Noname manuscript No.
(will be inserted by the editor)

Two-time Scale Learning Automata: An Efficient Decision Making Mechanism for Stochastic Nonlinear Resource Allocation

Anis Yazidi, Hugo L. Hammer and Tore M. Jonassen

Received: date / Accepted: date

Abstract The Stochastic Non-linear Fractional Equality Knapsack (NFEK) problem is a substantial resource allocation problem which admits a large set of applications such as web polling under polling constraints, and constrained estimation. The NFEK problem is usually solved by trial and error based on noisy feedback information from the environment. The available solutions to NFEK are based on the traditional family of Reward-Inaction Learning Automata (LA) scheme where the action probabilities are updated based on *only* the last feedback. Such an update form seems counterproductive for two reasons: 1) it only uses the last feedback and does not consider the whole history of the feedback and 2) it ignores updates whenever the last feedback does not correspond to a reward. In this paper, we rather suggest instead a learning solution that resorts to the whole history of feedback using the theory of two time-scale separation. Through comprehensive experimental results we show that the proposed solution is not only superior to the state-of-the-art in terms of peak performance but is also robust to the choice of the tuning parameters.

Keywords Decision Making Under Uncertainty, Continuous Learning Automata, Two-Time Scale, Stochastic Non-linear Fractional Equality Knapsack, Resource Allocation.

A very preliminary conference version of this work appeared in IEA/AIE 2017, the 30th International Conference on Industrial, Engineering, Other Applications of Applied Intelligent Systems, held in Paris, June 2017. Prof. Tore Jonassen passed away on February 04, 2018 and the authors dedicate this manuscript to his memory.

Anis Yazidi
Dept. of Computer Science
Oslo Metropolitan University
E-mail: anis.yazidi@oslomet.no

1 Introduction

In this article, we deal with the Stochastic Non-linear Fractional Equality Knapsack (NFEK) Problem which is the central underlying problem pertinent to allocating resources based on incomplete and noisy information. Such situations are not merely hypothetical – rather, they constitute the vast majority of allocation problems in the real-world. Resource allocation problems which involve such incomplete and noisy information are particularly intriguing. They cannot be solved by traditional optimization techniques, rendering them ineffective.

The Stochastic NFEK Problem has attracted research attention due to its ability to provide a model for some real-life problems under uncertainty where the dynamics of the environments are affected by the actions of the decision maker. In [5, 6], the web polling problem was addressed and modelled as *Stochastic* NFEK. The aim was to maximize the number of changes detected given limited polling capacity. The frequency of changes of the web pages are supposed to be unknown. In [5, 6], it was shown that the probability to uncover an update monotonically decreases as the polling probability of the web page increases. In the literature, there is a large class of multi-armed bandit problems that can be modelled as *Stochastic* NFEK Problem where the reward probability decreases as the probability of polling the arm increases. Examples of those problems include congestion monitoring under limited bandwidth [1], adaptive link monitoring in software defined networks [10] and dynamic probing for intrusion detection under resource constraints [8].

In a nutshell, the Stochastic NFEK problem has two main characterizing facets, namely, that the unit volume values of each material are *stochastic* variables with *unknown* distributions, and that the expected value of a material could decrease as additions are made to the knapsack.

Few optimal solutions to the NFEK problem have been suggested in the literature. The first reported optimal solution [5,6] utilizes a hierarchy of two-action discretized Learning Automata (LA). Although this solution is elegant, its implementation is complex because it involves updates at different levels of a balanced binary tree. It is worth mentioning that the latter work represents the first optimal solution reported in the literature to the Stochastic NFEK problem. A subsequent solution was recently reported in [25] and was devised by the authors of the current manuscript. The latter solution tries to reduce the complexity of the aforementioned hierarchical solution [5, 6]. It is based on the traditional family of Reward-Inaction Learning Automata ($LR-I$). The $LR-I$ based solution [25] reckoned as the Continuous Multi-action Learning Automata Solution (CMLS) can be seen as the counter-part solution of the work [5, 6] but in continuous probability space rather than in discretized space. Moreover, both legacy solutions hierarchical solution [5, 6] and CMLS [25] fall under the family of Reward-Inaction LA which means that the actions probabilities are only updated in case of Reward. Therefore CMLS solution does not use the whole history of feedback from the environment and rather incorporates only the last feedback in case of reward. In

simple terms, no update is performed whenever the last feedback does not correspond to a reward and thus the action probabilities are left unchanged. Such an update form seems counterproductive as very limited information is used: 1) only information based on only the last feedback 2) and whenever the last feedback is penalty no update is performed. In order to circumvent those advantages, we propose in this paper a solution that resorts the whole feedback history from each action. As we will observe in the experimental results, our solution yields higher performance. The theoretical fundamentals of our solution are based on the theory of two-time scale separation.

The contributions of this paper are the following:

- We propose a solution called the Two-Time Scale based Learning Automata Solution (TTS-LA) which introduces the concept of two-time scale to the field of LA. According to our solution, the polling probabilities are updated on the "slower time scale" while the feedback is estimated on a "faster time scale". We hope that this concept can boost more research interest in this type of LA design specially for non-stationary environments.
- The proposed two-time scale paradigm exploits much more information than the classical NFEK solutions proposed in the literature [5, 6, 25]. In fact, the available solutions to the NFEK problem fall both under the class of under the family of Reward-Inaction LA which means that the actions probabilities are only updated in case of reward. In contrast, our solution not only uses both reward and penalty to update the action probabilities but also the whole feedback history.
- We prove that the TTS-LA is is asymptotically optimal based on the theory of stochastic approximation [2].
- Our experimental results show TTS-LA is not only robust to the choice of the tuning parameters but also superior to legacy solution in terms of accuracy.

The paper is organized as follows. In Section 2 we survey the state-of-the-art solutions to the Stochastic NFEK problem. In Section 3 we present the novel TTS-LA solution to the problem, and prove its asymptotic optimality. We proceed in Section 4 to empirically verify that the TTS-LA solution provides superior convergence results to H-TRAA and CMLS. Furthermore, we also compare our proposed solution to two legacy heuristic solutions: proportional LA and LAKG. Finally, we conclude the paper in Section 5.

2 Stochastic NFEK: State-of-the-Art

In this Section, we survey the state-of-the-art solutions to the Stochastic NFEK problem. It is worth mentioning that the only available legacy solutions [5, 6, 25] as well as the current proposed solution are based on the principles of Learning Automata (LA) [11]. LA are stochastic machines that have been used to model biological systems [24]. They have attracted considerable interest in the last few decades because they are able to learn the optimal ac-

tions when operating in (or interacting with) unknown stochastic environments. Furthermore, they combine rapid and accurate convergence with low computational complexity. The theory of LA has found numerous applications in the field of computer science. One of the most recent applications of LA include sampling algorithms for stochastic graphs [19], trust propagation in online social networks [22], allocation hub location problem [4], selecting caching nodes in delay tolerant networks [9] and feature subset selection [23] to mention a few. For an updated overview over the theory and applications of LA we refer the reader to the following book [20] and to a recent special issue [21] dedicated to the applications of LA.

The state-of-the-art scheme for hierarchically solving n-material problems [5, 6] involves a primitive module, namely the Twofold Resource Allocation Automaton (TRAA) for the *two-material* problem. This module has been proven to be asymptotically optimal. The authors of [5, 6] then used the primitive TRAA as a building block, and arranged a set of TRAAs in a hierarchy so as to solve *multi-material* Stochastic NFEK Problems.

The hierarchy of TRAAs, referred to as H-TRAA, assumes that $n = 2^\gamma, \gamma \in \mathbb{N}^+$. If the number of materials is less than this, one trivially assumes the existence of additional materials whose values are "zero", and which are, thus, not able to contribute to the final optimal solution. The hierarchy is organized as a balanced binary tree with depth $D = \log_2(n)$. Each node in the hierarchy can be related to three entities: (1) a set of materials, (2) a partitioning of the material set into two subsets of equal size, and (3) a dedicated TRAA that allocates a given amount of resources among the two subsets. At depth D, then, each individual material can be separately assigned a fraction of the overall capacity by way of recursion, using a subtle mechanism described, in detail, in [6]. The principal theorem that guarantees the convergence of the H-TRAA [5, 6] has cleverly shown that if all the individual TRAAs converge to their *local* optimum, the overall system attains to the global optimum.

The CMLS method [25] can be seen as the counter part of the discretized H-TRAA solution but in continuous probability space. The update form of the CMLS is based on the principles of the Linear Reward-Inaction (L_{RI}) scheme. The main difference with classical L_{RI} is that the CMLS solution enforces a minimal value on the action probability that is strictly positive. Therefore, the CMLS introduces artificial barriers that have the effect that they prevent the instantaneous allocation's probability vector from getting trapped in a n exclusive choice of one of the actions.

The LA Knapsack Game (LAKG) proposed in [7] is considered the first reported solution in the literature to the NFEK problem. The amount x_i material i is defined in the LAKG approach as $x_i(t) = s_i(t)^\gamma / N^\gamma$ where $s_i(t)$ takes values from the set of $N+1$ values: $\{1, 2, \ldots, N-1, N\}$. Here N controls the resolution of the scheme while $\gamma > 0$ controls the non-linearity of the discretized solution space. It is easy to observe that $x_i(t)$ takes values from $\{1/N^\gamma, 2^\gamma/N^\gamma, \ldots, (N-1)^\gamma/N^\gamma, 1\}$ where the value 0 is excluded in order to ensure that each material will be chose with non zero probability. Furthermore, each material i is accessed with a probability proportional to x_i by way

of normalization. We should underline that LAKG is a heuristic solution for solving the Stochastic NFEK problem in contrast to the H-TRAA and CMLS solutions which are proven to be asymptotically optimal.

When it comes to resource polling under noisy environment with incomplete information as in the settings of NFEK, an intuitive strategy is to allocate resources to materials according to the concept of proportionality. In this perspective, Papadimitriou and his collaborators [12–16, 18] proposed a proportional allocation strategy that found a large set of applications and which operates according to two in tandem operations: one involving updating the reward probability and the other involving choosing the next resource to be polled. Whenever a resource is accessed, based on the feedback from the environment, the reward probability is estimated according to some LA inspired incremental update form. Subsequently, the next resource to be accessed is chosen with a probability that it is proportional to its estimated reward probability. However, the latter proportional solution is not guaranteed to converge to the optimal allocation vector.

3 A Two-Time Scale Solution to Resource Allocation

The Stochastic NFEK problem involves n materials, $1 \leq i \leq n$, where each material is available in a certain amount $x_i \leq b_i$. Let $h_i(x_i)$ denote the value of the amount x_i of material i [26]. The problem is to fill a knapsack of fixed volume c with the material mix $\mathrm{x} = [x_1, \ldots, x_n]$ of maximal value $\sum_1^n h_i(x_i)$ [3]. The material value per unit volume for any x_i is a *probability* function $p_i(x_i)$, and to render the problem non-trivial, the distribution of $p_i(x_i)$ is assumed to be *unknown*.

From this perspective, the expected value of the amount x_i of material i, $1 \leq i \leq n$, is given by $h_i(x_i) = \int_0^{x_i} p_i(u)du$[1]. At each time instant, an amount x_i of material i is placed in the knapsack. The complexity of the problem arises because we are only allowed to observe an instantiation of $p_i(x_i)$ at x_i, and not $p_i(x_i)$ itself. The solution should be able to converge to a mixture of the materials of maximal *expected* value, through a series of informed guesses.

The rest of this section is dedicated to presenting the design and the theoretical fundaments of our TTS-LA solution that optimally solves the Stochastic NFEK problem. For the sake of clarity, we separate the presentation of the algorithm from the formal analysis. Section 3.1 describes the TTS-LA algorithm while Section 3.2 presents the theoretical results together with their corresponding formal analysis.

3.1 Description of the TTS-LA Solution

The *Stochastic Environment* for the n materials case can be characterized by:

[1] We hereafter use $h_i'(x_i)$ to denote the derivative of the expected value function $h_i(x_i)$ with respect to x_i. Accordingly, the expected value per unit volume of material i becomes $h_i'(x_i) = p_i(x_i)$.

1. The capacity $c = 1$ of the knapsack;
2. $n - material$ unit volume value probability functions $[p_1(x_1), \ldots, p_n(x_n)]$.

Note that we allow only instantiations of the material values per unit volume to be observed. In brief, if the amount x_i of material i is suggested to the Stochastic Environment, the Environment replies with a unit volume value $v_i = 1$ with probability $p_i(x_i)$ and a unit volume value $v_i = 0$ with probability $1 - p_i(x_i)$. To render the problem both interesting and non-trivial, we assume that $p_i(x_i)$ is unknown to the LA.

The Stochastic NFEK problem is described formally as:

$$\text{maximize} f(\mathrm{x}) = \sum_{1}^{n} h_i(x_i), \tag{1}$$

$$\text{where } h_i(x_i) = \int_0^{x_i} p_i(u)du, \text{ and } p_i(x_i) = h_i'(x_i), \tag{2}$$

$$\text{subject to} \sum_{1}^{n} x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0. \tag{3}$$

Before, we proceed to the solution, we shall characterize the optimal solution of the Stochastic NFEK problem.

Lemma 1 *The material mix* $\mathrm{x} = [x_1, \ldots, x_n]$ *is a solution to a given Stochastic NFEK Problem if (1) the derivatives of the expected material amount values are all equal at* x*, (2) the mix fills the knapsack, and (3) every material amount is positive, i.e.:*

$$h_1'(x_1) = \cdots = h_n'(x_n)$$
$$\sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0.$$

The above lemma is based on the well-known principle of Lagrange Multipliers, and its proof is therefore omitted here for the sake of brevity [5,6].

The idea behind our TTS-LA is to resort to a two-time scale based approach, where the polling probabilities x_i are updated on the "slower time scale" while $p_i(x_i)$ are estimated on a "faster time scale". In practice, the updating parameter (in this case θ) used for updating the probabilities x_i should be much smaller than the corresponding updating parameter λ for the task of estimation of the p_i. Thus, we can say that the fast-evolving dynamics of p_i sees x_i as almost constant, while the slowly evolving dynamics of x_i sees p_i as almost equilibriated [2][2].

We denote the decision variable for selecting an action at time instant t, $R(t)$ that is, for $i \in [1..n]$. We say that the event $\{R(t) = i\}$ has occurred if the action i is polled.

[2] Another possible manner to implement a two-time scale approach is to execute one update on the slower time scale loop for every few iterations on the faster time scale loop, i.e., the slower time scale loop is run less frequently.

Once the action i is polled, the estimate $\hat{p_i}(t+1)$ of the reward probability of the polled action is immediately updated using an adaptive estimator, namely exponential moving average:

$$\hat{p_i}(t+1) = \hat{p_i}(t) + \lambda(v_i(t) - \hat{p_i}(t)) \tag{4}$$

where $v_i(t)$ is a random variable that takes a value 1 with $p_i(x_i(t))$ and 0 with $1 - p_i(x_i(t))$.

The reward estimates for the other actions are left unchanged, i.e.,

$$\hat{p_j}(t+1) = \hat{p_j}(t) \text{ for } j \neq i, j \in [1, n] \tag{5}$$

Now, we proceed to presenting the update equations for the polling probabilities x_i for $i \in [1..n]$.

$$x_i(t+1) = x_i(t) - \theta \left(\frac{1}{n} \sum_{k=1}^{n} \hat{p_k}(t) - \hat{p_i}(t) \right) \tag{6}$$

To start with, we initialize the probabilities of all actions at time 0 to $x_i(0) = \frac{1}{n}$ for $1 \leq i \leq n$. In the absence of prior information about the estimates of the reward probabilities, i.e., the $\hat{p_i}$'s, can be merely initialized to 0.5. For the sake of clarity, we give an algorithmic description of the TTS-LA in Algorithm 1. As any LA algorithm, we proceed in step 1 of Algorithm 1 to polling an action according to the action probability vector. We suppose that the chosen action is the one with index i. Once this action is chosen, the Environment returns either reward or penalty. In step 2, we update the estimate of the reward probability $\hat{p_i}$ of the chosen action using an exponential moving average update. In simple terms, $\hat{p_i}$ is either incremented or decremented depending on whether the feedback was a reward or penalty respectively. The estimates for rest of the actions with index $j \neq i, j \in [1, n]$ are not updated and thus they are left unchanged. In the next step 3, we update recursively the polling probabilities for all actions by directly incorporating the reward estimates into the update formula. The update form is designed such that, for an action j, the polling probability $x_j(t+1)$ at the next time step is increased or decreased depending on whether the corresponding reward estimate $\hat{p_j}(t)$ at the previous time step t is smaller or larger than the average of the instantaneous reward estimates of all actions given by $\frac{1}{n} \sum_{i=1}^{n} \hat{p_i}(t)$. In this manner, the update form tries to equalize the $\hat{p_j}(t)$'s and the quantity $\frac{1}{n} \sum_{i=1}^{n} \hat{p_i}(t)$. As a consequence, the the polling probabilities will converge to a point for which all the reward estimate probabilities are equal to the latter fixed quantity as time proceeds and for certain conditions on the update parameters guaranteeing two-time scale separation as will be seen in Theorem 1.

Algorithm 1 The Two-Time Scale based Learning Automata Solution (TTS-LA)

Loop

1. Poll an action at time instant t according to the probability vector $[x_1, x_2, \ldots, x_n]$, suppose $R(t) = i$, observe $v_i(t)$.

2. Updating the reward estimates.

- Update reward estimate of the chosen action:

$$\hat{p_i}(t+1) = \hat{p_i}(t) + \lambda(v_i(t) - \hat{p_i}(t))$$

- The reward estimates for the other actions are kept unchanged, i.e.,

$$\hat{p_j}(t+1) = \hat{p_j}(t) \text{ for } j \neq i, j \in [1, n]$$

3. Update the polling probabilities for the next time instant $t+1$ according to:

$$x_1(t+1) = x_1(t) - \theta \left(\frac{1}{n} \sum_{i=1}^{n} \hat{p_i}(t) - \hat{p_1}(t) \right)$$

$$x_2(t+1) = x_2(t) - \theta \left(\frac{1}{n} \sum_{i=1}^{n} \hat{p_i}(t) - \hat{p_2}(t) \right)$$

$$\vdots$$

$$x_n(t+1) = x_n(t) - \theta \left(\frac{1}{n} \sum_{i=1}^{n} \hat{p_i}(t) - \hat{p_n}(t) \right)$$

3.2 Theoretical Results and Formal Analysis

In the previous subsection, we have presented the the details of the TTS-LA algorithm. In this subsection, we proceed to characterizing the convergence of the TTS-LA paradigm. We provide the main theorem of this paper documenting the convergence of the TTS-LA to the solution of the Stochastic NFEK problem.

Theorem 1 *Algorithm 1 describing the update equations for $x_i(t)$ and the updates equations for $\hat{p_i}(t)$, for $1 \leq i \leq n$, converges to the fixed point characterized by:*

$$p_i(x_i^*) = p_j(x_j^*),\ i \neq j, \text{ for } i \in [1, n]$$

For $\theta > 0$ much smaller that λ and for $\lambda \to 0$.

According to Theorem 1 presented above, the TTS-LA algorithm given in Algorithm 1 converges asymptotically to the optimal solution of Stochastic NFEK characterized by Lemma 1. The convergence takes place whenever θ is much smaller that λ and for λ approaching zero. Since θ is much smaller that

λ, the x_i's evolve at a slower time scale compared to $\hat{p}_i$'s and thus the two-time scale separation. At this juncture, we shall proceed to prove the theorem. As a part of the proof, we will give insights into the intuition behind invoking the two-time scale separation in the update form of the the TTS-LA algorithm and how this leads to convergence to an optimal fixed point.

Proof The proof of Theorem 1 is based on the two-time scale approach [2] and includes two main steps. In the first step, we investigate the convergence of the reward estimates using two-time scale separation technique. In the second step, we establish the convergence of action probabilities using a deterministic ODE equation.

Part 1 of the proof We proof that for $1 \leq i \leq n$, $\hat{p}_i(t)$ converges to $p_i(x_i(t))$.

The proof of the two-time scale approach is based on the theory of stochastic approximation [2]. Let us consider the reward estimation scheme. We can write, for a positive integer M and for $1 \leq i \leq n$

$$\hat{p}_i(t+M) = \hat{p}_i(t) + \lambda \sum_{k=0}^{M-1} I_{\{R(t+k+1)=1\}}(v_i(t+k) - \hat{p}_i(t+k))$$

where $v_i(t)$, for $1 \leq i \leq n$ is binomial random variable that takes value 1 with probability $p_i(t)$ and 0 with probability $1 - p_i(t)$.

Whenever λ is small enough, the vector $[\hat{p_1}(t), \hat{p_2}(t), ..., \hat{p_N}(t)]$ is assumed to remain almost unchanged in the discrete interval $\{t, t+1, \ldots, t+M\}$. Thus, we can write the following approximate equations for $1 \leq i \leq n$

$$\hat{p}_i(t+M) \approx \hat{p}_i(t) + M\lambda(S_i(t,M) - Q_i(t,M)\hat{p}_i(t)) \tag{7}$$

For $i \in [1, n]$ when the values of estimates $\hat{p_1}(.), \hat{p_2}(.), \ldots, \hat{p_n}(.)$ are considered fixed at $\hat{p_1}(t), \hat{p_2}(t), \ldots, \hat{p_n}(t)$, and M is large, we will try to approximate the quantities

$$S_i(t,M) = \frac{\sum_{k=0}^{M-1} I_{\{R(t+k+1)=i\}} v_i(t+k)}{M}$$

as well as

$$Q_i(t,M) = \frac{\sum_{k=0}^{M-1} I_{\{R(t+k+1)=i\}}}{M}$$

The probability vector $x_1(.), x_2(.), \ldots, x_n(.)$, too, can be regarded essentially constant in the interval $\{t, t+1, \ldots, t+M\}$, because we supposed that x_i evolves at slower time scale compared to $\hat{p}_i$ and since the probabilities are continuous functions of the reward estimates. Note that the fact that θ is much smaller that λ permits the separation in time scale. The informed reader observes that by virtue of the two-time scale separation we are able to extend the deterministic NFEK solution presented in [26] to solve the Stochastic NFEK problem.

Now, assuming that M is large enough such that the law of large numbers takes place, the average $Q_i(t, M) = \frac{\sum_{k=0}^{M-1} I_{\{R(t+k+1)=i\}}}{M}$, which is the fraction of time the action i was chosen in the interval $[t, t+M]$ converges to $x_i(n)$.

With the actions probabilities fixed, the reward processes $v_i(.)$, can converge to a stationary distribution, with the mean being denoted by $p_i(x_i(n))$.

Further, the quantities $S_i(t, M) = \frac{\sum_{k=0}^{M-1} I_{\{R(t+k+1)=i\}} v_i(t+k)}{M}$ can be approximated by $x_i(n)p_i(x_i(t))$.

Employing the approximations as described above, we notice from Eq. (7) that the evolution of the vector $[\hat{p_1}(.), \hat{p_2}(.), \ldots, \hat{p_n}(.)]$ reduces to the following ODE system when λ is small enough:

$$\frac{\hat{p_i}(t)}{dt} = x_i(t).(p_i(x_i(t)) - \hat{p_i}(t)) \tag{8}$$

Eq. (8), reduces to having the reward estimates $[\hat{p_1}(.), \hat{p_2}(.), \ldots, \hat{p_n}(.)]$ converging to a steady state $[p_1(x_1(t)), p_2(x_2(t)), p_n(x_n(t))]$ whenever λ tends to 0. This ends the first part of the proof.

Part 2 of the proof Therefore using the ODE approximation obtained in the first part of the proof, we can approximate the system of equations (6) using a deterministic ODE equation:

$$x_i(t+1) = x_i(t) - \theta \left(\frac{1}{n} \sum_{i=1}^{n} p_i(x_i(t)) - p_i(x_i(t)) \right) \tag{9}$$

It is easy to note that any fixed point of the above systems is characterized by:

$$p_i(x_i^*) = p_j(x_j^*),\ i \neq j,\ \text{for} i \in [1, n] \tag{10}$$

We shall prove that the above ODE equation (Eq. (9)) admits a unique fixed point that is stable and attracting.

Uniqueness of the fixed point: The uniqueness of x^* is proven by contradiction. Suppose there exists $y^* = (y_1^*, y_2^*, ..., y_n^*)$ that verifies Eq. (10) such that $x^* \neq y^*$.

Without loss of generality since x^* and y^* are two probability vectors such that $x^* \neq y^*$, we are guaranteed[3] that they have at least two components i and j such that $x_i^* > y_i^*$ and $x_j^* < y_j^*$. Intuitively this means, that if we increase any one component of a probability vector, we should decrease another component so as to ensure that the sum of the components is still unity.

Suppose now that $x_i^* > y_i^*$. Then, by invoking the strict monotonicity of the function $p_i(.)$, we obtain that $p_i(x_i^*) < p_i(y_i^*)$. On the other hand, the condition $x_j^* < y_j^*$ implies that $p_j(x_j^*) > p_j(y_j^*)$, where this is obtained by virtue

[3] Please note that the result is general and applies for any two distinct probability vectors.

of the monotonicity of $p_j(.)$. But since x^* and y^* are equilibrium points, we know that $p_i(x_i^*) = p_j(x_j^*)$ and that $p_i(y_i^*) = p_j(y_j^*)$. This forces a contradiction since it is impossible to simultaneously maintain that : $p_i(x_i^*) < p_i(y_i^*)$ which is equivalent to $p_j(x_j^*) < p_j(y_j^*)$ and $p_j(x_j^*) > p_j(y_j^*)$.

Therefore x^* is unique.

Jacobian Properties at x^* Let $w(x(t)) = x(t+1) - x(t)$. The Jacobi matrix of $w(x(t))$ at a point $x = (x_1, x_2, \ldots, x_n)$ is given by

$$J_w(x) = \begin{bmatrix} \frac{\partial w_1}{\partial x_1} & \frac{\partial w_1}{\partial x_2} & \cdots & \frac{\partial w_1}{\partial x_n} \\ \frac{\partial w_2}{\partial x_1} & \frac{\partial w_2}{\partial x_2} & \cdots & \frac{\partial w_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial w_n}{\partial x_1} & \frac{\partial w_n}{\partial x_2} & \cdots & \frac{\partial w_n}{\partial x_n} \end{bmatrix}$$

Hence the Jacobi matrix has the form

$$J_w(x) = \begin{bmatrix} 1 + \frac{(n-1)\theta}{n}\frac{\partial p_1}{\partial x_1}(x_1) & -\frac{\theta}{n}\frac{\partial p_2}{\partial x_2}(x_2) & \cdots & -\frac{\theta}{n}\frac{\partial p_n}{\partial x_n}(x_n) \\ -\frac{\theta}{n}\frac{\partial p_1}{\partial x_1}(x_1) & 1 + \frac{(n-1)\theta}{n}\frac{\partial p_2}{\partial x_2}(x_2) & \cdots & -\frac{\theta}{n}\frac{\partial p_n}{\partial x_n}(x_n) \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\theta}{n}\frac{\partial p_1}{\partial x_1}(x_1) & -\frac{\theta}{n}\frac{\partial p_2}{\partial x_2}(x_2) & \cdots & 1 + \frac{(n-1)\theta}{n}\frac{\partial p_n}{\partial x_n}(x_n) \end{bmatrix}$$

We will now look at the structure of this matrix, let

$$\epsilon_j = -\frac{\theta}{n}\frac{\partial p_j}{\partial x_j}(x_j)$$

Hence the structure of the Jacobi matrix is

$$M_{J,n} = \begin{bmatrix} 1-(n-1)\epsilon_1 & \epsilon_2 & \cdots & \epsilon_n \\ \epsilon_1 & 1-(n-1)\epsilon_2 & \cdots & \epsilon_n \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_1 & \epsilon_2 & \cdots & 1-(n-1)\epsilon_n \end{bmatrix}$$

It is easy to note that the matrix $M_{J,n}$ has (at least) one eigenvalue equals 1. Now, we will resort to a perturbation argument.

Let us assume that $\epsilon_i \approx \epsilon_j$ for all $1 \leq i, j \leq n$. Define ϵ to be the average of ϵ_i, $1 \leq i \leq n$,

$$\epsilon = \frac{1}{n}\sum_{i=1}^{n}\epsilon_i$$

and define the matrix $M_{\epsilon,n}$ by

$$M_{\epsilon,n} = \begin{bmatrix} 1-(n-1)\epsilon & \epsilon & \cdots & \epsilon \\ \epsilon & 1-(n-1)\epsilon & \cdots & \epsilon \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon & \epsilon & \cdots & 1-(n-1)\epsilon \end{bmatrix}$$

Clearly $M_{\epsilon,n}$ is symmetric, that is, $M_{\epsilon,n} = M_{\epsilon,n}^{\mathrm{T}}$. We know that $\theta_1 = 1$ is an eigenvalue with with eigenvector $u_1 = (1, 1, 1, \ldots, 1)$. Furthermore, $\theta = 1-n\epsilon$ is an eigenvalue of algebraic multiplicity $n-1$, hence we have

$$\theta_2 = \theta_3 = \cdots = \theta_n = 1 - n\epsilon$$

We claim that the geometric multiplicity is $n-1$. To see this, let

$$u_j = (-1, 0, 0, \ldots, 1, 0, \ldots, 0) \text{ where } 2 \leq j \leq n$$

where the first component is -1, the j-th component is 1 and all other components is 0 in u_j. Consider $M_{\epsilon,n}u_j$, then a simple calculation shows that

$$M_{\epsilon,n}u_j = (1 - n\epsilon)u_j$$

This shows that $1 - n\epsilon$ is an eigenvalue with associated eigenvector u_j for $2 \leq j \leq n$. Furthermore, the set $\{u_i\}_{i=1}^{n}$ forms a basis for $\mathbb{R}^n$, in fact $\det(U) = n$, where U is the matrix with u_i as columns. Hence we conclude that $\theta = 1 - n\epsilon$ is an eigenvalue with algebraic and geometric multiplicity $n-1$.

The fixed point equations, for $1 \leq i \leq n$, are given by

$$\frac{1}{n}\sum_{j=1}^{n} p_j(x_j(t)) - p_i(x_i(t)) = 0$$

independent of the parameter θ.
However, we note that each element in the matrix $J_{\boldsymbol{w}}(x)$ is dependent on θ. We have obtained that $\xi_1 = 1$ is always an eigenvalue, and that the other eigenvalues are approximated by $\xi_i = 1 - n\epsilon$ where $\epsilon \to 0^+$ when $\theta \to 0^+$.

Therefore, one of the eigenvalues of $J_{\boldsymbol{w}}(x)$ at the fixed point is equal to 1 while the rest are less than 1 in norm. As the eigenspace corresponding to 1 is transversal to the invariant hyperplane, it follows that the fixed point is stable and attracting which concludes the proof. □

4 Empirical Results

In this Section, we perform a systematic comparison of the performance of the TTS-LA, H-TRAA and CMLS algorithms to solve the Stochastic NFEK Problem. We measure performance against the true optimal solution that can be found using Lemma 1. Furthermore, we will compare our scheme against two other heuristic solutions, namely, the LA based proportional allocation due to Papadimitriou [17] and LAKG [7].

4.1 Problem Specification

We have tested our algorithm against some datasets used in the past so that they can serve as benchmarks. We consider two different objective functions $w_i(x_i)$ and refer to them as $E_i(x_i)$ and $L_i(x_i)$. $E_i(x_i)$ and $L_i(x_i)$ were used in the literature as benchmarks as reported in [5,6,25] and are particularly useful in the sense that they appropriately model a large family of distinct material unit value functions. Furthermore, they are representative of the class of concave objective functions addressed here.

$$E_i(x_i) = \frac{0.7}{i}(1 - e^{-ix_i}) \tag{11}$$

$$L_i(x_i) = 0.7 \cdot x_i - \frac{1}{2} i \cdot x_i^2, \qquad \text{If } x_i \leq \frac{0.7}{i} \tag{12}$$

$$= \frac{0.7^2}{i}. \qquad \text{If } x_i > \frac{0.7}{i}. \tag{13}$$

To ease the readability, we have used the notation that the profitability of materials that have a smaller index decreases *slower* than the profitability of materials that have higher indices.

The constants in the above functions (Eq. (11)-(13)) are based on the boundary conditions due the contributions of x_i at the boundary values, and are not crucial in the optimization process. This is because the corresponding unit value functions are the respective derivatives of the functions, and these derivatives fall exponentially and linearly as per Eq. (14) and (15) respectively:

$$E_i'(x_i) = 0.7 \cdot e^{-i \cdot x_i} \tag{14}$$

$$L_i' = \text{Max}\,[0.7 - i \cdot x_i, 0]. \tag{15}$$

It is expedient to glean some input about the significance of these unit value functions. To understand this, consider the functions $E_i'(x_i)$, where the relative profitability of material i decreases with x_i, its presence in the mixture, exponentially. Indeed, if $x_2 = 0.3$ (i.e., material 2 fills 30% of the knapsack), the marginal profitability of increasing the amount of x_2 is $e^{-2 \cdot (0.3)} = e^{-0.6}$.

Unlike the exponential function, the linear function, $L_i'(x_i)$ has an interesting peculiarity that the function for material i intersects the X-axis at a finite point, implying that the function being optimized is quadratic. Thus, it attains a maximum value at this point, after which it remains constant. Clearly, after this intersection point, it is futile to add any additional quantity of material i.

4.2 Experimental setup

We consider the case of quaternary and hexadecimal number of primitive materials. We assume a dynamic system, which means that the reward probabilities for the different materials vary with time. More specifically, after a

Environment type	Periodicity type	Environment Switch Instant
Fast	Fixed	Each $T = 2000$
Slow	Fixed	Each $T = 2 \cdot 10^4$
Rand	Variable Periodicity	Next change is either after 2000, 8000 or $4 \cdot 10^4$ iterations with prob $20/26, 5/26$ and $1/26$ respectively

Table 1 Summary of the different types of dynamically changing environments.

period of T iterations, the reward for the different materials are randomly switched. For example, let's assume a quaternary number of materials, and that the rewards for materials 1, 2, 3 and 4 are given by $E_1(x_1)$, $E_2(x_2)$, $E_3(x_3)$ and $E_4(x_4)$, respectively. After the period T, we randomly switch the reward functions such that the rewards for material 1, 2, 3 and 4 are given by e.g. $E_3(x_1)$, $E_1(x_2)$, $E_4(x_3)$ and $E_2(x_4)$, respectively. Please note that the switch is implemented by shuffling randomly the indexes of the reward functions.

We consider three different cases for the period T between every switch:

- $T = 2000$ iterations. We refer to this scenario as SHORT below.
- $T = 2 \cdot 10^4$ iterations which is referred to as LONG below.
- We assume that T is a stochastic variable with possible outcomes 2000, 8000 and $4 \cdot 10^4$. The probabilities are $P(T = 2000) = 20/26$, $P(T = 8000) = 5/26$ and $P(T = 4 \cdot 10^4) = 1/26$ which means that, in average, the estimation process spends an equal amount of time in each of the states 'fast' ($T = 2000$), 'medium' ($T = 8000$) and 'slow' ($T = 4 \cdot 10^4$). We refer to this scenario as RAND below. In fact, the average time to spend in a fast environment before a switch takes place is $2000 \cdot 20/26 = 4 \cdot 10^4/26$ iterations, which is the same average time for the case of medium environment $8000 \cdot 5/26 = 4 \cdot 10^4/26$ and the same average time as well for a slow environment $4 \cdot 10^4 \cdot 1/26 = 4 \cdot 10^4/26$.

The motivation for the RAND scenario, is that an algorithm should be able handle environments where the dynamics change arbitrarily with time, i.e. changing between slowly and fast variations. Naturally, the optimal values of tuning parameters of an algorithm depend on the dynamics of the environment, but ideally the performance of the algorithm should not be too sensitive with respect to these dynamics. Table 1 summarizes the three types of dynamically changing environments used in the experiments. Similarly, Algorithm 2 gives the pseudo-code that describes the simulation procedure.

Algorithm 2 Simulation Steps

Pick Initial Periodicity T
Initialize instant $SwitchT$ to T ($SwitchT = T$)
loop For $time \leftarrow 0$ to $Max\ iterations$
  Poll an action according to the probability vector $[x_1, x_2, \ldots, x_n]$
  Receive Environment Feedback, reward or penalty
  Update action probability vector according to corresponding algorithm
  Update error based on distance between $x(t)$ and the optimal x^*
  Increment $time$
  if $time == SwitchT$ **then**
    Pick new Periodicity T (Random variable if Environment is Rand)
    Update next switch instant $SwitchT = t + T$
    Shuffle the indexes of the objective function (E or L according to model type)
    Update optimal x^* corresponding to shuffled objective function
  end if
end loop

We started the algorithm with initial material amounts as outcomes from the Dirichlet distribution with all parameter values equal to one. This is referred to as the flat Dirichlet distribution and the probability distribution is uniformly distributed over the simplex of possible material amounts, i.e. the vectors satisfying $x_1, x_2, \ldots, x_n > 0$ and $\sum_{i=1}^{n} x_i = 1$.

In a dynamic environment the aim is to achieve as precise estimate as possible in every iteration. We compute estimation error using the root mean squared error (RMSE) over all iterations.

$$L(\widehat{x}, x; \theta) = \frac{1}{n} \sum_{i=1}^{n} \sqrt{\frac{1}{R} \sum_{t=1}^{R} (\widehat{x}_{it} - x_{it})^2} \tag{16}$$

where R refers to the number of iterations while x_{it} refers the true and optimal amount of material i at iteration t (computed using Lemma 1) and $\widehat{x}_{it}$ denotes the estimate. We computed the RMSE for each material and then took the average. To remove Monte Carlo error, we ran the experiment for $R = 10^7$ iterations for every case and for every choice of the tuning parameters in the algorithms.

Based on the two-time scale theory described in this paper, we expect that using a low value of λ compared to θ would be the best alternative. In the experiments we tried the following values for the ratio λ/θ: $1/50$, $1/20$, $1/10$, $1/5$, $1/3$ and 1.

Let $p(\theta)$ denote a probability distribution for our prior belief on the tuning parameter θ in the algorithms. As a total representation of the performance

of an algorithm, we define the Bayesian expected loss

$$E_\theta\left(L\left(\widehat{x}, x; \theta\right)\right) = \int_0^1 L\left(\widehat{x}, x; \theta\right) p(\theta)\, \mathrm{d}\theta \tag{17}$$

The Bayesian expected loss computes the expected loss where our uncertainty in the knowledge about the tuning parameter is taken into account. Thus if the performance of an algorithm is sensitive to the choice of the tuning parameter and our prior knowledge is limited, we expect the Bayesian expected loss to be high. However, if the performance of the algorithm is high for a wide range of values of the tuning parameter or if we have good knowledge about the optimal value of the tuning parameter, the Bayesian expected loss typically will be small.

4.3 Comparison against legacy optimal solutions

We report here comparison results of our proposed TTS-LA solution against the only two available optimal solutions to the Stochastic NFEK problem reported in the literature, namely CMLS and H-TRAA.

Figures 1 and 2 show the results of the algorithms for all possible values of the tuning parameters. For both the exponential and linear decay reward functions (Figures 1 and 2, respectively) and for all the six cases, we see that the TTS-LA algorithm outperforms both the H-TRAA and CMLS algorithms with respect to peak performance. Optimal performance of the TTS-LA algorithm is achieved using a value of θ around 0.2. Please note that the error curves for the H-TRAA have a global minimum for a small value of the tuning parameter (interval width) and a local minimum for an interval width around 0.5. This is not due to Monte Carlo estimation error, but is an actual effect.

Table 2 shows the Bayesian expected loss in Eq. (17) assuming no prior knowledge of the tuning parameter, i.e. $p(\theta)$ in Eq. (17) is the uniform distribution on the $[0, 1]$ interval. For the abbreviation in the second column, e.g. EXP4 refers to exponential decay reward case ($E_i(x_i)$) with $n = 4$ materials. We see that by using a low value of the ratio λ/θ, the TTS-LA algorithm outperforms both the H-TRAA and CLMS algorithms with a clear margin. This documents that the performance of the TTS-LA algorithm is far less sensitive to the choice of the tuning parameter θ than the H-TRAA and CLMS algorithms. Such a robustness in performance is important since the optimal values of the tuning parameters generally are unknown in a dynamical environment.

The conclusion is that the TTS-LA documents better peak performance than both the CMLS and H-TRAA algorithms for all the six cases. In addition, using a small λ/θ ratio, the performance of the performance of the TTS-LA algorithm is far more robust than the CMLS and H-TRAA algorithms for all the six cases.

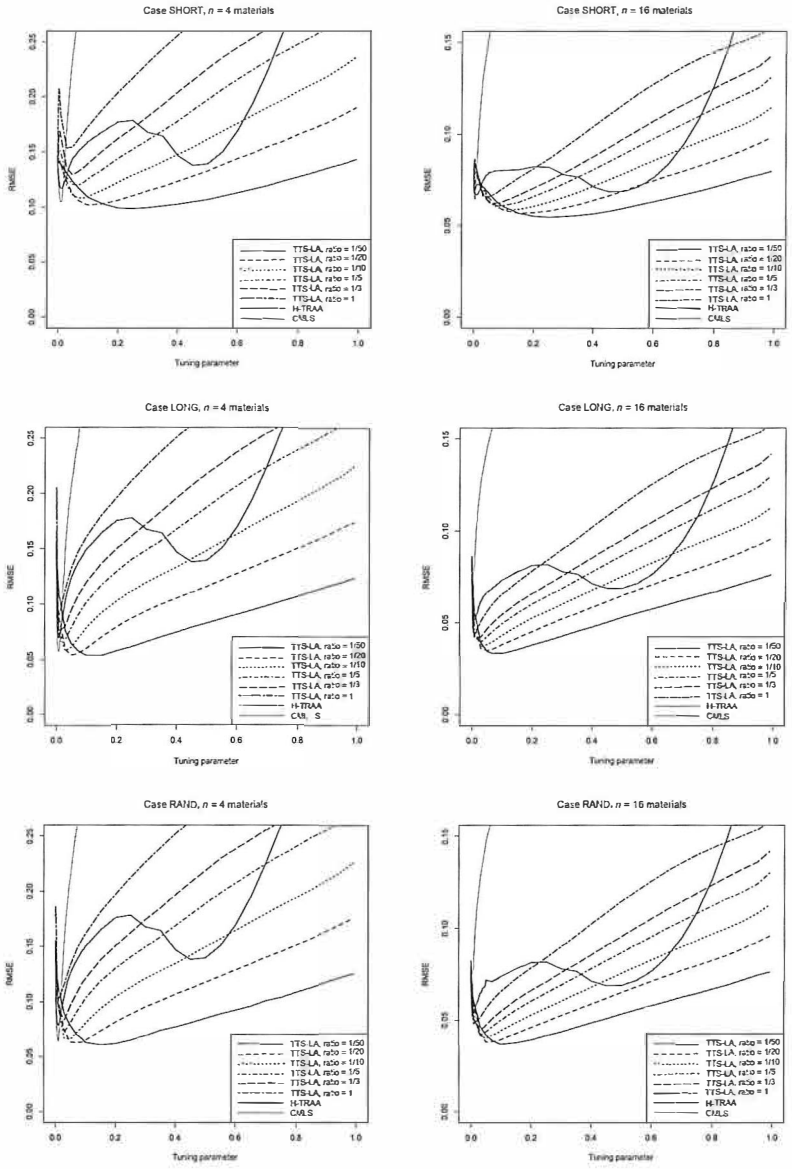

Fig. 1 Exponential decay reward case ($E_i(x_i)$): Estimation error (RMSE) for the TTS-LA, H-TRAA and CMLS algorithms for a wide range of values of the tuning parameters. The tuning parameter on the x axis refers to the interval width (inverse of the leaf node resolution) for the H-TRAA algorithm and to θ for the CMLS and TTS-LA algorithms. The left and right columns refer to cases with 4 and 16 materials, respectively. The rows from top to bottom refer to the cases SHORT, LONG and RAND, respectively.

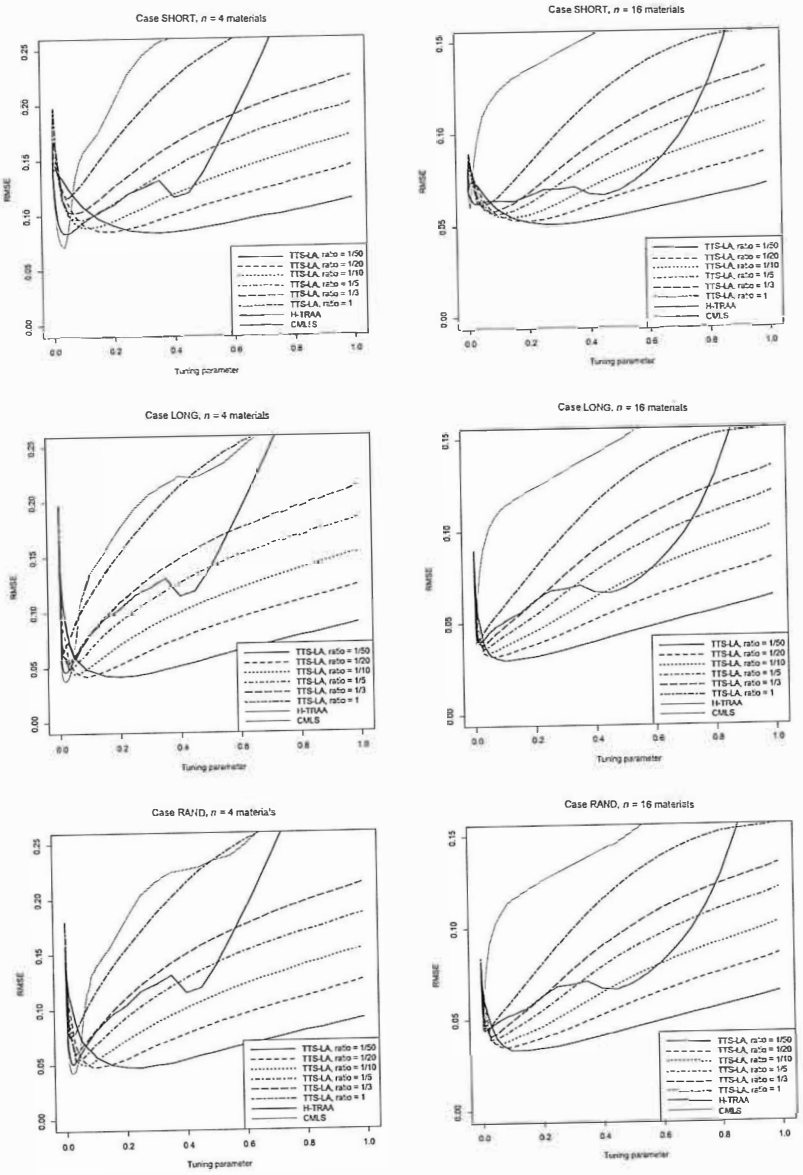

Fig. 2 Linear decay reward case ($L_i(x_i)$): Estimation error (RMSE) for the TTS-LA, H-TRAA and CMLS algorithms for a wide range of values of the tuning parameters. The tuning parameter on the x axis refers to the interval width (inverse of the leaf node resolution) for the H-TRAA algorithm and to θ for the CMLS and TTS-LA algorithms. The left and right columns refer to cases with 4 and 16 materials, respectively. The rows from top to bottom refer to the cases SHORT, LONG and RAND, respectively.

				Ratio					
		H-TRAA	CMLS	1/50	1/20	1/10	1/5	1/3	1
SHORT	EXP4	21.32	40.35	11.43	13.66	16.28	19.48	21.89	26.55
	EXP16	9.90	21.40	6.35	7.18	8.02	8.98	9.73	11.24
	LINE4	19.04	26.70	9.73	11.07	12.82	15.02	16.93	22.50
	LINE16	9.46	16.36	5.88	6.77	7.72	8.87	9.80	11.88
LONG	EXP4	21.10	36.36	8.45	11.42	14.51	18.07	20.67	25.72
	EXP16	9.75	20.25	5.27	6.37	7.37	8.46	9.29	10.91
	LINE4	18.68	24.06	6.21	8.32	10.49	13.01	15.14	21.36
	LINE16	9.23	15.47	4.48	5.71	6.90	8.24	9.31	11.56
RAND	EXP4	21.13	37.17	8.79	11.64	14.68	18.19	20.78	25.79
	EXP16	9.78	20.34	5.39	6.45	7.43	8.51	9.33	10.94
	LINE4	18.71	24.51	6.63	8.58	10.69	13.17	15.29	21.45
	LINE16	9.27	15.54	4.63	5.81	6.97	8.30	9.35	11.59

Table 2 Bayesian expected loss based on Eq. (17). E.g. the abbreviation EXP4, refers to exponential decay reward case ($E_i(x_i)$) with $n = 4$ materials. The values in the table are the computed Bayesian expected loss multiplied by 100.

4.4 Comparing against heuristic solutions

At this juncture, after having compared our TTS-LA solution against the only two available optimal solutions to the Stochastic NFEK problem namely CMLS and H-TRAA, we compare our solution to two heuristic solutions. The heuristic solutions we shall compare to are the LA based proportional allocation due to Papadimitriou [17] and the LAKG algorithm [7]. We consider a static environment and we ran an ensemble of 1000 experiments, each experiment consisting of 10^5 iterations. We shall report the peak performance [25] of each scheme which is obtained by running each scheme for a set of tuning parameters and report too the parameter and corresponding performance yielding the smallest root mean square error. When it comes to the LAKG, there are two tuning parameters as seen in Section 2, the resolution N and the non-linearity parameter γ which makes the tuning more difficult than the case of unique tuning parameter. As done in [7] for the case of a static environment, we run our simulation for the same configuration parameters as Granmo and Oommen, namely, $[N = 1500, \gamma = 1.3]$, $[N = 2500, \gamma = 1.2]$ and $[N = 5000, \gamma = 1.2]$. For our proposed scheme TTS-LA, the tuning parameter θ is chosen in $[0, 1]$ while λ depends on the choice of θ to ensure the time scale separation by constraining the ratio λ/θ to admit one of the following values: 1/50, 1/20, 1/10, 1/5, 1/3 and 1 also used in the previous experiments above. We have chosen 10 discrete values of θ, namely $\{0.1, 0.2 \ldots, 1\}$. When it comes to the LA based proportional allocation due to Papadimitriou [17] , the only tuning parameter is the learning factor which takes values from $[0, 1]$ which we denote by λ here. We have tested 10 values for λ for the proportional LA scheme contained in the set $\{0.1, 0.2 \ldots, 1\}$.

The tests are performed for 8 materials and 4 materials for both linear and exponential decaying functions resulting into the four cases: LIN4, EXP4, LIN8 and EX8 which are reported in Table 3, Table 4, Table 3 and Table 6 respectively. Interestingly, we observe that our TTS-LA outperforms the two

	LAKG	Proportional LA	TTS-LA
Error	0.323	0.0898	0.041
Optimal tuning parameter(s)	$[N = 2500, \gamma = 1.2]$	$\lambda = 0.1$	$\lambda/\theta = 1/50$, $\theta = 0.1$

Table 3 Peak performance for a static environment for the case of LIN4 for two heuristic solutions and the proposed TTS-LA.

	LAKG	Proportional LA	TTS-LA
Error	0.4383	0.1934	0.047
Optimal Tuning parameter(s)	$[N = 2500, \gamma = 1.2]$	$\lambda = 0.1$	$\lambda/\theta = 1/50$, $\theta = 0.1$

Table 4 Peak performance for a static environment for the case of EXP4 for two heuristic solutions and the proposed TTS-LA.

	LAKG	Proportional LA	TTS-LA
Error	0.3699	0.1481	0.037
Optimal tuning parameter(s)	$[N = 2500, \gamma = 1.2]$	$\lambda = 0.1$	$\lambda/\theta = 1/50$, $\theta = 0.1$

Table 5 Peak performance for a static environment for the case of LIN8 for two heuristic solutions and the proposed TTS-LA.

	LAKG	Proportional LA	TTS-LA
Error	0.438	0.2228	0.035
Optimal tuning parameter(s)	$[N = 2500, \gamma = 1.2]$	$\lambda = 0.1$	$\lambda/\theta = 1/50$, $\theta = 0.2$

Table 6 Peak performance for a static environment for the case of EXP8 for two heuristic solutions and the proposed TTS-LA.

heuristics by a large margin. The optimal tuning parameters for TTS-LA are $\lambda/\theta = 1/50$, $\theta = 0.1$ for all the cases except for EXP8 (Table 6) where $\theta = 0.2$ yields the peak performance. The proportional LA performs better than LAKG and yields its peak performance for a value of the tuning parameter equal to 0.1.

5 Conclusion

In this paper, we present an optimal solution to the Stochastic Non-linear Fractional Equality Knapsack (NFEK) Problem based on the two-time scale paradigm. Our solution exploits much more feedback information than the classical NFEK solutions proposed in the literature [5, 6, 25]. In fact, while the polling probabilities are updated based on only the last feedback in all classical NFEK solutions including H-TRAA and CMLS, our approach uses incremental averaging of the whole history of the feedback. Furthermore, we

provide sound theoretical results that show that our solution is asymptotically optimal. Through comprehensive experimental results, we show that our devised TTS-LA scheme outperforms the legacy solutions in terms of peak performance and robustness to the choice of the tuning parameters.

References

1. A. A. Al Islam, S. I. Alam, V. Raghunathan, and S. Bagchi. Multi-armed bandit congestion control in multi-hop infrastructure wireless mesh networks. In *2012 IEEE 20th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 31–40. IEEE, 2012.
2. A. Benveniste, P. Priouret, and M. Métivier. *Adaptive Algorithms and Stochastic Approximations*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.
3. P. E. Black. Fractional knapsack problem. *Dictionary of algorithms and data structures*, 2004.
4. M. Ghavipour and M. R. Meybodi. Trust propagation algorithm based on learning automata for inferring local trust in online social networks. *Knowledge-Based Systems*, 2017.
5. O.-C. Granmo and B. J. Oommen. Optimal sampling for estimation with constrained resources using a learning automaton-based solution for the nonlinear fractional knapsack problem. *Applied Intelligence*, 33(1):3–20, 2010.
6. 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*, 59(4):545–560, 2010.
7. 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*, 37(1):166–175, 2007.
8. K. Liu, Q. Zhao, and A. Swami. Dynamic probing for intrusion detection under resource constraints. In *Proceedings of IEEE International Conference on Communications, ICC 2013, Budapest, Hungary, June 9-13, 2013*, pages 1980–1984, 2013.
9. Z. Ma, H. Wang, K. Shi, and X. Wang. Learning automata based caching for efficient data access in delay tolerant networks. *Wireless Communications and Mobile Computing*, 2018, 2018.
10. M. Malboubi, L. Wang, C.-N. Chuah, and P. Sharma. Intelligent sdn based traffic (de) aggregation and measurement paradigm (istamp). In *2014 Proceedings IEEE INFOCOM*, pages 934–942. IEEE, 2014.
11. K. S. Narendra and M. A. L. Thathachar. *Learning automata: an introduction*. Courier Corporation, 2012.
12. P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis. Learning automata-based polling protocols for wireless lans. *IEEE Transactions on Communications*, 51(3):453–463, 2003.
13. P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis. Distributed protocols for ad hoc wireless lans: a learning-automata-based approach. *Ad Hoc Networks*, 2(4):419–431, 2004.
14. M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis. An efficient adaptive bus arbitration scheme for scalable shared-medium atm switch. *Computer Communications*, 24(9):790–797, 2001.
15. 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)*, 32(6):815–820, 2002.
16. G. I. Papadimitriou and A. S. Pomportsis. Dynamic bandwidth allocation in wdm passive star networks with asymmetric traffic. *Photonic Network Communications*, 2(4):383–391, 2000.
17. G. I. Papadimitriou and A. S. Pomportsis. Learning-automata-based tdma protocols for broadcast communication systems with bursty traffic. *IEEE Communications Letters*, 4(3):107–109, 2000.
18. G. I. Papadimitriou and A. S. Pomportsis. On the use of learning automata in medium access control of single-hop lightwave networks. *Computer Communications*, 23(9):783–792, 2000.
19. A. Rezvanian and M. R. Meybodi. Sampling algorithms for stochastic graphs: a learning automata approach. *Knowledge-Based Systems*, 127:126–144, 2017.

20. A. Rezvanian, A. M. Saghiri, S. M. Vahidipour, M. Esnaashari, and M. R. Meybodi. *Recent Advances in Learning Automata*, volume 754. Springer, 2018.
21. A. Rezvanian, S. M. Vahidipour, and M. Esnaashari. New applications of learning automata-based techniques in real-world environments. *Journal of Computational Science*, 24:287 – 289, 2018.
22. A. M. Saghiri and M. R. Meybodi. Open asynchronous dynamic cellular learning automata and its application to allocation hub location problem. *Knowledge-Based Systems*, 139:149–169, 2018.
23. S. H. Seyyedi and B. Minaei-Bidgoli. Estimator learning automata for feature subset selection in high-dimensional spaces, case study: Email spam detection. *International Journal of Communication Systems*, 2018.
24. M. L. Tsetlin. *Automaton theory and modeling of biological systems*. Academic Press, New York, 1973.
25. A. Yazidi and H. Hammer. Solving stochastic nonlinear resource allocation problems using continuous learning automata. *Applied Intelligence*, (To Appear), 2018.
26. A. Yazidi, T. M. Jonassen, and E. Herrera-Viedma. An aggregation approach for solving the non-linear fractional equality knapsack problem. *Expert Systems with Applications*, 110:323 – 334, 2018.