Determining Optimal Polling Frequency Using a Learning Automata-based Solution to the Fractional Knapsack Problem

Ole-Christoffer Granmo
Department of ICT
Agder University College
Grimstad, Norway
ole.granmo@hia.no

B. John Oommen*
School of Computer Science
Carleton University
Ottawa, Ontario, Canada
oommen@scs.carleton.ca

Svein A. Myrer
Morten G. Olsen
C/o. Dr. Granmo
Agder University College
Grimstad, Norway

Abstract— **Recent approaches to resource allocation in web monitoring target *optimal* performance under restricted capacity constraints [1], [2]. The resource allocation problem is generally modelled as a knapsack problem with *known deterministic* properties. However, for practical purposes the web must often be treated as *stochastic* and *unknown*. Unfortunately, estimating unknown knapsack properties (e.g., based on an estimation phase [1], [2]) delays finding an optimal or near-optimal solution. *Dynamic* environments aggravate this problem further when the optimal solution changes with time. In this paper, we present a novel solution for the nonlinear fractional knapsack problem with a separable and concave criterion function [3]. To render the problem realistic, we consider the criterion function to be *stochastic* with an *unknown* distribution. At every time instant, our scheme utilizes a series of informed guesses to move, in an online manner, from a "current" solution, towards the optimal solution. At the heart of our scheme, a *game* of deterministic learning automata performs a controlled random walk on a discretized solution space. Comprehensive experimental results demonstrate that the discretization resolution determines the precision of our scheme. In order to yield a required precision, the current resource allocation solution is consistently improved, until a near-optimal solution is found. Furthermore, our proposed scheme quickly adapts to periodically switching environments. Thus, we believe that our scheme is qualitatively superior to the class of estimation-based schemes.**

Keywords—**nonlinear knapsack problems, learning automata, resource allocation, stochastic optimization**

I. Introduction

Problem Motivation: We consider the problem of *optimizing* the allocation of resources in web monitoring when the monitoring capacity is restricted [1], [2]. Consider a basic web monitoring resource allocation system. Such a system may involve n web pages that are updated periodically, although with different periods. Clearly, each web page can be polled with a maximum frequency — and this would, effectively "flood" the system with status-probing polls as opposed to information-delivering searches. The problem which we study involves determining the web page polling frequencies that the system must use to maximize the number of web page updates detected, without exceeding the available monitoring capacity.

*Professor and Fellow of the IEEE

Dealing with the web is particularly fascinating because it leads to numerous extremely interesting problems involving real-life resource allocation and scheduling. Such problems, including the one we study, are particularly intriguing because access to the resources is constrained, and the parameters of the underlying system are unknown and not easily estimated. Indeed, in some cases the parameters can not be estimated *at all*. One such classic problem is that which involves "optimal search" [4]. An user has to determine how much time to allocate to search a particular "site" (which may be a web-site, a library, or in general, a geographical area). However, the probability of locating the object searched for in that site can not be estimated because the underlying event is unobservable. The web polling resource allocation problem involves a similar hurdle — how do we allocate resources for polling when the characteristics of the web pages to be polled are unknown, not easily estimated, and perhaps even dynamically changing.

Recent approaches to resource allocation in web monitoring target an *optimal* performance under a restricted capacity constraint [1], [2]. The resource allocation is generally modelled as a so-called knapsack problem with *known deterministic* properties. In this paper we shall see that a *stochastic* version of the knapsack problem with *unknown* properties is a more appropriate model.

The Linear Fractional Knapsack (FK) Problem: Several variants of the knapsack problem have been treated in the literature [5]. The linear FK problem is a classical continuous optimization problem which also has applications within the field of resource allocation. The problem involves n materials of different value v_i per unit volume, $1 \leq i \leq n$. Each material is available in a certain amount $x_i \leq b_i$. Let $f_i(x_i)$ denote the value of the amount x_i of material i, i.e., $f_i(x_i) = v_i x_i$. The problem is to fill a knapsack of fixed volume c with the material mix $\vec{x} = [x_1, \ldots, x_n]$ of maximal value $\sum_1^n f_i(x_i)$ [6].

The Nonlinear Equality FK (NEFK) Problem: One important extension of the above classical problem is the *nonlinear equality* FK problem with separable and concave objective functions. The problem can be stated as below [5]:

$$\begin{array}{ll} \text{maximize} & \sum_1^n f_i(x_i) \\ \text{subject to} & \sum_1^n x_i = c \\ & 0 \leq x_i, \qquad i = 1, \ldots, n. \end{array}$$

Since the objective function is considered to be concave, the Material Value Function (MVF), $f_i(x_i)$, of each material is also concave. The derivatives of the MVFs[1] with respect to x_i, are hereafter denoted by f_i'. The latter problem has been solved using Lagrange multipliers. The optimal value occurs when the derivatives f_i' of the MVFs are equal, subject to the knapsack constraints [3], [7]:

$$\begin{array}{l} f_1'(x_1) = \cdots = f_n'(x_n) \\ \sum_1^n x_i = c \\ 0 \leq x_i, i = 1, \ldots, n. \end{array}$$

The Stochastic Nonlinear Equality FK Problem: The fundamental problem that we solve generalizes the above NEFK problem. First of all, we let the material value per unit volume be an independent *stochastic* function $F_i'(x_i)$. Furthermore, we consider the case when the distribution $F_i'(x_i)$ is *unknown*. That is, each time an amount x_i of material i is placed in the knapsack, we are only allowed to observe an *instantiation* of $F_i'(x_i)$ at x_i, and not $F_i'(x_i)$ itself. Given this stochastic environment, we intend to devise an online (also referred to as a real-time) incremental scheme that learns the mix of materials of maximal *expected* value, through a series of informed guesses.

Stochastic Knapsack Problems - State-of-the-Art: To the best of our knowledge, our targeted stochastic NEFK problem has not been addressed in the literature before. However, several related problems have been addressed. For example, the works of [8], [9] consider solution policies for stochastic generalizations of the so-called NP-hard *linear* integer knapsack problem in which value distributions are considered known and constant, making dynamic programming a viable solution. Another variant of the knapsack problem is found in [10] where a deterministic knapsack is used, however, with objects arriving to and departing from the knapsack at random times. The optimization problem considered is to accept/block arriving objects so that the average value of the knapsack is maximized.

Contributions of This Paper: We base our work on the principles of Learning Automata (LA) [11], [12]. Learning Automata have been used to model biological systems [13], and have attracted considerable interest in the last decade because they can learn the optimal actions when operating in (or interacting with) unknown stochastic environments. Furthermore, they combine rapid and accurate convergence with low computational complexity. Recently, Oommen *et al.* [14] proposed a Learning Automata based mechanism for locating the optimal point on a line. An environment tells the learning mechanism whether its point guesses lie to the left or to the right of the optimal point, possibly erroneously, and by interacting with the environment the learning mechanism converges towards selecting the optimal point with probability $1 - \epsilon$, (where ϵ is arbitrarily small) rendering the scheme to be ϵ-optimal. While the latter work considers stochastic optimization of a single parameter, we target optimization of n parameters (n material amounts), where the different parameters are mutually constrained by the constraints imposed by the "knapsack". Furthermore, our environment is less informative than the one introduced in [14], because our environment does not tell us whether a parameter guess x_i lies to the left or to the right of the unknown optimal value. The environment only replies with an instantiated unit volume value at x_i, drawn from the stochastic unit volume value function, $F_i'(x_i)$.

[1]The derivatives represent the material values *per unit volume*. These are constant in the linear case, but non-increasing (they decrease with the quantity of the material concerned) in the nonlinear case.

Format of the Paper: In Section II we present a *game* of learning automata that is designed to solve our stochastic version of the nonlinear equality knapsack problem incrementally and online. In Section III we detail the polling frequency determination problem, and present a solution based on the Learning Automata Knapsack Game (LAKG) scheme from Section II. We evaluate the LAKG scheme empirically based on simulating stationary and non-stationary environments with as many as 500 materials. We conclude the paper in Section IV and suggest areas for further work.

II. A Learning Automata Knapsack Game (LAKG)

To present our work in the right perspective, we first briefly discuss the solution concepts for more basic variants of the knapsack problem. As indicated in the introduction, solving the classical linear FK problem involves finding the most valuable mix $\vec{x} = [x_1, \ldots, x_n]$ of n materials that fits within a knapsack of fixed capacity c. Because a fraction of each material can be placed in the knapsack, the following greedy algorithm from [6] finds the most valuable mix: *Take as much as possible of the material that is most valuable per unit volume. If there is still room, take as much as possible of the next most valuable material. Continue until the knapsack is full.*

Let us now assume that unit volume values are *stochastic* variables $\{V_1, \ldots, V_n\}$ with *constant* and *known* distribution. Furthermore, for the sake of conceptual clarity, let us only consider binary variables V_i that either instantiates to the value of 0 or 1. Let p_i^0 denote the probability that $V_i = 0$ and $p_i^1 = 1 - p_i^0$ denote the probability that $V_i = 1$. Then the value $F_i(x_i)$ of the amount x_i of material i also becomes stochastic: $F_i(x_i) = V_i \times x_i$. In other words, $F_i(x_i)$ takes the value 0 with probability p_i^0 and the value x_i with probability p_i^1. Under such conditions, the above greedy strategy can be used to maximize the *expected* value of the knapsack $\sum_1^n E[F_i(x_i)]$, simply by selecting material based on expected unit volume values, $E[V_i] = p_i^1$, rather than actual unit volume values.

However, the above solution approach is obviously inadequate when the distributions of the $F_i(x_i)$s and the V_is are unknown. The problem becomes even more challenging when the distribution of the material value per unit volume is no longer constant, but instead depends on the amount x_i. Let $F_i'(x_i)$ denote the stochastic function that determines the unit

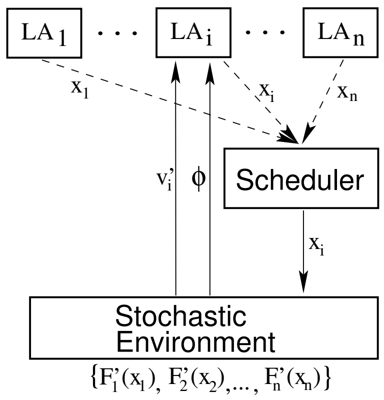

Fig. 1. Learning Automata interacting with a scheduler and an unknown stochastic environment.

volume value of material i. To elaborate, $F'_i(x_i)$ takes the value 0 with probability $p_i^0(x_i)$ and the value 1 with probability $p_i^1(x_i)$. Under the latter conditions, our aim is to find a scheme that moves towards optimizing the following NEFK problem online:

$$\begin{array}{ll} \text{maximize} & \sum_1^n E[F_i(x_i)] \\ \text{subject to} & \sum_1^n x_i = c \\ & 0 \le x_i, \qquad\qquad i = 1, \ldots, n. \end{array}$$

Note that we allow only instantiations of the material value per unit volume $F'_i(x_i)$ to be observed. That is, each time an amount x_i of material i is placed in the knapsack, an instantiation of $F'_i(x_i)$ at x_i is observed. We shall assume that x_i is any number in the interval $(0, 1]$ and generalize this later.

The crucial issue that we have to address is that of determining how to change our current guesses on $x_i,\ 1 \le i \le n$. We shall attempt to do this in a discretized manner by subdividing the unit interval into N points $\{\frac{1^\lambda}{N^\lambda}, \frac{2^\lambda}{N^\lambda}, \ldots, \frac{(N-1)^\lambda}{N^\lambda}, 1\}$, where N is the resolution of the learning scheme and $\lambda > 0$ determines the linearity of the discretized solution space[2]. We will see that a larger value of N will ultimately imply a more accurate solution to the knapsack problem.

Although, at the outset, it is clear that the rationale for this updating is the stochastic point location solution proposed by Oommen in [14], we emphasize the following differences:

(i) The scheme in [14] is linear.

(ii) The method proposed in [14] assumes the existence of an Oracle which informs the LA whether to go "right" or "left". In our application domain, this has now to be *inferred* by the system.

(iii) The method proposed in [14] assumes that there is only a single LA in the picture. Here we specifically understand that there are multiple LA — each of them being constrained to work together with the others. Thus, at the heart of our scheme is a game between n deterministic LA that interact with a scheduler and a stochastic environment as shown in Fig. 1. We describe each of the components below.

[2]The importance of λ will become evident in Section III where we present empirical results.

Stochastic Environment: The environment consists of a set of stochastic unit volume value functions $\mathcal{F}' = \{F'_1(x_1), F'_2(x_2), \ldots, F'_n(x_n)\}$. If amount x_i of material i is suggested to the environment, $F'_i(x_i)$ takes the unit volume value $v'_i = 0$ with probability $p_i^0(x_i)$ and the unit volume value $v'_i = 1$ with probability $p_i^1(x_i)$. In addition, the environment provides a signal ϕ indicating whether the knapsack is full:

$$\phi = \begin{cases} \mathit{true} & \text{if } \sum_{i=1}^n x_i \ge c \\ \mathit{false} & \text{otherwise} \end{cases}$$

Scheduler: The scheduler takes material amounts $\vec{x} = [x_1, \ldots, x_n]$ as input. The purpose of the scheduler is to (1) linearize accesses to the stochastic environment and (2) make sure that the unit volume value functions $F'_1(x_1), F'_2(x_2), \ldots, F'_n(x_n)$ are accessed with frequencies proportional to $\vec{x}$[3]. For the sake of simplicity, we choose to access the functions randomly by sampling from a probability distribution proportional to $\vec{x}$.

Learning Automata: Each material i is assigned a deterministic fixed structure automaton LA_i with the states $1, \ldots, N$. Let the current state of automaton LA_i be $s_i(t)$. When the automaton acts, it suggests the amount $x_i(t) = \frac{s_i(t)^\lambda}{N^\lambda}$ of material i to the scheduler, which in turn interacts with the stochastic environment. Assume that $v'_i(t)$ and $\phi(t)$ is the resulting feedback from the stochastic environment. Then the state of the automaton is updated as follows:

$$\begin{array}{l} s_i(t+1) := s_i(t) + 1 \\ \quad \textbf{if } v'_i(t) = 1 \textbf{ and } 1 \le s_i(t) < N \textbf{ and not } \phi(t) \\ s_i(t+1) := s_i(t) - 1 \\ \quad \textbf{if } v'_i(t) = 0 \textbf{ and } 1 < s_i(t) \le N \textbf{ and } \phi(t) \\ s_i(t+1) := s_i(t) \\ \quad \textbf{otherwise}. \end{array}$$

Notice that although the above state updating rules are deterministic, because the knapsack is stochastic, the state transitions will also be stochastic.

The purpose of the above Learning Automata game is to form a stochastic race between the n automata so that the race directs the automata towards the optimal solution. In essence, the race is governed by two situations: (1) There is still room for more material in the knapsack, and (2) The knapsack is full. In the first situation, the race consists of increasing ones material amount guesses as quickly as possible. However, a guess x_i can only be increased when the corresponding unit volume value function $F'_i(x_i)$ instantiates to 1. Accordingly, if $p_i^1(x_i) < p_j^1(x_j)$ for material i and j, then automaton j will have an edge in the race. The second situation is the opposite of the first one. In the second situation, each automaton i *decreases* its material amount guess x_i whenever the corresponding unit volume value function $F'_i(x_i)$ instantiates to 0. Accordingly, if $p_i^1(x_i) < p_j^1(x_j)$ for material i and j, automaton i will have a competitive edge over automaton j.

[3]We emphasize that our scheme does not rely on accessing the unit volume value functions sequentially with frequency proportional to $\vec{x}$ for solving the knapsack problem, however, this restriction is obviously essential for solving the problem *incrementally* and *online*.

In combination, the two situations of the race are designed to stochastically move the current allocation of materials towards the knapsack capacity, with the aim of approaching the optimal expected knapsack value.

Note that because each automaton LA_i acts (e.g., polls a web page) with average frequency proportional to $x_i(t)$, the F_i's are accessed with a frequency corresponding to the current knapsack solution. In other words, our scheme actually applies the *current solution* when seeking its improvement. As we will see in Section III the latter property enable us to use the scheme incrementally and online.

III. Application: Polling Frequency Determination

Recent approaches to resource allocation in web monitoring attempt to *optimize* the performance of the system when the monitoring capacity is restricted [1], [2]. The principle cited in the literature essentially invokes the principles of Lagrange multipliers to solve a *nonlinear equality* knapsack problem with a separable and concave objective function [5]. Thus, for example, a basic web monitoring resource allocation problem may involve n web pages that are updated periodically, although with different periods. Clearly, each web page can be polled with a maximum frequency - which would result in a sluggish system. The problem which we study involves determining the web page polling frequencies that maximize the number of web page updates detected, without exceeding the available monitoring capacity.

The approaches that represent the state-of-the-art, assume that the knapsack problem is *deterministic* and fully *known*. However, from a web monitoring perspective the web must often be seen as a stochastic and more or less unknown environment. In such cases, the parameters of the knapsack problem could be *estimated* instead, e.g., based on a tracking phase where web pages are polled mainly for estimation purposes [1], [2]. A main disadvantage of the latter approach is that the parameter estimation phase significantly delays the implementation of the optimal solution. This disadvantage is further aggravated in *dynamic* environments where the optimal solution may change over time, introducing the need for parameter re-estimation.

A. Problem Background

Although several nonlinear MVFs for measuring web monitoring performance have been proposed in the literature (e.g., see [1], [2]), from a broader viewpoint they are mainly built around the basic concept of *update detection probability*, i.e., the probability that polling a web page results in new information being discovered. Therefore, for the purpose of clarification and for the sake of conceptual clarity, we will use the update detection probability as the token of interest in this paper. To further define our notion of web monitoring performance, we consider that time is discrete with the time interval length T to be the atomic unit of decision making. In each time interval every single web page i has a constant probability q_i of remaining *unchanged* [4]. Furthermore, when a web page is updated/changed, the update is available for detection only until the web page is updated again. After that the original update is considered lost. For instance, each time a newspaper web page is updated, previous news items are replaced by the most recent ones.

In the following we will denote the update detection probability of a web page i as d_i. Under the above conditions, d_i depends on the frequency that the page is polled with, and is modelled using the following expression:

$$d_i(x_i) = 1 - {q_i}^{\frac{1}{x_i}}.$$

By way of example, consider the scenario that a web page remains unchanged in any single time step with probability 0.5. Then polling the web page uncovers new information with probability $1 - 0.5^3 = 0.875$ if the web page is polled every 3^{rd} time step (i.e., with frequency $\frac{1}{3}$) and $1 - 0.5^2 = 0.75$ if the web page is polled every 2^{nd} time step. As seen, increasing the polling frequency reduces the probability of discovering new information on each polling.

Given the above considerations, our aim is to find the page polling frequencies $\vec{x}$ that maximize the expected number of pollings uncovering new information per time step:

$$\begin{array}{ll} \text{maximize} & \sum_1^n x_i \times d_i(x_i) \\ \text{subject to} & \sum_1^n x_i = c \\ & 0 \leq x_i, \qquad\qquad i = 1, \ldots, n. \end{array}$$

Note that in the general web monitoring case, we are not able to observe $d_i(x_i)$ or q_i directly — polling a web page only reveals whether the web page has been updated *at least once* since our last poll[5]. As such, web monitoring forms a proof-of-concept application for resource allocation in unknown stochastic environments.

B. Learning Automata Knapsack Game Solution

In order to find a LAKG Solution to the above problem we must define the Stochastic Environment that the LA are to interact with. As seen in Section II, the Stochastic Environment consists of the stochastic unit volume value functions $\mathcal{F}' = \{F_1'(x_1), F_2'(x_2), \ldots, F_n'(x_n)\}$. We find these functions by applying the principles of Lagrange multipliers to the above maximization problem. In short, after some simplification, it can be seen that the following conditions characterize the optimal solution:

$$\begin{array}{l} d_1(x_1) = d_2(x_2) = \cdots = d_n(x_n) \\ \sum_1^n x_i = c \\ 0 \leq x_i, \; i = 1, \ldots, n. \end{array}$$

Since we are not able to observe $d_i(x_i)$ or q_i directly, we base our definition of $\mathcal{F}'$ on the result of polling web pages.

[4]Note that in our empirical results, we also report high monitoring performance even with changing q_i. The high performance can be explained by the ability of our scheme to adaptation.

[5]Some web pages are also annotated with the time of last update. However, this information is not generally available/reliable [15], and is therefore ignored in our scheme.

Briefly stated, we want $F_i'(x_i)$ to instantiate to the value 0 with probability $1 - d_i(x_i)$ and to the value 1 with probability $d_i(x_i)$. Accordingly, if the web page i is polled and i has been updated since our last polling, then we consider $F_i'(x_i)$ to have been instantiated to 1. And, if the web page i is unchanged, we consider $F_i'(x_i)$ to have been instantiated to 0.

C. Empirical Results

In this section we evaluate our learning scheme by comparing it with four classical policies using synthetic data. We have implemented the following classical policies:

Uniform: The uniform policy allocates monitoring resources uniformly across all web pages. This is the only classical policy of the four that can be applied directly in an unknown environment.

Proportional: In the proportional policy, the allocation of monitoring resources to web pages is proportional to the update frequencies of the web pages. Accordingly, this policy requires that the web page update frequencies are known.

Estimator: The estimator policy handles unknown web update frequencies by polling web pages *uniformly* in a parameter estimation phase, with the purpose of estimating update frequencies. After the parameter estimation phase, the proportional policy is applied, however, the latter is based on the estimated update frequencies rather than the true ones.

Optimal: The optimal policy requires that web page update frequencies are known, and finds the optimal solution based on the principles of Lagrange multipliers [1], [2].

To evaluate web resource allocation policies, recent work uses the Zipf distribution [16] to generate realistic web page update frequencies [1], [2]. In short, web pages are considered ranked according to their update frequencies, and the update probability of a web page is calculated from its rank. We use the following distribution to determine the update probability of each web page:

$$q_k(\alpha, \beta) = \frac{\alpha}{k^\beta}.$$

Here, k refers to the web page of rank k, the parameter β determines the skewedness of the distribution (e.g., for $s = 0$ the distribution is uniform), and $\alpha \in [0.0, 1.0]$ represents the magnitude of the update probabilities (i.e., the web page of rank 1 is updated with probability α each time step).

Without loss of generality, we normalize the web page polling capacity in our experiments to 1.0 poll per time step, and accordingly, we vary the average total number of web page updates per time step instead.

The results of our experiments are truly conclusive and confirm the power of the LAKG. Although several experiments were conducted using various α, β, and number of automata, we report for the sake of brevity, only the results for 500 web pages (the main case from [1]) within the following environments.

- $\alpha = 0.3$; $\beta = 1.5$, where the average number of updates per time step is 0.76 and accordingly, below the web page polling capacity. The web page update distribution is highly skewed, as explored in [1].

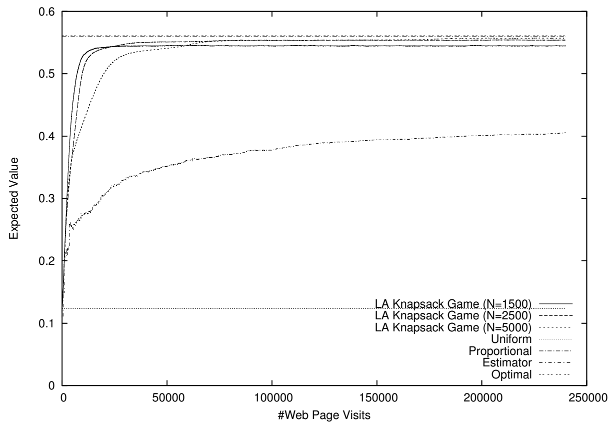

Fig. 2. In the $(\alpha = 0.3, \beta = 1.5)$-environment, the LA Knapsack scheme is superior to the estimator scheme, allowing increased convergence accuracy at the cost of convergence speed, without any parameter estimation phase.

- $\alpha = 0.3$; $\beta = 1.0$, where the average number of updates per time step is increased to 2.0 (twice the web page polling capacity) by making the web page update distribution less skewed (the normal Zipf distribution).
- $\alpha = 0.9$; $\beta = 1.5$, where the average number of updates is set to 2.3 by increasing the web page update probability. Because of the high values of both α and β, this environment turns out to be the most challenging one, discriminating clearly between the optimal policy and the proportional policy.

For these values, an ensemble of several independent replications with different random number streams was performed to minimize the variance of the reported results.

1) Static Environments: We see from Fig. 2 that the proportional policy and the optimal policy provide more-or-less the same solution — a solution superior to the uniform policy solution. We also observe that the performance of the estimator scheme increases steadily with the length of the parameter estimation phase. The figure also shows the performance of the three different LAKG configurations reported in this section, i.e., $[N = 1500, \gamma = 1.3]$, $[N = 2500, \gamma = 1.3]$, and $[N = 5000, \gamma = 1.2]$. As seen, the performance of the LAKG increases significantly quicker compared to the Estimator scheme, and in contrast to the Estimator scheme, the performance is improved online (in a real-time manner) *without* invoking any parameter estimation phase. We also observe that, as expected, increasing the resolution, N, improves the convergence accuracy at the cost of convergence speed.

As seen in Fig. 3, a less skewed web page update distribution function makes the uniform policy more successful, mainly because a larger number of web pages will have a significant probability of being updated. For the same reason, the estimator scheme is able to lead to an improved performance quicker. In spite of this, the LAKG yields a superior performance.

The most difficult class of environments we simulate is an environment with a highly skewed web page update distribution $(\beta = 1.5)$ combined with a high update probability

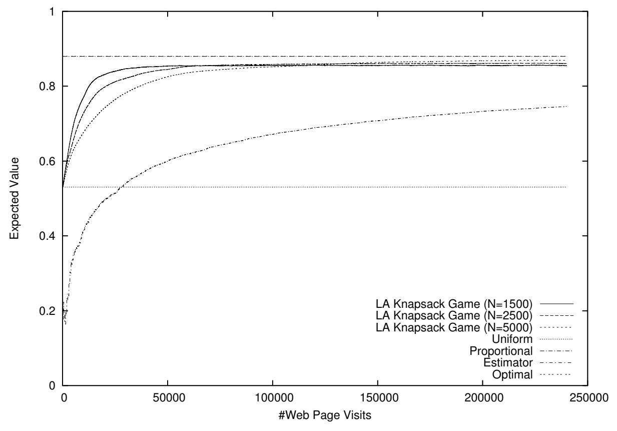

Fig. 3. In the $(\alpha = 0.3, \beta = 1.0)$-environment, a less skewed web page update distribution makes the uniform policy as well as the estimator policy more successful, mainly because of more widely and abundant updating of web pages.

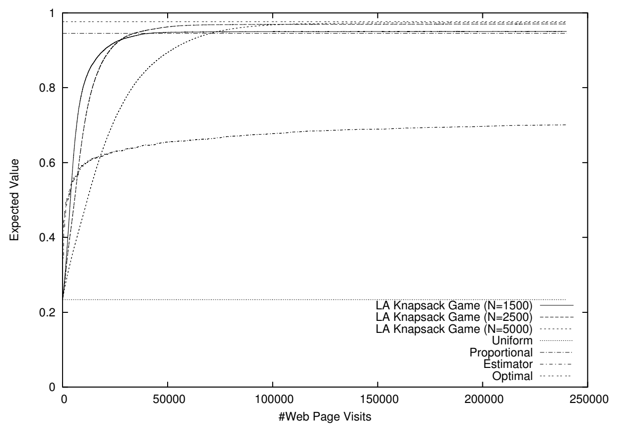

Fig. 4. In the $(\alpha = 0.9, \beta = 1.5)$-environment, the LAKG scheme breaches the performance boundary set by the proportional policy, converging towards near-optimal solutions.

$(\alpha = 0.9)$. In such an environment, the optimal policy performs significantly better than the proportional policy, and so any scheme that converges towards a proportional policy solution will not reach optimal performance. As seen in Fig. 4 the LAKG breaches the performance boundary set by the proportional policy, and converges towards near optimal solutions.

2) Dynamic Environments: A dynamically changing environment is particularly challenging, because the optimal solution is time dependent. In such cases, the current resource allocation solution should be modified according to the environmental changes. When, additionally, the environment and its characteristics are unknown, any changes must first be learnt before any meaningful modification can take place.

In order to simulate a dynamic environment, we change the ranking of the web pages at every r^{th} web page poll — a single web page is selected by sampling from the current Zipf-distribution, and this web page switches rank with the succeeding web page in the ranking. As a result, the Zipf-distribution also changes. This means that the web monitor is

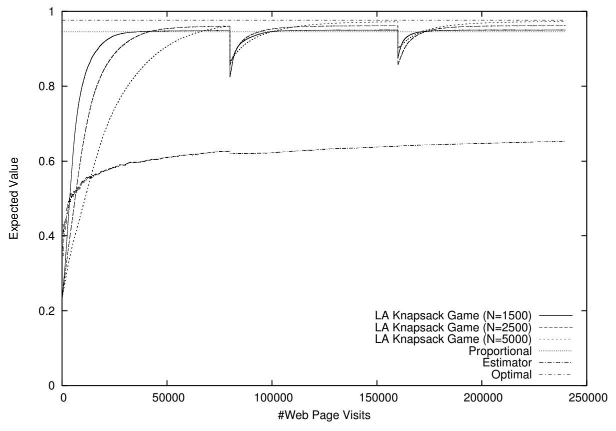

Fig. 5. In the $(\alpha = 0.9, \beta = 1.5)$-environment where the web page ranking changes at every $80,000^{th}$ web page poll. The LAKG finds near-optimal solutions initially, and recovers quickly after environmental changes.

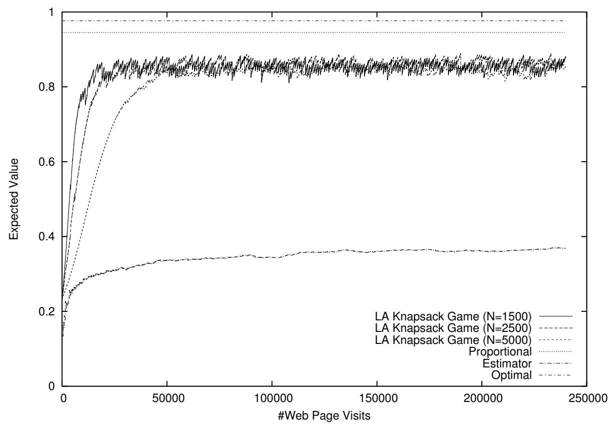

Fig. 6. In the $(\alpha = 0.9, \beta = 1.5)$-environment where the web page ranking changes every $1,000^{th}$ poll. Observe that the LAKG is able to steadily improve the initial solution, but is never allowed to reach an optimal solution.

allowed to conduct r web page polls before the environment changes. Fig. 5 demonstrates the ability of our scheme to re-learn in a switching environment for $r = 80,000$. As seen in the figure, the LAKG quickly recovers after the environment has changed, and then move towards a new near optimal solution. The reader should again note how the speed of recovery depends on the resolution of the automata, N.

In the previous dynamic environment, the LAKG was able to fully recover to a near-optimal solution because of the low frequency of environmental changes. Fig. 6 demonstrates the behavior of the automata in a case when this frequency is increased to $r = 1,000$. As seen, the automata still quickly and steadily improve the initial solution, but are obviously never allowed to reach an optimal solution. However, note how the quickly changing environment is not able to hinder the automata stabilizing on a solution superior to the solutions found by the estimator scheme.

IV. Conclusion and Further Work

In this paper we have considered the problem of resource allocation in web monitoring when the capacity is restricted.

This problem has been modelled in the literature as a knapsack problem with *known deterministic* properties. We have extended the fractional knapsack problem in two ways. First of all, we have treated the unit volume value of each material as a stochastic variable having an unknown distribution. Secondly, we assume that the expected value of a material may decrease after each addition to the knapsack. We have proposed to solve this knapsack problem by using a team of LA that perform a controlled random walk on a discretized fraction space. Comprehensive experimental results have demonstrated the superiority of our scheme that also adapts to dynamic environments.

In our further work we aim to develop a so-called *variable structure* automaton for the knapsack problem, and to verify its ϵ-optimality.

REFERENCES

[1] S. Pandey, K. Ramamritham, and S. Chakrabarti, "Monitoring the Dynamic Web to Respond to Continuous Queries," in *12th International World Wide Web Conference*. ACM Press, 2003, pp. 659–668.

[2] J. L. Wolf, M. S. Squillante, J. Sethuraman, and K. Ozsen, "Optimal Crawling Strategies for Web Search Engines," in *11th International World Wide Web Conference*. ACM Press, 2002, pp. 136–147.

[3] K. M. Bretthauer and B. Shetty, "The Nonlinear Knapsack Problem — Algorithms and Applications," *European Journal of Operational Research*, vol. 138, pp. 459–472, 2002.

[4] Q. Zhu and B. J. Oommen, "Some Fundamental Results on Optimal Search with Unknown Target Distributions," *Submitted for Publication*, 2005.

[5] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, 2004.

[6] P. E. Black, "Fractional knapsack problem," *Dictionary of Algorithms and Data Structures*, 2004.

[7] B. Fox, "Discrete optimization via marginal analysis," *Management Sciences*, vol. 13, no. 3, pp. 211–216, 1966.

[8] B. C. Dean, M. X. Goemans, and J. Vondrdk, "Approximating the Stochastic Knapsack Problem: the benefit of adaptivity," in *45th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 2004, pp. 208–217.

[9] E. Steinberg and M. S. Parks, "A Preference Order Dynamic Program for a Knapsack Problem with Stochastic Rewards," *The Journal of the Operational Research Society*, vol. 30, no. 2, pp. 141–147, 1979.

[10] K. W. Ross and D. Tsang, "The stochastic knapsack problem," *IEEE Transactions on Communications*, vol. 37, no. 7, 1989.

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

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

[13] M. L. Tsetlin, *Automaton Theory and Modeling of Biological Systems*. Academic Press, 1973.

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

[15] C. Castillo, "Effective web crawling," Ph.D. dissertation, University of Chile, 2004.

[16] G. Zipf, *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.