# Gaussian Process Based Optimistic Knapsack Sampling with Applications to Stochastic Resource Allocation

**Sondre Glimsdal and Ole-Christoffer Granmo**
Department of ICT
University of Agder
Norway
{sondre.glimsdal, ole.granmo}@uia.no

**Abstract**

The stochastic non-linear fractional knapsack problem is a challenging optimization problem with numerous applications, including resource allocation. The goal is to find the most valuable mix of materials that fits within a knapsack of fixed capacity. When the value functions of the involved materials are fully known and differentiable, the most valuable mixture can be found by direct application of Lagrange multipliers. However, in many real-world applications, such as web polling, information about material value is uncertain, and in many cases missing altogether. Surprisingly, without prior information about material value, the recently proposed Learning Automata Knapsack Game (LAKG) offers arbitrarily accurate convergence towards the optimal solution, simply by interacting with the knapsack on-line.

This paper introduces Gaussian Process based Optimistic Knapsack Sampling (GPOKS) — a novel model-based reinforcement learning scheme for solving stochastic fractional knapsack problems, founded on Gaussian Process (GP) enabled Optimistic Thompson Sampling (OTS). Not only does this scheme converge significantly faster than LAKG, GPOKS also incorporates GP based learning of the material values themselves, forming the basis for OTS supported balancing between exploration and exploitation. Using resource allocation in web polling as a proof-of-concept application, our empirical results show that GPOKS consistently outperforms LAKG, the current top-performer, under a wide variety of parameter settings.

## 1 Introduction

The Internet can be seen as a massive collection of ever-changing information, continuously evolving as web resources are created, edited, deleted, and replaced (Pandey, Ramamritham, & Chakrabarti 2003). Obtaining adequate information from the Internet is crucial for many tasks, including social media analytics, counter terrorism, and business intelligence. It is thus important that the applied search engines and web-monitoring frameworks are able to keep their indexes and caches complete and up-to-date. Achieving this, of course, relies on detecting the changes that the web resources undergo, typically by means of polling.

The problem of balancing polling capacity optimally among web resources, with limited prior information, was essentially unsolved until the Learning Automata Knapsack Game (LAKG) was introduced in 2006 as a generic and adaptive solution to the so-called *Stochastic Non-linear Equality Fractional Knapsack (NEFK) Problem* (Granmo *et al.* 2006). Before that, the simplest and perhaps most common polling approach was to allocate the available polling capacity uniformly among the web resources being monitored, polling them all with the same fixed frequency, constrained by the available polling capacity. This uniform polling strategy is clearly sub-optimal since web resources evolve at different speed. For slowly changing web resources, a high polling frequency translates into a correspondingly large number of unfruitful polls. Conversely, for quickly evolving web resources, a too low polling frequency leads to potential loss of information or acting on out-dated information. In brief, without balancing the allocation of the available polling capacity, wasting resources polling one resource may in turn prevent us from polling another more attractive resource, thus degrading overall performance.

A two phase strategy has been proposed to address the latter inefficiency: In the first phase, the uniform strategy is applied, which allows the update probability of monitored web resources to be estimated. By treating these probability estimates as the true ones, Lagrange multipliers can be applied to find an allocation of capacity that is optimal for the *estimated* values (Pandey, Ramamritham, & Chakrabarti 2003). However, this method needs an arbitrary long estimation phase to approach the optimal solution in the second phase. That is, one either has to accept a sub-optimal final solution because the update probability estimates are inaccurate, or one must wait an extensive amount of time till the estimates have become sufficiently accurate, allowing a better solution in the second phase. Also note that evolving update probabilities render the solution found with the latter approach progressively more inaccurate.

This paper introduces Gaussian Process based Optimistic Knapsack Sampling (GPOKS) — a novel scheme for solving stochastic knapsack problems founded on Gaussian Process (GP) (Rasmussen & Williams 2006) based Thompson Sampling (TS) (Thompson 1933; Granmo 2010), enhanced by the principles of *Optimistic* TS (May *et al.* 2012). As we shall see, not only does this scheme converge significantly faster than LAKG, GPOKS also incorporates GP based learning of the material unit values themselves, form-

ing the basis for TS based exploration and exploitation. This allows GPOKS to gradually shift from estimation to optimization, starting with pure estimation and converging towards pure optimization.

In (Granmo 2010) we reported a *Bayesian* technique for solving bandit like problems, revisiting the *Thompson Sampling* (Thompson 1933) principle pioneered in 1933. This revisit lead to novel schemes for handling multi-armed and dynamic (restless) bandit problems (Granmo & Berg 2010; Gupta, Granmo, & Agrawala 2011a; 2011b), and empirical results demonstrated the advantages of these techniques over established top performers. Furthermore, we provided theoretical results stating that the original technique is instantaneously self-correcting and that it converges to only pulling the optimal arm with probability as close to unity as desired. We now expand this principle to support Thompson Sampling for Stochastic NEFK Problems.

### 1.1 Formal Problem Formulation

In order to appreciate the qualities of the Stochastic NEFK Problem, it is beneficial to view the problem in light of the classical *linear* Fractional Knapsack (FK) Problem. Indeed, the Stochastic NEFK Problem generalizes the latter problem in two significant ways. Both of the two problems are *briefly* defined below.

**The Linear Fractional Knapsack (FK) Problem**: 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, $1 \leq i \leq n$, where 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 $\sum_1^n f_i(x_i)$ (Black 2004).

**The Nonlinear Equality FK (NEFK) Problem**: One important extension of the above classical problem is the *Nonlinear Equality* FK problem with a separable and concave objective function. The problem can be stated as follows (Kellerer, Pferschy, & Pisinger 2004):

$$\begin{array}{ll} \text{maximize} & f(\vec{x}) = \sum_1^n f_i(x_i) \\ \text{subject to} & \sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0. \end{array}$$

Since the objective function is considered to be concave, the value function $f_i(x_i)$ of each material is also concave. This means that the derivatives of the material value functions $f_i(x_i)$ with respect to $x_i$, (hereafter denoted $f_i'$), are non-increasing. In other words, the material value *per unit volume* is no longer constant as in the linear case, but decreases with the material amount, and so the optimization problem becomes:

$$\begin{array}{ll} \text{maximize} & f(\vec{x}) = \sum_1^n f_i(x_i), \\ & \text{where } f_i(x_i) = \int_0^{x_i} f_i'(x_i) dx_i \\ \text{subject to} & \sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0. \end{array}$$

Efficient solutions to the latter problem, based on the principle of Lagrange multipliers, have been devised. In short, the optimal value occurs when the derivatives $f_i'$ of the material value functions are equal, subject to the knapsack constraints (Bretthauer & Shetty 2002):

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

**The Stochastic NEFK Problem**: In this paper we generalize the above nonlinear equality knapsack problem. First of all, we let the material value per unit volume for any $x_i$ be a *probability* function $p_i(x_i)$. Furthermore, we consider the distribution of $p_i(x_i)$ to be *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 $p_i(x_i)$ at $x_i$, and not $p_i(x_i)$ itself.[1] Given this stochastic environment, we intend to devise an on-line incremental scheme that learns the mix of materials of maximal *expected* value, through a series of informed guesses. Thus, to clarify issues, we are provided with a knapsack of fixed volume $c$, which is to be filled with a mix of $n$ different materials. However, unlike the NEFK, in the Stochastic NEFK Problem the unit volume value of a material $i$, $1 \leq i \leq n$, is a random quantity — it takes the value 1 with probability $p_i(x_i)$ and the value 0 with probability $1-p_i(x_i)$, respectively. As an additional complication, $p_i(x_i)$ is nonlinear in the sense that it decreases monotonically with $x_i$, i.e., $x_{i_1} \leq x_{i_2} \Leftrightarrow p_i(x_{i_1}) \geq p_i(x_{i_2})$.

Since unit volume values are random, we operate with expected unit volume values rather than the actual unit volume values. With this understanding, and the above perspective in mind, the expected value of the amount $x_i$ of material $i$, $1 \leq i \leq n$, becomes $f_i(x_i) = \int_0^{x_i} p_i(u)du$. Accordingly, the expected value per unit volume[2] of material $i$ becomes $f_i'(x_i) = p_i(x_i)$. In this stochastic and non-linear version of the FK problem, the goal is to fill the knapsack so that the expected value $f(\vec{x}) = \sum_1^n f_i(x_i)$ of the material mix contained in the knapsack is maximized. Thus, we aim to:

$$\begin{array}{ll} \text{maximize} & f(\vec{x}) = \sum_1^n f_i(x_i), \\ & \text{where } f_i(x_i) = \int_0^{x_i} p_i(u)du, p_i(x_i) = f_i'(x_i) \\ \text{subject to} & \sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0. \end{array}$$

A fascinating property of the above problem is that the amount of information available to the decision maker is limited — the decision maker is only allowed to observe the current unit value of each material (either 0 or 1). That is, each time a material mix is placed in the knapsack, the unit value of each material is provided to the decision maker. The actual outcome probabilities $p_i(x_i), 1 \leq i \leq n$, however, remain *unknown*. As a result of the latter, the expected value of the material mix must be maximized by means of trial-and-error, i.e., by experimenting with different material mixes and by observing the resulting random unit value outcomes.

[1] For the sake of consistency with previous work on the Stochastic NEFK Problem, we here model stochastic material unit values using Bernoulli trials. However, since GPOKS is based on Gaussian Processes, the central limit theorem opens up for addressing a number of other distributions too. Furthermore, there exist dedicated kernel functions for a variety of distributions.

[2] We hereafter use $f_i'(x_i)$ to denote the derivative of the expected value function $f_i(x_i)$ with respect to $x_i$.

### 1.2 Paper Contributions

The contributions of this paper can be summarized as follows:

1. We combine Bayesian modeling with reinforcement learning to provide a novel solution to the Stochastic NEFK Problem.
2. We propose the first reinforcement learning scheme that combines Gaussian Processes (Rasmussen & Williams 2006) with Thompson Sampling (Thompson 1933; Granmo 2010).
3. We introduce GP based sampling mechanisms in the spirit of Optimistic Thompson Sampling (May *et al.* 2012) for increased performance.
4. The resulting scheme persistently outperforms state-of-the-art approaches when applied to resource allocation in web polling.

These contributions form the first steps towards establishing a new family of reinforcement learning schemes that provide on-line solutions to stochastic versions of classical optimization problems. This is achieved by carefully designing Bayesian models that capture the nature of the optimization problems, applying TS principles to address the exploration/exploitation dilemma in on-line learning and control.

### 1.3 Paper Outline

In Section 2, we present our scheme for Gaussian Process Based Optimistic Knapsack Sampling (GPOKS). We start with a brief introduction to Gaussian Processes before we propose how Gaussian Processes can enable Thompson Sampling — the current leader when it comes to solving Bernoulli Bandit Problems (Granmo 2010) — for exploration and exploitation when solving on-line Stochastic NEFK problems. Then, in Section 3, we define the web resource allocation polling problem in more detail, following up with an evaluation of GPOKS compared with state-of-the-art. We conclude in Section 4 and present pointers for further work.

## 2 Gaussian Process Based Optimistic Knapsack Sampling (GPOKS)

The conflict between exploration and exploitation is a well-known problem in reinforcement learning, and other areas of artificial intelligence. The multi-armed bandit problem captures the essence of this conflict, and has thus occupied researchers for over fifty years (Wyatt 1997). In brief, an agent sequentially pulls one of multiple arms attached to a gambling machine, with each pull resulting in a random reward. The reward distributions are unknown, and thus, one must balance between exploiting existing knowledge about the arms, and obtaining new information.

We are here facing a similar problem, however, instead of seeking the singly best material (bandit arm), we need to find a mixture of materials, also referred to as a *mixed strategy* in Game Theory. Recently, GP optimization has been addressed from a bandit problem perspective (Srinivas N. & M. 2010), allowing the GP to be explored globally with as few evaluations as possible based on so-called upper confidence bounds. Inspired by the success of GP based optimization, we here propose a novel GP based model for stochastic NEFK problems, where a *collection* of GPs captures the individual material unit values. Based on the GP colletion, Thompson Sampling is applied to sample likely deterministic NEFK problem instances from the GPs. These, in turn, are solved based on Lagrange Multipliers, producing a *potential* solution to the problem at hand.

### 2.1 Gaussian Processes based Representation of Material Unit Value

A Gaussian Process (GP) is a stochastic process that represents a function as a multivariate Gaussian distribution (Rasmussen & Williams 2006). It is specified as a tuple $\mathcal{GP} = (\mu(\vec{x}), K(\cdot,\cdot))$ where $\mu(\cdot)$ is the mean function, typically assigned $\mu(\vec{x}) = \vec{0}$, and $K(\cdot,\cdot)$ is a kernel that specifies the covariance matrix for the random vector $\vec{x}$. In this paper, we use the one dimensional *Squared Exponential* kernel (eq. 1), configured by the hyper parameters $\vec{\theta} = \{l, \sigma_f^2, \sigma_n^2\}$.

$$K(x_p, x_q) = \sigma_f^2 \exp(-\frac{1}{2l^2}(x_p - x_q)^2)) + \sigma_n^2 \delta_{pq} \quad (1)$$

Here $l$ is the characteristic length-scale parameter that determines how rapidly the correlation should decay as the distance between $x_p$ and $x_q$ increases, $\sigma_f^2$ is the signal variance and $\sigma_n^2$ is white noise (note that $\delta_{pq}$ here denotes the Kronecker delta between $x_p$ and $x_q$). For further information on GPs we refer to (Rasmussen & Williams 2006).

By way of example, Figure 1 illustrates how the posterior distribution over possible material unit value functions for a given material $i$ can be represented by means of a GP. The $x$-axis measures the amount of material, $x_i$, while the $y$-axis provides the material unit value $f_i'(x_i)$. The mean and 95% confidence interval is included, as well as four samples indicating possible candidates for $f_i'(x_i)$. Note that since the Stochastic NEFK problem deals with non-increasing unit value functions, $f_i'(x_i)$, we apply Rejection Sampling to sample from the distribution of non-increasing functions. Similarly, "optimistic" sampling, as pioneered by May et al. (May *et al.* 2012), is realized by rejecting sampled functions that drop below the estimated mean.

### 2.2 Architectural Overview of GPOKS

Figure 2 provides an architectural overview of our scheme. As illustrated in the figure, GPOKS operates as follows:

1. A collection of GPs, one Gaussian Process, $\mathrm{GP}_i$, for each material $i$, attempts to estimate the material unit value functions, $f_i'(x_i), 1 \leq i \leq n$.
2. One candidate material unit value function, $\hat{f}_i'(x_i), 1 \leq i \leq n$, is then sampled from each $\mathrm{GP}_i$ , thus applying the TS principle of sampling functions proportionally to their likelihoods.
3. The *DET-KS* component in the architectue finds the optimal material mixture $\hat{\mathbf{M}} = [\mathbf{x_1}, \ldots, \mathbf{x_n}]$ for the sampled material unit value functions, $\hat{f}_i'(x_i), 1 \leq i \leq n$, using Lagrange multipliers.

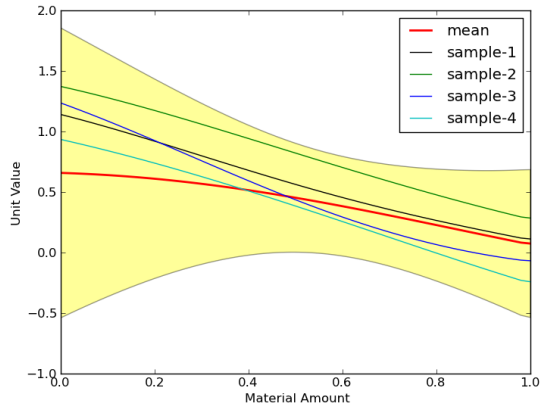


Figure 1: Gaussian Process based representation of material unit value

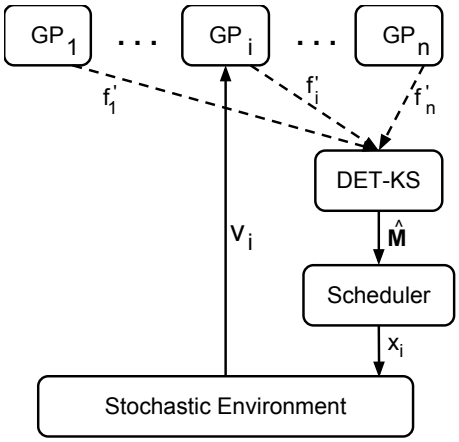


Figure 2: GPOKS Architectural Overview

4. One of the materials is then selected by the *Scheduler* component for evaluation, ensuring that each material $i$ is selected with a frequency that is proportional to the amount of material, $x_i$, assigned by $\hat{\mathbf{M}}$.
5. Finally, the *Stochastic Environment*, i.e., the Stochastic NEFK, samples the true outcome probability function, $p_i(x_i)$, at $x_i$, providing feedback $v_i$ to the corresponding $\mathrm{GP}_i$, which updates its Bayesian estimate of $f_i'(x_i)$.

By following the above steps our goal is to gradually improve our "best guesses" so that each iteration successively brings us closer to the optimal solution of the targeted Stochastic NEFK problem.

### 2.3 Example Steps

Figure 3 and 4 show the GP based estimates for the unit value of two materials, $f_1'(x_1)$ and $f_2'(x_2)$, after only 5 material value observations. As can be seen, uncertainty about the material unit value functions is significant, and the estimated optimal material amounts $\hat{M} = [\hat{x}_1, \hat{x}_2]$ are far from

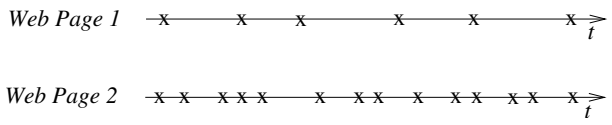


Figure 6: Web resource changes occurring over time. An 'x' on the time-lines denotes that the respective web resource has changed.

the optimal amounts $M = [x_1, x_2]$.

However, after 193 iterations of the GPOKS algorithm, we observe a number of fascinating properties in Figure 5. First of all, the Bayesian estimates of the material unit values, $f_1'(x_1)$ and $f_2'(x_2)$, have become more accurate. Furthermore, we observe that the estimated optimal material mixture is now much closer to the optimal mixture. Finally, observe that the uncertainty concerning $f_1'(x_1)$ and $f_2'(x_2)$ varies with $x_1$ and $x_2$. The beauty of Thompson Sampling is that the observations are collected with gradually increasing exploitation, zooming in on the areas that are most likely to contain the optimal material mixture.

## 3 Application: Web Polling

Having obtained a solution to the model in which we set the NEFK, we shall now demonstrate how we can utilize this solution for the current problem being studied, namely, the optimal web-polling problem.

Web resource monitoring consists of repeatedly polling a selection of web resources so that the user can detect changes that occur over time. Clearly, as this task can be prohibitively expensive, in practical applications, the system imposes a constraint on the *maximum* number of web resources that can be polled per time unit. This bound is dictated by the governing communication bandwidth, and by the speed limitations associated with the processing. Since only a fraction of the web resources can be polled within a given unit of time, the problem which the system's analyst encounters is one of determining which web resources are to be polled. In such cases, a reasonable choice of action is to choose web resources in a manner that maximizes the number of changes detected, and the optimal allocation of the resources involves trial-and-error. As illustrated in Figure 6, web resources may change with varying frequencies (that are unknown to the decision maker), and changes appear more or less randomly. Furthermore, as argued elsewhere, (Granmo & Oommen 2006; Granmo *et al.* 2006; 2007), the probability that an individual web resource poll uncovers a change on its own decreases monotonically with the polling frequency used for that web resource.

Although several nonlinear criterion functions for measuring web monitoring performance have been proposed in the literature (e.g., see (Pandey, Ramamritham, & Chakrabarti 2003; Wolf *et al.* 2002)), from a broader viewpoint they are mainly built around the basic concept of *update detection probability*, i.e., the probability that polling a web resource results in new information being discovered. Therefore, for the purpose of conceptual clarity, we will use

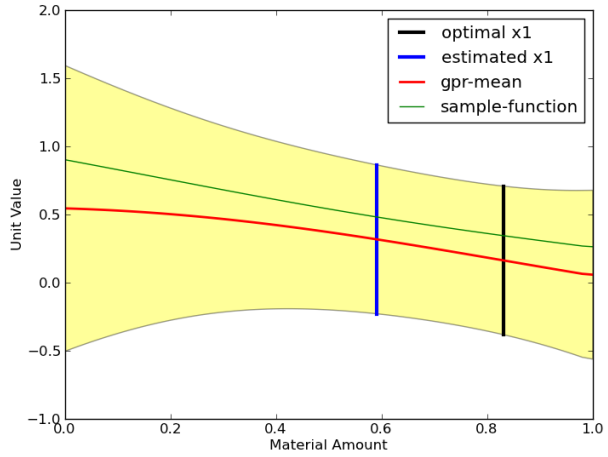


Figure 3: Estimate of material unit value $f_1'(x_1)$ after 7 observations, with optimal and estimated material amounts $x_1$.

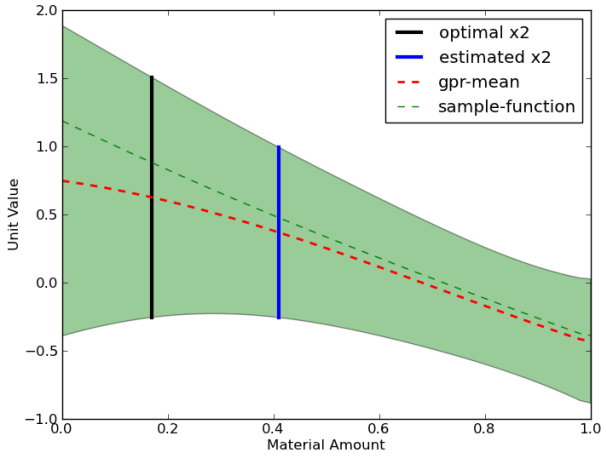


Figure 4: Estimate of material unit value $f_2'(x_2)$ after 7 observations, with optimal and estimated material amounts $x_2$.

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 resource $i$ has a constant probability $q_i$ of remaining *unchanged*. Furthermore, when a web resource is updated/changed, the update is available for detection only until the web resource is updated again. After that, the original update is considered lost. For instance, each time a newspaper web resource 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 resource $i$ as $d_i$. Under the above conditions, $d_i$ depends on the frequency, $x_i$, that the resource is polled with, and is modeled 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 resource remains unchanged in any single time step with probability $0.5$. Then polling the web resource uncovers new information with probability $1 - 0.5^3 = 0.875$ if the web resource is polled every $3^{rd}$ time step (i.e., with frequency $\frac{1}{3}$) and $1 - 0.5^2 = 0.75$ if the web resource 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 resource polling frequencies $\vec{x}$ that maximize the expected number of pollings uncovering new information per time step:

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

### 3.1 GPOKS Solution

In order to find a solution to the above problem we must define the Stochastic Environment that GPOKS is to interact with. As seen in Section 2, the Stochastic Environment consists of the unit volume value functions $\{f_1'(x_1), f_2'(x_2), \ldots, f_n'(x_n)\}$, which are unknown to GPOKS. We identify the nature of these functions by applying the principle 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:

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

Since we are not able to observe $d_i(x_i)$ or $q_i$ directly, we base our definition of $\{f_1'(x_1), f_2'(x_2), \ldots, f_n'(x_n)\}$ on the result of polling web resources. 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 resource $i$ 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 resource $i$ is unchanged, we consider $f_i'(x_i)$ to have been instantiated to $0$.

### 3.2 Empirical Results

In this section we evaluate GPOKS and compare its performance with the currently best performing algorithm, LAKG. While H-TRAA possesses better scalability than LAKG (Granmo & Oommen 2010), for two material problems, their performance is identical because the hierarchical setup of H-TRAA does not come into play. For clarification we will also include some promising variants of GPOKS. Here follows an overview of a selection of the policies that we have investigated:

**Uniform**: The uniform policy allocates monitoring resources uniformly across all web resources. This classical

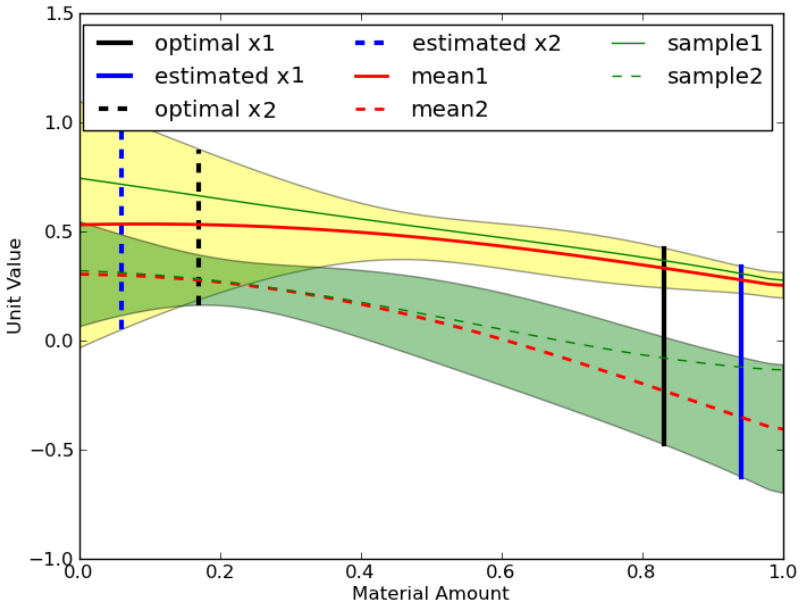


Figure 5: Estimate of material unit values $f_1'(x_1)$ and $f_2'(x_2)$ after 193 observations, with optimal and estimated material amounts $x_1$ and $x_2$.

policy can, of course, be applied directly in an unknown environment.

**LAKG**: The LAKG scheme is basically a game between so-called Learning Automata (Narendra & Thathachar 1989). They start off from a uniform policy and gradually improves toward the optimal configuration through a sequence of small jumps across a discretized search space. In all our experiments the resolution of LAKG is set to 100 states.

**Optimal**: This policy requires that update frequencies are known, and finds the optimal solution based on the principle of Lagrange multipliers (Pandey, Ramamritham, & Chakrabarti 2003; Wolf *et al.* 2002).

**GPOKS - Mean**: To highlight the advantage of our Optimistic Thompson Sampling approach, we also test a simpler scheme where we use the mean of the GPs when estimating the optimal solution rather than sampling functions from the GPs.

We have conducted numerous experiments using various configurations, such as different noise parameters and update probabilities. Here, we present a representative subset of these, as they all show the same trend. Performance is measured as the average *accumulated* number of web resource updates found.

For these experiments, we used an ensemble of 1000 independent replications, each random generator seeded with a unique number, to maximize the precision of the reported results. In order to provide a robust overview of the performance of GPOKS, we investigated three radically different update probability configurations for web resource pairs. In the first one, $q_1 = 0.9/q_2 = 0.1$, one web resource is updated significantly more often than the other. A more moderate version of the latter configuration, $q_1 = 0.75/q_2 = 0.25$, was also investigated. Furthermore, we measured performance when the two web resources have almost equal update probability, $q_1 = 0.55/q_2 = 0.45$. Finally, we also investigated the robustness of GPOKS by adding increasing amount of white-noise, ($w_\sigma$), to the feedback given to GPOKS. Note that, for the sake of fairness, we applied the same kernel hyper-parameters, $\theta = \{1.0, 1.0, 0.1\}$, for all the GP based strategies, without further optimization.

Table 1 reports the performance of the different policies[3]. As can be seen, GPOKS clearly outperforms LAKG when facing the $q_1 = 0.9/q_2 = 0.1$ configuration, with GPOKS detecting on average approximately 8 more updates than LAKG over 1000 time steps. Also note how remarkably close GPOKS gets to the optimal performance, missing on average merely 7 web resource updates over 1000 time steps. We observe similar results for the $q_1 = 0.75/q_2 = 0.25$ configuration. Finally, for the $q_1 = 0.55/q_2 = 0.45$ configuration, we observe that the performance of LAKG and GPOKS becomes more similar. This can be explained by the prior bias of LAKG, starting from a uniform allocation of resources. This gives LAKG an advantage over GPOKS, which are largely unbiased when it comes to prior belief about update probabilities. Finally, notice the performance loss caused by using the mean of the GPs (GPOKS-Mean) instead of TS. This trend is further explored in Ta-

[3]Note that all of the setups apply a small degree of white noise ($w_\sigma = 0.1$).

ble 2, where we increase the amount of white noise affecting feedback. We then observe that GPOKS is surprisingly robust towards noisy feedback compared to GPOKS-Mean. This can be explained by the greedy nature of GPOKS-Mean, which is less inclined to explore the space of functions encompassed by the GPs, thus being more easily mislead by noise.

## 4 Conclusions and Further Work

The stochastic non-linear fractional knapsack problem is a challenging optimization problem with numerous applications, including resource allocation. The goal is to find the most valuable mix of materials that fits within a knapsack of fixed capacity. When the value functions of the involved materials are fully known and differentiable, the most valuable mixture can be found by direct application of Lagrange multipliers.

In this paper we introduced Gaussian Process based Optimistic Knapsack Sampling (GPOKS) — a novel model-based reinforcement learning scheme for solving stochastic fractional knapsack problems. The scheme is founded on Gaussian Process (GP) enabled Optimistic Thompson Sampling (OTS). Our empirical results demonstrates that this scheme converge significantly faster than LAKG. Furthermore, GPOKS incorporates GP based learning of the material unit values themselves, forming the basis for OTS supported balancing between exploration and exploitation. Using resource allocation in web polling as a proof-of-concept application, our empirical results show that GPOKS consistently outperforms LAKG, the current top-performer, under a wide variety of parameter settings.

In our further work, we will address games of interacting GPOKS for solving networked and hierarchical resource allocation problems. Furthermore, we are investigating techniques for decomposing the GP calculations for increased computational performance.

## References


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

Bretthauer, K. M., and Shetty, B. 2002. The Nonlinear Knapsack Problem — Algorithms and Applications. *European Journal of Operational Research* 138:459–472.

Granmo, O.-C., and Berg, S. 2010. Solving Non-Stationary Bandit Problems by Random Sampling from Sibling Kalman Filters. In *Proceedings of the Twenty Third International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems (IEA-AIE 2010)*, 199–208. Springer.

Granmo, O.-C., and Oommen, B. J. 2006. On Allocating Limited Sampling Resources Using a Learning Automata-based Solution to the Fractional Knapsack Problem. In *Proceedings of the 2006 International Intelligent Information Processing and Web Mining Conference (IIS:IIPW'06)*, Advances in Soft Computing. Springer.

Granmo, O.-C., and Oommen, B. J. 2010. Solving Stochastic Nonlinear Resource Allocation Problems Using a Hierarchy of Twofold Resource Allocation Automata. *IEEE Transactions on Computers* 59(4):545–560.

Granmo, O.-C.; Oommen, B. J.; Myrer, S. A.; and Olsen, M. G. 2006. Determining Optimal Polling Frequency Using a Learning Automata-based Solution to the Fractional Knapsack Problem. In *Proceedings of the 2006 IEEE International Conferences on Cybernetics & Intelligent Systems (CIS) and Robotics, Automation & Mechatronics (RAM)*. IEEE.

Granmo, O.-C.; Oommen, B. J.; Myrer, S. A.; and Olsen, M. G. 2007. 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* 37(1):166–175.

Granmo, O.-C. 2010. Solving Two-Armed Bernoulli Bandit Problems Using a Bayesian Learning Automaton. *International Journal of Intelligent Computing and Cybernetics* 3(2):207–234.

Gupta, N.; Granmo, O.-C.; and Agrawala, A. 2011a. Successive Reduction of Arms in Multi-Armed Bandits. In *Proceedings of the Thirty-first SGAI International Conference on Artificial Intelligence (SGAI 2011)*. Springer.

Gupta, N.; Granmo, O.-C.; and Agrawala, A. 2011b. Thompson Sampling for Dynamic Multi-Armed Bandits. In *Proceedings of the Tenth International Conference on Machine Learning and Applications (ICMLA'11)*. IEEE.

Kellerer, H.; Pferschy, U.; and Pisinger, D. 2004. *Knapsack Problems*. Springer.

May, B. C.; Korda, N.; Lee, A.; and Leslie, D. S. 2012. Optimistic bayesian sampling in contextual-bandit problems. *J. Mach. Learn. Res.* 8:2069–2106.

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

Pandey, S.; Ramamritham, K.; and Chakrabarti, S. 2003. Monitoring the Dynamic Web to Respond to Continuous Queries. In *12th International World Wide Web Conference*, 659–668. ACM Press.

Rasmussen, C. E., and Williams, C. K. I. 2006. *Gaussian Processes for Machine Learning*. The MIT Press.

Srinivas N., Krause A., K. S., and M., S. 2010. Gaussian process optimization in the bandit setting: No regret and experimental design. In Fürnkranz, J., and Joachims, T., eds., *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 1015–1022. Haifa, Israel: Omnipress.

Thompson, W. R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25:285–294.

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

Wyatt, J. 1997. *Exploration and Inference in Learning from Reinforcement*. Ph.D. Dissertation, University of Edinburgh.

| Scheme | $p_1/p_2$ | Avg[#Updates] t=10 | Avg[#Updates] t=100 | Avg[#Updates] t=1000 |
|---|---|---|---|---|
| Optimal | 0.90/0.10 | 9.1 | 91.0 | 909.9 |
| Uniform | 0.90/0.10 | 5.9 | 59.0 | 590.0 |
| LAKG | 0.90/0.10 | 6.0 | 71.6 | 874.9 |
| GPOKS | 0.90/0.10 | 8.0 | 88.9 | 903.0 |
| GPOKS-Mean | 0.90/0.10 | 8.5 | 89.7 | 902.9 |
| Optimal | 0.75/0.25 | 8.1 | 81.2 | 812.5 |
| Uniform | 0.75/0.25 | 6.9 | 68.8 | 687.5 |
| LAKG | 0.75/0.25 | 6.9 | 74.1 | 793.1 |
| GPOKS | 0.75/0.25 | 7.4 | 78.8 | 807.9 |
| GPOKS-Mean | 0.75/0.25 | 6.6 | 69.6 | 792.2 |
| Optimal | 0.55/0.45 | 7.5 | 75.2 | 752.5 |
| Uniform | 0.55/0.45 | 7.5 | 74.8 | 747.5 |
| LAKG | 0.55/0.45 | 7.5 | 74.8 | 749.8 |
| GPOKS | 0.55/0.45 | 7.0 | 73.5 | 749.4 |
| GPOKS-Mean | 0.55/0.45 | 5.4 | 52.8 | 725.3 |

Table 1: Average number of updates at different times, $w_\sigma = 0.1$

| Scheme | $p_1/p_2$ | $w_\sigma = 0.0$ | $w_\sigma = 0.2$ | $w_\sigma = 0.4$ |
|---|---|---|---|---|
| GPOKS | 0.75/0.25 | 808.2 | 804.5 | 804.1 |
| GPOKS-Mean | 0.75/0.25 | 793.9 | 787.2 | 769.1 |

Table 2: The performance of GPOKS variants under different levels of white noise