t+1) &\leftarrow \hat{p}_j(t) \text{ for } j \neq i(t) \end{aligned} \tag{7}$$

where $\alpha_t \in [0, 1]$ is a learning parameter. As we can see from the above equation, $\hat{p}_i(t)$ is an estimate of the reward probability obtained using an exponential moving average update with α_t as a learning parameter. $\hat{p}_i(t)$ are probability estimates. The initial values of $\hat{p}_i(t)$ should be in the interval $[0, 1]$. Although the initial values do not affect the asymptotic convergence they can influence the convergence speed during the first iterations. We suggest to choose 0.5 as initial value in the absence of a priori knowledge about the underlying functions $p_i(.)$.

b) Updating $x(t)$: We suppose that initially, $x_i(0) = 1/n$ for $1 \le i \le n$. The value of $x_i(t)$, $1 \le i \le n$ is updated as per the following rule:

$$x_i(t+1) \leftarrow x_i(t) + \theta_t \left[\hat{p}_i(t) - x_i(t) \sum_{j=1}^{n} \hat{p}_j(t)\right] \tag{8}$$

The stepsizes α_t, θ_t in (7) and (8) are positive scalars satisfying:

$$\sum_t \alpha_t = \sum_t \theta_t = \infty, \quad \sum_t \alpha_t^2 + \theta_t^2 < \infty, \quad \theta_t = o(\alpha_t)$$

The first two terms are standard for the stochastic approximation literature. The final term is what makes the approach a two-timescale one [57]: parameter θ_t, which governs the magnitude of the $x(t)$ updates should be considerably smaller than parameter α_t used to track the reward probability estimates $\hat{p}(t)$, with the consequence that from the perspective of the fast dynamics of $\hat{p}_i$, x_i seems to be "almost constant", while when considering the slow dynamics of x_i, $\hat{p}_i$ seems to be almost always "almost equilibriated" [58].

It is easy to see that the above update form $x_i(t)$, $1 \leq i \leq n$ preserves the property that the components of the probability $x(t)$ still sums to 1:

$$\begin{aligned}
\sum_{i=1}^{n} x_i(t+1) &= \sum_{i=1}^{n} \left[x_i(t) + \theta_t(\hat{p}_i(t) - x_i(t) \sum_{j=1}^{n} \hat{p}_j(t)) \right] \\
&= 1 + \theta_t \sum_{i=1}^{n} \hat{p}_i(t) - \theta_t \sum_{j=1}^{n} \hat{p}_j(t) \sum_{i=1}^{n} x_i(t) \\
&= 1 + \theta_t \sum_{j=1}^{n} \hat{p}_j(t) - \theta_t \sum_{j=1}^{n} \hat{p}_j(t) \\
&= 1
\end{aligned}$$

B. *Remark: Bounds of the sampling probability*

Although (7–8) is the scheme we would like to run, it is necessary to ensure that all actions are selected sufficiently often so that all of the $\hat{p}_i$ estimates are reasonable. We therefore do not use strategy $x(t)$ to select the actions, but use a modified strategy x' satisfying

$$x'_i(t) = \frac{\epsilon}{n} + (1-\epsilon)x_i(t). \tag{9}$$

Choosing $\epsilon > 0$ ensures that all actions are played infinitely often, while choosing ϵ small enough ensures that the system is only perturbed to a small extent. Note, however, that, by playing strategy $x'(t)$ instead of $x(t)$, the reward $v(t)$ is now sampled from a Bernoulli distribution with a success probability of $p_{i(t)}(x'_{i(t)}(t))$. We denote $q_i^\epsilon(x_i(t)) = p_i(x'_i(t)) = p_i\left(\frac{\epsilon}{n} + (1-\epsilon)x_i(t)\right)$.

At this juncture, we note that there is some sensitivity to choosing sufficiently small learning parameters so that the $x_i(t)$ probabilities remain positive. In particular, assuming that $x_i(t) > 0$,

$$\begin{aligned}
x_i(t+1) &= x_i(t) + \theta_t[\hat{p}_i(t) - x_i(t) \sum_{j=1}^{n} \hat{p}_j(t)] \\
&= x_i(t)(1 - \theta_t \sum_{j=1}^{n} \hat{p}_j(t)) + \theta_t \hat{p}_i(t) \\
&\geq x_i(t)(1 - \theta_t n)
\end{aligned}$$

since $\sum_{j=1}^{n} \hat{p}_j(t) \leq n$. Hence a sufficient condition is $\theta_t < \frac{1}{n}$ for all t.

Similarly, if $\epsilon \in [0,1]$, then $x'_i(t)$ is a convex combination of $1/n$ and $x_i(t) \in [0,1]$ and therefore lies in $[\epsilon/n, 1-(1-1/n)\epsilon]$.

C. *Convergence proofs*

In this section, we resort to the theory of two-timescale stochastic approximation [57]–[59]. The "almost constant" and "almost calibrated" nature of the system means that we should consider continuous time ODE's corresponding to the 'fast' timescale $\hat{p}(t)$ as if the 'slow' timescale $x(t)$ is constant, and, for the $x(t)$ system, as if the $\hat{p}(t)$ system is fully calibrated. We start by proving the convergence of the slow system.

Let $x(t) = (x_1(t), \ldots, x_n(t))$, $\hat{p}(t) = (\hat{p}_1(t), \ldots, \hat{p}_n(t))$ and $q(x) = (q(x_1), \ldots, q_n(x_n))$ where $q_i^\epsilon(x_i) = p_i(\epsilon/n + (1-\epsilon)x_i)$ denotes the composition of the smoothing from x to x' followed by the proportional polling function p_j. It is easy to check that the $q_i^\epsilon(.)'$ s are strictly decreasing, continuous, non-negative functions of the x_i's because the $p_i(.)$'s possess the same property.

Consider the following ODE for $i \in [1, n]$:

$$\dot{x}_i = q_i^\epsilon(x_i) - x_i \sum_{j=1}^{n} q_j^\epsilon(x_j). \tag{10}$$

Let $x^*(\epsilon)$ be the fixed point of the ODE given by (10). The uniqueness of $x^*(\epsilon)$ is guaranteed by following the same lines as Lemma 1 for a modified Stochastic NFEK problem where p_i is replaced by q_i^ϵ in the original Stochastic NFEK problem (2).

Proposition 1. *The fixed point of the ODE given by Eq. (10) is asymptotically stable and unique.*

Proof. Throughout this proof, we fix ϵ and suppress it from the notation (in both q_i^ϵ and $x^*(\epsilon)$ to simplify the notation. Consider the following Lyapunov function.

$$V(x) = -\sum_{i=1}^{n} x_i^* \ln(\frac{x_i}{x_i^*}), \tag{11}$$

where x^* is the fixed point of the ODE (10). We can see that $V(x^*) = 0$.

By applying the Jensen's inequality, we move the log outside the parentheses:

$$\ln(\sum_{i=1}^{n} x_i^* \frac{x_i}{x_i^*}) \geq \sum_{i=1}^{n} x_i^* \ln(\frac{x_i}{x_i^*})$$

$$\ln(\sum_{i=1}^{n} x_i) \geq -V(x)$$

$$\ln(1) \geq -V(x)$$

$$V(x) \leq 0$$

Because the function $\ln(.)$ is strictly concave, the equality $V(x) = 0$ holds whenever:

$$\frac{x_1}{x_1^*} = \cdots = \frac{x_n}{x_n^*} \tag{12}$$

Let the above ratio equal σ.

Therefore,

$$1 = \sum_{i=1}^{n} x_i = \sigma \sum_{i=1}^{n} x_i^* = \sigma \tag{13}$$

Then $\sigma = 1$. Thus, $V(x) = 0$, yields that $x_i = x_i^*$ $\forall i \in \{1, \dots, n\}$.

We know that $\sum_{i=1}^{n} \dot{x}_i = 0$.

$$\begin{aligned}
\frac{dV(x)}{dt} &= \sum_{i=1}^{n} \frac{\partial V(x)}{\partial x_i} \dot{x}_i \\
&= \sum_{i=1}^{n} (1 + \frac{\partial V(x)}{\partial x_i}) \dot{x}_i \quad \left(\text{since } \sum_{i=1}^{n} \dot{x}_i = 0\right) \\
&= \sum_{i=1}^{n} (1 - \frac{x_i^*}{x_i})(q_i(x_i) - x_i \sum_{j=1}^{n} q_j(x_j)) \\
&= \sum_{i=1}^{n} (x_i - x_i^*) \left(\frac{q_i(x_i)}{x_i} - \sum_{j=1}^{n} q_j(x_j) \right) \\
&= \sum_{i=1}^{n} (x_i - x_i^*) \left(\frac{q_i(x_i)}{x_i} - \sum_{j=1}^{n} q_j(x_j^*) \right), \quad (14)
\end{aligned}$$

where (14) follows because $\sum_{i=1}^{n} (x_i - x_i^*) \sum_{j=1}^{n} q_j(x_j) = (1-1) \sum_{j=1}^{n} q_j(x_j) = 0 = (1-1) \sum_{j=1}^{n} q_j(x_j^*) = \sum_{i=1}^{n} (x_i - x_i^*) \sum_{j=1}^{n} q_j(x_j^*)$.

Recall that $q_i(x_i)$ is non-negative decreasing and $x_i > 0$, so that $q_i(x_i)/x_i$ is also decreasing. We will show that $x_i - x_i^*$ and $\frac{q_i(x_i)}{x_i} - \sum_{j=1}^{n} q_j(x_j^*)$ have opposite signs. In fact, suppose $x_i > x_i^*$, then $q_i(x_i)/x_i < q_i(x_i^*)/x_i^* = \sum_{j=1}^{n} q_j(x_j^*)$ (Recall that x^* is the fixed point of the ODE given of Eq. (10) which satisfies $x_i^* = q_i(x_i^*) / \sum_{j=1}^{n} q_j(x_j^*)$). Therefore, $\frac{dV(x)}{dt} < 0$ for $x \neq x^*$ and $\frac{dV(x^*)}{dt} = 0$.

Given that a unique solution exists, x^* is also asymptotically stable, and the ODE will converge to the unique fixed point satisfying,

$$\begin{aligned}
x_i^* &= \frac{q_i(x_i^*)}{\sum_{j=1}^{n} q_j(x_j^*)} \\
&= \frac{p_i(\frac{\epsilon}{n} + (1-\epsilon) x_i^*)}{\sum_{j=1}^{n} p_j(\frac{\epsilon}{n} + (1-\epsilon) x_j^*)}.
\end{aligned} \tag{15}$$

Thus, we have proved the result. It is not out place to remark that we have resorted to the decreasing monotonicity of the functions p_i to prove that the derivative of the Lyapunov function is negative by showing that it is the sum of terms of opposite signs. In the proof of Lemma 1, only strict monotonicity is needed to prove the uniqueness of the optimal solution.

□

We now make use of this ODE result within the proof of convergence of the two timescale discrete time algorithm.

Theorem 1. *$(\hat{p}(t), x(t))$ converges almost surely to $(q^\epsilon(x^*(\epsilon)), x^*(\epsilon))$ which, for small ϵ, approximates $(p(x^{opt}), x^{opt})$.*

Proof. Let i be arbitrary and fix $x_i = x_i(t)$. Double expectation gives

$$\begin{aligned}
E[(\hat{p}_i(t+1) - \hat{p}_i(t))/\alpha_t \,|\, \mathcal{H}_t] &= \\
E\left[E[(\hat{p}_i(t+1) - \hat{p}_i(t))/\alpha_t \,|\, \mathcal{H}_t, i(t)] \,|\, \mathcal{H}_t\right]
\end{aligned}$$

where $\mathcal{H}_t$ denotes the history up to immediately prior to action $i(t)$ being selected at time t.

Since $\hat{p}_i(t)$ is updated only if $i(t) = i$,

$$\begin{aligned}
E[(\hat{p}_i(t+1) - \hat{p}_i(t))/\alpha_t \,|\, \mathcal{H}_t, i(t) = j] &= \\
\mathbb{I}_{j=i} \left\{ E[v(t) \,|\, \mathcal{H}_t, i(t) = j] - \hat{p}_i(t) \right\} &= \\
\mathbb{I}_{j=i} \left\{ q_i^\epsilon(x_i) - \hat{p}_i(t) \right\}
\end{aligned}$$

where $\mathbb{I}_{j=i} = 1$ if $i = j$ and $\mathbb{I}_{j=i} = 0$ otherwise. Therefore

$$\begin{aligned}
E[(\hat{p}_i(t+1) - \hat{p}_i(t))/\alpha_t \,|\, \mathcal{H}_t] &= \\
P(i(t) = i \,|\, \mathcal{H}_t) \left\{ q_i^\epsilon(x_i) - \hat{p}_i(t) \right\} &= \\
\left\{ \tfrac{\epsilon}{n} + (1-\epsilon) x_i \right\} \left\{ q_i^\epsilon(x_i) - \hat{p}_i(t) \right\}
\end{aligned}$$

where $P(i(t) = i \,|\, \mathcal{H}_t) = \frac{\epsilon}{n} + (1-\epsilon) x_i$ by (9). Thus

$$(\hat{p}_i(t+1) - \hat{p}_i(t))/\alpha_t = \left\{ \tfrac{\epsilon}{n} + (1-\epsilon) x_i \right\} \left\{ q_i^\epsilon(x_i) - \hat{p}_i(t) \right\} + M_{t+1}$$

for some martingale difference sequence $\{M_t\}$.

We obtain the following ODE system, if (as is proscribed by the two-timescale approach [57]) the strategies are fixed at x_i:

$$\dot{\hat{p}}_i = \left(\frac{\epsilon}{n} + (1-\epsilon) x_i \right) (q_i^\epsilon(x_i) - \hat{p}_i). \tag{16}$$

Since x_i is fixed, and $\left(\frac{\epsilon}{n} + (1-\epsilon) x_i \right) > 0$, this system has a unique globally asymptotically stable fixed point at $\hat{p}_i = q_i^\epsilon(x_i)$.

Now we consider the slow dynamics, under the assumption that $\hat{p}_i$ is fully calibrated (i.e. $\hat{p}_i = q_i^\epsilon(x_i(t))$):

$$\begin{aligned}
x_i(t+1) &= x_i(t) + \theta_t [\hat{p}_i(t) - x_i(t) \sum_{j=1}^{n} \hat{p}_j(t)] \\
&= x_i(t) + \theta_t [q_i^\epsilon(x_i(t)) - x_i(t) \sum_{j=1}^{n} q_j^\epsilon(x_j(t))],
\end{aligned}$$

The corresponding ODE is simply that given in (10), so Proposition 1 tells us that the slow system also has a unique globally asymptotically stable fixed point. Applying the results of [59] shows that $(\hat{p}(t), x(t))$ converges almost surely to $(q(x^*(\epsilon)), x^*(\epsilon)$, where $x^*(\epsilon)$ satisfies (15). In Appendix A, we give a summary of the theory behind two-timescale framework and how it applies to our case.

We now show that $x^*(\epsilon)$ is continuous in ϵ, proving the final part of the theorem. Suppose not, so that there exists an ϵ_∞ and a sequence $\epsilon_m \to \epsilon_\infty$ such that $x^*(\epsilon_m) \not\to x^*(\epsilon_\infty)$. Since $x \in [0,1]^n$, a compact space, there exists a subsequence ϵ_{m_k} and $x^\infty \neq x^*(\epsilon_\infty)$ such that $x^*(\epsilon_{m_k}) \to x^\infty$. Now recall that $x^*(\epsilon)$ is the unique solution of $0 = Q_i^\epsilon(x) := q_i^\epsilon(x_i) - x_i \sum_j q_j^\epsilon(x_j)$ for all i, and note that the

Q_i^ϵ are continuous as functions of both ϵ and x. Hence for each i

$$Q_i^{\epsilon_\infty}(x_i^\infty) = \lim_{k\to\infty} Q_i^{\epsilon_{m_k}}(x_i^*(\epsilon_{m_k})) = \lim_{k\to\infty} 0 = 0.$$

By uniqueness of $x^*(\epsilon_\infty)$, this implies that $x^\infty = x^*(\epsilon_\infty)$, which leads to a contradiction. So $x^*(\epsilon)$ is continuous in ϵ. Note that this is an implicit function theorem [60], but the need to prove continuity at $\epsilon = 0$ on the boundary of the space precludes the use of standard implicit function theorems.

□

V. Experimental Results

In this Section, we compare the performance of the suggested algorithm with the algorithm of Papadimitriou et al. [13], [14]. We consider two different reward functions:

- **Linear**: For the linear case, we assume the following reward probabilities

$$p_i(x_i(t)) = \max\{0.7 - x_i(t)i, 0\},\ i = 1, 2, \ldots, n$$

- **Exponential**: For the exponential case, we assume the following reward probabilities

$$p_i(x_i(t)) = 0.7e^{-x_i(t)i},\ i = 1, 2, \ldots, n$$

We consider fixed steps sizes, meaning $\theta_t = \theta$, $\alpha_t = \theta$, and $\alpha << \theta$. We used $\epsilon = 0.001$. We considered the linear case for $n = 4$ materials and the exponential case for $n = 2$ and $n = 4$. We refer to the three cases as LINE4, EXP2, and EXP4. Furthermore, We consider both static and dynamic environments. Please note that the linear function is only decreasing and not strictly decreasing. The strictly monotonically decreasing condition is a sufficient but not necessary condition to ensure the uniqueness of the optimal point. Please note that the only part where we use the condition is when, by contradiction, we used the uniqueness of the optimal point. In our case, we have verified numerically that the fixed point is unique.

For the sake of completeness, we provide a description of Papadimitriou's algorithm [13], [14] which we use for comparison purposes in the experiments. In simple terms, Papdimitriou's algorithm runs an estimate of the reward probability for each action and polls that action in a proportional manner to its estimated reward. Initially, at time instant 0, we start from an initial probability vector that is in the simplex. The algorithm runs the following loop over time starting from arbitrarily initial estimates of the reward probabilities $\hat{p}_i(t)$, $1 \le i \le n$ that are in the unit interval.

1) Poll a random action $i \in [1,..,n]$, according to the probability

$$x_i(t) = \frac{\hat{p}_i(t)}{\sum_{j=1}^n \hat{p}_j(t)} \tag{17}$$

2) Update the reward probability for the polled action i, in same manner as (7).

A. Dynamic environment

In this experiment, we investigate situations where the feedback from the system varies with time. We consider three different cases:

1. Every $D = 500$ iteration the reward probabilities for the different materials are shuffled. More specifically, at every $D = 500$ iteration we draw a random permutation $\bar{n}_1, \ldots, \bar{n}_n$ of the material indexes $[1, 2, \ldots, n]$. The reward probability for amount x_i of material i, then becomes $p_{\bar{n}_i}(x_i)$. We refer to this case as SHORT.
2. The same as 1., with $D = 5000$. We refer to this case as LONG.
3. We now let D be a stochastic variable with the outcomes 500, 2000, and 10 000 with the probabilities

$$P(D = 500) = {}^{20}\!/_{26}$$
$$P(D = 2000) = {}^{5}\!/_{26}$$
$$P(D = 10000) = {}^{1}\!/_{26}$$

which means that the amount of time spent in a rapidly ($D = 500$), medium ($D = 2000$), and slowly ($D = 10\,000$) changing environment is equal. We refer to this case as RAND. The objective of this case is to investigate the extent to which our algorithm is robust and can handle environments that randomly alternate between rapid and slow changes.

In order to initialize our scheme, we use a flat Dirichlet distribution that ensures that the initial values are uniformly distributed in the simplex.

In a dynamic environment, the aim is to obtain polling probabilities as close as possible to the underlying optimal values in every iteration. For a given choice of the tuning parameters and for each of the three cases above, we ran a chain for $N = 10^6$ iterations. We measured the error using the RMSE.

$$L\left(\widehat{x'^*}, x^{opt}; \alpha\right) = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(x'_{ij} - x_i^{opt}\right)^2} \tag{18}$$

where x'_{ij} refers to the value of the chain at iteration j for material i and x^{opt} to the optimal values of the chains.

Let $p(\alpha)$ denote a probability distribution for our prior belief in the tuning parameter α in the algorithms. As a key performance indicator, we define the Bayesian expected loss

$$E_\alpha\left(L\left(\widehat{x'^*}, x^{opt}; \alpha\right)\right) = \int_0^1 L\left(\widehat{x'^*}, x^{opt}; \alpha\right) p(\alpha)\,\mathrm{d}\alpha \tag{19}$$

As described above, it is not possible to know beforehand which values for the tuning parameters will perform best, and we therefore computed the RMSE in (18) for a large set of values. We tested the following ratios: θ/α: 1/50, 1/20, 1/10, 1/5 and 1/3. Using higher ratios than 1/3 resulted in convergence problems since the material amounts sometimes became negative. Table I shows results under optimal values of the tuning

			Ratio				
		Pap.	1/50	1/20	1/10	1/5	1/3
SHORT	EXP2	4.76	4.67	4.69	4.79	4.93	5.01
	EXP4	3.02	2.90	2.98	3.03	3.13	3.17
	LINE4	4.07	4.53	4.05	4.11	4.26	4.37
LONG	EXP2	2.73	2.74	2.82	2.77	2.85	2.89
	EXP4	1.70	1.74	1.75	1.79	1.85	1.89
	LINE4	2.33	2.26	2.35	2.42	2.43	2.56
RAND	EXP2	3.05	3.08	3.11	3.12	3.22	3.23
	EXP4	1.90	1.90	1.93	1.96	2.03	2.08
	LINE4	2.64	2.51	2.58	2.60	2.72	2.76

TABLE I: Dynamic environment. RMSE multiplied by 100 under optimal choices of the tuning parameter α.

			Ratio				
		Pap.	1/50	1/20	1/10	1/5	1/3
SHORT	EXP2	23.70	5.28	5.10	5.69	7.11	8.82
	EXP4	14.12	3.16	3.52	4.35	5.79	7.43
	LINE4	17.19	6.12	5.03	4.82	5.49	6.84
LONG	EXP2	23.61	3.16	3.74	4.82	6.57	8.42
	EXP4	14.07	2.17	2.96	4.00	5.57	7.26
	LINE4	17.08	2.94	2.85	3.36	4.54	6.15
RAND	EXP2	23.63	3.48	3.92	4.93	6.63	8.46
	EXP4	14.08	2.30	3.02	4.04	5.59	7.28
	LINE4	17.08	3.44	3.10	3.52	4.64	6.22

TABLE II: Bayesian expected loss based on Equation (19) in dynamic environments. The Bayesian expected losses are scaled in the tables by a factor of 100.

parameter α. Figure 3 in the Appendix B shows additional results under all choices of the tuning parameter α. From Table I, we see that, under optimal values of the tuning parameters, Papadimitriou's algorithm and the suggested algorithm perform about equally well. However, from Figure 3, we see that the performance of Papadimitriou's algorithm is much more sensitive to the choice of the tuning parameter α than the suggested algorithm. This is further demonstrated in Table II, which shows the Bayesian expected loss on the assumption of no prior knowledge of α, i.e., $p(\alpha)$ in (19) is the uniform distribution on the $[0, 1]$ interval. We see that the suggested algorithm outperforms the algorithm of Papadimitriou by a large margin. Robustness is important since we usually have little knowledge of suitable values for α.

B. Static environment

In a static environment, we assume that the reward probabilities do not vary with time. In this section we demonstrate how the algorithm in this paper can also be used for such cases. The basic idea is to take the average over time instead of picking the last material estimates. In a static environment, the aim is to get as precise estimates of the true material amounts $x_1^{opt}, x_2^{opt}, \ldots, x_n^{opt}$ as possible within a given set of iterations N. If we use high values for the tuning parameters, α and θ, the estimates will rapidly converge towards the true material amounts, but the marginal variance of the chain after convergence will be high. If we use low values for the tuning parameters, the estimates will converge slowly toward the true estimates, but the marginal variance of the chain after convergence will be low. In other words, it is not easy to know which values of the tuning parameters will result in the least estimation error based on a set of iterations. In the experiments below, we therefore measured the estimation error for a large number of different choices of parameters. To estimate the true material amounts, we use two estimation strategies.

1. In the first strategy, we simply take the average of all the values of the material amounts in the chain up to iteration N. A disadvantage of this strategy is that we also include the values before convergence, resulting in estimation bias.
2. The second strategy uses a simple procedure to detect when a chain has converged by running two chains in parallel, starting with highly different initial estimates. For example if $n = 4$, we could start the first chain with the initial state $x_1 = 0.5$, $x_2 = 0$, $x_3 = 0.5$ and $x_4 = 0$ and start the second chain from $x_1 = 0.25$, $x_2 = 0.25$, $x_3 = 0.25$ and $x_4 = 0.25$. When the two chains have crossed at least once for every material, we assume that the chains have converged. Since we run two chains, to make the comparison with the first strategy fair, we only run each of the chains for $N/2$ iterations in this strategy. We estimate the true material amounts by taking the average of all the values from the two chains after convergence is detected.

For a given choice of the tuning parameters, we computed the estimation error using the root mean squared error measure (RMSE). In order to reduce the Monte Carlo error in the results, we repeated the experiment $M = 1000$ times. For the first strategy the RMSE becomes

$$L\left(\widehat{x'^*}, x^{opt}; \alpha\right) = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{1}{M}\sum_{k=1}^{M}\left(\widehat{x'^*_{ki}}(\alpha) - x_i^{opt}\right)^2} \quad (20)$$

where $\widehat{x'^*_{ki}}(\alpha)$ is the average of all the values of the amount of material i in the chain using the value α for the tuning parameter, i.e.

$$\widehat{x'^*_{ki}}(\alpha) = \frac{1}{N}\sum_{j=1}^{N} x_{ijk}$$

where x^*_{ijk} is the value of the chain from experiment k at iteration j for material i. Results under optimal choices of the tuning parameters are shown in Tables III and IV. Figures 1 and 2 in the Appendix B present additional results under all choices of the tuning parameter α. We start with the results based on the first estimation strategy (average of all values of the chain) shown in Figure 1 and Table III. From Figure 1, for $N = 1000$ iterations, and to some extent for $N = 10\,000$ iterations, using a ratio of $1/50$ results in a high estimation error for low values of α. Using a ratio of $1/3$ for the LINE4 case, we observe high estimation errors for high values of α. Overall, it seems that using ratios of $1/5$, $1/10$, $1/20$ results in good estimation performance for almost any choice of α. This is an important property since,

		Ratio					
		Pap.	1/50	1/20	1/10	1/5	1/3
$N = 1000$	EXP2	7.40	12.70	8.27	7.41	7.12	7.32
	EXP4	4.29	5.40	4.46	4.38	4.39	4.33
	LINE4	3.83	5.19	3.58	3.55	3.53	3.61
$N = 10\,000$	EXP2	2.30	2.46	2.25	2.27	2.29	2.27
	EXP4	1.34	1.39	1.38	1.37	1.37	1.37
	LINE4	1.04	1.01	0.97	0.99	0.98	0.96

TABLE III: Static environment. Estimation based on using all values of the chains. The table shows RMSE multiplied by 1000 under optimal choices of the tuning parameter α.

		Ratio					
		Pap.	1/50	1/20	1/10	1/5	1/3
$N = 1000$	EXP2	7.18	9.16	7.24	7.23	7.09	7.02
	EXP4	4.28	4.50	4.30	4.22	4.37	4.35
	LINE4	2.97	2.97	2.83	2.85	2.89	2.88
$N = 10\,000$	EXP2	2.29	2.16	2.21	2.22	2.13	2.22
	EXP4	1.31	1.33	1.33	1.33	1.33	1.35
	LINE4	0.88	0.86	0.86	0.87	0.87	0.86

TABLE IV: Static environment. Estimation based on using all values of the chains after convergence. The table shows RMSE multiplied by 1000 under optimal choices of the tuning parameter α.

in a real situation, it is hard to know which values of the tuning parameters are preferable. From Table III, we see that Papadimitriou's algorithm and the suggested algorithm perform about equally well under optimal choices of the tuning parameters. However the performance of Papadimitriou's algorithm is highly sensitive to the choice of α, which can be critical in a real situation since we do not know which value of α to use. In fact, by choosing a value of α that deviates slightly from the optimal value, the performance of Papadimitriou's algorithm drops dramatically. Inspecting the results for the second strategy (average after convergence), shown in Figure 2 and Table IV, we can draw many of the same conclusions. Using a ratio of $1/5$, $1/10$ or $1/20$ yields very robust results. Again, we see that the performance of Papadimitriou's algorithm is critically sensitive to the choice of α.

Tables V and VI summarize the Bayesian expected loss for the results in Figures 1 to 2. We usually have minimal knowledge about preferable values of α, and the results in Tables V and VI are computed on the assumption that the prior distribution $p(\alpha)$ is the uniform distribution on the interval $[0, 1]$. From the tables, we see that especially for $N = 10\,000$ iterations, the suggested method outperforms Papadimitriou's algorithm by a large margin. We also see that the suggested method performs a little better for strategy two (average of values after convergence) compared to strategy one (average of all values), while Papadimitriou's algorithm performs about equally well for the two strategies.

		Ratio					
		Pap.	1/50	1/20	1/10	1/5	1/3
$N = 1000$	EXP2	1.58	4.08	2.24	1.53	1.17	1.04
	EXP4	0.79	1.61	0.95	0.72	0.62	0.63
	LINE4	3.47	1.87	1.01	0.73	0.75	1.09
$N = 10\,000$	EXP2	1.35	0.72	0.43	0.33	0.31	0.36
	EXP4	0.63	0.30	0.20	0.18	0.22	0.31
	LINE4	3.42	0.31	0.18	0.19	0.41	0.86

TABLE V: Bayesian expected loss based on equation (19) in a static environment using estimation based on all values of the chain. Evaluation after $N = 1000$ and $N = 10\,000$ iterations. The Bayesian expected losses are scaled in the tables by a factor of 100.

		Ratio					
		Pap.	1/50	1/20	1/10	1/5	1/3
$N = 1000$	EXP2	1.56	2.86	1.65	1.20	0.97	0.92
	EXP4	0.79	1.16	0.75	0.60	0.55	0.59
	LINE4	3.46	1.92	0.98	0.68	0.69	1.07
$N = 10\,000$	EXP2	1.35	0.50	0.36	0.29	0.28	0.34
	EXP4	0.63	0.22	0.19	0.17	0.21	0.30
	LINE4	3.41	0.26	0.19	0.18	0.40	0.85

TABLE VI: Bayesian expected loss based on equation (19) in a static environment using estimation based on values after convergence. Evaluation after $N = 1000$ and $N = 10\,000$ iterations. The Bayesian expected losses are scaled in the tables by a factor of 100.

VI. Conclusion

In this article, we have presented an efficient and novel solution to the NSPP problem, which was first introduced by Papadimitriou and colleagues. Our solution outperforms the state-of-the-art solution by a large margin in terms of Bayesian expected loss. Furthermore, our solution is robust to the choice of the tuning parameter. Moreover, the paper advocates the concept of two-timescale LA, paving the way for more research on this novel type of LA design. In future work, we would like to generalize our solution to the NSPP problem to tackle the case of continuous random variables. Furthermore, designing a discretized LA algorithm to solve the NSPP problem is an interesting research direction that is worth pursuing.

References

[1] P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis, "Learning automata-based polling protocols for wireless lans," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 453–463, 2003.

[2] M. S. Obaidat, G. I. Papadimitriou, A. S. Pomportsis, and H. Laskaridis, "Learning automata-based bus arbitration for shared-medium atm switches," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 6, pp. 815–820, 2002.

[3] G. I. Papadimitriou, M. S. Obaidat, and A. S. Pomportsis, "On the use of learning automata in the control of broadcast networks: a methodology," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 6, pp. 781–790, 2002.

[4] G. I. Papadimitriou and A. S. Pomportsis, "Learning-automata-based tdma protocols for broadcast communication systems with bursty traffic," *IEEE Communications Letters*, vol. 4, no. 3, pp. 107–109, 2000.

[5] ——, "Self-adaptive tdma protocols for wdm star networks: A learning-automata-based approach," *IEEE Photonics Technology Letters*, vol. 11, no. 10, pp. 1322–1324, 1999.

[6] ——, "Self-adaptive tdma protocols: a learning-automata-based approach," in *Proceedings of the 1999 IEEE International Conference on Networks (ICON'99)*. IEEE, 1999, pp. 85–90.

[7] ——, "Dynamic bandwidth allocation in wdm passive star networks with asymmetric traffic," *Photonic Network Communications*, vol. 2, no. 4, pp. 383–391, 2000.

[8] P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis, "Distributed protocols for ad hoc wireless lans: a learning-automata-based approach," *Ad Hoc Networks*, vol. 2, no. 4, pp. 419–431, 2004.

[9] M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis, "An efficient adaptive bus arbitration scheme for scalable shared-medium atm switch," *Computer Communications*, vol. 24, no. 9, pp. 790–797, 2001.

[10] G. I. Papadimitriou and A. S. Pomportsis, "On the use of learning automata in medium access control of single-hop lightwave networks," *Computer Communications*, vol. 23, no. 9, pp. 783–792, 2000.

[11] O.-C. Granmo and B. J. Oommen, "Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata," *IEEE Transactions on Computers*, vol. 59, no. 4, pp. 545–560, 2010.

[12] ——, "Optimal sampling for estimation with constrained resources using a learning automaton-based solution for the nonlinear fractional knapsack problem," *Applied Intelligence*, vol. 33, no. 1, pp. 3–20, 2010.

[13] G. I. Papadimitriou and D. G. Maritsas, "Wdm passive star networks: receiver collisions avoidance algorithms using multifeedback learning automata," in *Local Computer Networks, 1992. Proceedings., 17th Conference on*. IEEE, 1992, pp. 688–697.

[14] ——, "Learning automata-based receiver conflict avoidance algorithms for wdm broadcast-and-select star networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 4, no. 3, pp. 407–412, 1996.

[15] A. Rezvanian, A. M. Saghiri, S. M. Vahidipour, M. Esnaashari, and M. R. Meybodi, "Learning automata theory," in *Recent Advances in Learning Automata*. Springer, 2018, pp. 3–19.

[16] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*. Prentice-Hall, Inc., 1989. [Online]. Available: http://portal.acm.org/citation.cfm?id=64802

[17] M. L. Tsetlin, "Finite automata and models of simple forms of behaviour," *Russian mathematical surveys*, vol. 18, no. 4, pp. 1–27, 1963.

[18] ——, *Automaton theory and modeling of biological systems*. New York: Academic Press, 1973.

[19] K. S. Narendra and M. A. Thathachar, "On the behavior of a learning automaton in a changing environment with application to telephone traffic routing," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 10, no. 5, pp. 262–269, 1980.

[20] K. S. Narendra and M. A. L. Thathachar, *Learning automata: an introduction*. Courier Corporation, 2012.

[21] K. S. Narendra and M. A. Thathachar, *Learning automata: an introduction*. Courier Corporation, 2012.

[22] O.-C. Granmo, B. J. Oommen, S. A. Myrer, and M. G. Olsen, "Learning automata-based solutions to the nonlinear fractional knapsack problem with applications to optimal resource allocation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 1, pp. 166–175, 2007.

[23] A. Abouzeid, O.-C. Granmo, C. Webersik, and M. Goodwin, "Socially fair mitigation of misinformation on social networks via constraint stochastic optimization," *arXiv preprint arXiv:2203.12537*, 2022.

[24] A. Yazidi, H. L. Hammer, and T. M. Jonassen, "Two-time scale learning automata: an efficient decision making mechanism for stochastic nonlinear resource allocation," *Applied Intelligence*, vol. 49, no. 9, pp. 3392–3405, Sep 2019. [Online]. Available: https://doi.org/10.1007/s10489-019-01453-0

[25] A. Yazidi and H. L. Hammer, "Solving stochastic nonlinear resource allocation problems using continuous learning automata," *Applied Intelligence*, vol. 48, no. 11, pp. 4392–4411, 2018.

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

[27] E. A. Billard, "Stability of adaptive search in multi-level games under delayed information," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 26, no. 2, pp. 231–240, 1996.

[28] S. M. Vahidipour, M. R. Meybodi, and M. Esnaashari, "Learning automata-based adaptive petri net and its application to priority assignment in queuing systems with unknown parameters," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 10, pp. 1373–1384, 2015.

[29] N. Baba and Y. Sawaragi, "On the learning behavior of stochastic automata under a nonstationary random environment," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 2, pp. 273–275, 1975.

[30] H. Morshedlou and M. R. Meybodi, "A new local rule for convergence of icla to a compatible point," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 12, pp. 3233–3244, 2017.

[31] P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis, "Using learning automata for adaptive push-based data broadcasting in asymmetric wireless environments," *IEEE Transactions on vehicular technology*, vol. 51, no. 6, pp. 1652–1660, 2002.

[32] G. I. Papadimitriou and A. S. Pomportsis, "Adaptive mac protocols for broadcast networks with bursty traffic," *IEEE Transactions on communications*, vol. 51, no. 4, pp. 553–557, 2003.

[33] V. L. Kakali, G. I. Papadimitriou, P. Nicopolitidis, and A. S. Pomportsis, "A new class of wireless push systems," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4529–4539, 2009.

[34] G. I. Papadimitriou and G. D. Pallas, "Performance evaluation of a new multiaccess protocol for local area networks," *Computer Communications*, vol. 26, no. 15, pp. 1800–1803, 2003.

[35] P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis, "Exploiting locality of demand to improve the performance of wireless data broadcasting," *IEEE Transactions on vehicular technology*, vol. 55, no. 4, pp. 1347–1361, 2006.

[36] ——, "A mac protocol for bursty traffic ad-hoc wireless lans with energy efficiency," *Wireless Personal Communications*, vol. 67, no. 2, pp. 165–173, 2012.

[37] ——, "Continuous flow wireless data broadcasting for high-speed environments," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 260–269, 2009.

[38] A. G. Sarigiannidis, P. Nicopolitidis, G. I. Papadimitriou, P. G. Sarigiannidis, M. D. Louta, and A. S. Pomportsis, "On the use of learning automata in tuning the channel split ratio of wimax networks," *IEEE Systems Journal*, vol. 9, no. 3, pp. 651–663, 2015.

[39] V. L. Kakali, P. G. Sarigiannidis, G. I. Papadimitriou, and A. S. Pomportsis, "A novel adaptive framework for wireless push systems based on distributed learning automata," *Wireless Personal Communications*, vol. 57, no. 4, pp. 591–606, 2011.

[40] D. Ye, M. Zhang, and A. V. Vasilakos, "A survey of self-organization mechanisms in multiagent systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, pp. 441–461, 2016.

[41] L. Bu, R. Babu, B. De Schutter *et al.*, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[42] P. Rakshit, A. Konar, P. Bhowmik, I. Goswami, S. Das, L. C. Jain, and A. K. Nagar, "Realization of an adaptive memetic algorithm using differential evolution and q-learning: a case study in multirobot path planning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 814–831, 2013.

[43] Y. Lv and X. Ren, "Approximate nash solutions for multiplayer mixed-zero-sum game with reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[44] A. Yazidi, I. Hassan, H. L. Hammer, and B. J. Oommen, "Achieving fair load balancing by invoking a learning automata-based two-time-scale separation paradigm," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[45] A. Yazidi, H. L. Hammer, and T. M. Jonassen, "Two-time scale learning automata: an efficient decision making mechanism for stochastic nonlinear resource allocation," *Applied Intelligence*, vol. 49, no. 9, pp. 3392–3405, 2019.

[46] P. Laforgue, G. Clerici, N. Cesa-Bianchi, and R. Gilad-Bachrach, "A last switch dependent analysis of satiation and seasonality in bandits," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 971–990.

[47] L. Leqi, F. Kilinc Karzan, Z. Lipton, and A. Montgomery, "Rebounding bandits for modeling satiation effects," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4003–4014, 2021.

[48] X. Wang, H. Xie, and J. C. Lui, "Multiple-play stochastic bandits with shareable finite-capacity arms," in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 181–23 212.

[49] X. Wang, H. Xie, and J. Lui, "Multi-player multi-armed bandits with finite shareable resources arms: Learning algorithms & applications," *arXiv preprint arXiv:2204.13502*, 2022.

[50] O. Papadigenopoulos, C. Caramanis, and S. Shakkottai, "Non-stationary bandits under recharging payoffs: Improved planning with sublinear regret," *arXiv preprint arXiv:2205.14790*, 2022.

[51] R. Kleinberg and N. Immorlica, "Recharging bandits," in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2018, pp. 309–319.

[52] D. Simchi-Levi, Z. Zheng, and F. Zhu, "Dynamic planning and learning under recovering rewards," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9702–9711.

[53] W. Tang, C.-J. Ho, and Y. Liu, "Bandit learning with delayed impact of actions," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 804–26 817, 2021.

[54] C.-Y. Chang, G.-J. Yu, T.-L. Wang, and C.-Y. Lin, "Path construction and visit scheduling for targets by using data mules," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 10, pp. 1289–1300, 2014.

[55] X. Chen, "Police patrol optimization with security level functions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 5, pp. 1042–1051, 2013.

[56] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[57] V. S. Borkar, "Stochastic approximation with two time scales," *Systems & Control Letters*, vol. 29, no. 5, pp. 291–294, 1997.

[58] A. Benveniste, P. Priouret, and M. Métivier, *Adaptive Algorithms and Stochastic Approximations*. New York, NY, USA: Springer-Verlag New York, Inc., 1990.

[59] D. S. Leslie and E. Collins, "Convergent multiple-timescales reinforcement learning algorithms in normal form games," *The Annals of Applied Probability*, vol. 13, no. 4, pp. 1231–1251, 2003.

[60] S. G. Krantz and H. R. Parks, *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.

[61] M. Benaïm, "Dynamics of stochastic approximation algorithms," in *Seminaire de probabilites XXXIII*. Springer, 1999, pp. 1–68.

Acknowledgment

This work was supported by the Andalusian Government through the project P20 00673 and by the project PID2019- 103880RB-I00 funded by MCIN / AEI / 10.13039/501100011033.

Appendix A
Two-timescale framework

Theorem 5 of [59] presents the results of [57] in a more useful framework for our context. It considers a system:

$$\theta^{(j)}_{n+1} = \theta^{(j)}_n + \lambda^{(j)}_n \left\{ F^{(j)}(\theta^{(1)}_n, \theta^{(2)}_n) + M^{(j)}_{n+1} \right\}$$

for $j = 1, 2$, with $F^{(j)}$ globally Lipschitz, the $\{\theta^{(j)}_n\}$ sequences bounded, $\sum \lambda_n = \infty$ and $\sum \lambda_n^2 < \infty$, with the M_n martingale difference sequences (so that $\sum \lambda_n M_{n+1}$ converges a.s.), and $\lambda^{(1)}_n / \lambda^{(2)}_n \to 0$ as $n \to \infty$ (the two-timescales assumption). If for each fixed $\theta^{(1)}$ the ODE $\dot{Y} = F^{(2)}(\theta^{(1)}, Y)$ has a globally attracting fixed point $\xi(\theta^{(1)})$ where ξ is Lipschitz, then (almost surely):

$$\|\theta^{(2)}_n - \xi(\theta^{(1)}_n)\| \to 0$$

and an interpolation of the iterates $\theta^{(1)}_n$ is an asymptotic pseudotrajectory of the ODE:

$$\dot{X} = F^{(1)}(\theta^{(1)}_n, \xi(X)). \tag{21}$$

Results from Benaïm [61], culminating with Proposition 6.4, tell us that if there is a Lyapunov function for a unique fixed point of this fast timescale ODE (21) then the iterates $\{\theta^{(1)}_n\}$ converge to the fixed point.

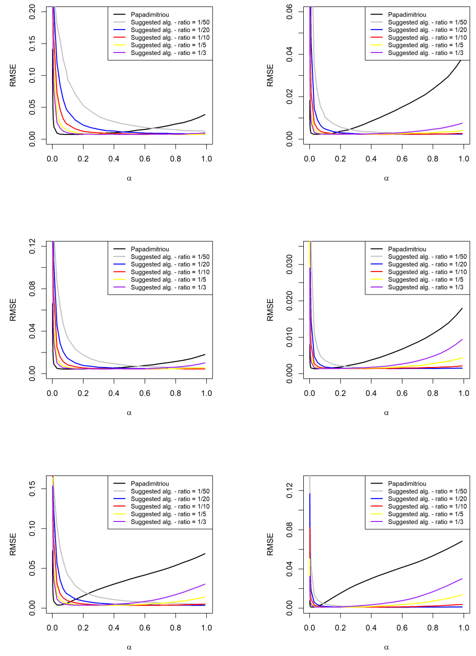

Fig. 1: Static environment based on using all values of the chains. The left and the right refer to evaluation after $N = 1000$ and $N = 10\,000$ iterations, respectively. The rows from top to bottom refer to the cases EXP2, EXP4 and LINE4, respectively.

In our article we map $\theta^{(2)}$ to the vector of $\hat{p}$ values, $\theta^{(1)}$ to the vector of x values, $\lambda^{(1)}$ to θ and $\lambda^{(2)}$ to α. The conditions on α_t and θ_t immediately following Eq. (8) ensure they meet the requirements of the general theory. Furthermore, by the assumption that p_i are Lipschitz function of x_i, and using the fact that the composition of two Lipschitz functions is Lipschitz, and the sum of Lipschitz functions is again Lipschitz, we deduce that $q_i(x_i) - x_i \sum_j q_j(x_j)$ is Lipschitz function x_i. We can also see that $(\epsilon/n + (1-\epsilon)x_i)(q_i(x_i) - \hat{p}_i)$ is Lipschitz function of $\hat{p}_i$ since any linear function is Lipschitz.

The proof given of Theorem 1 verifies the convergence of the requisite ODEs.

Appendix B
Supplementary Experimental Results

Please recall the experiments leading to the results in Tables III and IV. Figures 1 and 2 show estimation error under all possible values of α.

Please recall the experiments leading to the results in Table I. Figures 3 in Appendix B show tracking error under all possible values of α.

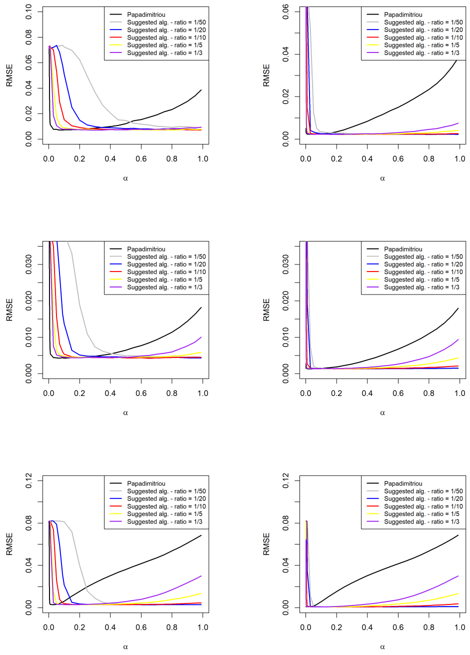

Fig. 2: Static environment based on using values of the chains after convergence. The left and the right column refer to evaluation after $N = 1000$ and $N = 10\,000$ iterations, respectively. The rows from top to bottom refer to the cases EXP2, EXP4 and LINE4, respectively.

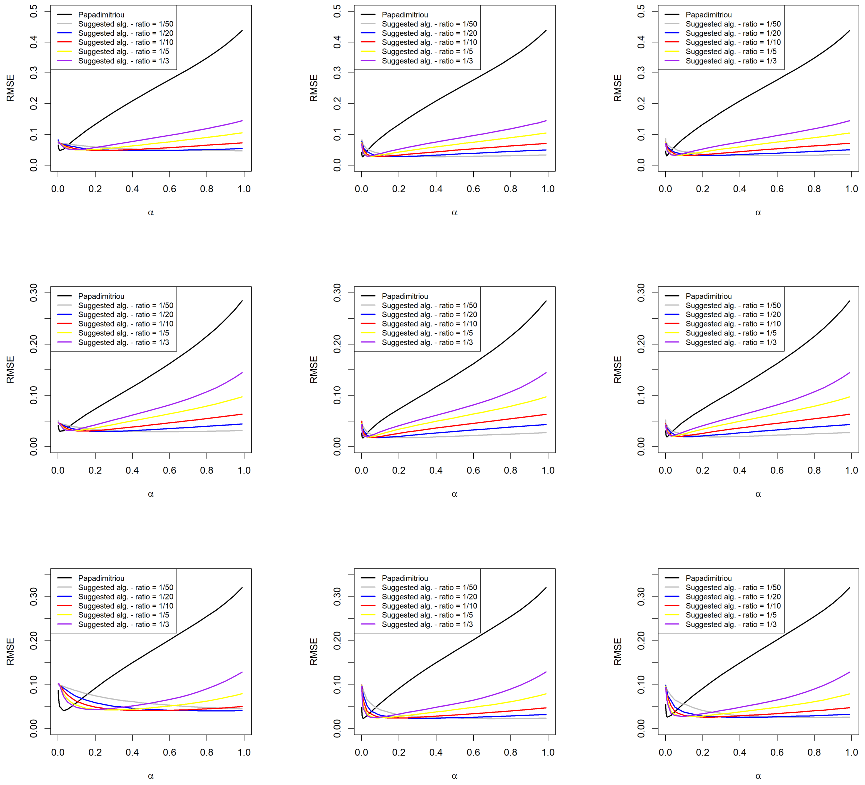

Fig. 3: Dynamic environment. RMSE under different choices of the tuning parameters. The columns from left to right refer to the cases SHORT, LONG and RAND, respectively. The rows from top to bottom refer to the cases EXP2, EXP4 and LINE4, respectively.