Towards Thompson Sampling for Complex Bayesian Reasoning

Sondre Glimsdal

UiA University of Agder

Doctoral Dissertations at the University of Agder 275

Sondre Glimsdal

Towards Thompson Sampling for Complex Bayesian Reasoning

Doctoral Dissertation for the Degree *Philosophiae Doctor (PhD)* at
the Faculty of Engineering and Science, Specialisation in Information and
Communication Technology

University of Agder
Faculty of Engineering and Science
2020

Doctoral Dissertation at the University of Agder 275
ISSN: 1504-9272
ISBN: 978-82-7117-976-2

Printed by Wittusen & Jensen
Oslo

Preface

This dissertation presents the results of the research I have carried out in my Ph.D. project at the Department of Information Comunication Institute, Faculty of Engineering and Science, University of Agder, Norway. The research goal was to develop and expand the family of Thompson Sampling techniques beyond the domain of vanilla bandit problems, to address more complex optimization problems requiring Bayesian reasoning. The work has been carried out under the supervision of Professor Ole-Christoffer Granmo and Associate Professor Svein Olav Nyberg.

Acknowledgements

I would like to begin by thanking my two supervisors: Professor Ole-Christoffer Granmo and Associate Professor Svein Olav Nyberg. It is an often-used cliché, but in this case, it is no overstatement to say that without the consistent guidance, tutelage, support, unparalleled knowledge, and encouragement of my supervisors, this thesis would never have existed. In particular, I would like to thank Ole-Christoffer who went above and beyond to assist me throughout my Ph.D.

I would also like to thank Professor Andreas Prinz for taking the chance of putting a dyslexic (me) into the integrated Ph.D. program, well knowing that writing is not my strong suit. I am also grateful for the support from my colleges at Confirmit, letting me work on my Ph.D. between juggling different projects.

On a more personal level, I would like to thank all the players and friends at GSI Grizzlies for continuously giving me a venue to output my frustration and excess energy over these years.

Above all, I would like to thank my wife Ann Kristin for her love and constant support, for all the late nights and early mornings, and for keeping me sane over the years. But most of all, thank you for being my best friend. I owe you everything.

Abstract

Thompson Sampling (TS) is a state-of-art algorithm for bandit problems set in a Bayesian framework. Both the theoretical foundation and the empirical efficiency of TS is well-explored for plain bandit problems. However, the Bayesian underpinning of TS means that TS could potentially be applied to other, more complex, problems as well, beyond the bandit problem, if suitable Bayesian structures can be found.

The objective of this thesis is the development and analysis of TS-based schemes for more complex optimization problems, founded on Bayesian reasoning. We address several complex optimization problems where the previous state-of-art relies on a relatively myopic perspective on the problem. These includes stochastic searching on the line, the Goore game, the knapsack problem, travel time estimation, and equipartitioning. Instead of employing Bayesian reasoning to obtain a solution, they rely on carefully engineered rules. In all brevity, we recast each of these optimization problems in a Bayesian framework, introducing dedicated TS based solution schemes. For all of the addressed problems, the results show that besides being more effective, the TS based approaches we introduce are also capable of solving more adverse versions of the problems, such as dealing with stochastic liars.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Introduction

The human capability to make sound decisions in a new situation is based on the ability to reason from knowledge and to learn from previous experience. For instance, when a doctor faces a new patient, she needs to use her medical knowledge, as well as her previous experience with similar symptoms and patients, to set a diagnosis and prescribe a treatment. Similarly, an automated recommender system needs to use customer models (knowledge) and available data, such as the customer's previous purchases, current trends and browsing history (experience), to display advertisements that maximise click-rate.

Both of the above decision problems also introduce another trait that is prevalent for many real-world problems, namely, that decisions need to be made under uncertainty, and in particular, that the effect of the decisions may be partially or even entirely unknown. In the medical decision scenario, for instance, an important indicator symptom might not be present, or a symptom that is not related to the underlying disease could be observed as a red herring. To aggravate, there might be several diseases that correspond with the observed symptoms, and a treatment may have different effects on different patients. Prescribing the most effective treatment can thus be highly difficult.

In some cases, one can fully quantify both uncertainty and the effect of actions. That is, one can quantify how likely each possible scenario is, and one can specify the value (effect) of the available decisions for each of the scenarios. Then the decision that maximizes expected value can be exactly identified using Bayesian reasoning and decision theory [1]. However, for many real-world decision problems, the value of decisions are only partially known, or may even be entirely unknown. The remaining option is then *trial and error*, to learn which decision is best. To avoid unnecessary loss, this decision should be identified as quickly as possible. Of course, identifying the best decision becomes even more challenging when their effects are stochastic.

Decision problems with an intrinsic uncertainty, and that require both reasoning and trial-and-error to make good decisions, arise frequently in practice; for instance: What advertising should be presented to maximize revenue? How often should a web page be visited to observe all changes? Which version of a drug should a patient receive? How should the layout of a warehouse be organized to enable the optimal gathering of orders?

The Multi-Armed Bandit Problem. The latter family of decision problems is both so pervasive and difficult that its most simple form has been formalized as the *Multi-Armed Bandit Problem (MABP)*, still challenging researchers. In MABPs, an agent takes actions sequentially (pulling bandit arms) in an environment, with each action stochastically eliciting a reward. The agent tries to maximize its rewards over time by identifying the optimal action, i.e., the one that has the highest expected reward value. Initially, the agent only knows which actions are offered by the environment, and the only way to learn how the environment responds to the actions is to interact with it.

Interacting with the environment solely for learning the responses can be costly. For instance, nobody wants to run a series of inefficient ads to make certain that the ads are indeed inefficient. So, there is a trade-off between

- gathering new information about the environment and
- executing the action that looks most profitable at the decision point.

Performing this trade-off optimally is the essence of the MABP.

The Exploration-Exploitation Dilemma. The MABP has a long history. The first application of MABP is credited to William R. Thompson in 1933 [2], while the formal definition of MABP is due to Robbins and Herbert [2]. To further analyze this problem, Bush et al. introduced a simple lab experiment called the T-Maze [3], illustrated in Figure 1.1. In this experiment, a rat is placed at the start location, and as it traverses down the limb of the maze, it is faced with a decision: go left or go right? The challenge is that the food is only placed in one of the two locations, and the probability of receiving food is higher for one of the locations. Which location provides food most frequently is initially unknown to the rat. It thus has to learn which decision maximizes the amount of food it receives in subsequent trials.

To repeat this experiment on human subjects, a two-armed bandit machine was commissioned at Harvard by Frederick Mosteller and Robert Bush [3]. They named the machine as a tribute to the single-lever operated slot machine (or bandit, as it stole your money), thus coining the term *Multi-Armed Bandit* (MAB). The original multi-armed bandit machine had two arms, where one arm was more rewarding than the other. The subject was then tasked with playing the machine so as to maximize his reward.

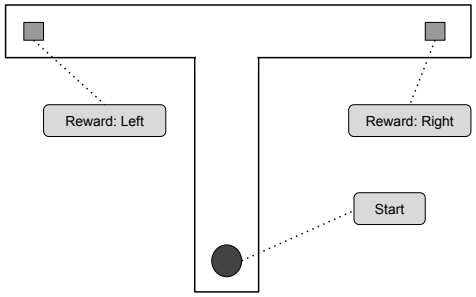

Figure 1.1: The T-Maze problem. A rat is placed at the start of the maze with the objective of learning what arm, left or right, is most rewarding to visit.

The subject is as such faced with a dilemma. The first option is to explore the other arm by pulling it, trying to learn the identity of the superior arm. The other option is to exploit the knowledge that he already has accumulated and continue pulling the arm he just pulled, even though it might be inferior to the other arm. This dilemma is the essence of the MABP and is called the *exploration–exploitation* dilemma or trade-off. The exploration-exploitation trade-off problem was formalized during the second world war by allied researchers and it was thought to be so difficult that it should be given to the German researchers to make them waste their resources on it [4].

Learning Automata. One of the simplest approaches to solving the MAB problem is the concept of Learning Automata. Narendra and Thathachar describes in their book [5] that: *"The concept of an LA grew out of a fusion of the work of psychologists in modeling observed behavior, the efforts of statisticians to model the choice of ex- periments based on past observations, the attempts of operation researchers to im- plement optimal strategies in the context of the two-armed bandit problem, and the endeavors of system theorists to make rational decisions in random environments"*. The origin of LA can be traced back to the two-action Tsetlin Automaton, proposed by M. L. Tsetlin in 1961 [6].

The Tsetlin Automaton has a state space of $2N$ states, where the first N states indicate that one should pull the first arm, and conversely, the last N indicate that the second arm should be pulled. Governed by simple rules to update the state, shown in Figure 1.2, each round of action selection and state updating is extremely fast.

Yet, due to their versatile structure, Tsetlin Automata are simultaneously adaptable to more complex problems, such as resource allocation [7], decentralized control [8], knapsack problems [9], searching on the line [10], meta-learning [11], the satisfiability problem [12], graph colouring [13], preference learning [14], frequent item set mining [15], adaptive sampling [16], equi-partitioning [17], streaming sampling for social activity networks [18], routing bandwidth-guaranteed paths [19], faulty dichotomous search [20], and learning in deceptive environments [21], to give a few examples.

State-of-the-art MABP Algorithms. Over the last decades, specialized MABP

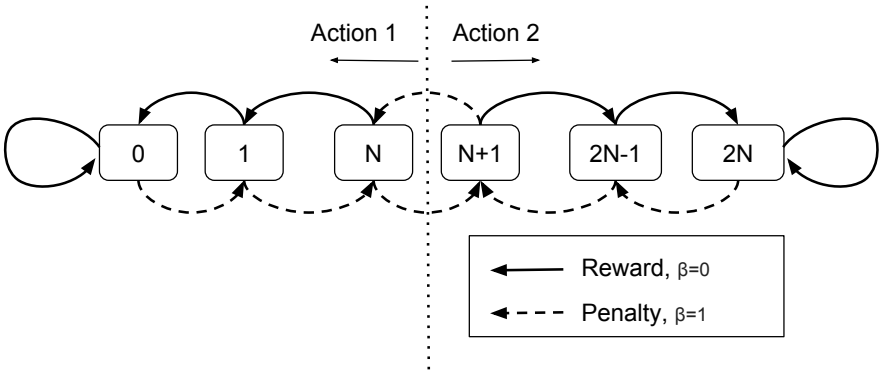

Figure 1.2: The Tsetlin Automaton, if the current state is on the left of the middle, select action α_1 else select action α_2.

solvers have emerged, providing increasingly better trade-offs for the exploration-exploitation dilemma. These advances can be attributed to better models of MABP, with increasing specialization. For instance, one of the more popular MABP algorithms, Upper Confidence Bound (UCB) [22, 23] is based on deriving a Hoeffding bound on the reward for each arm, and selecting the arm with the highest bound. The Exponential-weight algorithm for Exploration and Exploitation (Exp3) [24] works by maintaining a list of weights for each arm and decreasing or increasing the weights according to the rewards observed. Exp3 then selects an arm randomly based on the weights, such that an arm with a high weight is more likely to be selected. To further increase the robustness of Exp3, an egalitarianism factor is presented that controls how much the weights should affect the selection process. This ranges from uniformly random, i.e. weights have no impact, to greedy, i.e. always selecting the arm with the highest weight.

Another popular and successful approach leverages Bayesian reasoning combined with sampling, referred to as Thompson Sampling (TS). TS places a posterior distribution over the arm rewards, and selects an arm with a probability proportional to the likelihood of that arm being optimal.

These different arm representations are illustrated in Figure 1.3. As seen, for UCB, Exp3 and TS, the pure MABP setting is an inherent part of the schemes. Indeed, it has turned out that applying these algorithms outside the MABP paradigm is non-trivial. In contrast to Tsetlin Automata based schemes, examples of successful applications are more sparse. A prominent example is the usage of UCT in Monte Carlo Tree Search [25], one of the key components in the much celebrated Alpha Zero framework[26]. Another example, is the usage of TS as the driving force in Bayesian Optimization (BO) [27] and an similar BO optimisation based on UCB [28]. TS has also been used for action selection in context-aware sequential decision making [29], and in addition, Ortega and Braun [30] note that *"Thompson sampling provides a natural strategy for causal induction*

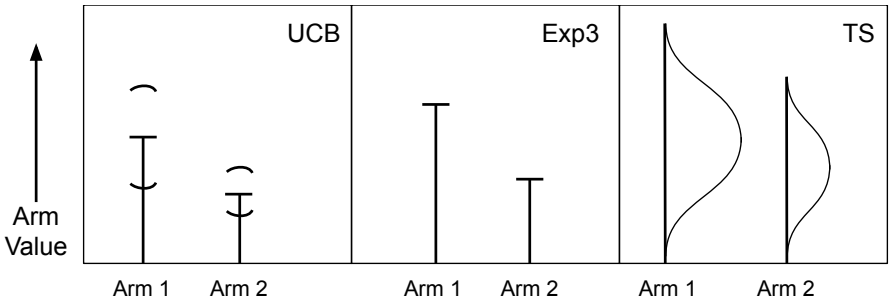

Figure 1.3: The MABP representation for UCB, Exp3 and TS.

when interacting with an environment with unknown causal structure." and as such apply TS together with do-calculus [31] for casual inference. This is in contrast to the work by Finn et. al. [32] that uses fully specified value functions, focusing on the problem of calculating the expected value of each action. We, however, focus on problems where the value of actions are unknown and stochastic and need to be estimated as quickly and accurately as possible, thus involving the exploration-exploitation trade-off.

Motivating Example. To further illuminate advantages of Bayesian reasoning in the context of the bandit problem, we here consider the well-known Monty Hall problem. This problem is drawn from the game show "Let's make a deal" hosted by Monty Hall. The game goes as follows: First, the contestant is asked to select one out of three doors, with one of the doors concealing a car, while the other two, each hide a goat. The objective for the contestant is to obtain the car. After the contestant has made his selection of a door, but before it is opened to reveal its content, Mr.Hall will open one of the other doors, one that contains a goat. Thus, there are now only two available doors, with one hiding a goat and the other hiding a car. The contestant is then asked if he wants to switch from the door he initially selected, to the other remaining door. After the contestant has made his choice, the door is opened and the contestant receives the prize concealed behind the door.

As illustrated in Figure 1.4, the problem can be represented by a Bayesian Network containing three variables. The first one is "Prize". This variable captures our prior on where the car is located before we acquire any additional information. For simplicity, we here assume a uniform prior, i.e., each location is equally likely. The second variable is "First Selection", which is the door that the contestant chooses. Finally, the variable "Monty Opens" refers to the door hiding a goat, which is the one that Monty opens after seeing the contestant's selection. A naive take on the Monty Hall problem would be to assume that since we are left with two potential doors after Monty opens the door containing a goat, the two remaining doors have the same probability with regards to hiding the car. If we model this as a MABP problem where each of the three doors is an arm, and the door Monty opens is unchoosable, then this is exactly a problem where

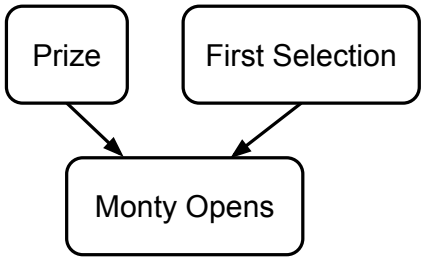

Figure 1.4: The Bayesian Network composing the Monty Hall problem. An arrow indicate dependencies between the variables.

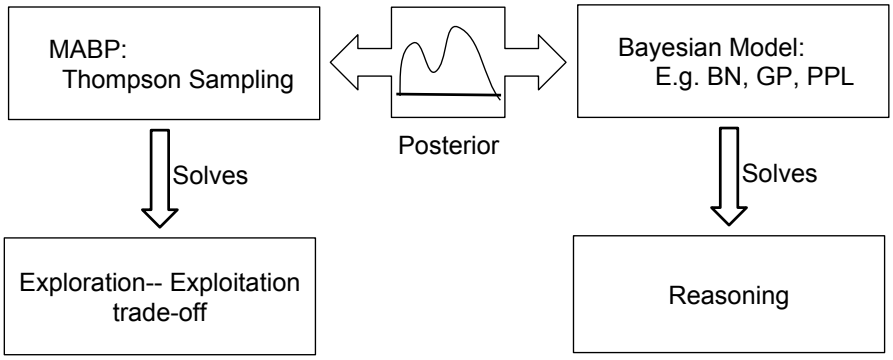

Figure 1.5: The posterior representation of Thompson Sampling bridges the gap between the MABP solver and Bayesian Reasoning.

methods like LA and MABP algorithms struggle to make the correct choice. Namely that the doors are not equiprobable since they were drawn under different circumstances, and obtaining information about one arm affects what is known about the other arms. That is, the probability of a reward for the initial selection is one-third, and after Monty opens a door, the *other door* has a reward probability of two-thirds.

Overall Thesis Contribution and Approach. The efficiency of the specialized MABP algorithms compared to LA approaches in the bandit setting raises an intriguing question: *Is it possible to get the best of both worlds by mixing the versatility of LA with the efficiency of a MABP solver?*

As detailed in the following, in this thesis we explore the above question by proposing novel solutions to several problems that to date mainly LA-based approaches have handled well. In all brevity, we will introduce approaches that leverage TS and its Bayesian perspective. We achieve this by expanding the internal representation of TS in several directions, and by proposing inference algorithms designed for each new representation. In this manner, we introduced TS to significantly more complex decision problems than the MABP.

Figure 1.5 depicts our overarching strategy for designing the TS based solutions. As

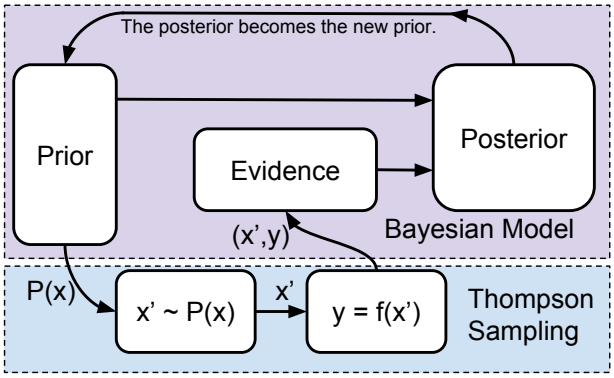

Figure 1.6: The core idea of this thesis, a Bayesian Model provides reasoning for exploitation while Thompson Sampling provides exploration. A distribution $p(x)$ is exposed to the Thompson Sampling section that samples a location x' for further investigation. We then obtain new evidence (x', y) from querying the underlying problem $f(x)$ at the location x'. The prior and the new evidence is then used to formulate a posterior. The posterior distribution subsequently becomes the prior for the next iteration of the algorithm.

seen, at the core of each solution, we propose a novel posterior state space representation that enables effective Bayesian reasoning, combined with trading exploration against exploitation with TS. Accordingly, each solution goes beyond the classical MABP, addressing problems where the bandit arms are not independent, but are interacting in complex ways. A key part of this approach is the Bayesian modelling of the interaction between bandit arms, using Bayesian Networks (BN), Gaussian Processes (GP), and Probabilistic Programs (PPL), as well as more specialized approaches. In this principled manner, we will recast several decision problems that currently only have effective LA based solutions, in a Bayesian framework designed for TS.

1.2 Motivations and Objectives

1.2.1 Motivations

Being one of the state-of-the-art approaches to the *classical* MABP [33, 34, 35], TS has outperformed LA-based MABP solvers. However, as discussed earlier, LA based solutions are not only limited to the classic MABP setting. The applications of LA are numerous and LA schemes have successfully addressed a wide range of complex problems. In this thesis, we attack five such problems from a TS perspective, with the intent of broadening the impact of TS:

1. **The Stochastic Fractional Non-Linear Knapsack Problem**
   In the Stochastic Fractional Non-Linear Knapsack Problem (SFNKP) the objective

is to fill a knapsack with objects whose value is stochastic and is subject to non-linear diminishing returns, so that the value of the knapsack is maximized. The existing LA solution is based on linking the amount of each object with a linear function and continuously swapping a fraction of the object that provides the least value per unit with the object that provides the most value per unit. This can be seen as a bandit problem where pulling arm k is equivalent to adding object k to the knapsack.

This problem is both interesting and challenging as it extends the classical bandit problem with rewards that diminishes with the frequency accessed. That is, the return per amount of each object is decreasing monotonically. Further, the problem introduces a global constraint – that the capacity of the knapsack cannot be exceeded.

2. **Goore Game**
   In the Goore Game each player votes *yes* or *no* in each round, with the objective of finding the optimal fraction of yes/no votes. This fraction is not known beforehand, and the only feedback given to each player is a noisy binary signal based on the distance from the optimal fraction i.e. the closer the optimal fraction the higher probability of a reward.

   The LA solution maintains a two-armed bandit for each player with one arm representing a yes vote, the other a no vote. And essentially keep each player unaware of the other players, trusting instead the bandits to guide the LA to the optimal fraction.

   The main drawback with this approach is that there is no way of knowing how confident the remaining players are in their vote. This is especially paramount in settings where new players are introduced as a replacement for existing players, introducing uncertainty in the system.

3. **Searching on the Line**
   In Searching on the Line (SPL), the objective is to locate a specific point on a line by issuing a sequence of location queries to an oracle, where the oracle responds with whether or not the point is to the left or the right of the query location. The oracle can be either informative or divergent. That is, if the oracle is informative, it will on average provide the correct direction; if the oracle is divergent it will on average provide the wrong direction. The key metric for discriminating between the different solution schemes is: the number of queries for convergence, as well as, the sum of the distances between the point and the query locations.

The SPL turns into a bandit problem by realizing that each location on the line is an arm, and the feedback from pulling an arm is the direction provided by the oracle. Thus, we have a scenario where the classical MABP independence between arms is invalidated. Further, the total reward i.e. the sum of distances between queries and the optimal point, is only observable after the algorithm finishes. This turns out to be particularly challenging in scenarios where the player does not know beforehand if the oracle is informative or deceptive since as it necessary to both explore the nature of the oracle and finding the point while simultaneously try to query locations close to the point.

4. **Equipartioning Problem**
   In the Object Partition Problem (OPP), the goal is to divide a set of objects into different partitions so as to maximize some objective function over the partitions. The Equipartition Problem (EPP) is a variant of the OPP where we constrain the cardinality of the partitions to be equal. With the relations between the objects given as a stream of object pairs, the objective function is to maximize the probability that the next pair of objects are in the same partition.

   The state-of-the-art LA solution for this problem is based on the Object Migration Automata (OMA) [17], a simple yet effective scheme that operates by having each object represent how strongly they are connected with their current partition as an integer. If a pair of objects that are located in the same partition are observed, then the LA strengthens their connection to the partition, otherwise it decreases the connection strength. If the connection strength of two objects is zero and they are observed together, then they are migrated into the same partition. The basic OMA scheme has been improved by various heuristic improvements [36, 37].

   The main challenge presented in EPP is that the relationships between the object are stochastic, and it is not efficient to only consider the information observed, one must also take into account the information gained from the fact that certain pairs of objects are rarely observed together. This holds true especially as the number of objects grows large.

5. **Distance Estimation**
   In Distance Estimation (DE), the objective is to estimate the distance between two points based solely on global coordinates, e.g. GPS, longitude & latitude, of the point of interest. That is, without access to local features such as rivers, roads, and hills. For example, the traveled distance between two points on the Manhattan is quite different from the distance between two points in an open desert.

   The LA solution for this problem is based on estimating the hyperparameters of a

fixed metric function in order to minimize the error in the dataset. This estimation is done using an SPL variant for each parameter, and is in essence applying a Coordinate Descent algorithm where one parameter is optimized while keeping the other parameters fixed.

A novel point of interest is the selection of travel destinations to learn the underlying metric function as fast and efficient as possible. This is an form of bandit based active-learning where a traveler is at an location and have to select a next destination from a list of locations (the arms) and observe the feedback in the form of travel time.

The overall drawback with the existing LA solutions is that they operate under a myopic view of the problem, where instead of considering the problem as a whole they take small steps towards the solution. If an LA encounter noisy feedback, it assumes that the feedback on average is correct and thus by following the feedback directly, it will asymptotically converge towards a solution.

Bayesian reasoning do not suffer from this weakness as it interact directly with the posterior distribution.

1.2.2 Objectives

The overall objective of this thesis is to demonstrate how the basic principle of TS can be leveraged to solve complex decision problems, previously mainly addressed by LA-based schemes:

1. **Stochastic Fractional Non-Linear Knapsack Problem**
   In this thesis, we will replace the non-linear value function found in the state-of-the-art LA solution with a Gaussian Process (GP), a Bayesian nonparametric prior over functions, to explicitly model the characteristics of each material, including the uncertainty associated with the material. To control the exploration-exploitation trade-off we will apply TS over the posterior GP's. The objective is to let TS deal with the exploration of the solution space where the GP is uncertain, while simultaneously exploiting the more well-explored areas. Consequentially, providing a better and more robust solution.

   Application: Web Mining:
   In Web Mining, the objective is to acquire up to date information from websites as fast as possible given a limited set of resources (web crawlers). We assume that there exists an underlying function that gives the relation between how often one visits a particular site and the probability that a visit results in fresh information.

Existing web-crawling solutions implicitly model this function by trial and error, and do not utilize the structure of the function. Therefore, if we model the update probability vs time between each visit for each website as a GP, we can exploit the websites' individual characteristics to maximize our update rate.

2. **Searching on the Line**
   In this thesis, we aim to develop a Bayesian model over the interaction between the line and the responses from the Oracle. This model gives us a posterior distribution over the solution space that TS can utilize to find the next query point. Thus, handling the exploration-exploitation trade-off inherent in the SPL problem. Also, as a side-effect, since the Bayesian model is dependent on the nature of the oracle, it should seamlessly handle the case where the Oracle is deceptive.

   Application: Stochastic Root Finding:
   In Stochastic Root Finding the objective is to find the root of a function based on noisy observations of the function. A Bayesian model is capable of performing optimal probabilistic reasoning. Thus by modeling the relationship between the observations and the noise we are able to quickly reduce the search space to feasible regions and therefore find the root faster and more robust than existing methods.

3. **Equipartioning Problem**
   In this thesis, we intend to develop a BN to model the interaction between the objects and the partitions in such a way that it enables us to not only react to observed data but reason using the not-observed data as well. As a side effect, a BN permits us to specify additional probabilistic constraints on the partitions and the objects to better handle real-world problems.

   Application: Warehouse optimization:
   In the Warehouse optimization setting, one strives to place the wares of a warehouse so as to minimize the time it takes to collect orders, thus maximizing the number of orders that can be collected. We will investigate two objectives:

   (1) To make order picking as efficient as possible, we will investigate how to place wares that are often ordered together in close proximity of each other based on the order history.

   (2) A warehouse is also often faced with practical real-world constraints such as that heavy goods needs to be placed on the floor level, and items that need to be frozen should be in the freezer. We will investigate how to model this problem such that these constraints become first-class citizens in a Bayesian Network representation, thus ensuring that the constraints are handled in a flexible manner instead of as

ad-hoc post process feature.

4. **Goore Game**
   The Multi-Armed Bandit Problem is most commonly expressed as a one-to-one interaction between the bandit and the environment. However, in a decentralized setting, a bandit interacts not only with the environment, but also indirectly with all the other bandits that concurrently perform actions on the environment. In this thesis, we aim to explicitly encode the probabilistic relationships between the players into our likelihood, thus enabling TS to take the decentralized component of Goore Game into account when making decisions, consequentially accelerating the learning.

   Application: Quality of Service for Wireless Sensor Networks:
   In Quality of Service (QoS) for Wireless Sensor Networks (WSN), the objective is to deploy sensors over an area such that the coverage of the entire area is maximized. A particular scenario of interest is randomly deployed networks, whose applications include environmental monitoring and battlefield surveillance & reconnaissance. However, due to the strain the real world places on these sensors, they might break down, and since they are deployed at random, it is more cost effective to simply replace sensors, e.g by airdrop, than tracking down and repairing sensors. We will investigate how we can accelerate the cooperation between the sensors to maximize the QoS of the entire network using an explicit model of the prior distribution.

5. **Distance Estimation**: In DE, the objective is to give an estimate of the distance between points A and B without using a map, instead, the estimate is based on a sequence of empirical observations where each observation is a pair of GPS coordinates and the actual distance traveled. In this thesis, we will develop a Bayesian model (implemented as a Probabilistic Program), over the parameters of the estimator. This enables us to measure the uncertainty associated with the model, thus allowing us to select the most efficient spots to observe, as to minimize the number of observations needs to calibrate the estimator.

1.3 Research Approach

The complex nature of the stochastic interactions presented in the objectives indicates the need for new knowledge and the necessity of exploration and trials through empirical and theoretical studies. Consequently, we will employ a mixed method approach, combining the theory-, experimentation-, and design research paradigms of computing [38].

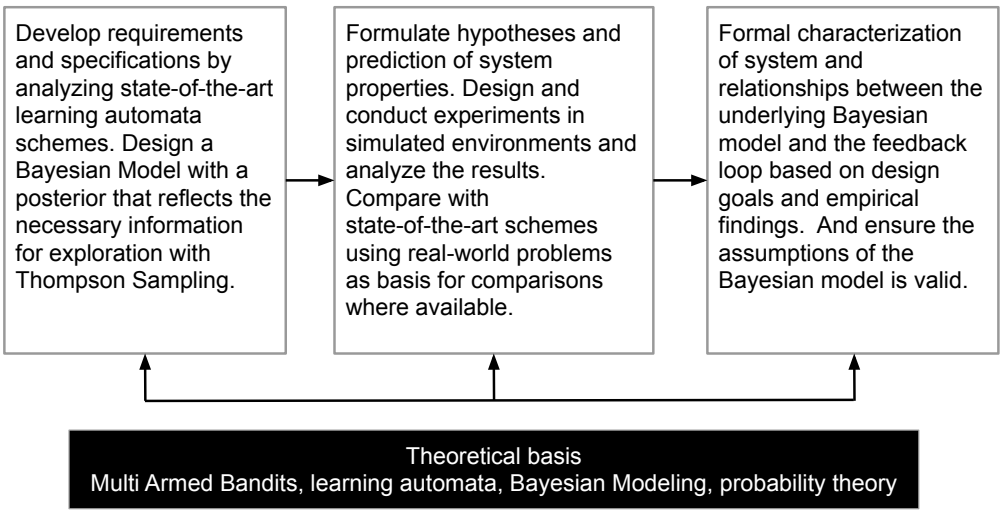

Figure 1.7: An overview of the research process.

In Figure 1.7, an overview of the research process is presented. To put it succinctly, we will demonstrate that combining various Bayesian Models with Thompson Sampling improves upon the existing Learning Automata by designing several new algorithms. These algorithms will take advantage of the ability to see the global picture provided by the posterior as opposed to their Learning Automata counterparts. To guide this research, the theory of Multi-Armed Bandits, Learning Automata and Bayesian Modeling, all underpinned by probability theory, will be applied.

To further confirm the efficiency of these new algorithms, we will not restrict ourselves to the simulated environments that many state-of-art algorithms are demonstrated in. Rather, we will include investigation and evaluation of their performance in challenging real-world problems such as warehouse layout optimization.

The experience and knowledge we gain from these empirical studies will, in turn, form the basis for an iterative process for developing algorithms and intelligent systems. While obtaining insights from design and experimentation, we will develop a robust approach to common problems such as the inherent numerical instability found in many probabilistic algorithms.

To summarize, as illustrated in Figure 1.7, we will iteratively develop algorithms that take advantage of Bayesian reasoning to enable efficient Thompson Sampling exploration and study these through experimentation and theoretical analysis.

1.4 Publications

The following papers are appended and will be referred to by the letters **A-F**. The papers are printed in their originally published state.

Paper A Sondre Glimsdal and Ole-Christoffer Granmo. A Bayesian Network Based Solution Scheme for the Constrained Stochastic On-Line Equi-Partitioning Problem. Applied Intelligence, 48(10):3735–3747, 2018.

Paper B Sondre Glimsdal and Ole-Christoffer Granmo. Thompson Sampling Guided Stochastic Searching on the Line for Deceptive Environments with Applications to Root-Finding Problems. Journal of Machine Learning Research, 52 (20):1–24, 2019.

Paper C Sondre Glimsdal and Ole-Christoffer Granmo. Thompson Sampling Guided Stochastic Searching on the Line for Non-Stationary Adversarial Learning. In International Conference on Machine Learning and Applications, pages 687–692, IEEE, 2015.

Paper D Ole-Christoffer Granmo and Sondre Glimsdal. A Two-Armed Bandit Based Scheme for Accelerated Decentralized Learning. Modern Approaches in Applied Intelligence, pages 532–541, Springer, 2011.

Paper E Sondre Glimsdal and Ole-Christoffer Granmo. Gaussian Process Based Optimistic Knapsack Sampling with Applications to Stochastic Resource Allocation. Proceedings of the 24th Midwest Artificial Intelligence and Cognitive Science Conference, pages 43–50, 2013.

Paper F Sondre Glimsdal and Ole-Christoffer Granmo. Thompson Sampling Based Active Learning in Probabilistic Programs with Application to Travel Time Estimation. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pages 71–78, Springer, 2019.

1.5 Organization of the Thesis

In this section, we present the overall organization of this doctoral dissertation:

1. **Chapter 2: Background**
   This chapter presents the necessary background material for this thesis. We give an overview of the various Bayesian inference methods that we will employ in conjunction with TS. These methods include Gaussian Processes, Bayesian Networks, and Probabilistic Programming. We will also survey the MABP and some of its many popular variants such as Upper Confidence Bound (UCB), ϵ-Greedy and Thompson Sampling. In addition, we also review the relevant LA literature.

2. **Chapter 3: Contributions**
   This chapter from summarizes the main contributions of this thesis. The published results include:

- The existing solutions for Searching on a Line (SPL) is based on making slow discrete steps towards the solution. We will demonstrate that TS can be used to solve this in conjunction with a BN, in a more efficient manner. We also demonstrate that SPL and the Probabilistic Bisection search are nearly identical, and by this provide both an overview and novel connection between these fields. In addition, we expand the TS model to also handle the case when the objective location is moving. The corresponding paper is found in Appendix B and C.
- By directly encoding the uncertainty in the decentralized Goore Game, we show that TS can use this knowledge to greatly accelerate the speed of convergence. The corresponding paper is found in Appendix D.
- We combine a GP model with TS to solve the SFNP and demonstrate the applications to Web Polling. The corresponding paper is found in Appendix E.
- Modeling a Warehouse layout as a Bayesian Network, and using TS to optimize the BN based on the incoming orders. The corresponding paper is found in Appendix A.
- The use of Probabilistic Programming to model the Distance Estimation problem shows how powerful these techniques can be. We also demonstrate that by applying TS, we can solve cases where the data are not available apriori, but have to be collected in an online manner while simultaneously providing an anytime solution. The corresponding paper is found in Appendix F.

3. **Chapter 4: Conclusion**
   Concludes the thesis with a summary and directions for further research.

Chapter 2

Background

In Chapter 1, we introduced the overarching theme of the thesis. In this chapter, we will provide a more in-depth description of the research that we build upon, as well as connecting the thesis research to the relevant state-of-the-art.

2.1 Thompson Sampling

The Multi-Armed Bandit Problem (MABP) is arguably the most condensed version of the exploration-exploitation dilemma in sequential decision making. As earlier introduced, the MABP manifests the challenge of optimally balancing between staying with the previously most rewarding action and exploring new, potentially more rewarding, actions.

William R. Thompson is credited with the first MABP solution, later called Thompson Sampling (TS) [39, 40]. In his pioneering work, Thompson considered the unethical practice in clinical trials taking place when a drug is shown at an early stage to have a high effect, yet the control group is not provided access to the drug, thus not being helped.

The concept of TS was largely ignored in the academic literature until recently, despite being independently rediscovered several times [41, 42]. However, after several researchers documented strong empirical performance [43, 42, 34], interest has increased steadily.

TS is best understood in a Bayesian setting as follows [44, 45, 33]: A slot machine has N arms. For each time step $t = 0, 1, \ldots$ a gambler selects an arm $i \in [1, N]$ to be played. As arm i is played, it yields a real-valued reward randomly drawn from an (unknown) fixed underlying reward distribution associated with arm i. The gambler immediately observes the reward and each reward from arm i is assumed to be i.i.d.

Measurements of Performance. An algorithm addressing the MABP must make its decision using the previously recorded observations (arm-reward pairs). Let μ_k be the mean reward from arm k, and $i(t)$ the arm pulled at time t. A common optimization

criterion is to maximize the expected reward over T time steps: $\sum_{t=1}^{T} \mathbb{E}[\mu_{i(t)}]$.

Another measurement of performance is the notion of *expected total regret*, that is, the total reward lost by not playing the optimal arm. Let $\mu^* = \max_k \mu_k$ and $\Delta_k = \mu^* - \mu_k$. Also, let $n_k(t)$ be the number of times arm k has been played up to time $t-1$. Then we can define the expected total regret as:

$$\mathbb{E}[\mathcal{R}(T)] = \mathbb{E}[\sum_{t=1}^{T}(\mu^* - \mu_{i(t)})] = \sum_{k} \Delta_k \cdot \mathbb{E}[n_k(T+1)] \tag{2.1}$$

.

The TS Algorithm. The basic idea of TS is to model the (unknown) reward distribution of each bandit arm with a prior distribution. The posteriors of the distributions are then calculated from the observations obtained thus far. At each time step, we play each arm with frequency proportional to the belief of the arm being optimal, by comparing samples from the posterior distributions.

Even though the original approach was specifically targeting clinical trials, we will here use the term TS to refer to the general sampling based approach embodied in the clinical trials. Let $P(\cdot)$ denote a probability density. The general structure of TS contains the following components [34, 44, 45, 33]:

1. A set Ψ of parameters $\widetilde{\mu}$.

2. An assumed prior distribution $P(\widetilde{\mu})$ on these parameters.

3. Historical observations $\mathcal{D}$ on previous rewards for the different arms played.

4. An assumed likelihood function $P(r|\widetilde{\mu})$ that gives the probability of a reward given a parameter $\widetilde{\mu}$.

5. A posterior distribution $P(\widetilde{\mu}|\mathcal{D}) \propto P(\mathcal{D}|\widetilde{\mu})P(\widetilde{\mu})$, where $P(\mathcal{D}|\widetilde{\mu})$ is the likelihood function.

As seen, TS maintains a posterior distribution over the expected reward for each arm i, i.e., a posterior distribution over μ_i. To determine which arm to play for a given round, TS draws a sample from the posterior distribution of each arm $P(\mu_i|\mathcal{D})$ and play the arm i that produced the highest reward. In effect, this means that TS selects each arm with the probability of that arm being optimal, that is, the probability that μ_i equals μ^*. To exemplify, the concrete TS algorithm for Bernoulli bandits with Beta priors is given in Algorithm 1 the Beta and Gaussian distributions are the most commonly used

distributions due to their efficiency as conjugate priors).

Algorithm 1: Thompson Sampling with Beta Priors

For each arm $k = 1, \ldots, N$ set $S_k = F_k = 0$.

foreach *t = 1, 2, ...* **do**

  For each arm k sample $\theta_k(t)$ from the Beta$(S_k + 1, F_k + 1)$ distribution

  Play arm $i(t) := \arg\max_k \theta_k(t)$ and obtain reward r_t

  if r_t *is a reward* **then**

    $S_{i(t)} := S_{i(t)} + 1$

  else

    $F_{i(t)} = F_{i(t)} + 1$

  end

end

A wide range of variations of TS exists. For instance, a variant called Optimistic TS by May et al. [46] restricts the sampling to the upper half of the posterior distribution to make TS consider the reward distribution more optimistically. Another efficient variant of TS is the application of Kalman filters to allow for restless bandits, i.e., the best arm changes over time instead of remaining fixed [42].

While the empirical properties of TS are well studied, it was not until the work of Agrawal et al. [44, 45] that a finite time theoretical regret bound on TS was established (for the case of Beta Priors and for Gaussian Priors). They proved that TS achieves the problem independent regret bounds of $O(\sqrt{NT \ln T})$ and $O(\sqrt{NT \ln N})$ for these priors respectively, and thus matches the known $\Omega(\sqrt{(NT)}$ problem independent bound [47]. Furthermore, Dong et al. proved sharp regret bounds for large finite action spaces, based on information theory [48], while Srinivas et al. [49] addressed continuous action spaces with a Gaussian Process prior. Finally, Kaufmann et al. [50] showed analytically that TS achieves sub-linear regret in stochastic MAB settings.

Armed with both a theoretical and empirical foundation for the efficacy of TS, practitioners have employed it for a wide range of domains, such as planning under uncertainty [51], revenue management [52], selecting web advertisement in challenging environments [53], web site optimization [54], click-through optimization for web advertising [55], recommender systems [56], balancing exploration vs exploitation in Arcade Games [57], and multi-armed bandit experiments [58, 43]. A more complete survey of general bandit techniques in is found in [33].

2.2 Learning Automata

An LA is an adaptive decision-making unit that tries to determine the optimal action out of a set of allowable actions. The learning is performed as a sequence of interaction

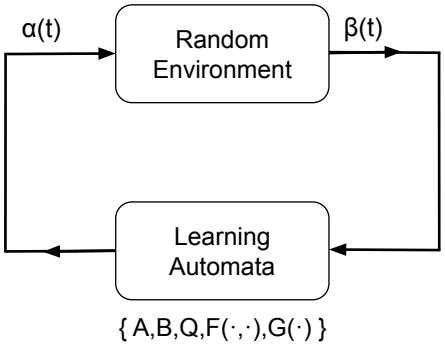

Figure 2.1: Feedback connection of LA and environment.

cycles where the LA perform a action, a environment takes this action as its input and returns an output based on the action chosen. This cycle is illustrated in Figure 2.1. The idea is that based on the knowledge obtained from the previous actions the LA should incrementally learn to make *better* decisions.

As illustrated in Figure 2.1, an LA operates in a random environment and continually updates its preferred action based on the response received from the environment.

At each iteration t, the LA selects an action $\alpha(t)$ from the set of r actions $\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$. The environment receives $\alpha(t)$ as the input, and outputs a response $\beta(t) \in \{0, 1\}$. Based on the response, the LA updates its internal state, based on previously acquired responses, such that the next action $\alpha(t+1)$ minimizes the number of penalty responses. The core idea is that even though the LA lacks a complete overview of the environment, the LA will still adapt itself towards the optimal solution by repeated interactions with the environment.

The entire field of LA is too large to be covered in its full extent here, so we will instead limit the exposition to the original LA structure, and then proceed to detail the specific LA architectures that we build upon in this thesis, for more info see [59, 5, 6, 60, 61]. .

Definition 1. An LA is defined by a 5-tuple [5] $(A, B, Q, F(\cdot,\cdot), G(\cdot))$ where:

1. $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the set of possible actions that the LA must choose from, and $\alpha(t)$ is the action selected by the automaton at instant t.

2. $B = \{\beta_1, \beta_2, \beta_m\}$ is the set of possible responses from the environment and is thus, from the perspective of the LA, the set of possible inputs. In illustration 2.1, the environment provided a binary response/input $\beta(t) \in B = \{0, 1\}$.

3. $Q = \{q_1, q_2, \ldots, q_s\}$ is the set of states. Traditionally this set is considered finite. However, some approaches also handle the non-finite state spaces. The state at time t is denoted $Q(t)$.

4. $F : Q \bigtimes B \mapsto Q$ is the recurrent state transition function that maps a state $q \in Q$ and an input $\beta \in B$ to a new state $q' = F(q, \beta)$.

5. $G : Q \mapsto A$ is the output function that maps a state $q_i \in Q$ into an action $\alpha_j \in A$.

Note that if the sets A, B and Q are finite, the LA is finite.

The environment E that the LA interacts with can be defined correspondingly.

Definition 2. An Environment is defined by a 3-tuple (A,B,C) where:

1. $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the set of actions that can be performed on the environment.

2. $B = \{\beta_1, \beta_2, \beta_m\}$ is the output set of the environment. For the binary case, $m = 2$, we consider $\beta = 0$ to be a reward, and $\beta = 1$ to be a penalty. This is contrary to the typical MABP definition where β is an indicator function of a reward, i.e., $\beta = 1$ is a reward.

3. $C = \{c_k \mid k = 1 \ldots r\}$ is a set of penalty probabilities, where each c_k maps to a corresponding action a_k.

Thus, the learning interaction between the LA and the Environment is based on a recurrent sequence. At time t the LA selects an action $\alpha(t)$, and the environment gives a response $\beta(t)$ to the LA. The response is a reward with probability $1 - c_\alpha(t)$, otherwise it is a penalty. Upon receiving a response $\beta(t)$, the LA updates its internal state $q(t+1) = F(q(t), \beta(t))$, and selects a new action $\alpha(t+1) = G(q(t+1))$. This recurrent process is repeated with $\alpha(t+1)$ as the new starting point.

2.2.1 Classification of Learning Automata

An LA is either *stochastic* or *deterministic*.

Definition 3. A *Deterministic Learning Automaton* is an LA where both the transition function $F(\cdot, \cdot)$ and output function $G(\cdot)$ are deterministic.

Definition 4. A *Stochastic Learning Automaton* is an LA where either the transition function $F(\cdot, \cdot)$ or the output function $G(\cdot)$ is stochastic.

Furthermore, it either has a *fixed* structure or a *variable* structure.

Definition 5. A Fixed Structure Stochastic Automaton (FSSA) is an stochastic automaton that is time invariant. That is, both the transition function $F(\cdot, \cdot)$ and the output function $G(\cdot)$ is time invariant.

Definition 6. A Variable Structure Stochastic Automaton (VSSA) is a stochastic automaton where the structure of the LA varies with time. That is, one or both of the functions (transition function $F(\cdot, \cdot)$, output function $G(\cdot)$) are dependent on time.

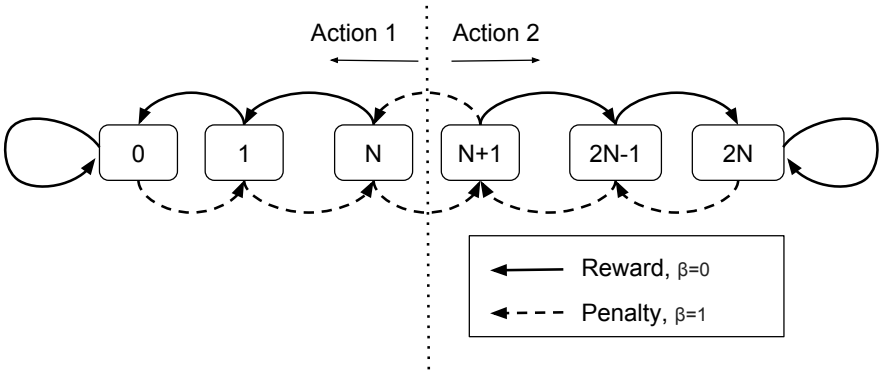

Figure 2.2: The Tsetlin Automaton. If the current state is on the left of the middle, select action α_1. Else, select action α_2.

2.2.2 Learning Automata Schemes Addressed

The process of applying an LA to a learning problem requires the design of an LA-Environment interaction loop, such that the LA can learn the best course of action. In this section, we will give an overview of some of the most successful LA schemes, related techniques, and their applications. In particular, we will cover the Tsetlin Automaton; the Object Migration Automaton; the Stochastic Searching on the Line Automaton; and the Learning Automata Knapsack Game. These schemes, their related approaches, and the applications they tackle, form the starting point and focus for our research.

2.2.3 The Tsetlin Automaton (TA)

The *Tsetlin Automaton* (TA) is the first, and yet, perhaps most elegant LA scheme, pioneered by M. L. Tsetlin [62]. Being the first LA, it forms a natural starting point for our investigation. In all brevity, the TA is designed to solve the two-armed bandit problem, as depicted in Figure 2.2 (the figure illustrates the transition function $F(\cdot,\cdot)$ and the output function $G(\cdot)$ of the TA).

The learning principle of the TA is quite straightforward. If you get a reward, strengthen the belief that the arm you pulled is the optimal arm by moving toward the end state ($q = 1$ for action 1, $q = 2N$ for action 2). Conversely, if you get a penalty, change state towards the middle states, ultimately switching action. This scheme is asymptotically optimal given that the reward probability of the optimal action is larger than 0.5.

Formally, a TA can be defined as follows. First of all, the outputs of the TA correspond to the MAB arms, $A = \{\alpha_1, \alpha_2\}$, while the feedback $B = \{\beta_0, \beta_1\}$ are the rewards and penalties, respectively. Table 2.1 specifies the state-transition function, while the output function $G(q_i)$ is given as:

$$G(q_i) = \alpha_1, \text{ if } i = 1, \ldots, N$$
$$= \alpha_2, \text{ if } i = N+1, \ldots, 2N$$

State:	Reward β_0	Penalty β_1
$q_2, \ldots, q_N$	$q_i \rightarrow q_{i-1}$	$q_i \rightarrow q_{i+1}$
q_1	$q_1 \rightarrow q_1$	$q_1 \rightarrow q_2$
$q_{N+1}, \ldots, q_{2N-1}$	$q_i \rightarrow q_{i+1}$	$q_i \rightarrow q_{i-1}$
q_{2N}	$q_{2N} \rightarrow q_{2N}$	$q_{2N} \rightarrow q_{2N-1}$

Table 2.1: The state transition function $F(q_i, \beta_j)$ for the Tsetlin Automaton.

2.2.4 Object Migration Automata

The Object Partitioning Problem (OOP) is concerned with partitioning a set of objects such that some objective function over the partitions is optimized. In the specific problem we consider we assume that there is an underlying *true* partitioning, and that the objective is to minimize the difference between the learned partitioning and the true, underlying one. When the partitions are required to have the same cardinality, the problem is referred to as equi-partitioning.

In the Stochastic On-line Equi-Partitioning Problem (SO-EPP) the partitions are inferred purely from observing an online sequence of object pairs. The sequence contains paired objects that belong to the same partition with probability p, and to different partitions with probability $1-p$, with p also being unknown. Figure 2.3 illustrates the process of SO-EPP. The only observable relationship between the objects in the stream is that objects that originate from the same underlying partition occur together more frequently than objects from two different partitions.

While several LA-based solutions have been suggested, the most efficient ones are the Object Migration Automaton (OMA) [17] that incorporates the changes by Gale et al. [36], and the filtering method of Abdolreza et al. [37].

A basic overview of OMA is provided in Figure 2.4. As seen, the structure of the OMA is similar to a Tsetlin Automaton in the sense that each partition is allocated a sequence of N states, with the distance from the center states measuring confidence. The difference, however, is that there are multiple objects moving from state to state, and that it is finding the optimal partitioning of the objects that is the goal, rather than finding the optimal action.

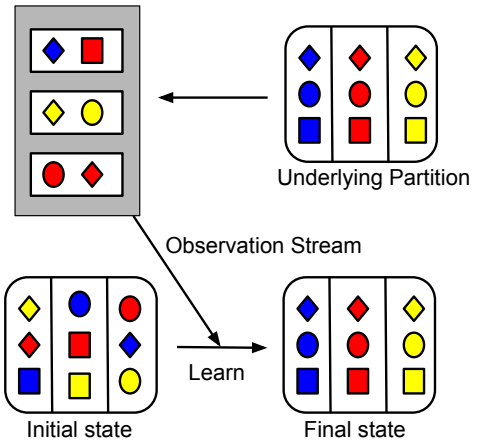

Figure 2.3: The SO-EPP with 3 partitions and 9 objects. The objective is to transform the initial, random, configuration into the underlying solution by learning from the stream of object tuples. The topmost tuple in the stream (the blue rhombus and red square) is a noisy observation as the objects do not originate from the same underlying partition, contrary to the other two informative tuples. Note that the coloring and shape are added here for illustration purposes, while in the learning problem they have identical appearance, apart from a unique label.

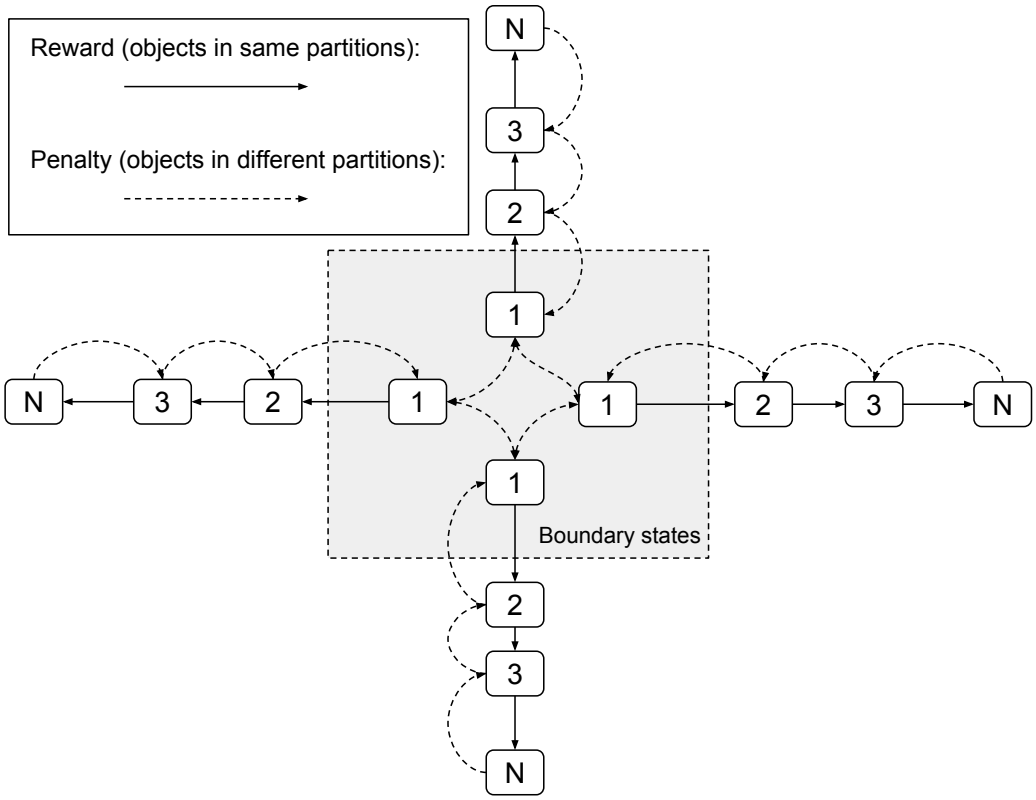

Figure 2.4: The OMA with 4 partitions and N states for each partition. The learning scheme is quite simple: If two objects are observed together and they are in the same partition move the objects away from the boundary; otherwise, move them towards the boundary.

That is, there are W different objects that are to be partitioned into R different partitions, with each object O_k assigned to an initial random partition. Let the current partition for an object O_i at time t be given by a function $Pa^{(t)}(O_i)$. The transition function can then be described with the simple rule that upon observing a pair of objects O_i, O_j, if the objects are in the same partition, $Pa^{(t)}(O_i) = Pa^{(t)}(O_j)$, then one gives a reward to both of the objects (the solid lines in Figure 2.4); otherwise, the two objects are given a penalty (the dashed lines in Figure 2.4).

The single exception to the above simple rule happens when one or both of the objects are in the *boundary state* and receive a penalty (i.e., they are in different partitions). We then have a *boundary transition*, which happens as follows. Firstly, if both objects O_i and O_j are in a boundary state, switch one object, for instance O_i, to the partition of O_j by swapping it with the object O_k that is closest to O_j in partition $Pa^{(t)}(O_j)$, so as to preserve the number of objects in each partition. If only one of the objects are in a boundary state, say O_i is in a boundary state and O_j is not, then check if there is another object O_k in the boundary state of partition $Pa^{(t)}(O_j)$. If such an object O_k is found, switch the objects O_k and O_j. This last rule is the improvement made by Gale et al. [36], and significantly improves the efficiency of the scheme. The algorithm is presented in Algorithm 2.

2.2.5 Pursuit Object Migration Automaton (POMA)

The state-of-the-art solution scheme for SO-EPP is the Pursuit Object Migration Automaton (POMA), introduced by Shirvani et al. in 2018 [37]. POMA is based on the Object Migration Automaton (OMA) [17, 36]. The basic OMA is a statistics free scheme, meaning that it does not try to estimate object co-occurrence frequencies. Instead, each object navigates a finite state machine according to a few simple fixed rules, allowing the objects to migrate between the different partitions, gradually converging to a solution.

The main difference between OMA and POMA comes from dividing the observations of OMA into two categories: converging and diverging, where OMA's speed of convergence is dependent on the fraction of convergent observations. The fewer diverging queries, the quicker the convergence.

POMA takes advantage of this phenomenon by learning a filter that sits in front of OMA and discards diverging queries, rendering OMA to work solely on convergent observations thus greatly enhancing OMA's speed of convergence. To learn this filter POMA employs two-phase scheme:

1. The first phase consists of gathering co-occurrence frequency estimates between the objects while simultaneously using OMA to generate an initial solution.

2. Once sufficient statistics have been collected, a filter is learned using a Pursuit Automata. This filter only passes through observations that have a high probability of being convergent, therefore, in the second phase, OMA only operates on convergent queries.

In all brevity, the POMA scheme, after an initial learning phase, lets the OMA operate in a noise-free environment, and consequently, converge quicker than the regular OMA.

2.2.6 Stochastic Point Location Automata

Stochastic Point Location (SPL) is a challenging problem that was independently solved in the field of Learning Automata and the field of Operational Research (where it is known as the Probabilistic Bisection Search [PBS]).

The objective of the SPL problem is to locate a point x^* on a line guided solely by noisy feedback given by an Oracle through a sequence of queries. In each step, the SPL queries a point x, and the Oracle responds as to whether the root lies to the *Left* or to the *Right* of x. However, with probability $1-p$ the oracle gives the wrong answer. If $p > 0.5$ we call the oracle *informative*, and if $p < 0.5$ we call it *deceptive*. We shall without loss of generality assume that the point x^* is located in the unit interval $[0, 1]$. If it is not, then a simple mapping function should be applied.

The original SPL algorithm handles only the case of informative oracles [17], i.e., the Oracle is expected on average to give correct information. In all brevity, the unit interval is discretized into N learning automaton states, $Q = \{0, 1/N, 2/N, \ldots, (N-1)/N, 1\}$ where N is the resolution of the learning scheme. Thus, a higher N will lead to a more accurate convergence to the unknown x^*. Let $\lambda(n) \in Q$ be the state of the algorithm at time step n, and let $\beta(\lambda(n)) \in \{\text{Left}, \text{Right}\}$ be the response from the oracle at time n. Then the state is updated as follows:

$$\lambda(n+1) := \lambda(n) + 1/N \text{ if } \beta = \text{Right}.$$
$$\lambda(n+1) := \lambda(n) - 1/N \text{ if } \beta = \text{Left}.$$

Finally, we clamp the value of $\lambda(n+1)$ to be in the unit interval.

2.2.7 Hierarchical Stochastic Searching on the Line

The Hierarchical Stochastic Searching on the Line (HSSL) is another scheme for solving the SPL problem that significantly outperform the simple SPL scheme [63]. The basis for HSSL is a recursive tree search, where, based on the feedback from the oracle, one either stays in the current tree node, traverse deeper, or traverse upwards in the tree.

Accordingly, one performs a random walk in the tree space. An example of a HSSL tree is given in Figure 2.5.

When performing a random walk in the tree space, we do not only need to know if we got a penalty or a reward but in the case of a reward, we need to know what child node i.e. left or right, to visit. To this end, the traditional bijection found in SPL between the feedback from the environment and state change must be revisited. Therefore, a sampling technique that uses the feedback from three samples or queries instead of a single query as in SPL is applied. The oracle is queried for a direction at both the endpoints and the middle of the interval, these three directions is then used as a index in Table 2.2 that gives us the link to traverse in the tree.

Next Node	Feedback	Condition
Parent	Penalty	[R ,R, R] $\vee$ [L, R, R] $\vee$ [L, L, R] $\vee$ [L, L, L]
Left Child	Reward	[R, L, R] $\vee$ [R, L, L]
Right Child	Reward	[R, R, L] $\vee$ [L, R, L]

Table 2.2: The transition function is governed by the three directions (**L**eft / **R**ight) obtained by querying the current node interval at the extreme left, at the centre, and at the extreme right.

The intuition behind the the transition function in Table 2.2 can be summarized as follows. If we obtain a penalty, it means that the interval of the current node does not contain the point. E.g. if we receive the response [R, R, R], then obviously the point is located to the right of the current interval. However, by definition the interval to the right is not contained in this node's interval, thus we penalize the system. If we assume, as is done in the original paper [63], that the system is informative then we know that we will *on average* be moving in the right direction. Conversely, if the responses corroborate the interval, we reward the system and traverse to one of the child nodes. The child note that must contain x^* assuming the data is correct is selected assuming the data was correct. However, a limitation of HSSL is that it has been proven that it converges only if p, the probability of an informative response, is greater than $\frac{\sqrt{5}-1}{2}$, the golden ratio conjugate [63].

The Symmetrical HSSL (Sym-HSSL) [21] extends HSSL to handle deceptive environments by creating a parallel tree that moves towards the root on a reward, as opposed to the HSSL scheme (scheme is illustrated in Figure 2.6). The idea is that if the environment is deceptive we should do the opposite of what the responses from the environment tell us. However, this technique also inherits the limitations of the HSSL, now requiring that $p \notin ((1 - \frac{\sqrt{5}-1}{2}), \frac{\sqrt{5}-1}{2}) \approx (0.38, 0.61)$. As a p value close the 0.5 will make the Sym-HSSL

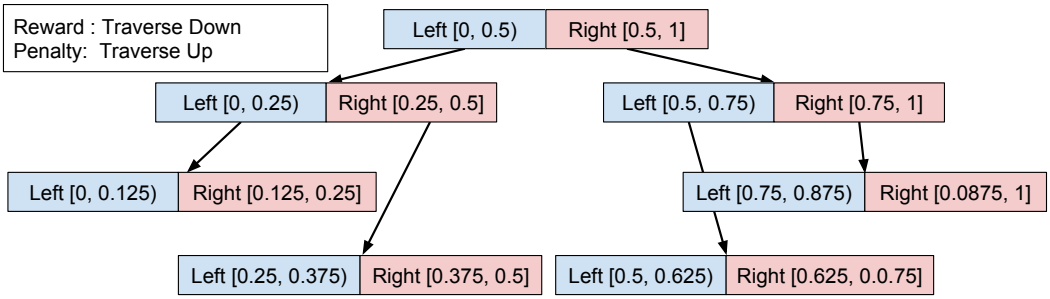

Figure 2.5: The 7 topmost nodes of a HSSL tree. Notice how the path we traverse no longer is solely a question of going to the left or right, but now also includes the option of going upwards, to the parent.

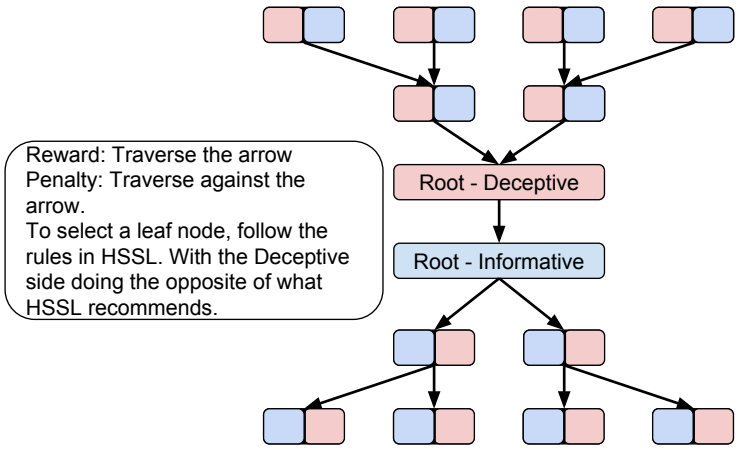

Figure 2.6: Extending the HSSL scheme to handle deceptive environments by adding a symmetrical tree. On a reward, the system follows the arrows, and on a penalty the system goes in the opposite direction of the arrows.

alternate between the root nodes of the two trees.

2.2.8 Probabilistic Bisection Search

The goal of Probabilistic Bisection Search (PBS) [64, 65, 66] is to locate an unknown point $x^* \in [0, 1]$. To acquire intelligence on the location of x^*, one queries an Oracle about the relation between a point x and x^*. The Oracle responds by informing whether x is on the left or the right side of x^*. If we assume that the Oracle always tells the truth, then the well known deterministic Bisection Search that halves the search space with each query will efficiently find x^*. However, in PBS we assume that the Oracle provides correct answers with probability $p \in (0.5, 1.0]$ and erroneous ones with probability $1 - p$.

To exemplify the general applicability of this search problem, we reformulate it as a two player game where Player A thinks of a number and player B tries to find the number using a sequence of guesses. Each guess is of the type: *is the number less or more*

than the number x. If player A always answers truthfully, then the optimal scenario is a bisection search. However, if Player A is allowed to lie, then we obtain the PBS. We further distinguish the behavior of Player A based on the number of times he lies: if he on average is truthful, he is deemed *informative*. Else he is *deceptive*. In the special case where he is neither informative nor deceptive, the problem is intractable as Player A only provides white noise.

The origin of PBS can be traced back to Horstein [64], where the PBS is applied to handle a noisy communication channel between two agents trying to transmit a number. An important assumption is that $1-p$, the noise probability, is known. We then generate a probability distribution over the search space that we gradually refine towards singularity, using a Bayesian update rule, with the median of the posterior distribution as the point of interest. It has been shown that PBS has a geometric rate of convergence under the latter assumptions [65].

More formally, let $Z(x) \in \{\text{left}, \text{right}\}$ be the signal obtained from querying at point x, independent of all previous queries. The signal indicates the likely direction of x^* (left or right), relative to the queried point x. If $x^* < x$ then the response is $Z(x) = \text{left}$ with probability p and $Z(x) = \text{right}$ with probability $1 - p$. Likewise, if $x^* > x$ then the response is $Z(x) = (left$ with probability $1 - p$ and $Z(x) = \text{right}$ with probability p.

The PBA assumes a prior density f_0 on $[0, 1]$ that is positive everywhere. Let F_0 denote the corresponding cdf. Then for $n = 0, 1, 2, \ldots$, PBA follows these inference steps:

1. Identify the median of f_n, $X_n = F_n^{-1}(\frac{1}{2})$, which is the next query point.

2. Query the oracle at point X_n, to obtain the signal $Z(X_n)$.

3. Apply the Bayesian update rule to the current posterior f_n:

$$\text{if } Z(X_n) = \text{right}, f_{n+1}(y) = \begin{cases} 2pf_n(y) & \text{if } y \geq X_n \\ 2(1-p)f_n(y) & \text{else} \end{cases} \tag{2.2}$$

$$\text{if } Z(X_n) = \text{left}, f_{n+1}(y) = \begin{cases} 2(1-p)f_n(y) & \text{if } y \geq X_n \\ 2pf_n(y) & \text{else} \end{cases} \tag{2.3}$$

4. $n \leftarrow n + 1$

In other words, the idea is quite simply to move the mass of the distribution in the perceived direction of x^*, and lower the mass in the opposite direction. Thus, for each iteration, we obtain an increasingly sharper distribution surrounding x^*, the point of interest.

2.2.9 Learning Automata Knapsack Game

The Stochastic Fractional Non-linear Knapsack (SNEFK) problem is a challenging optimization problem with numerous applications, including resource allocation. The objective is to find the optimal mixture of materials that fit within a knapsack of a fixed, finite capacity. Assuming that the value function, the function that maps a set of materials into a scalar value is known, then the solution can be found through a direct application of Lagrange multipliers. However, in many real-world applications, such as resource allocation in web polling, the value function is uncertain, and in many cases unavailable altogether, and must therefore be learned. This learning will happen simultaneously with our optimization of the knapsack material mixture.

Thus, the particular type of uncertainty considered by the SNEFK problem is a value function giving a binary signal for each material in the knapsack, with the probability of a reward for a particular material directly tied to the amount of that material in the knapsack. Due to the stochastic nature, we try to optimize the expected knapsack value instead of directly optimizing the value of the materials.

More formally, let x_i be the amount of material i in the knapsack. Furthermore let $p_i(x_i)$ be its non-decreasing reward probability function such that by adding x_i about of i into the knapsack, we have a probability $p(x_i)$ of observing a reward for material i. Let $F_i(x_i)$ take the value 1 with probability $p_i(x_i)$ and conversely the value 0 with probability $1 - p_i(x_i)$. For a knapsack with capacity c the objective of SNEFK is then:

$$\begin{aligned} &\text{maximize} \sum_{1}^{n} E[F_i(x_i)] \\ &\text{such that} \sum_{1}^{n} x_i = c \\ &\text{where } x_i \geq 0,\ i = 1, \ldots, n. \end{aligned} \tag{2.4}$$

The Learning Automata Knapsack Game (LAKG) [67, 9] is based on two simple rules:

1. If material k receives a reward and the knapsack is not full, add more of material k.
2. If material k receives a penalty, and the knapsack is full, decrease the amount of material k.

To implement these rules, the LAKG scheme models each material as a Tsetlin automaton, and updates the automaton with a reward or penalty, as dictated by the knapsack value function. Each material k is assigned a Tsetlin automaton LA_k with states $1, \ldots, N$ where $s_k(t)$ is the current state of LA_k. The knapsack should at time t be filled with $\{s_1(t)/N, s_2(t)/N, \ldots, s_n(t)/N\}$. and receives a feedback $\{v_i(t), \}_1^n$, with

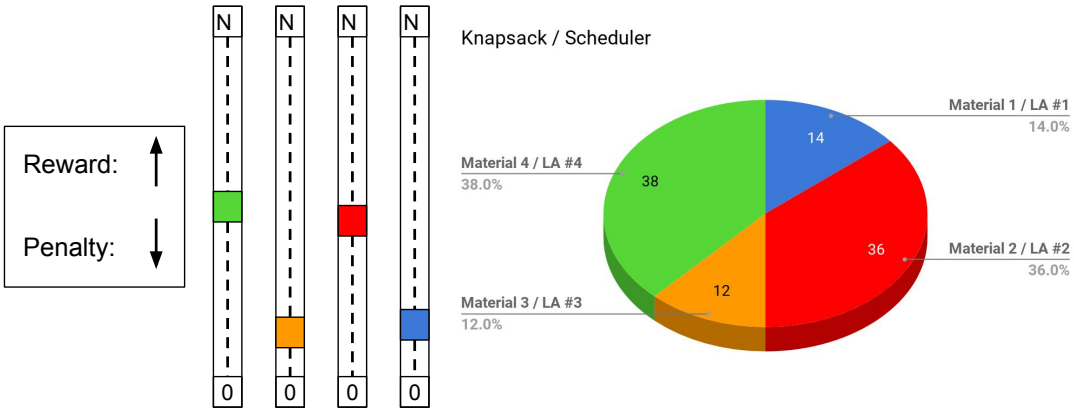

Figure 2.7: The LAKG scheme consists of a collection of TAs that together form the content of a knapsack. The figure depicts a team of four TAs with states from 0 to N, with the state giving the mix of the corresponding material in the knapsack.

$v_i(t) \in [0, 1], i = 1, \ldots, n$. The signal ϕ says if the knapsack is full or not, and is defined as:

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

The update scheme for each LA_k in LAKG is therefore as follows:

$$\begin{aligned} &\textbf{if } v_k(t) = 1 \textbf{ and } 1 \leq s_k(t) \leq N \textbf{ and not } \phi(t) \\ &\qquad s_k(t+1) := s_k(t) + 1 \\ &\textbf{if } v_k(t) = 0 \textbf{ and } 1 \leq s_k(t) \leq N \textbf{ and } \phi(t) \\ &\qquad s_k(t+1) := s_k(t) - 1 \\ &\textbf{otherwise} \\ &\qquad s_k(t+1) := s_k(t) \end{aligned} \tag{2.5}$$

An illustration of the LAKG scheme applied to a SNEFK problem with four materials is shown in Figure 2.7.

Finally, it is clear that the idea behind the LAKG scheme is based on the SPL scheme [10], however, the LAKG have some major differences[67]:

1. The Tsetlin automaton based SPL scheme in [10] is linear.

2. The SPL scheme [10] assumes the availability of an Oracle which informs the LA whether to go "left" or "right". In the SNEFK problem, the Oracle feedback must be inferred from observations.

Algorithm 2: Enhanced OMA

Input: The abstract set of objects, a number of states per action, a sequence of random queries in form (O_i, O_j)

Output: A periodic clustering of the objects into R partitions

Notation: ζ_i is the state of the abstract object O_j. It is an integer in the range $1, \ldots, RN$, where, if $(h-1)N + 1 \leq \zeta_i \leq hN$, then the object O_i is assigned to α_h

Method:

Initialize ζ_i for $1 \leq i \leq W$ randomly among the boundary state of classes, each class having W/R objects.

for *a sequence of* T *queries* **do**
- Read a query (A_i, A_j)
- **if** $(\zeta_i$ *div* $N) = (\zeta_j$ *div* $N)$ **then**
  - **if** ζ_i *mod* $N \neq 1$ **then**
    - $\zeta_i = \zeta_i - 1$
  - **if** ζ_j *mod* $N \neq 1$ **then**
    - $\zeta_j = \zeta_j - 1$
- **else**
  - **if** ζ_i *mod* $N \neq 0$ *and* $(\zeta_j$ *mod* $N) \neq 0$ **then**
    - $\zeta_i = \zeta_i + 1$
    - $\zeta_j = \zeta_j + 1$
  - **else if** ζ_i *mod* $N \neq 0$ **then**
    - **if** O_v*: unaccessed object in group of* O_i *where* ζ_v *mod* $N = 0$ **then**
      - $\zeta_j, \zeta_v = \zeta_v, \zeta_j$
    - $\zeta_i = \zeta_i + 1$
  - **else if** ζ_j *mod* $N \neq 0$ **then**
    - **if** O_v*: unaccessed object in group of* O_j *where* ζ_v *mod* $N = 0$ **then**
      - $\zeta_i, \zeta_v = \zeta_v, \zeta_i$
    - $\zeta_j = \zeta_j + 1$
  - **else**
    - temp $= \zeta_i$
    - $\zeta_i = \zeta_j$
    - $t =$
    - index of an unaccessed object in group of O_j where O_t is closest to ζ_j
    - $\zeta_t =$ temp

return *Partitions based on the states* $\{\zeta_i\}$

Chapter 3

Contributions

The main focus of this thesis is developing a completely new family of solution techniques, unifying Thompson sampling with advanced Learning Automata based solution schemes. The purpose is to address the inherent exploitation–exploration dilemma of the Learning Automata based solutions from a Bayesian perspective, thus increasing robustness, accuracy and speed of learning.

3.1 Contributions in the Stochastic Fractional Non-Linear Knapsack Problem

A natural application of Stochastic Fractional Non-Linear Knapsack Problem (SNEFK) is in the context of polling in web crawling. For web monitoring frameworks and search engines it is important to keep their indices and caches up-to-date, typically by means of polling. Achieving this, of course, relies on detecting the changes that the web resources undergo, typically by means of polling. The naive, and perhaps most common way of approaching this problem is to divide the available polling capacity equally among all the web resources, a clearly sub-optimal strategy, except in the case where the update frequencies of the web resources are equal.

This is different from the Bandits with Knapsacks (BwK) [68, 33] problem in many ways, but most significant is the fact that in BwK the feedback includes the consumption of resources (as a vector), instead of it being a part of the arm selection process. So the amount of resources consumed is not a part of the action space as in SNEFK.

To apply the SNEFK to the domain of web polling as is done by Granmo et al. [67], each web resource corresponds to a material, and the polling frequency of each web resource is proportional to the fraction of that "material" in the knapsack. The value of a knapsack is here equal to the expected number of updates detected in a time interval. As with most stochastic functions, we cannot directly observe this function; we can only

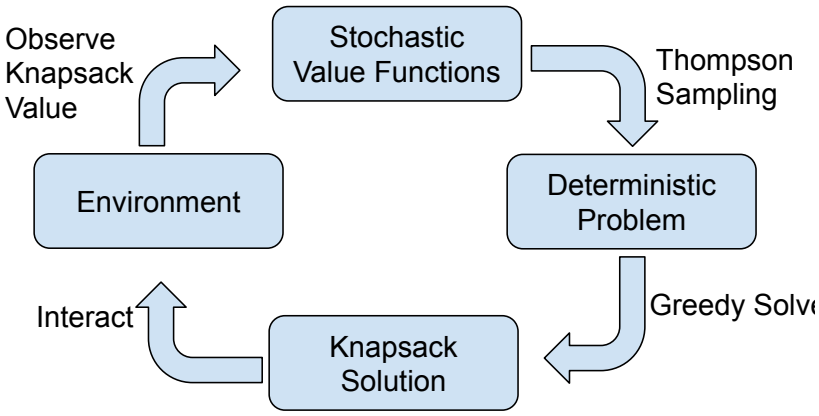

Figure 3.1: The GPOKS scheme is based on progressively updating the GP model of the value functions as more information is obtained. A deterministic problem is then sampled from the GP model using TS and solved using a greedy solver. This solution is then applied to the Environment to obtain a knapsack value that is subsequently used to refine the model. The main idea is that the closer the GP model is to the underlying environment, the closer the solution to the sampled deterministic problem is to the solution of the stochastic knapsack problem.

observe an instantiation, that is, the number of updates detected in a interval. However, we do assume that the probability of detecting an update is non-decreasing with time. That is, waiting a long time before polling a particular web site does not decrease the chance of detecting an update.

The paper presented in Appendix E introduces Gaussian Process based Optimistic Knapsack Sampling (GPOKS) for solving the SNEFK problem. The GPOKS is based on the observation that solving the SNEFK problem with known value functions is significantly easier than solving it with stochastic value functions.

Thus, GPOKS introduce a Gaussian Process (GP) based model of the stochastic value functions and applies Thompson Sampling (TS) to extract a deterministic problem that can be solved exactly with a greedy solver. The idea is therefore to tie the accuracy of the GP model directly to the quality of the SNEFK problem. However, to ensure that we explore the solution space sufficiently, we employ TS to handle the trade-off between exploring the model that we currently got, and exploring new options to potentially find a better solution.

As the GPOKS obtains more information about the value functions, the stochastic mapping grows closer to the underlying mapping, thus making the solution of the deterministic problem approach the solution of the original problem. The GPOKS scheme is illustrated in Figure 3.1.

3.1.1 Related Work

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 SNEFK Problem [67].

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 to acting on outdated 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 [69]. 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 may render the solution found with the latter approach progressively more inaccurate.

The LAKG scheme is described in Section 2.2.9, and the main difference between LAKG and GPOKS is the way in which the two schemes move in the solution space. The LAKG operates as a game between a set of LAs where each individual LA is restricted to take only small steps towards a solution. Therefore, the difference between each solution generated by it is also small, and thus LAKG can be seen as a slow but steady approach to solving the SNEFK problem. Our proposed TS-based scheme on the other hand, can make large jumps in the solution space based on te current posterior probability distribution over solutions.

3.1.2 Summary of the Contributions

In the paper presented in Appendix E, we improve upon the LAKG solution for SNEFK by recognizing that the value functions may be modeled as GPs that in turn can be solved using TS and a greedy solver.

To further enhance the performance of GPOKS, we apply Optimistic Thompson Sampling (OTS) as we sample the deterministic functions that represent a optimistic estimate. That is, its is sampled from Half-Normal Multivariate Gaussian distribution induced by the observations.

The resulting scheme, GPOKS significantly outperforms the state-of-art approached when applied to resource allocation in web polling.

3.2 Contributions in the Goore Game Problem

One of the most fascinating games studied in the field of artificial games is the Goore Game (GG). We describe it using the following informal formulation given in [5].

Imagine a large room containing N cubicles and a raised platform. One person (voter) sits in each cubicle, and a Referee stands on the platform. The Referee conducts a series of voting rounds as follows: On each round, the voters vote "Yes" or "No" (the issue is unimportant) simultaneously and independently (they do not see each other), and the Referee counts the fraction λ of "Yes" votes. The Referee has a unimodal performance criterion $G(\lambda)$, which is optimized when the fraction of "Yes" votes is exactly λ^*. The current voting round ends with the Referee awarding a dollar with probability $G(\lambda)$ and taking a dollar with probability $1 - G(\lambda)$ to every voter independently. On the basis of their individual gains and losses, the voters then decide, again independently, how to cast their votes on the next round.

The naive approach to these problems is to model each voter as a bandit problem. In decentralized decision making problems, however, a phenomenon regarding variance renders current bandit problem based solutions sub-optimal. Specifically, multiple decentralized decision makers are simultaneously exploring a collection of interacting bandits. This means that the variance of the reward distributions of each bandit problem is governed by the current level of exploration being manifested in the system as a whole. In other words, the variance of the reward distributions will be fluctuating with the degree of exploration taking place. Thus, initially, when exploration typically is significant, each decision maker should be correspondingly more conservative or cautious when interpreting the received rewards than they would be if exploring alone. Otherwise, by being too reckless, the decision maker may be led astray early on, converging to a sub-optimal decision.

The traditional approach to dealing with the above described fluctuation of reward distribution variance is to make learning more conservative. The purpose is to minimize the chance of decision makers converging prematurely. Obviously, the disadvantages of this approach is the corresponding loss in learning speed, caused by staying too conservative when exploration calms down.

In Appendix D, we present a paper that we have published addressing the conservative learning needed to solve GG, since GG is at heart, a decentralized learning problem. The core idea is to exploit that variance is an additive quantity, (cf. Bienayme formula: $\sigma^2 = \sum \sigma_i^2$). Then having each decision maker broadcast their uncertainty, expressed as a variance, allows each decision maker to gauge the total uncertainty in the system. The learning for an individual decision maker can then be adjusted for the total uncertainty, thus potentially greatly *accelerating* the learning when the rest of the decision makers are done exploring.

3.2.1 Related Work

The literature concerning the GG is sparse. It was initially studied in the general learning domain, and, as far as we know, was for a long time merely considered as an interesting but pathological game. Recently, however, the GG has found important applications within two main areas, namely, in Quality of Service (QoS) control in wireless sensor networks [70, 71, 72], and in cooperative mobile robotics, as summarized in [73].

The initial solution to the Goore Game by Narendra and Thathachar [5] is based on modeling each decision maker as a Tselin automaton with $2N$ states. Here, the first $[1, N]$ states denote that the player votes "no", and the last $[N + 1, 2N]$ states denote that the player votes "yes". After each round, if the player obtained a reward, it updates its state by strengthening its belief in its current vote. Conversely, if it obtained a penalty, it lessens its belief in its current vote.

More recently, Thathachar et al. [60] introduced a L_{R-I} based algorithm to the Goore Game and proved that it would converge to the Nash equilibrium of the game. A parallel version of the algorithm was also presented that improved its speed of convergence in the Goore game. They do, however, still note that the learning rate must be drastically reduced if good accuracy is to be maintained. Thus, they corroborate our initial assessment of the bottleneck for the convergence speed of the Goore Game.

The GG has found applications within the field of wireless sensor networks, as explained briefly here. Consider a base station that collects data from a sensor network. The sensors of the network are battery driven and have been dropped from the air, leaving some of them non-functioning. The functioning sensors can be switched either on or off, and since they are battery-driven, it is expedient to turn them off whenever possible.

The base station, on the other hand, has been set to maintain a certain resolution (i.e., QoS), and therefore requires that Q sensors are switched on. Unfortunately, it does not know the number of functioning sensors, and it is only able to contact them by means of a broadcast, leaving it unable to address them individually. This leaves us with the following challenge: *How can only the base station turn on exactly Q sensors, by means of its limited broadcast capability?*

Iyer et al. [72] proposed a scheme where the base station provided broadcasted QoS feedback to the sensors of the network. Using this model, the above problem was solved by modeling it as a GG [8, 71].

From the GG perspective, a sensor is seen as a voter that chooses between transmitting data and remaining idle in order to preserve energy. Thus, in essence, each sensor takes the role of a GG player that votes either "On" or "Off", and acts accordingly. The base station, on the other hand, is seen as the GG Referee with a uni-modal performance function $G(\cdot)$ whose maximum is found at Q normalized by the total number of sensors available. The "trick" is to let the base station (1) count the number of sensors that have turned on, and (2) use the broadcast mechanism to distribute, among the sensors, the corresponding reward based on the probability obtained from $G(\cdot)$. The application of the GG solution to the field of sensor network is thus both straightforward and obvious.

For the case when the target resolution Q is non-stationary, e.g. it is changing over time, Li et al. [71] proposed an LA based estimator scheme that both handles a non-stationary Q and gives theoretically proved convergence for stationary environments.

Tung and Kleinrock [8] have demonstrated how the GG can be used for coordinating groups of mobile robots (also called "mobots") that have a restricted ability to communicate. The main example application described in [8] consists of a fixed number of mobots that can either (1) collect pieces of ore from a landscape, or (2) sort already collected ore pieces. The individual mobots vary with respect to how fast they collect and how fast they sort these pieces of ore. In this context, the GG is used to make sure that the mobots choose their action so as to maximize the throughput of the overall collection and sorting system.

Another mobot application is that of combating anti-personnel mines by searching in an unpredictable environment, using a GG model to coordinate the mobots [74].

3.2.2 Summary of the Contributions

In the paper included in Appendix D, we propose a novel scheme for solving one particular class of decision making problems, namely, the Goore Game [62, 5].

The proposed scheme, *Accelerated Decentralized Learning in Two-Armed Bandit Based Decision Making (ADL-TAB)* directly and specifically addresses fluctuating reward dis-

tribution variances. To achieve this, we derive theoretical results that characterize the variance of the random rewards each individual decision maker experiences. Based on these theoretical results, each decision maker is able to accelerate its own learning as follows: When a decision maker chooses which arm to pull, it also submits a measurement of its degree of exploration expressed as a variance, which we refer to as arm selection variance.

Then, along with the random reward it receives from the arm pull, it also receives a signal that reflects the current aggregate level of exploration in the system. Using this signal, each decision maker accelerates its learning by taking advantage of the increasingly more reliable feedback that can be obtained as exploration gradually turns into exploitation.

Through a series of empirical test we verify that ADB-TAB outperforms existing schemes in the regular Goore Game. The tests control for a variety of $G(\cdot)$ objective functions with different characteristics, number of players and a white noise on the reward signal.

We additionally test ADB-TAB in a QoS management scenario, where the number of sensors is controlled through a stochastic birth-death process. And we demonstrate that the decentralized arm selection signal greatly accelerates the speed of convergence in an already stable system. In particular, we observe that if the system is close to stable, the replacement of a sensor has little to no impact on the stability of the system as a whole, since the new sensor quickly convergences, due to the stability of the remainder of the sensors. And again, greatly outperforming existing schemes.

3.3 Contributions in the Equipartition Problem

As explained in Section 2.2.4, the objective of SO-EPP is to partition a set of objects to reflect their underlying dependency, as inferred from a sequence of observations, while simultaneously keeping the cardinality of each partition equal. In real-life scenarios, we observe the need to refine the solution ever further, for example: fix an object to a subset of partitions, force a set of objects to be in the same partition, or conversely, insist that some objects should not be in the same partition. The existing OMA scheme does not allow these types of restrictions. We coin this new problem as the Constrained Stochastic Online Equi-Partitioning Problem (CSO-EPP).

As such, we give a real-life example of the CSO-EPP in the context of *order picking*. Order picking is defined as "the process of retrieving products from storage (or buffer areas) in response to a specific customer request" [75]. Order picking occurs both in warehouses employing an Automated Storage/Retrieval System (AS/RS), and in those

depending on manual labor. Tompkins et al. identified travel time as the main factor for optimizing order-picking [76]. For this reason, to facilitate efficient retrieval of products, frequently ordered products should be placed in easy to reach locations. Additionally, products that are often ordered together should be placed in near proximity of each other. By doing so, we can systematically reduce the total travel time needed to collect orders. Examples of constraints in this scenario are for instance that that all frozen objects should be in a freezer, even though they are rarely purchased together. Other constraints in real life are that all products from a brand must be co-located on the request of the manufacturer, or that fragile or heavy objects must be placed on shelves close to the floor.

The OMA scheme operates by taking short, discrete steps towards the solution. That is; if two objects are seen together, each object takes one step towards each other. This increase the proximity of the two objects. To deal with noise, it is assumed that the signal observations outnumber the noisy observations. We then, see that OMA is essentially an random walk with drift from an initial state A to the solution state B, where the drift factor is given by the noise-to-signal ratio. Even though the OMA scheme is effective, it leaves a lot of information unused, such as the state of objects not in the current query. In addition, the ability to do only small steps prevents the solution from exploring solutions that are likely but far from the current setup.

To alleviate the disadvantages in OMA/POMA, we present a paper in Appendix A, which details a Bayesian Network based scheme (BN-EPP) that encodes both optional restrictions on the solution space as well as finding the optimal solution for the SO-EPP. For larger problems, finding the optimal solution is not feasible, and thus we also introduce an approximate solution finder for these cases.

3.3.1 Related Work

The OPP is already a thoroughly studied problem [77, 78]. Yet, research on its fascinating variant, SO-EPP [37, 79, 80, 17, 81, 36], is surprisingly sparse despite its many real-world applications. To cast further light on the unique properties of SO-EPP, we will here relate it to two similar problems, namely, the Poset Ordering Problem (POP) and the Graph Partitioning Problem (GPP), before reviewing approaches and applications that are specifically designed for SO-EPP.

The Poset Ordering Problem (POP). A poset is defined as a set of elements with a transitive partial order, where some elements may be incomparable [82]. A binary relation that is reflexive, antisymmetric, and transitive defines this ordering, referred to as a less-than-or-equal relation ($\leq$). The standard less-than- or-equal relation for integers is an example of such a partial ordering on the set of integers. In the poset ordering problem, the goal is to establish the partial ordering of a poset by comparing

pairs of elements, while simultaneously using the less-than-or-equal relation as few times as possible. Accordingly, both in SO-EPP and in POP, one must learn from paired elements to uncover an underlying, more complex structure. That is, in POP, the less-than- or-equal relation is applied iteratively to pairs of elements, while in SO-EPP, an in-the-same-partition relation of SO-EPP is used instead. Whereas the less-than-or-equal relation found in POP is both reflexive and transitive, it is not symmetric, i.e., $A \leq B$ does not imply $B \leq A$. The in-the-same-partition relation, on the other hand, is symmetric. This means that the solution of SO-EPP is not a partial ordering, but a set of equivalence classes, leading to unique solution schemes.

The Graph Partitioning Problem (GPP). The GPP is in its most general form an NP-complete problem [83]: Let $G = (V, E)$ be a graph with a set of vertices V and a set of weighted edges E. In graph equipartitioning, the goal is to partition V into k subsets $V_1, V_2, \ldots, V_k$ of equal cardinality. In all brevity, a solution to a GPP instance is a partitioning that minimizes the sum of those edge weights that cross between different vertex sets, (V_i, V_j), $i \neq j$ [84]. The SO-EPP can thus be cast as a GPP if the frequencies of object co-occurrence are known for all object pairs. We could then form a complete graph, $G = (V, E)$, where each vertex in V represents an object. Further, the weight of an edge between a pair of objects is simply the frequency with which we observe that particular pair. The resulting GPP can then be solved by any GPP solver [85, 86, 87].

Query Statistics and Spectral Clustering (SC). The naive approach to solving the SO-EPP is the usage of query statistics [81]. That is, we introduce a two-step procedure: First, for a sufficient long period, count the number of times each object is paired with the other objects. Second, based on the counts, cluster the objects based on the counts. This approach does, however, suffer from first requiring a exponentially long estimation period, and is thus unfeasible for larger problems. In addition, it is also not an online method.

Another approach to solving GPPs is based on Markov Random Walks. By defining a Markov Random Walk over the graph G, one can perform clustering based on the eigenvalues of the resulting transition matrix [88].

This method is based on using query statistics to generate a transition matrix from the frequency counts. The baseline for this approach is Spectral Clustering (SC) where the eigenvalues are used as a low-dimensional embedding of the problem space. SC can then effectively generate clusters using the MultiClass Normalized Cuts scheme [89].

In Section 5.2 in Appendix A, we introduce a simple yet strong baseline based on SC and query statistics.

OMA and POMA The OMA scheme [17, 36] and its improved version, the POMA scheme [37], described in Section 2.2.4 and Section 2.2.5, are the leading schemes for

SO-EPP. The interaction between these is that OMA will converge quickly in a noise-free environment. The SO-EPP environment, however, gives no such guarantees. Thus, to reduce the noise, POMA introduces a filter mechanism where it, based on a query estimation, removes noisy queries. It thus allows OMA to converge significantly faster.

In Mamaghani et al. [90], they apply OMA to partition a Module Dependency Graph (MDG) that models the different dependencies between software modules. In MDG, each node represents a system module such as a file or a class, and the edges are their relationships, for example function calls and inheritance relationships. The aim is to produce partitions of system modules that concurrently minimize the inter-connectivity (the connections between two partitions) and maximize the intra-connectivity (the connections within a cluster). The utilization of OMA is enabled by casting each node in the graph as a object in the OMA and using the relationships as a stream of observations.

Another application of OMA is to solve the NP-hard problem of mapping a finite alphabet A onto a set of keys B on a keyboard with the limitation that the $|A| > |B|$ [91]. From the pigeonhole principle it follows that at least one key must have multiple symbols assigned to it. The problem is then to assign to each $a \in A$ a key $b \in B$ so as to minimize the chance of ambiguous words. This problem is solved with the OMA scheme by realizing that each key represents a partition, each symbol in A an object, and then penalizing ambiguous words and rewarding words with a unique representation.

3.3.2 Summary of the Contributions

In the paper included in Appendix A, we construct a Bayesian Network based scheme BN-EPP to capture the intricasies of the CSO-EPP in such a manner that, given sufficient computational capability, the solution is optimal. For large scale problems, we provide an efficient approximate solution called Walk-BN-EPP that builds upon the well-known and effective WalkSAT [92] solver for the NP-complete Boolean satisfiability (SAT) problem.

Moreover, we perform a thorough review of techniques using both artificial data and a real-world warehouse order picking problem, and show that BN-EPP outperform all existing techniques. In addition, we introduce a strong baseline based on Spectral Clustering, and demonstrate that its performance is close to BN-EPPs state-of-the-art performance on artificial data.

3.4 Stochastic Point Location and Stochastic Root Finding

As shown in Chapter 2, the two fields of Learning Automata and Operation Research have independently discovered and developed the SPL and PBS problems. Our work in Appendix B is the first work published that bridges the two fields, and presents a unifying view of both fields by applying a wide range of empirical tests to give a clear picture of the state-of-art when combining research from both fields. Note that for consistency in notation, during the rest of the thesis, we will henceforth apply the terminology from the SPL formulation of the problem.

The objective in SPL is finding an optimal point λ^* on an interval $I = [0, 1]$ by a sequence of queries $\lambda(n)n = 1, \ldots$ where $\lambda(n)$ is the nth query, and the environment E responds to $\lambda(n)$ with $\beta_n \in \{\text{LEFT, RIGHT}\}$, indicating the correct direction of λ^* from $\lambda(n)$ with probability p and the wrong direction with probability $1 - p$.

The original version of SPL is a direct application of the Tsetlin automaton, and progresses from the initial position $\lambda(0)$ towards λ^* by taking small, discrete steps in the direction indicated by the oracle. Evidently, assuming the oracle on average returns the true direction, the SPL will eventually converge. However, due to the small size of the fixed steps, the number of interactions with the environment can be excessive. For example, if the first k interactions all indicate that λ^* is to the left, it could be much more efficient to query the environment for a point located further left than $\lambda(n) + \epsilon$ with ϵ being a small value.

To address this problem, we observe that the structure of the problem can be modeled as a BN, where the BN represents an encoding of a probability distribution over the interval I. Furthermore, if we generate the BN for a fixed value of p, we obtain the PBS algorithm. Clearly, knowing the value of p a priori is an unrealistic assumption, and we thus introduce TS-SPL, a model that include the p value as part of the learnable model. Since the p value is a part of the model, we, unlike previous approaches, in both SPL and PBS can remove the limitation on p, and let p encompass the unit domain $[0, 1]/\{\frac{1}{2}\}$. The special case of $p = \frac{1}{2}$ means the feedback is white noise. The remaining question is then where to query the environment next? In PBS, the median of the distribution is used as the next query. However, this approach fails when taking the possibility of $p < 0.5$ into consideration. We thus, model the selection problem as a MABP, and utilize Thompson Sampling to determine the next query point $\lambda(n + 1)$ based on the posterior distribution.

Another important question is how to deal with with SPL when the target location λ^* is non-stationary. That is, if λ^* move along the line in jumps or with continuous motion. To handle the additional complications introduced by a non-stationary target

λ^*, we introduce the Non-Stationary TS-SPL (TS-NSPL).

The idea behind TS-NSPL is to first partition the observations into two sections: observations = *tail* $\|$ *head*. With the m most recent observations located in the *head*, the remaining observations in *tail*, thus *head* is a sliding window of size m. We then employ two independent TS-SPL algorithms, one operating on all observations and one operating on the *head* observations denoted O and H respectively. The core idea is then to apply the Jensen–Shannon divergence between the O and H posteriors to measure their similarity. If the two distributions differ, we can draw the conclusion that the point λ^* has changed location. And consequentially, we flush the observations located in O, and utilize H as the new O, effectively restarting the two TS-SPL algorithms. The details of the algorithm is presented in Appendix C.

3.4.1 Related Work

Adaptive Step Searching (ASS) [93] is currently the leading approach to solving SPL problems in the LA domain, although it is outperformed by Hierarchical Stochastic Searching on the Line (HSSL) [63] in highly volatile non-stationary environments [93]. Optimal Computing Budget Allocation (OCBA) has also been applied to SPL [94], and provides stable solutions while converging slightly slower than ASS. Unfortunately, these state-of-the-art schemes fail when the majority of obtained directions mislead rather than guide. Indeed, by naively following the directions provided under such circumstances, one is systematically led away from the optimal point. We refer to these kinds of problem environments as *deceptive* environments, as opposed to *informative* ones, which are explained in more detail below.

To the best of the author' knowledge, the CPL-AdS [95] was the first known approach handling deceptive SPL environments. CPL-AdS has two phases. In the first phase, a sequence of intelligently selected questions is used to classify the environment as either informative or deceptive. By spending a sufficient amount of time in this phase, the classification can be made arbitrarily accurate. In the second phase, a regular SPL scheme is applied, except that the directions obtained are reversed if the problem environment was classified as deceptive in the first phase. This means that the scheme may have to remain in the first phase for an an extensive amount of time to ensure that the problem environment is correctly classified, or else, one risks being systematically misled in the second phase. These properties largely render CPL-AdS inappropriate for online or anytime problem-solving.

Recently, HSSL has been extended by Zhang et al. to cover both informative and deceptive environments, using a Symmetric HSSL (SHSSL) [21]. This scheme essentially runs two HSSL schemes in parallel: one regular, which handles informative environments,

and one which inverts all feedback from the environment to handle deceptive environments. The hierarchy navigation capabilities of HSSL are then exploited to allow SHSSL to switch between the two HSSLs, depending on the nature of the environment. However, a significant limitation of HSSL, namely, that π must be larger than the conjugate of the golden ratio, carries over to SHSSL. Indeed, SHSSL fails to converge for $\pi \in [0.382, 0.618]$, which amounts to approximately 30% of the feasible values for π. This is in contrast to the approach we propose in this paper, as well as to CPL-AdS [95], since both of these schemes work well in the entire range of π (apart from $\pi = \frac{1}{2}$).

From the perspective of PBS, the original PBS method by Horstein [64] is still a serious contender to solve the SPL problem. In the context of Active Learning [96, 97], the Burnashev-Zigangirov (BZ) Algorithm [98] has been widely used.

The Generalized Binary Search (GBS) problem can be formulated as follows [99, 100]. Consider a collection of unique binary-valued functions H defined on a domain X. Each $h \in H$ is defined as a mapping from X to $\{-1, 1\}$. Assume that there exists in the collection an optimal function $h^* \in H$ that produces the correct binary labeling for each $x \in X$. For each query, $x \in X$, the value of $h^*(x)$ is observed, possibly corrupted by independent binary noise. The objective is then to determine the function h^* using as few queries as possible. Restricting H to the class of threshold binary functions has the effect of turning the GBS into the SPL problem. If the feedback is noiseless, then the problem boils down to the combinatorial problem of finding an optimal decision tree in the H space, a problem that Hyafil and Rivest showed to be NP-complete [101, 99]. The Soft-Decision Generalized Binary Search (SDGB-Search) [100, 99] is the *state-of-art* algorithm for finding $h^*(x) \in H$, when the binary reward signal is corrupted by noise.

For the case when p is not known a priori, the PBS, a recent paper by Frazier et al. [102] demonstrate an alternative approach to removing the dependency of PBS on knowing the fixed noise probability p. Instead of applying a Bayesian Prior over p, as done in TS-SPL, they introduce a frequency based approached, referred to as PowerTest-PBS (PT-PBS). PT-PBS is based on repeatedly sampling the underlying function $g(x)$ until a pre-specified confidence α is achieved on a hypothesis test over the sign of the feedback of $g(x)$. This can be seen as the PBS version of the CPL-AdS scheme found in LA literature [95]. They further demonstrated that the asymptotic convergence of PT-PBS is similar to that of Stochastic Approximation [103, 104].

3.4.2 Summary of the Contributions

The paper included in Appendix B introduces a formula for combining TS and a Bayesian update scheme in the TS-SPL algorithm. More specifically, the contributions of the paper can be summarized as follows:

1. We introduce the novel TS-SPL scheme that represents the solution space of N-Door Puzzles, and SPL problems, in terms of a Bayesian model. As opposed to competing solutions that merely maintain and refine a single candidate solution, our Bayesian model encompasses the complete space of candidate solutions at every time instant. This Bayesian representation of the problem opens up for efficient exploration and exploitation of the solution space with Thompson Sampling.

2. We formulate a compact and scalable Bayesian representation of the solution space that simultaneously captures both the location of the optimal point (arm), as well as the probability of receiving correct feedback.

3. We link TS-SPL to so-called Stochastic Bisection Search, and unify accompanying methods under the umbrella of Thompson Sampling.

4. Similarly, we enhance Soft Generalized Binary Search (SGBS), Probabilistic Bisection Search (PBS) and Burnashev-Zigangirov Algorithm (BZ) by introducing novel parameter free solutions that take advantage of our Bayesian model of the N-Door Puzzle/SPL problem. This approach eliminates previous reliance on prior knowledge of the degree of noise affecting the system to be optimized.

5. We provide the first unified empirical comparison of the key state-of-the-art SPL/SRF solvers.

6. We finally demonstrate the empirical performance of TS-SPL for both SPL and SRF problems. TS-SPL outperforms state-of-the-art algorithms in both informative and deceptive environments, except that it is beaten by the SGBS and BZ schemes with correctly specified observation noise.

3.5 Travel Time Estimation

An important part of planning a journey is estimating the travel time. Without knowing the time it will take to travel between two locations it can be difficult to plan ahead and ensure that things go according to plan. Many services already provide good estimates for well-known scenarios such as car travel and public transport. These routes are typically calculated by first determining a route and then adding up the individual components of that route to obtain the total travel time.

However, in many situations, the navigation system may fail to provide adequate information to form a route, leaving the it unable to provide a travel time estimate. These situations could occur for instance when hiking cross country or traveling in an area where shortcuts and obstacles that do not appear on maps are frequent, such as in

urban city centers. An alternative approach focuses on estimating the *true road distance*, and while this is an interesting approach, this type of data is significantly harder to gather from real-world data, requiring not only a timekeeping device but also some way to accurately track velocity. Thus, we avoid the above complications by instead focusing on the actual time it takes to travel between two points.

Active Learning (AL) [105, 106] tries to alleviate the problem of data sparsity by actively selecting samples that minimize the total number of samples needed to do accurate inference. To guide the AL, an important factor is making the model uncertainty a first-class citizen, and thus, we employ a Bayesian model to allow us to measure uncertainty directly. For a complex model that does not directly follow a well-known distribution, we turn to a Probabilistic Programming Language (PPL) that not only gives us the opportunity to explicitly specify the model in terms of a data generation process, but also allow us to apply a powerful Bayesian inference to model problems involving uncertainties.

In the paper presented in Appendix F, we present an algorithm for travel time estimation that based on not only learning an estimator for travel time but also to provide guidance to what locations should be visited to obtain the best results using the least amount of observations.

3.5.1 Related Work

Probabilistic Programming

Probabilistic Programming (PP) is an attempt to close the representation gap between the much celebrated probabilistic graphical models (PGM) such as Bayesian Networks and Markov Networks and the more specialized algorithms that are typically represented as a mixture of pseudo code, natural language, and mathematics. The idea is thus to express the entire model, from sample generation to the joint distribution, and let the underlying framework handles the inference. This alleviates the need for highly specialized algorithms and lets the designer focus on designing a correct model rather than a model that it is easy to do inference on. With the advances in computational power, numerous PPLs has appeared in the literature, for instance, PyMC3 [107] that is built on top of the Theano framework [108], or Edward [109] that is built on top of Tensorflow [110].

Active Learning

In the highly effective Query By Committee (QBC) [111, 112] algorithm, a committee of unique learners label each potential data point, that is, it is a in a pool-based setting where each data point in the pool is labeled by each learner. The next point that queries the oracle for its true label will be as the point where the learners have a maximal

disagreement. For the simple case of binary labeled points and two learners, any point where the two learners disagree is therefore a possible next query point. In cases where the labels are not binary, or even discrete, an alternative approach is to select the point that is expected to reduce prediction error the most [112]. For real-valued regression problems the point that maximizes the variance is selected [113].

A critical aspect of the QBC algorithm is the disagreement between the learners. In the original work [111], a randomized algorithm was used. However, a more general approach is to train the same algorithm on different subsets of the data, as in query by bagging and query by boosting [114].

Bandit based active learning is also well-explored in the literature [115, 116, 117]. These approaches are ill-suited to travel time estimation problems, since they require a pool based approach where one can track the uncertainty for each possible query-point as part of the active learning.

Distance Estimation

The field of Distance Estimation (DE) has primarily been dominated by the use of parameterized Distance Estimating Functions (DEFs) of a simple yet effective form. These functions are calibrated using a set of inter connected points and their distances [118], maximizing the Goodness of Fit (GoF) between the observed values and the underlying function [119]. Other approaches have been created by adding various modifications to the DEFs such as rotational robustness by rotating the coordinate axes [120], or using non-parametric DEF for greater representative flexibility [121]. This is in contrast to mapless path planning [122], for instance for Autonomous Underwater Vehicles (AUV) that mainly focus on obstacle avoidance while reaching the target, not on how long it takes.

From an application point of view, DEFs have been used in travel time estimation for delivery scheduling [120] as well as in other operational models, such as service systems and strategic decision processes [123].

Recently, an Adaptive Tertiary Search (ATS) based method that does not explicitly depend on GoF was proposed [124]. The, ATS instead depends on the sign of the difference between the estimated distance and the actual, observed distance, and can be seen as a form of stochastic gradient descent.

3.5.2 Summary of the Contributions

The included paper in Appendix F shows how a powerful PPL model combined with TS for exploration can rapidly calibrate a travel time model.

We have proposed TS-PPL, an effective scheme for performing Active Learning in Probabilistic Programs. We have shown that TS-PPL can be applied to both a simple regression problem, and to a more complex problem like the Travel Time Estimation problem. Our method significantly outperforms the strong baseline of Query by Committee as well as passive learning for Travel Time Estimation, and gives comparatively better results in the case of regression. We hope that our results will inspire more researchers to apply probabilistic programming in their research.

Chapter 4

Conclusion and Future Research

In this thesis, we presented a Bayesian perspective on several efficient LA schemes, with the goal of providing a way to accelerate learning even further by incorporating Bayesian Inference over a posterior coupled with Thompson Sampling for action selection.

The traditional LA approach assumes that the environment on average guides the state of the LA toward convergence to an optimal state. With this assumption, even in the case of a truthful, non-noisy environment, the LA will still walk the same path as in the scenario where noise is injected in to the observations from the environment, with the exception that occasionally the LA will take a diverging step due to noise. However, since the environment on average guides the LA toward convergence, asymptotically this noise does not provide a hindrance to convergence.

On the other hand, for a finite time scenario, it is critical to fully exploit the knowledge about both the environment one operates in, as well as about any other prior knowledge. In this case, a Bayesian model allows a global picture of the problem space, allowing the explorer to jump from location to location while simultaneously quantifying the uncertainty. In this scenario, we show that Thompson Sampling is highly suited for navigating a Bayesian model, handling the exploration-exploitation trade-off in such a way that the algorithm quickly converges without making too many severe mistakes.

Several overall research directions for further investigation arise from this thesis. They are discussed below.

4.1 Further work in the Stochastic Fractional Non-Linear Knapsack Problem

1. In this thesis, we solved the SNEFK problem by assuming that the underlying knapsack value functions remained constant. However, we did not consider the case where the value functions contains a temporal dependency. Clearly, in situations

where the assumption of constant value functions is false, the GPOKS algorithm will struggle. One possible line of research to integrate temporal differences could be to extend the GP for each material to explicitly model the time component.

2. An interesting venue of research is to consider not only a single GPOKS but instead a game of interacting GPOKS for solving networked and hierarchical resource allocation problems. In this scenario, the Bayesian underpinning of GPOKS could be used to create problem-specific likelihoods and prior functions to handle these complex interactions.

4.2 Further work in the Goore Game Problem

1. In this thesis, we solved the Goore Game problem for a fixed stationary reward function using ADL-TAB. By incorporating a decentralized abnormality detection to track changes, we could handle the non-stationary behavior in a principled manner.

2. The current ADL-TAB scheme is based on solving a single Goore Game. However, there is the case of multiple, overlapping Goore Games, where each player participates in several co-current Goore Games. By directly modeling this multi-game interaction in ADL-TAB, we could potentially handle even more complex QoS scenarios.

4.3 Further work in the Equipartition Problem

1. In this thesis, we presented a solution for the Equipartition problem. However, the BN-EPP scheme could be expanded to cover other classes of stochastic optimization problems such as graph partitioning and poset ordering problems, potentially outperforming generic off-line techniques such as Particle Swarm Optimization (PSO) [125], Genetic Algorithm (GA) [126], and Ant Colony Optimization (ACO) [127].

4.4 Further work in Stochastic Point Location and Stochastic Root Finding

1. In this thesis, we presented a unified view combining the field of LA with the literature surrounding the PBS algorithm. An important venue for further research would be to investigate if new and improved solutions could be crafted by merging the advantages from each field.

2. Another important avenue for future work is the establishment of theoretical results, including proofs of convergence, to corroborate the purely empirical findings presented in this paper. We suggest that a promising starting point for such an endeavor would be to combine the theoretical properties of TS [44, 45, 48] with the theoretical results of PBS [65, 102], as they are closely related.

3. The TS-NSPL scheme was developed to handle the case of non-stationary SPL problems. It introduces some necessary parameters to tune the abnormality detection behavior. These parameters could potentially be integrated into the likelihood equivalent to how TS-SPL integrates the noise parameter found in PBS, thus obtaining a more robust method.

4.5 Further work in Travel Time Estimation

1. The TS-PPL scheme is based on the assumption that all travel is performed in a similar manner. However, an interesting venue for further work would be to explicitly model different forms of travel, such as taking the train or walking. This could open up for even more precise models of travel time estimation.

2. Just as in the case of SPL and SRF, another important avenue for future work is the establishment of theoretical results, including proof of convergence, to corroborate the purely empirical findings presented in this paper. We suggest that a promising starting point for such an endeavor would be to combine the theoretical properties of TS [44, 45, 48] with the theoretical results of PBS [65, 102] as they are closely related.

Bibliography

[1] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[2] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[3] Robert R Bush and Frederick Mosteller. A stochastic model with applications to learning. *The Annals of Mathematical Statistics*, pages 559–585, 1953.

[4] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):148–177, 1979.

[5] Kumpati S Narendra and Mandayam AL Thathachar. *Learning Automata: An Introduction*. Prentice Hall, Inc, 1989.

[6] Michael L Tsetlin. On behaviour of finite automata in random medium. *Avtom I Telemekhanika*, 22:1345–1354, 1961.

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

[8] Brian Tung and Leonard Kleinrock. Using finite state automata to produce self-optimization and self-control. *IEEE Transactions on Parallel and Distributed Systems*, 7(4):439–448, 1996.

[9] 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*, 37(1):166–175, 2007.

[10] B John Oommen. Stochastic searching on the line and its applications to parameter learning in nonlinear optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(4):733–739, 1997.

[11] B John Oommen, Sang-Woon Kim, Mathew T Samuel, and Ole-Christoffer Granmo. A Solution to the Stochastic Point Location Problem in Metalevel Nonstationary Environments. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(2):466–476, 2008.

[12] O.-C. Granmo and N Bouhmala. Solving the Satisfiability Problem Using Finite Learning Automata. *International Journal of Computer Science and Applications*, 4(3):15–29, 2007.

[13] Noureddine Bouhmala and Ole-Christoffer Granmo. Stochastic Learning for SAT-Encoded Graph Coloring Problems. *International Journal of Applied Metaheuristic Computing*, 1(3):1–19, 2010.

[14] A. Yazidi, O.-C. Granmo, and B.J. Oommen. Service Selection in Stochastic Environments: A Learning-Automaton based Solution. *Applied Intelligence*, 36(3):617–637, 2012.

[15] Vegard Haugland, Marius Kjølleberg, Svein-Erik Larsen, and Ole-Christoffer Granmo. A Two-Armed Bandit Collective for Hierarchical Examplar based Mining of Frequent Itemsets with Applications to Intrusion Detection. *Transactions on Computational Collective Intelligence XIV*, 8615:1–19, 2014.

[16] Ole-Christoffer Granmo and B John 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.

[17] B. John Oommen and Daniel C. Y. Ma. Deterministic learning automata solutions to the equipartitioning problem. *IEEE Transactions on Computers*, 37(1):2–13, 1988.

[18] Mina Ghavipour and Mohammad Reza Meybodi. A streaming sampling algorithm for social activity networks using fixed structure learning automata. *Applied Intelligence*, 48(4):1054–1081, 2018.

[19] B John Oommen, Sudip Misra, and Ole-Christoffer Granmo. Routing bandwidth-guaranteed paths in mpls traffic engineering: A multiple race track learning approach. *IEEE Transactions on Computers*, 56(7):959–976, 2007.

[20] Anis Yazidi and B John Oommen. On the analysis of a random walk-jump chain with tree-based transitions and its applications to faulty dichotomous search. *Sequential Analysis*, 37:31–46, Jan 2018.

[21] Junqi Zhang, Yuheng Wang, Cheng Wang, and MengChu Zhou. Symmetrical hierarchical stochastic searching on the line in informative and deceptive environments. *IEEE transactions on cybernetics*, 47(3):626–635, 2017.

[22] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[23] Peter Auer and Ronald Ortner. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

[24] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[25] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[26] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[27] Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Parallelised bayesian optimisation via thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 133–142, 2018.

[28] Andreas Krause and Cheng S Ong. Contextual gaussian process bandit optimization. In *Advances in neural information processing systems*, pages 2447–2455, 2011.

[29] Kevin Lloyd and David S Leslie. Context-dependent decision-making: a simple bayesian model. *Journal of the Royal Society Interface*, 10(82):20130069, 2013.

[30] Pedro A Ortega and Daniel A Braun. Generalized thompson sampling for sequential decision-making and causal inference. *Complex Adaptive Systems Modeling*, 2(1):2, 2014.

[31] Judea Pearl. *Causality: Models, Reasoning and Inference*, volume 29. Springer, 2000.

[32] Finn V Jensen and Jianming Liang. drhugin a system for value of information in bayesian networks. 1994.

[33] Shipra Agrawal. Recent advances in multiarmed bandits for sequential decision making. In *Operations Research & Management Science in the Age of Analytics*, pages 167–188. INFORMS, 2019.

[34] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257, 2011.

[35] Ole-Christoffer Granmo. Solving two-armed bernoulli bandit problems using a bayesian learning automaton. *International Journal of Intelligent Computing and Cybernetics*, 3(2):207–234, 2010.

[36] William Gale, Sumit Das, and Clement T. Yu. Improvements to an algorithm for equipartitioning. *IEEE Transactions on Computers*, 39(5):706–710, 1990.

[37] Abdolreza Shirvani and B John Oommen. On enhancing the object migration automaton using the pursuit paradigm. *Journal of Computational Science*, 24:329–342, 2018.

[38] Douglas E Comer, David Gries, Michael C Mulder, Allen Tucker, A Joe Turner, Paul R Young, and Peter J Denning. Computing as a discipline. *Communications of the ACM*, 32(1):9–23, 1989.

[39] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[40] William R Thompson. On the theory of apportionment. *American Journal of Mathematics*, 57(2):450–456, 1935.

[41] Malcolm Strens. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pages 943–950, 2000.

[42] Ole-Christoffer Granmo and Stian Berg. Solving non-stationary bandit problems by random sampling from sibling kalman filters. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 199–208. Springer, 2010.

[43] Steven L Scott. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

[44] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012.

[45] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial intelligence and statistics*, pages 99–107, 2013.

[46] Benedict C May, Nathan Korda, Anthony Lee, and David S Leslie. Optimistic bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research*, 13(Jun):2069–2106, 2012.

[47] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[48] Shi Dong and Benjamin Van Roy. An information-theoretic analysis for thompson sampling with many actions. In *Advances in Neural Information Processing Systems*, pages 4161–4169, 2018.

[49] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[50] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

[51] Aijun Bai, Feng Wu, and Xiaoping Chen. Bayesian mixture modelling and inference based thompson sampling in monte-carlo tree search. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 1646–1654. Curran Associates, Inc., 2013.

[52] Kris Johnson, David Simchi-Levi, and He Wang. *Online Network Revenue Management using Thompson Sampling*. Harvard Business School, 2015.

[53] Eric M Schwartz, Eric T Bradlow, and Peter S Fader. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4):500–522, 2017.

[54] Daniel N Hill, Houssam Nassif, Yi Liu, Anand Iyer, and SVN Vishwanathan. An efficient bandit algorithm for realtime multivariate optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1813–1821. ACM, 2017.

[55] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. Omnipress, 2010.

[56] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient thompson sampling for online matrix-factorization recommendation. In *Advances in Neural Information Processing Systems*, pages 1297–1305, 2015.

[57] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, pages 4026–4034, 2016.

[58] Steven L Scott. Multi-armed bandit experiments in the online service economy. *Applied Stochastic Models in Business and Industry*, 31(1):37–45, 2015.

[59] Kumpati S Narendra and Mandayam AL Thathachar. Learning automata - a survey. *IEEE Transactions on systems, man, and cybernetics*, (4):323–334, 1974.

[60] MAL Thathachar and MT Arvind. Solution of goore game using modules of stochastic learning automata. *Journal of the Indian Institute of Science*, 77(1):47, 2013.

[61] B John Oommen and Mariana Agache. Continuous and discretized pursuit learning schemes: Various algorithms and their comparison. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(3):277–287, 2001.

[62] ML Tsetlin. *Automaton Theory and Modeling of Biological Systems*. New York: Academic Press, 1973.

[63] Anis Yazidi, Ole-Christoffer Granmo, B John Oommen, and Morten Goodwin. A hierarchical learning scheme for solving the stochastic point location problem. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 774–783. Springer, 2012.

[64] Michael Horstein. Sequential transmission using noiseless feedback. *Information Theory, IEEE Transactions on*, 9(3):136–143, 1963.

[65] Rolf Waeber, Peter I Frazier, and Shane G Henderson. Bisection search with noisy responses. *SIAM Journal on Control and Optimization*, 51(3):2261–2279, 2013.

[66] Robert Nowak. Generalized binary search. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 568–574. IEEE, 2008.

[67] Ole-Christofier Granmo and B John Oommen. On allocating limited sampling resources using a learning automata-based solution to the fractional knapsack problem. In *Intelligent Information Processing and Web Mining*, pages 263–272. Springer, 2006.

[68] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216. IEEE, 2013.

[69] Sandeep Pandey, Krithi Ramamritham, and Soumen Chakrabarti. Monitoring the dynamic web to respond to continuous queries. In *Proceedings of the 12th International Conference on World Wide Web*, pages 659–668. ACM, 2003.

[70] Dazhi Chen and Pramod K Varshney. Qos support in wireless sensor networks: A survey. In *International Conference on Wireless Networks*, volume 233, pages 1–7, 2004.

[71] Shenghong Li, Hao Ge, Ying-Chang Liang, Feng Zhao, and Jianhua Li. Estimator goore game based quality of service control with incomplete information for wireless sensor networks. *Signal Processing*, 126:77–86, 2016.

[72] Ranjit Iyer and Leonard Kleinrock. Qos control for sensor networks. In *IEEE International Conference on Communications, 2003.*, volume 1, pages 517–521. IEEE, 2003.

[73] Y Uny Cao, Alex S Fukunaga, and Andrew Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–27, 1997.

[74] Dragos Calitoiu. New search algorithm for randomly located objects: A non-cooperative agent based approach. In *Computational Intelligence for Security and Defense Applications.*, pages 1–6. IEEE, 2009.

[75] René de Koster, Tho Le-Duc, and Kees Jan Roodbergen. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2):481 – 501, 2007.

[76] James A Tompkins. *Facilities Planning*. Wiley, 2010.

[77] Rui Xu and Donald C Wunsch. Survey of clustering algorithms. 2005.

[78] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping Multidimensional Data*, pages 25–71. Springer, 2006.

[79] Michael Hammer and Arvola Chan. Index selection in a aelf-adaptive data base management system. In *Proceedings of the 1976 International Conference on Management of data*, pages 1–8. ACM, 1976.

[80] CT Yu, MK Siu, K Lam, and F Tai. Adaptive clustering schemes: General framework. In *Proceedings of the IEEE COMPSAC Conference*, pages 81–89, 1981.

[81] Daniel Chiu Yu Ma. *Object Partitioning by using Learning Automata*. PhD thesis, Carleton University, 1986.

[82] Constantinos Daskalakis, Richard M Karp, Elchanan Mossel, Samantha J Riesenfeld, and Elad Verbin. Sorting and selection in posets. *SIAM Journal on Computing*, 40(3):597–622, 2011.

[83] Konstantin Andreev and Harald Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39(6):929–939, 2006.

[84] Rainer E Burkard, Eranda Cela, Panos M Pardalos, and Leonidas S Pitsoulis. The quadratic assignment problem. In *Handbook of Combinatorial Optimization*, pages 1713–1809. Springer, 1998.

[85] Upavan Gupta and Nagarajan Ranganathan. A game theoretic approach for simultaneous compaction and equipartitioning of spatial data sets. *IEEE Transactions on Knowledge and Data Engineering*, 22(4):465–478, 2010.

[86] Philippe Galinier, Zied Boujbel, and Michael Coutinho Fernandes. An efficient memetic algorithm for the graph partitioning problem. *Annals of Operations Research*, 191(1):1–22, 2011.

[87] Jin Kim, Inwook Hwang, Yong-Hyuk Kim, and Byung-Ro Moon. Genetic approaches for graph partitioning: a survey. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 473–480. ACM, 2011.

[88] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems*, pages 873–879, 2001.

[89] Stella X Yu and Jianbo Shi. Multiclass spectral clustering. In *Proceedings of the Ninth IEEE International Conference on Computer Vision-Volume 2*, page 313. IEEE Computer Society, 2003.

[90] Ali Safari Mamaghani and Mohammad Reza Meybodi. Clustering of software systems using new hybrid algorithms. In *Computer and Information Technology, 2009. CIT'09. Ninth IEEE International Conference on*, volume 1, pages 20–25. IEEE, 2009.

[91] B John Oommen, Radhakrishna S Valiveti, and Jack R Zgierski. An adaptive learning solution to the keyboard optimization problem. *IEEE transactions on systems, man, and cybernetics*, 21(6):1608–1618, 1991.

[92] Bart Selman, Henry A Kautz, and Bram Cohen. Noise strategies for improving local search. In *AAAI*, volume 94, pages 337–343, 1994.

[93] Tongtong Tao, Hao Ge, Guixian Cai, and Shenghong Li. Adaptive step searching for solving stochastic point location problem. In *Intelligent Computing Theories*, pages 192–198. Springer, 2013.

[94] J. Zhang, L. Zhang, and M. Zhou. Solving stationary and stochastic point location problem with optimal computing budget allocation. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 145–150, Oct 2015.

[95] B John Oommen, Govindachari Raghunath, and Benjamin Kuipers. On how to learn from a stochastic teacher or a stochastic compulsive liar of unknown identity. In *AI 2003: Advances in Artificial Intelligence*, pages 24–40. Springer, 2003.

[96] Aarti Singh, Robert Nowak, and Parmesh Ramanathan. Active learning for adaptive mobile sensing networks. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 60–68. ACM, 2006.

[97] Rui M Castro and Robert D Nowak. Upper and lower error bounds for active learning. In *The 44th Annual Allerton Conference on Communication, Control and Computing*, volume 2, page 1, 2006.

[98] Marat Valievich Burnashev and Kamil'Shamil'evich Zigangirov. An interval estimation problem for controlled observations. *Problemy Peredachi Informatsii*, 10(3):51–61, 1974.

[99] Robert D Nowak. The geometry of generalized binary search. *IEEE Transactions on Information Theory*, 57(12):7893–7906, 2011.

[100] Robert Nowak. Generalized binary search. In *Annual Allerton Conference on Communication, Control and Computing*, pages 568–574. IEEE, 2008.

[101] Laurent Hyafil and Ronald L Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[102] Peter I Frazier, Shane G Henderson, and Rolf Waeber. Probabilistic bisection converges almost as quickly as stochastic approximation. *Mathematics of Operations Research*, 2019.

[103] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[104] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

[105] Robert Burbidge, Jem J Rowland, and Ross D King. Active learning for regression based on query by committee. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 209–218. Springer, 2007.

[106] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[107] John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016.

[108] Theano Development Team. Theano: A Python Framework for Fast Computation of Mathematical Expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[109] Dustin Tran, Alp Kucukelbir, Adji B. Dieng, Maja Rudolph, Dawen Liang, and David M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

[110] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[111] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational Learning Theory*, pages 287–294. ACM, 1992.

[112] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168, 1997.

[113] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

[114] Naoki Abe Hiroshi Mamitsuka et al. Query learning strategies using boosting and bagging. In *ICML 98)*, volume 1, 1998.

[115] Djallel Bouneffouf, Romain Laroche, Tanguy Urvoy, Raphael Féraud, and Robin Allesiardo. Contextual bandit for active learning: Active thompson sampling. In

International Conference on Neural Information Processing, pages 405–412. Springer, 2014.

[116] Ravi Ganti and Alexander G Gray. Building bridges: Viewing active learning from the multi-armed bandit lens. *arXiv preprint arXiv:1309.6830*, 2013.

[117] Thomas Osugi, Deng Kim, and Stephen Scott. Balancing exploration and exploitation: A new algorithm for active machine learning. In *Fifth IEEE International Conference on Data Mining*, pages 8–pp. IEEE, 2005.

[118] Robert F Love and James G Morris. Modelling inter-city road distances by mathematical functions. *Journal of the Operational Research Society*, 23(1):61–71, 1972.

[119] Wolfgang Berens. The suitability of the weighted lp-norm in estimating actual road distances. *European Journal of Operational Research*, 34(1):39–43, 1988.

[120] Jack Brimberg, Robert F Love, and John H Walker. The effect of axis rotation on distance estimation. *European Journal of Operational Research*, 80(2):357–364, 1995.

[121] Ethem Alpaydin, I Kuban Altinel, and Necati Aras. Parametric distance functions vs. nonparametric neural networks for estimating road travel distances. *European Journal of Operational Research*, 93(2):230–243, 1996.

[122] Yushan Sun, Junhan Cheng, Guocheng Zhang, and Hao Xu. Mapless motion planning system for an autonomous underwater vehicle using policy gradient-based deep reinforcement learning. *Journal of Intelligent & Robotic Systems*, pages 1–11, 2019.

[123] Jack Brimberg and John H. Walker. *Estimation of Travel Distance*, pages 1–17. American Cancer Society, 2010.

[124] Jessica Havelock, B John Oommen, and Ole-Christoffer Granmo. Novel distance estimation methods using "stochastic learning on the line" strategies. *IEEE Access*, 2018.

[125] James Kennedy. Particle swarm optimization. *Encyclopedia of machine learning*, pages 760–766, 2010.

[126] John H Holland. Genetic algorithms. *Scientific American*, 267(1):66–73, 1992.

[127] Marco Dorigo, Vittorio Maniezzo, Alberto Colorni, et al. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics*, 26(1):29–41, 1996.

Appendix A

A Bayesian Network Based Solution Scheme for the Constrained Stochastic On-line Equi-Partitioning Problem

A Bayesian Network Based Solution Scheme for the Constrained Stochastic On-line Equi-Partitioning Problem*

Sondre Glimsdal† and Ole-Christoffer Granmo‡

Abstract

A number of intriguing decision scenarios revolve around partitioning a collection of objects to optimize some application specific objective function. This problem is generally referred to as the Object Partitioning Problem (OPP) and is known to be NP-hard. We here consider a particularly challenging version of OPP, namely, the Stochastic On-line Equi-Partitioning Problem (SO-EPP). In SO-EPP, the target partitioning is unknown and has to be inferred purely from observing an on-line sequence of object pairs. The paired objects belong to the same partition with probability p and to different partitions with probability $1-p$, with p also being unknown. As an additional complication, the partitions are required to be of equal cardinality. Previously, only heuristic sub-optimal solution strategies have been proposed for SO- EPP. In this paper, we propose the first *Bayesian* solution strategy. In brief, the scheme that we propose, BN-EPP, is founded on a Bayesian network representation of SO-EPP problems. Based on probabilistic reasoning, we are not only able to infer the underlying object partitioning with superior accuracy. We are also able to simultaneously infer p, allowing us to accelerate learning as object pairs arrive. Furthermore, our scheme is the first to support a wide range of constraints on the partitioning (Constrained SO-EPP). Being Bayesian, BN-EPP provides superior performance compared to existing solution schemes. We additionally introduce Walk-BN-EPP, a novel WalkSAT inspired algorithm for solving large scale BN-EPP problems. Finally, we provide a BN-EPP based solution to the problem of order picking, a representative real-life application of BN-EPP.

1 Introduction

A number of intriguing decision scenarios revolve around grouping a collection of objects into partitions in such a manner that some application specific objective function is optimized. This type of grouping is referred to as the Object Partitioning Problem (OPP) and is in its general form known to be NP-hard.

*A preliminary version of parts of this paper was presented at ICMLA 2014 - the 13th International Conference on Machine Learning and Applications, Detroit, USA, December 2014.

†This author can be contacted at: Centre for Artificial Intelligence Research (CAIR), University of Agder, Postbox 422, 4604 Kristiansand, Norway. E-mail: sondre.glimsdal@uia.no.

‡Author's status: *Professor*. This author can be contacted at: Centre for Artificial Intelligence Research (CAIR), University of Agder, Postbox 422, 4604 Kristiansand, Norway. E-mail: ole.granmo@uia.no.

In this paper, we consider a particularly challenging variant of OPPs — the Constrained Stochastic Online Equi-Partitioning Problem (CSO-EPP). In CSO-EPP, objects arrive sequentially, in pairs[1]. Furthermore, the relationship between the arriving objects is stochastic: Paired objects belong to the same partition with probability p, and to different ones with probability $1-p$. As an additional complication, the partitioning is constrained, with the default constraint being that the partitions must be of equal cardinality, referred to as equi-partitioning. Unlike previous work, we relax this constraint and only require the size of each partition to be known beforehand. Under these challenging conditions, the overarching goal is to infer the underlying partitioning, that is, to predict which objects will appear together in future arrivals, from a history of object arrivals.

The CSO-EPP can be applied to solve a number of challenging tasks. We will here study a particularly fascinating one, *order picking*, which highlights the full spectrum of nuisances captured by CSO-EPP. Order picking is defined as *"the process of retrieving products from storage (or buffer areas) in response to a specific customer request"* [1]. Order picking occurs both in warehouses employing an Automated Storage/Retrieval System (AS/RS) and those depending on manual labor. Tompkins et al. identified travel time as the main factor when it comes to optimizing order-picking [2]. For this reason, to facilitate efficient retrieval of products, frequently ordered products should be placed in easy to reach locations. Additionally, products that are often ordered together should be placed in near-proximity of each other. By doing so, we can systematically reduce the total travel time needed to collect orders.

In more challenging order-picking scenarios, the governing product relationships may be unknown initially, and thus have to be learned over time by monitoring which products are ordered together. Additionally, non-related products may sporadically be ordered in conjunction, leading to *stochastic* order composition. This means that successful solution strategies must be able to operate in a stochastic environment. Furthermore, many order picking scenarios impose constraints when it comes to product placement. One could for instance require that a subset of the objects is located in a subset of the available locations, e.g., that all frozen objects should be in freezers, even when they are rarely purchased together. Other constraints could be that all products from a brand must be co-located on the request of the manufacturer, or that fragile objects must be placed in shelves close to the floor. To further exemplify the importance of dealing with constraints, several more are listed in Table 1[2]. Noting that each section of a warehouse can be represented as a CSO-EPP partition, and that products can be represented as CSO-EPP objects, we propose CSO-EPP as a model for order picking.

In this paper, we present the first *Bayesian* solution scheme for SO-EPP and CSO-EPP. Let $\mathcal{O} = \{O_1, O_2, \ldots, O_w\}$ be a set of W objects. These are to be partitioned into R different partitions $\mathcal{P} = \{P_1, P_2, \ldots, P_R\}$. The aim is to find some unknown underlying partitioning of the objects based on noisy observations. Succinctly, the problem can be described as a 2-tuple $(\mathcal{U}, p)$, where $\mathcal{U}$ is a set of tuples (O_i, O_j). If $(O_k, O_m) \in \mathcal{U}$

[1]Note that the arrival of objects in *pairs* can easily be generalized to arrival of objects in *sets*, which in turn are transformed into pairwise combinations of the objects contained in each set. See Section 5.4 for an example of this.

[2]These are based on real-world point-of-sale transaction data from a grocery outlet [3].

Table 1: Example constraints governing the placement of products in a warehouse.

Number	Products	Constraint
1	shopping bags	Must either be in the entrance- or counter section
2	whole milk, rolls/buns, tropical fruit	Cannot be in the same section
3	white wine, specialty chocolate	Must be in the same section
4	yogurt	Has to be in the cooler section
5	tropical fruit	Cannot be in the cooler section

then object k and m belong to the same underlying partition, otherwise, they belong to different ones. Constraints can then naturally be formulated in terms of: (1) the cardinality of each partition; (2) what objects must be, or must not be, in the same partition; and (3) which subset of objects must be in which subset of partitions. The two latter types of constraints can be expressed by formulating restrictions on object pairs in $\mathcal{U}$, while the first type of constraint can be specified as a cardinality vector of size R. Finally, p is the probability of a *convergent request* [4], i.e., the probability that a request (i.e., an observation) encompasses two objects from the same underlying partition. A request where the objects originate from different underlying partitions is called a *divergent request* [4], which occurs with probability $1 - p$.

Under the above model, an observation can be simulated by sampling from a Bernoulli distribution. With probability p, select a pair of objects randomly from U: $(O_i, O_j) \in \mathcal{U}, i \neq j$ (a *convergent* request). And with probability $1 - p$, randomly select a pair of objects not in U: $(O_k, O_m) \notin \mathcal{U}, k \neq m$ (a *divergent* request). This definition is equivalent to the definition given by Oommen et al. [5].

For completeness, we mention that solutions to SO-EPP are invariant to permutation of the partitions, as long as the objects grouped together inside each partition remain unchanged.

Previously, only heuristic sub-optimal solution strategies have been proposed for SO-EPP, and no solution exists for CSO-EPP. In this paper, for both of these problems, we propose the first *Bayesian* solution strategy. The solution strategy is based on a novel Bayesian network representation of CSO-EPP problems. To enable swifter computations with BN-EPP, we additionally introduce Walk-BN-EPP, an approximate reasoning approach that takes advantage of the unique structure of BN-EPP. The paper contribution can be summarized as follows:

1. We propose a novel Bayesian network model of the CSO-EPP problem (BN-EPP) that fully captures the nuances of CSO-EPP.

2. We provide a BN-EPP based algorithm for on-line object partitioning that outperforms the existing *state-of-the-art* SO-EPP solution schemes.

3. The BN-EPP scheme is highly flexible in the sense that we can encode a wide range of partitioning constraints, leveraging the representation capacity of Bayesian networks.

4. BN-EPP is parameter-free, which means that performance is maximized without any fine tuning of parameters.

5. In addition to predicting the correct partitioning of objects, BN-EPP also estimates the noise parameter p on-line.

6. We demonstrate that Walk-BN-EPP exhibits state-of-the-art performance on a large-scale real-world warehouse order picking problem.

7. We define a novel scheme that allows us to apply Spectral Clustering (SC) to the SO-EPP problem, demonstrating performance close to BN-EPPs state-of-the-art performance on artificial data.

The paper is organized as follows. In Sect. 2 we present related work. We then provide a brief overview of Bayesian networks in Sect. 3, before we proceed with providing the details of our BN-EPP scheme in Sect. 4. Then, in Sect. 5, we present our empirical results and demonstrate the superiority of BN-EPP when compared to existing state-of-the-art schemes. We conclude in Sect. 6 and provide pointers for further work.

2 Related Work

, The OPP is already a thoroughly studied problem [6, 7]. Yet, research on its fascinating variant, SO-EPP [8, 9, 10, 5, 11], is surprisingly sparse despite its many real-world applications, which includes software clustering [12] and keyboard layout optimization [13]. To cast further light on the unique properties of SO-EPP, we will here relate it to two similar problems, namely, the *Poset Ordering Problem* (POP) and the *Graph Partitioning Problem* (GPP).

The Poset Ordering Problem (POP). A *poset* is defined as a set of elements with a transitive partial order, where some elements may be incomparable [14]. A binary relation that is reflexive, antisymmetric, and transitive defines this ordering, referred to as a *less-than-or-equal* relation ($\leq$). The standard *less-than-or-equal* relation for integers forms for instance a partial ordering on the set of integers. In the poset ordering problem, the goal is to establish the partial ordering of a poset by comparing pairs of elements, typically using the *less-than-or-equal* relation as few times as possible. Accordingly, both in SO-EPP and POP, one must learn from paired elements to uncover an underlying more complex structure. That is, in POP, the *less-than-or-equal* relation is applied iteratively on pairs of elements, while in SO-EPP a *in-the-same-partition* relation is used instead. Whereas the *less-than-or-equal* relation found in POP is both reflexive and transitive, it is not symmetric, i.e., $A \leq B$ does not imply $B \leq A$. The *in-the-same-partition* relation, however, is symmetric. This means that the solution of SO-EPP is not a partial ordering, but a set of *equivalence classes*, leading to unique solution schemes.

The Graph Partitioning Problem (GPP) and Spectral Clustering (SC). The GPP is in its most general form an NP-complete problem [15]: Let $G = (V, E)$ be a graph with a set of vertices V and a set of weighted edges E. In graph equipartitioning, the goal is to partition V into k subsets $V_1, V_2, \ldots, V_k$ of equal cardinality. In all brevity, the solution to a GPP instance is the partitioning that minimizes the sum of those edge weights that cross different vertex sets, $(V_i, V_j), i \neq j$ [16]. The SO-EPP can thus be cast as a GPP if

the frequencies of object co-occurrence are known for all object pairs. Then we could form a complete graph, $G = (V, E)$, where each vertex in V represents an object. Further, the weight of an edge between a pair of objects is simply the frequency with which we observe that particular pair. The resulting GPP can then be solved by any GPP solver [17, 18, 19].

Another approach to solving GPPs is based on a Markov Random Walk. By defining a Markov Random Walk over G, one can perform clustering based on the eigenvalues of the resulting transition matrix [20]. This method is based on the usage of query statistics [11], e.g. to generate the transition matrix from the different frequency counts. The baseline for this approach is Spectral Clustering (SC) where the eigenvalues is used as a low-dimensional embedding of the problem space. SC can then effectively generate clusters using the MultiClass Normalized Cuts scheme [21]. In Section 5.2 we define a simple, yet effective scheme that allow us to apply SC to SO-EPP.

A main drawback of SC and other GPP solvers is that they do not support the type of real-world probabilistic constraints mentioned in the introduction (CSO-EPP), and as we shall see, do not either fully utilize of the problem specific characteristics of SO-EPP.

State-of-the-art solution schemes for SO-EPP. We now turn our attention to algorithms that are specifically designed to solve SO-EPP. The state-of-art solution scheme for SO-EPP is the Pursuit Object Migration Automaton (POMA), introduced by Shirvani et al. in 2017 [8]. POMA is based on the Object Migration Automaton (OMA) [5, 4]. The basic OMA is a statistics free scheme, meaning that it does not try to estimate object co-occurrence frequencies. Instead, each object navigates a finite state machine according to a few simple fixed rules, allowing the objects to *migrate* between the different partitions, gradually converging to a solution. POMA, on the other hand, leverages co-occurrence frequency estimates, through the following two phases:

1. An *estimation phase* adopts the previous state-of-art Object Migration Automaton (OMA) [5, 4] to generate an initial solution, while simultaneously estimating the pairwise-object frequencies.

2. A *fine-tuning phase* refines the initial solution by making use of the pursuit paradigm [22, 23] to filter diverging or noisy queries. Thus, the POMA is able to determine whether a pairwise query facilitates convergence.Theoretically, this would allow the underlying OMA to operate in a noise free environment, and, consequently, converge quickly.

However, POMA is still a heuristic rule based approach. While efficient, it is not optimal, which leads us to design the BN-EPP algorithm presented in this paper. BN-EPP is a probabilistic parameter-free algorithm that, as we shall see, is not only more flexible in terms of the requirements placed on the solution, but also able to infer the level of noise present in the environment.

3 A Bayesian Network Based Solution Scheme for the Constrained Stochastic On-line Equi-Partitioning Problem

In this section, we present our novel BN-EPP scheme — a generative modeling approach for solving CSO-EPP based on Bayesian networks (BNs). By taking advantage of the ability of BNs to construct interpretable models that encode probability distributions over complex domains [24], we capture the unique characteristics of CSO-EPP. We further propose an efficient reasoning algorithm for BN-EPP that allows uncertainty to be represented and managed explicitly.

A BN consists of a directed acyclic graph (DAG) representing the conditional dependencies between a set of random variables. When modeling causal relationships, an edge between the nodes A and B signifies that A *"causes"* B. Consider the BN shown in Figure 1. In this simple BN, we have three discrete random variables: *Weather*, *Sprinkler* and *Lawn*. Let us assume that the weather can have one of three different states: *Sunny*, *Cloudy*, or *Rainy*. Further, the lawn is either *Wet* or *Dry*, and the sprinkler can be *On* or *Off*. Adding directed edges, we can encode knowledge about cause and effect, such as the fact that rainy weather causes the lawn to be wet. Similarly, a long period of sunny weather triggers a need for turning the sprinkler on, hence weather indirectly causes the lawn to be wet though the sprinkler system.

The above qualitative description of cause and effect is further enriched with a quantitative description. The quantitative description takes the form of a probability distribution assigned to each node, conditioned on the state of the parents of the respective node. The purpose of the conditional probability distributions is to quantitatively describe the probabilistic independence relationships captured by the DAG. We assign these probabilities though Conditional Probability Tables (CPTs), one for each node in the graph. Note that a node without parents is assigned an unconditional probability distribution. A CPT for the sprinkler can be seen in Table 2, where the effect weather has on the state of the sprinkler is captured. The CPT here tells us, e.g., that the sprinkler turns on with probability 0.8 in sunny weather.

From the BN CPTs, we can conduct diagnostic and predictive reasoning, simply by asking questions about the state of the random variables. One could for instance ask: "if the lawn is wet, what are the chances that it was caused by rain or by the sprinkler?" or "if the sprinkler is on, does that indicate that there is sun outside?".

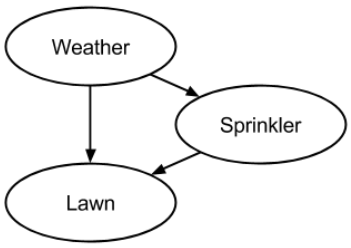

Figure 1: Simple BN.

Weather – State	Sunny	Cloudy	Rainy
P(Sprinkler = on|Weather)	0.8	0.15	0.05
P(Sprinkler = off|Weather)	0.2	0.85	0.95

Table 2: CPT of Sprinkler

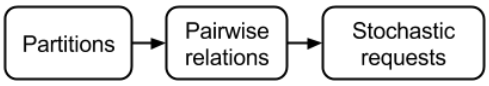

Figure 2: Overview of BN-EPP.

The generative model that we propose, BN-EPP, can be described in terms of three interacting BN fragments, as shown in Figure 2. Firstly, a dedicated BN fragment, referred to as "Partitions" in the figure, captures the actual placement of objects into partitions. This includes any constraints on the partitioning, such as equi-partitioning. Since the objects arrive in pairs, we need to further generate an intermediate BN fragment — the "Pairwise relations" fragment — that explicitly extracts all pairwise object relations from the "Partitions" fragment. Finally, an observation model is derived from "Pairwise relations", capturing generation of convergent and divergent request. This latter fragment, "Stochastic requests", is based on the noise parameter p and the "Pairwise relations" fragment.

Based on the BN-EPP, our on-line solution strategy for CSO-EPP can be summarized as follows. In operation, arriving object pairs (requests) are entered into the "Stochastic requests" part of the BN-EPP as observations (evidence). From these observations, pairwise relations are inferred in the intermediate fragment, which, finally, leads to a probability distribution over allowed partitions of objects in the "Partitions" fragment. Every object pair observed provides new information, and gradually, with successive observations, the probability distribution over object partitions converges to a single partitioning that solves the underlying CSO-EPP. In the case of multiple equally probable solutions, BN-EPP will arbitrarily select a single solution.

The detailed construction of BN-EPP is outlined in Algorithm 1. The BN-EPP needs to:

Requirement 1 Handle constraints, such as only considering partitions of equal cardinality.

Requirement 2 Infer whether two objects belong to the same partition.

Requirement 3 Correctly handle both converging and diverging requests.

Requirement 4 Encode the actual object partitioning.

We will now explain how the BN-EPP algorithm fulfills the above requirement. First of all, recall that $\mathcal{O} = \{O_1, O_2, \ldots, O_w\}$ is a set of W objects. These are to be partitioned into R different partitions $\mathcal{P} = \{P_1, P_2, \ldots, P_R\}$. The aim is to find some unknown underlying partitioning of the objects based on noisy observations of object pairs (convergent and divergent requests).

(Requirement 1) Only consider partitions that fulfill governing constraints (Lines 1-5)
The first part of the algorithm builds the "Partitions fragment" from Figure 2. Briefly stated, we represent

Algorithm 1: Constructing BN-EPP

Data: Objects $\mathcal{O} = \{O_1, O_2, \ldots, O_w\}$; Partitions $\mathcal{P} = \{P_1, P_2, \ldots, P_R\}$; and Noise resolution N
Result: A BN-EPP model β: Noise p_β; Partitions $\mathcal{O}_\beta$; Pairwise relations $\mathcal{A}_\beta$; Stochastic requests $\mathcal{X}_\beta$

`/* Create partitions fragment` $\mathcal{O}_\beta$`. */`
$\mathcal{O}_\beta := \emptyset$
for $i := 1$ *to* W **do**
  `/*` O_{β_i} `assigned to a partition in` $\mathcal{P}$`, given preceding assignments` $\mathcal{O}_\beta$ `*/`
  `/* and the CPT of object` i`:` $F_{O_{\beta i}}$ `*/`
  $O_{\beta_i} :=$ Node(States=$[P_1, P_2, \ldots, P_R]$, Parents=$\mathcal{O}_\beta$, Distr=$F_{O_{\beta i}}$)
  $\mathcal{O}_\beta := \mathcal{O}_\beta \cup O_{\beta_i}$
end
`/* Create pairwise relations fragment. */`
$\mathcal{A}_\beta := \emptyset$
for $i, j \in [W \times W]$ *s.t.* $i < j$ **do**
  `/*` $A_{\beta_{ij}}$ `is true if and only if` O_{β_i} `and` O_{β_j} `are in the same partition. */`
  $A_{\beta_{ij}} :=$ Node(States=[True, False], Parents=$\{O_{\beta_i}, O_{\beta_j}\}$, Distr=$F_{A_\beta}$)
  $\mathcal{A}_\beta := \mathcal{A}_\beta \cup \{A_{\beta_{ij}}\}$
end
`/* Create stochastic requests fragment. */`
$p_\beta :=$ Node(States=$[\frac{0}{N}, \frac{1}{N}, \ldots, \frac{N}{N}]$, Parents=$\emptyset$, Distr=$\mathrm{F}_{p_\beta}$) `// Noise probability.`
$\mathcal{X}_\beta := \emptyset$
for $i, j \in [W \times W]$ *s.t.* $i < j$ **do**
  $X_{\beta_{ij}} :=$ Node(States=$[\mathbb{N}_0]$, Parents=$\{A_{\beta_{ij}}, p_\beta\}$, Distr=$F_{X_\beta}$) `// Pair observation count.`
  $\mathcal{X}_\beta := \mathcal{X}_\beta \cup \{X_{\beta_{ij}}\}$
end

each EPP object, $O_i \in \mathcal{O}$, using a corresponding BN node, $O_{\beta i}$. Each BN node, $O_{\beta i} \in \mathcal{O}_\beta$, has one state per partition, $P_i \in \{P_1, \ldots, P_R\}$, representing the partition assigned to O_i. For instance, if we have two partitions then there will be two states per object, one for partition P_1 and one for partition P_2.

We now model the governing constraints, including equal cardinality of partitions, by means of the BN DAG. Because of the reciprocal relationships among objects (objects are either in the same partition or not), we can order the BN object nodes arbitrarily. Without loss of generality, assume that A is the first BN node in the ordering. This means that A can be freely placed in any partition (the placement does not depend on the placement of any other object, because none of the other objects have been placed yet). Then the next object in the ordering, object B, only needs to take into account object A's choice of partition. Likewise object C, the third object, is only restricted by the previous objects' choice of partition (the choices of object A and B). Continuing in this manner, we can always represent the partition of the next object as solely being dependent on the already partitioned objects. It is for this purpose we maintain the gradually increasing object set $\mathcal{O}_\beta$, containing all the already partitioned predecessor objects. This organization of objects is thus leading to a BN DAG structure, as exemplified in Figure 3, capturing two partitions and four objects.

The corresponding CPTs for the EPP objects ($F_{O_{\beta i}}$ in the algorithm) are generated as a function of the constraints set by CSO-EPP (the constraints governing the partitioning, e.g., equi-partitioning). As an example, the CPT of object C (the third object) can be seen in Table 3. From the table we observe for

A - State	P_1		P_2	
B - State	P_1	P_2	P_1	P_2
C in P_1	0.0	0.5	0.5	1.0
C in P_2	1.0	0.5	0.5	0.0

Table 3: CPT of object C

instance that $P(C = P_1|A = P_1, B = P_1) = 0.0$, that is, if object A and B is in P_1 then the probability of C being in P_1 is zero. On the other hand, if object A and B is located in different partitions then object C is equally likely to be in partition P_1 as in partition P_2. Thus, by constructing the CPT of each node (representing an object) in this manner, a solution that fulfills all of the constraints is always ensured because a partitioning that violates constraints is assigned a probability of zero.

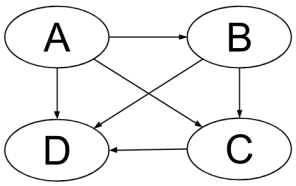

Figure 3: Object dependencies for 4 objects with 2 partitions.

(Requirement 2) Infer pairwise relations between the objects (Lines 6-10)
Now that the "Partitions fragment" has determined the partition of each object, it is a simple task to determine whether an object pair belongs to the same partition. In the "Pairwise relations" fragment, we represent every pair of objects as a deterministic node with two states: *True* if the pair is in the same partition, and *False* when they are not (distribution F_{A_β} in the algorithm).

Figure 4 provides an example of a "Pairwise relations" fragment, obtained following the above procedure for four objects and two partitions. The corresponding CPT for the pair node for object A and object C (node AC) can be found in Table 4. From the truth table it is evident that if object A and C belong to the same partition, the state of node AC state is *True*, and *False* otherwise.

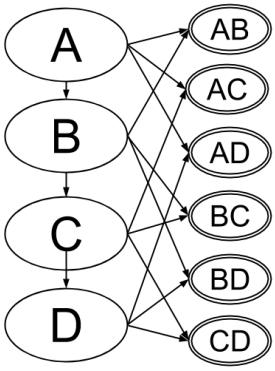

Figure 4: Pairwise object relations for 4 objects with 2 partitions.

A - State	P_1		P_2	
C - State	P_1	P_2	P_1	P_2
AC - State	True	False	False	True

Table 4: Truth-table for node AC

Table 5: The CPT of an observation node conditioned on the state of the parent pair $A_{\beta_{ij}}$ and the p_β node.

$P(X_{\beta ij} = n \mid A_{\beta_{ij}} = \text{True}, p_\beta = p)$	$B\left(n, p \cdot \frac{1}{P} \cdot \binom{\frac{W}{P}}{2} \frac{1}{\frac{W}{P}} \cdot \frac{1}{\frac{W}{P}-1}\right)$
$P(X_{\beta ij} = n \mid A_{\beta_{ij}} = \text{False}, p_\beta = p)$	$B\left(n, (1-p) \cdot \binom{P}{2} \frac{1}{P} \cdot \frac{1}{P-1} \cdot \frac{1}{\frac{W}{P}} \cdot \frac{1}{\frac{W}{P}}\right)$

The "Pairwise relations" fragment gives BN-EPP the capability to infer object relations from pairwise observations, such as in the following scenario: Given the above example, assume that we know that (1) Object A is known to be in partition P_1, and (2) Object B and object D should be in the same *unknown* partition, i.e., the BD-node is set to *True*. BN-EPP will then correctly infer the only possible partitioning, namely the two partitions $P_1 : \{A, C\}$ and $P_2 : \{B, D\}$. While similar result could have been obtained though the usage of a propositional logic solver, as we shall see, the stochastic nature of CSO-EPP rules out such a solution.

(Requirement 3) Stochastic requests (Lines 11-15)

The BN model obtained through the "Partitions"- and "Pairwise relations" fragments allows us to infer the correct object partitions, given that we know the state of a sufficient number of the pairwise relation nodes. However, the CSO-EPP involves both convergent and divergent requests. Consequently, we need a mechanism for handling noisy information.

Firstly, we introduce a BN node p_β representing p — the convergent request probability. The state space of p is a discretization of potential values for p, each with an equal prior probability. Attached to this p_β node is a series of *observation* nodes $X_{\beta_{ij}} \in \mathcal{X}_\beta$, each dependent on the state of the p_β node, and whether or not its corresponding pair node $A_{\beta ij}$ is *True* or *False*. The CPT for each observation node (F_{X_β} in the algorithm) is a function of the number times $n \in \mathbb{N}_0$ that particular pair has been observed, as well as the states of the p_β node, as shown in Table 5. As seen, F_{X_β} is distributed according to a Bernoulli distribution, $B(n, p)$.

(Requirement 4) Decode the object partitioning from the BN representation

While the BN correctly models the CSO-EPP, it does not directly present us with a solution in the form of a partition for each object. However, we obtain the partitioning indirectly by finding the Maximum a Posteriori (MAP[3]) configuration of the BN-EPP. In all brevity, a MAP query identifies the most probable solution given the observations [25, 24].

$$\text{solution(BN)} = \text{MAP}(\mathcal{O}_\beta \cup \mathcal{A}_\beta \cup p_\beta | \mathcal{X}_\beta) = \underset{\mathcal{O}_\beta \cup \mathcal{A}_\beta \cup p_\beta}{\arg\max}\ P(\mathcal{O}_\beta \cup \mathcal{A}_\beta \cup p_\beta | \mathcal{X}_\beta)$$

[3]Also known as Most Probable Explanation (MPE).

For an example of the outcome of the final step, see Figure 5. Note that the observation nodes for an object pair XY in the figure is denoted by $O(XY)$. The complete BN-EPP for four objects and two partitions is shown, ready for MAP inference. As can be seen, the resulting BN-EPP has a complex structure. In the next section we take advantage of this structure to propose an efficient and novel inference algorithm for large scale CSO-EPP problems.

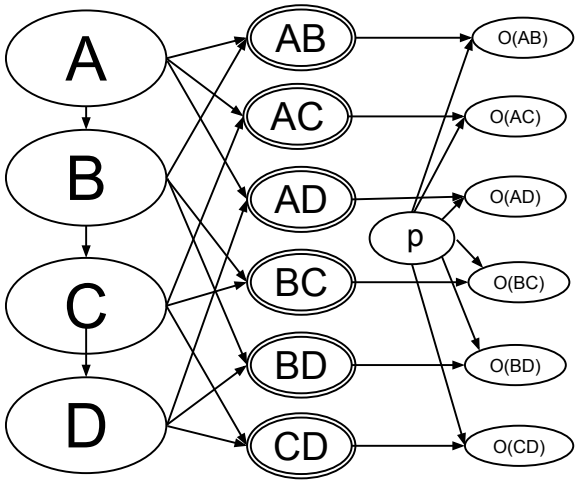

Figure 5: BN for solving an EPP with 4 objects, 2 partitions, and Binomially distributed observation nodes.

Note that the BN-EPP solution strategy is related to the Thompson Sampling (TS) principle that was introduced by Thompson in 1933 [26], and forms the basis for several of the leading solution schemes for so-called Multi-Armed Bandit (MAB) Problem. The classical MAB problem is a sequential resource allocation problem. At each time step, one pulls one out of multiple available bandit arms. Each arm pulled provides a reward with a certain probability, and the objective is to maximize the total number of rewards obtained through the sequence of arms pulled [27, 28]. In the Learning Automata (LA) literature this scheme is referred to as Bayesian Learning Automata (BLA) [28].

In TS, to quickly shift from exploring reward probabilities to reward maximization, one recursively estimates the reward probability of each arm using a Bayesian filter. To determine which arm to play, one obtains a reward probability sample from each arm, and the arm that provides the highest value is pulled. The selected arm triggers a reward, which in turn is used to perform a Bayesian update of the arm's reward probability estimate. As a result, TS selects arms with a frequency proportional to the posterior probability that the arm is optimal.[28]

TS has turned out to be among the top performers for traditional MAB problems [28, 29], supported by theoretical regret bounds [30, 31]. It has also been been successfully applied to contextual MAB problems [32], Gaussian Process optimization [33], Distributed Quality of Service Control in Wireless Networks [34], Cognitive Radio Optimization [35], as well as a foundation for solving the Maximum a Posteriori Estimation problem [36].

4 Walk-BN-EPP

The MAP problem is NP-complete [24], and thus cannot be solved efficiently for large networks in general. Accordingly, to allow solutions to be found for large CSO-EPPs, we will in this section introduce a novel inference scheme, Walk-BN-EPP. Walk-BN-EPP is designed to take advantage of the particular characteristics of the BN-EPP DAG structure and is based on WalkSAT [37], a well-known and effective solver for the NP-complete Boolean satisfiability (SAT) problem.

Note that our decision to design a dedicated algorithm for BN-EPP does not mean that existing general MAP solvers, such as Variable Elimination, Belief Propagation and the various evolutionary algorithms [38], cannot be used. On the contrary, they work quite well on small- and medium sized CSO-EPPs. However, since they do not take advantage of the BN-EPP's unique structure, they scale poorly. Thus, by introducing Walk-BN-EPP we expand the class of problems that can be solved with BN-EPP.

Walk-BN-EPP is based on WalkSAT [37], a successful algorithm for solving the NP-complete Boolean satisfiability (SAT) problem [39]. In all brevity, in SAT the goal is to find a truth value assignment for the variables of a Boolean expression that makes the overall expression evaluate to "True", thus *satisfying* the expression. The Boolean expression is a propositional logic formula that consists of a conjunction of Boolean clauses. The overall strategy of WalkSAT can be summarized as follows. One repeatedly selects one of the Boolean variables randomly, negate its value, and then observe whether the new truth value increases the total number of Boolean clauses satisfied. If the number of satisfied clauses does not increase, then with high probability one reverts the negated Boolean variable to its original state. Otherwise, the new state is kept. This simple iterative procedure is repeated until all the clauses are satisfied. Hence, one could say that WalkSAT performs a random walk with a drift towards "better" truth value assignments, that is, assignments with an increasing number of clauses satisfied.

Walk-BN-EPP is inspired by Walk-SAT in the sense that we divide Walk-BN-EPP into two steps: (1) Generate an initial configuration that partitions the objects by sampling from BN-EPP using forward sampling. (2) Improve the initial partitioning by applying a random walk with a drift towards more probable partitionings, that is, BN variable state configurations with higher MAP. The two steps are laid out in Algorithm 2, and we here explain them in more detail.

Initialization Step. In order to perform a Walk-SAT inspired random walk, we need an initial state configuration for the variables in BN-EPP. This initial configuration should ideally be as close as possible to the solution we seek, to reduce the length of the random walk. To achieve this, we sample an initial configuration from a rough estimate of the posterior probability distribution, one object O_{β_i} at a time, starting with O_{β_1}. That is, the state assigned to O_{β_i} is sampled from $P(O_{\beta_i}|\mathcal{X}_\beta, \{O_{\beta_k}\}_{k=1}^{i-1})$ using the traditional Likelihood-Weighted (LW) sampling algorithm [24]. Since the "Pairwise relations" fragment follows deterministically from the "Partitions" fragment, the states of the nodes $\mathcal{A}_\beta$ are then also given. When all of the object nodes, $O_{\beta_i} \in \mathcal{O}_\beta$, have been assigned a state in this manner, we use this configuration as an initial solution candidate for the random walk. The details of the initialization step are covered by

lines 1-5 in Algorithm 2.

Note that constraints forces the posterior probability of any violating assignment to zero, with the remaining probabilities renormalized. As an example, assume that we have 16 objects and 4 partitions. We have already placed 4 objects into partition number 3. To place the 5th object, use LW sampling and obtain $P(O_{\beta_5}) = \{0.1, 0.7, 0.2, 0.0\}$ from the BN. Note that the fourth probability becomes zero due to the previous assignment of objects to the corresponding partition, reflecting a full partition. To place the 5th object we then sample a partition from $P(O_{\beta_5})$. That is, we select partition 1 w.p. 0.1, partition 2 w.p. 0.7 and partition 3 w.p. 0.2. In this example, let us assume that we sampled partition 1. The 5th object is thus assigned to this partition. We repeat this process for each object, taking into account the choices of all previously assigned objects, until all objects have been assigned to a partition.

The Walk-SAT Based Search. In the second step of our algorithm (lines 6-22), we seek to iteratively improve the initial configuration from the initialization step. We do this by performing a Walk-SAT inspired random walk over the state space of candidate partitions. The random walk consists of iteratively swapping the partition of randomly selected pairs of objects, $(O_{\beta_i}, O_{\beta_j}) \in \mathcal{O}_\beta \times \mathcal{O}_\beta$, with the intent of gradually moving towards more probable object partitions, and ultimately, the most probable partitioning (i.e., the solution to the MAP problem). Let the set $\mathcal{O}_\beta^t = \{O_{\beta_1} = o_1, O_{\beta_2} = o_2, \ldots, O_{\beta_i} = o_i, \ldots, O_{\beta_j} = o_j, \ldots, O_{\beta_n} = o_n\}$ be the current configuration of the network before two randomly selected objects, O_{β_i} and O_{β_j}, swap partitions. Further, let $\mathcal{O}_\beta^{t+1} = \{O_{\beta_1} = o_1, O_{\beta_2} = o_2, \ldots, O_{\beta_i} = o_j, \ldots, O_{\beta_j} = o_i, \ldots, O_{\beta_n} = o_n\}$ be the configuration produced by the swap. Finally, let the log probability, C^q, of a configuration q be defined as follows:

$$C^q = \log P(\mathcal{O}_\beta^q) = \sum_{1 \leq k \leq N} \log P(O_{\beta_k} = o_k | \mathit{parents}(O_{\beta_k}))$$

with $\mathit{parents}(O_{\beta_k})$ being the parents of the node O_{β_k} in BN-EPP.

To systematically refine the current configuration, we always switch from configuration $\mathcal{O}_\beta^t$ to configuration $\mathcal{O}_\beta^{t+1}$ if the log probability C^t is greater than C^{t+1} (we accept the new configuration). If, on the other hand, the log probability decreases, we instead reject the new configuration, $\mathcal{O}_\beta^{t+1}$, with probability $1-\epsilon$. Otherwise, we accept the new configuration. Note that in the algorithm, $U(0,1)$ refers to a uniform distribution over the interval $[0, 1]$.

As an example assume that $C^4 = -15.3$, we then pick one objects from two different partitions, say object number 4 and 10 and swap their location. Calculating $C^5 = -14.9$ we observe that C^5 is greater than C^4, thus we accept the new state $\mathcal{O}_\beta^5$. For the next step, we select object 1 and 2 and swap their locations. However, calculating $C_6 = -20.2$ we see that the previous state, $\mathcal{O}_\beta^5$, has a larger log probability than the new configuration. Therefore we revert to the original configuration with probability $1 - \epsilon$, else, w.p. ϵ we keep the new, though inferior configuration. This process is then repeated for a predefined number of steps T and the best observed configuration is presented as the solution.

Algorithm 2: Walk-BN-EPP

Data: Bayesian network BN-EPP, ϵ - probability of accepting an inferior state, and T - the number of steps to execute.

Result: MAP configuration

for $i := 1$ *to* W **do**
  Estimate $\pi_i = P(O_{\beta_i}|\mathcal{X}_\beta, \{O_{\beta_k}\}_{k=1}^{i-1})$ using LW.
  Draw a single sample from π_i: $s \sim \pi_i$.
  Set the state of object i: $O_{\beta_i} = s$
end
$\mathcal{O}_\beta^0 = \{O_{\beta_1} = o_1, O_{\beta_2} = o_2, \ldots, O_{\beta_n} = o_n\}$
$C_0 = \text{CalculateLogProbability}(\mathcal{O}_\beta^0)$
$\mathcal{O}_\beta^{max} := \mathcal{O}_\beta^0$
$C^{max} := C^0$
for $t := 1$ *to* T **do**
  $O_{\beta_i}, O_{\beta_j} = \text{PickTwoRandomObjects}()$
  $\mathcal{O}_\beta^t := \text{SwapPartitionsOfObjects}(O_{\beta_i}, O_{\beta_j}, \mathcal{O}t - 1_\beta)$
  $C^t := \text{CalculateLogProbability}(\mathcal{O}_\beta^t)$
  |
  if $C^{max} < C^t$ **then**
    $\mathcal{O}_\beta^{max} := \mathcal{O}_\beta^t$
    $C^{max} := C^t$
  end
  if $C^t < C^{t-1}$ *and* $U_{(0,1)} < 1 - \epsilon$ **then**
    $\mathcal{O}_\beta^t := \mathcal{O}_\beta^{t-1}$
    $C^t := C^{t-1}$
  end
end
return $\mathcal{O}_\beta^{max}$

Table 6: Walk-BN-EPP results for different configurations on the r4w16 problem with 100 observations and p=0.75. Each data point is the average of a 1000 independent trials.

Walk Iterations	**50**	**100**	**500**	**1000**	**2000**	**4000**
Random Prior	0.005	0.02	0.039	0.05	0.057	0.064
TS with 50 samples	0.007	0.039	0.048	0.056	0.069	0.070
TS with 250 samples	0.061	0.066	0.063	0.085	0.11	0.10

5 Experimental Results on Walk-BN-EPP

To evaluate the on-line performance of BN-EPP and Walk-BN-EPP, we will here study convergence speed and accuracy empirically. The main question is how many observations, or queries, are required to obtain a correct partitioning of the objects, for various stochastic environments. Since the response to queries is stochastic, we will measure average performance over a large ensemble of independent trials.

We will explore two different kinds of stochastic environments. The first one is generated environments, where data is generated directly from an underlying SO-EPP problem. The data could for instance be generated from a SO-EPP with three partitions and nine objects (abbreviated r3w9) and a predefined level of noise. The second one is real-world environments, where an underlying perfect partitioning does not necessarily exist. Here data is separated into two parts, one part for training and one part for testing. The goal is then to solve the SO-EPP at hand using the training data, in such as manner that we maximize the performance on the unseen test data.

5.1 Impact of Walk-BN-EPP Parameter Settings

To evaluate the impact of the various parameters available in Walk-BN-EPP, we first solve the r4w16 (four partitions and 16 objects) problem using likelihood-weighted sampling with different number of random walk steps, as well as the number of samples used to estimate a maximum posterior initial configuration. Not surprisingly, as seen in Table 6, increasing the number of steps in the random walk significantly enhances the performance of Walk-BN-EPP. In addition, we observe that increasing the number of samples used to estimate an initial configuration increases performance further. Indeed, by applying our likelihood-weighted sampling algorithm by the modest number of 250 samples per object, we obtain an 1120% increase in probability of finding the configuration that provides the maximum posterior probability.

5.2 Applying Spectral Clustering (SC) to SO-EPP

The SC algorithm cannot be directly applied to SO-EPP. In order to compare our BN-EPP scheme with SC, we therefore here introduce a new variant of the SC algorithm. Vanilla SC takes a graph $G = (V, E)$ represented by a transition matrix T as input. By inspecting the eigen-vectors of T, SC then generates a predefined number, R, of clusters $C = \{C_1, C_2, \ldots, C_R\}$ [21]. To find T we simply row normalize the count matrix $M = \{m_{ij}\}$ where m_{ij} is the number of times object i and j have appeared together in a query.

However, the number of objects in each cluster is not constrained to be n as SO-EPP requires. So to balance C we find the subset of clusters $C_- \subset C$ that have an incorrect number of objects. Let μ_i be the euclidean mean of the objects of C_i as given by T. The fitness of a single object o_k in C_i is then defined as the cosine similarity between object o_k row in T and μ_i.

We then iteratively remove the least fitting object from all clusters that have a surplus of objects. Once all clusters have n or less objects, we greedily insert the removed objects, one-by-one, into the cluster where the object fits the most, and where there also is available space for the object.

This simple, yet effective, opens up for using SC to solve SO-EPP.

5.3 Empirical Comparison with Pursuit Object Migration Automaton (POMA) and Spectral Clustering

The Pursuit Object Migration Automaton (POMA) [8] represents state-of-the-art for solving EPP. We here compare our novel BN-EPP approach with POMA and other state-of-the-art approaches, focusing on:

- Accuracy of convergence, i.e., how many requests do we need to observe before we are able to correctly partition the objects.
- Probability of convergent requests (degree of noise).

For each experiment configuration, ten thousand individual trials were performed in order to minimize variance in our results. To avoid bias, we further independently selected a random optimal partitioning of the objects for every trial, and made sure that all algorithms were exposed to an identical sequence of incoming queries.

Unlike POMA, which requires a predetermined number of parameters (denoted N, τ and κ as in the original paper), BN-EPP is a parameter free scheme. Note that the Walk-BN-EPP scheme for doing inference on BN-EPP does require two parameters, namely, the number of steps for the random walk and the number of samples used to estimate an initial maximum posteriori distribution. We here report the results of POMA using the standard choice of 10 states ($N = 10$) [8, 5, 4] for all of the experiment configurations. For τ (noise tolerance) and κ (estimation phase length) we use the settings from the original paper [8]. For the warehouse experiment, a random search singles out the parameter values $\kappa = 9000$ and $\tau = 0.0003$ for high performance.

We have generated a diverse range of scenarios, and each scenario has been used to generate 1000 independent random trials to minimize variance. The results can be found in Table 7. Here, the notation rXwY refers to an EPP problem where X is the number of partitions and Y is the total number of objects. In the table, we observe that BN-EPP's accuracy on generated scenarios greatly exceed the state-of-the-art POMA as well as SC. The reasoning behind this is that POMA uses a very simple threshold scheme based on Maximum Likelihood to determine if a request is convergent or divergent. If this predefined threshold (τ) is wrong then POMA will either assume that all requests are convergent or that all are divergent. BN-EPP, on the other hand, directly quantifies the uncertainty associated with the requests by estimating p – the

probability of a convergent request. In Figure 6 we have plotted the probabilities BN-EPP assigned to the different p values from time step to time step. A major feature of BN-EPP is that it maintains a probability distribution spanning the whole object partitioning solution space, while POMA only works from a single configuration instance. We further believe that the ability of BN-EPP to track p explains why BN-EPP infers the correct partitioning significantly faster than POMA. From Table 7 it is clear that BN-EPP and SC the superior choices for solving generated SO-EPP scenarios where the data clearly forms a solution, with BN-EPP outperforming SC slightly. However, we shall see in Section 5.4 that SC does not exhibit this level of performance when faced with real-word data that does not conform as strictly to the problem definition. The reason for this is that SC implicitly imposes very strong bias onto how the data is formed, making it excellent when the problem data fits perfectly with the bias.

Table 7: The average number of objects that are wrongly placed for BN-EPP, POMA, and SC for different generated scenarios with $p = 0.6$. The results are average values, obtained from 1000 independent trials to minimize variance.

Scenario	T	BN-EPP	POMA	SC
r2w4	10	**0.30**	0.55	0.32
r3w6	50	**0.04**	0.87	**0.04**
r3w9	100	**0.05**	1.86	0.06
r6w12	200	**0.00**	1.48	0.02
r4w12	200	**0.01**	3.42	0.20
r2w12	200	**0.41**	1.89	1.29
r5w15	400	**0.00**	4.97	0.31
r3w15	400	**0.00**	4.85	0.08
r3w18	800	**0.00**	4.62	1.13
r6w18	800	**0.00**	6.50	0.16
r9w18	800	**0.00**	2.48	0.12

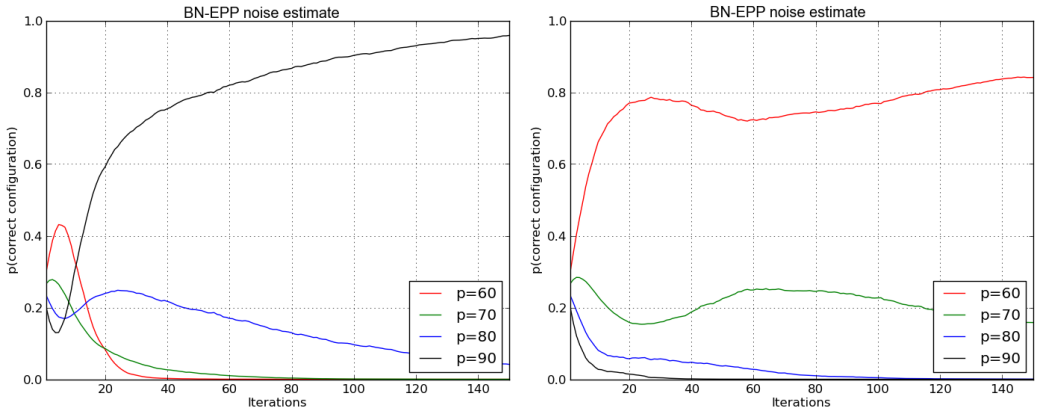

Figure 6: Probability of the different p values for $R = 3$ and $W = 9$. The left plot covers the scenario where the true probability of convergent requests p is 0.9, while the right plot shows the results for convergent request probability 0.6.

5.4 Empirical Results for Warehouse Optimization

To demonstrate the applicability of BN-EPP, we evaluate our scheme using one month (30 days) of real-world point-of-sale transaction data from a grocery outlet (collected in [3]). Each transaction T_k is a subset of the set of all unique articles O, where $|O| = 169$. In total there are 9835 transactions. Each article is labeled by its type of product, e.g. ice-cream instead of the actual brand. The number of articles per transaction vary wildly from orders of size 32 down to single article transactions. The mean number of objects per transaction is 4.4, with a standard deviation of 3.5.

Note that the above data-set does not provide a physical layout of the grocery outlet. Therefore, we here assume that the objects are to be partitioned among 13 different sections of the store in such a manner that the time each customer spend travelling between sections is minimized when collecting the articles on their shopping list (transaction T_k).

In addition, as discussed in the introduction, we introduce constraints on the placement of objects, listed in Table 1 transforming the problem from a SO-EPP problem to a CSO-EPP problem. However, as neither OMA/POMA nor SC does support CSO-EPP, we will include results for SO-EPP.

To measure solution effectiveness, we track how many warehouse sections, v, a consumer must visit to collect all the wares on his shopping list. We then assume that the experienced cost of travel doubles for each new unique section the consumer must visit. As an example, if a customer needs to visit 3 different sections the cost of that transaction becomes $2^3 = 8$.

We evaluate the effectiveness of BN-EPP using 5 fold cross validation, where we select 1 fold for training and 4 folds for testing. We report the mean cost of the transactions in the test set. The 5-fold cross validation is performed 1000 times to estimate expected effectiveness. For Walk-BN-EPP, we used the parameter settings of likelihood-weighted sampling with 100 samples per object and 1000 iterations for the walk phase. Table 8 demonstrate that Walk-BN-EPP significantly outperform the state-of-the-art by obtaining nearly half the loss compared to POMA and SC. Even when imposing the rules from Table 1, rendering the optimization problem significantly harder, we obtain comparable loss to the other schemes, with the other contenders (solving the easier SO-EPP) having no such rules imposed on their solution. One reason for the effectiveness of Walk-BN-EPP can be the Bayesian global perspective used to guide the search, which is in contrast to the heuristic local search employed by the competing approaches.

We would finally like to remark that our experiments show that POMA [8] seems to be highly dependent on the hyper-parameters. During our random search we often observed POMA obtaining a low loss on parts of the data with a particular set of hyper-parameters, only for the loss to be much higher than average for other part of the dataset with the same hyper-parameters.

Table 8: Effectiveness of Walk-BN-EPP, POMA and SC on the grocery dataset [3] as measured in number of sections traveled (5-fold cross validation). In physical terms, we can see that Walk-BN-EPP roughly halves the number of sections a customer has to visit on average to find all groceries on the shopping list.

	Walk-BN-EPP (CSO-EPP / SO-EPP)	POMA (SO-EPP)	OMA (SO-EPP)	SC (SO-EPP)
Mean	61.6 / **30.6**	56.0	68.1	61.6
Std.Dev	6.1 / **4.5**	7.0	6.5	14.4

6 Conclusion and Further Work

In this paper we have presented a novel approach to the Constrained Stochastic Online Equi-Partitioning Problem (CSO-EPP), namely, the Bayesian Network EPP model and inference scheme. We have demonstrated how the various components of BN-EPP interact and that BN-EPP significantly outperform existing state-of-art, not only in speed of convergence, but also in its ability to estimate the stochastic properties of the underlying environment. From a history of object arrivals, we are able to predict which objects will appear together in future arrivals. To enable BN-EPP to deal with larger data sets we introduced Walk-BN-EPP, a WalkSAT inspired solver for BN-EPPs. Walk-BN-EPP was then applied to a real-world warehouse problem and shown to significantly outperform state-of-the-art inference schemes, even when constraining the solution space in terms of real-world constraints.

We also introduced an adaption of Spectral Clustering (SC) for SO-EPP and showed that its performance on generated SO-EPP scenarios came close to BN-EPP, however it was clearly outperformed by BN-EPP on more complex real-world datasets [3].

In our future work, we intend to investigate how the BN-EPP approach can be expanded to cover other classes of stochastic optimization problems such as graph partitioning and poset ordering problems, potentially outperforming generic off-line techniques such as Particle Swarm Optimization (PSO) [40], Genetic Algorithm (GA) [41] or Ant Colony Optimization (ACO) [42].

References

[1] R. de Koster, T. Le-Duc, and K. J. Roodbergen, "Design and control of warehouse order picking: A literature review," *European Journal of Operational Research*, vol. 182, no. 2, pp. 481 – 501, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221706006473

[2] J. A. Tompkins, *Facilities planning.* Wiley, 2010.

[3] M. Hahsler, K. Hornik, and T. Reutterer, "Implications of probabilistic data modeling for mining association rules," in *From Data and Information Analysis to Knowledge Engineering.* Springer, 2006, pp. 598–605.

[4] W. Gale, S. Das, and C. T. Yu, "Improvements to an algorithm for equipartitioning," *Computers, IEEE Transactions on*, vol. 39, no. 5, pp. 706–710, 1990.

[5] B. Oommen and D. Ma, "Deterministic learning automata solutions to the equipartitioning problem," *Computers, IEEE Transactions on*, vol. 37, no. 1, pp. 2–13, Jan 1988.

[6] R. Xu, D. Wunsch *et al.*, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, 2005.

[7] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping multidimensional data*. Springer, 2006, pp. 25–71.

[8] A. Shirvani and B. J. Oommen, "On enhancing the object migration automaton using the pursuit paradigm," *Journal of Computational Science*, 2017.

[9] M. Hammer and A. Chan, "Index selection in a self-adaptive data base management system," in *Proceedings of the 1976 ACM SIGMOD international conference on Management of data*. ACM, 1976, pp. 1–8.

[10] C. T. Yu, M. K. Siu, K. Lam, and F. Tai, "Adaptive clustering schemes: general framework," in *Proc. IEEE COMPSAC Conf.* IEEE, 1981, pp. 81–89.

[11] D. Ciu and Y. Ma, "Object partitioning by using learning automata," Ph.D. dissertation, Carleton University, 1986.

[12] A. S. Mamaghani and M. R. Meybodi, "Clustering of software systems using new hybrid algorithms," in *Computer and Information Technology, 2009. CIT'09. Ninth IEEE International Conference on*, vol. 1. IEEE, 2009, pp. 20–25.

[13] B. J. Oommen, R. S. Valiveti, and J. R. Zgierski, "An adaptive learning solution to the keyboard optimization problem," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 6, pp. 1608–1618, 1991.

[14] C. Daskalakis, R. M. Karp, E. Mossel, S. J. Riesenfeld, and E. Verbin, "Sorting and selection in posets," *SIAM Journal on Computing*, vol. 40, no. 3, pp. 597–622, 2011.

[15] K. Andreev and H. Racke, "Balanced graph partitioning," *Theory of Computing Systems*, vol. 39, no. 6, pp. 929–939, 2006.

[16] R. E. Burkard, *Quadratic assignment problems*. Springer, 2013.

[17] P. Galinier, Z. Boujbel, and M. C. Fernandes, "An efficient memetic algorithm for the graph partitioning problem," *Annals of Operations Research*, vol. 191, no. 1, pp. 1–22, 2011.

[18] J. Kim, I. Hwang, Y.-H. Kim, and B.-R. Moon, "Genetic approaches for graph partitioning: a survey," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 473–480.

[19] U. Gupta and N. Ranganathan, "A game theoretic approach for simultaneous compaction and equipartitioning of spatial data sets," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 4, pp. 465–478, 2010.

[20] M. Meila and J. Shi, "Learning segmentation by random walks," in *Advances in neural information processing systems*, 2001, pp. 873–879.

[21] S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ser. ICCV '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 313–. [Online]. Available: http://dl.acm.org/citation.cfm?id=946247.946658

[22] M. Agache and B. J. Oommen, "Generalized pursuit learning schemes: New families of continuous and discretized learning automata," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 6, pp. 738–749, 2002.

[23] B. J. Oommen and M. Agache, "Continuous and discretized pursuit learning schemes: Various algorithms and their comparison," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 3, pp. 277–287, 2001.

[24] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[25] C. Yuan, T.-C. Lu, and M. J. Druzdzel, "Annealed map," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 2004, pp. 628–635.

[26] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[27] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[28] O.-C. Granmo, "Solving two-armed bernoulli bandit problems using a bayesian learning automaton," *International Journal of Intelligent Computing and Cybernetics*, vol. 3, no. 2, pp. 207–234, 2010.

[29] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2249–2257.

[30] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," in *Conference on Learning Theory, COLT*, 2012.

[31] ——, "Further optimal regret bounds for thompson sampling," in *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, 2013, pp. 99–107.

[32] ——, "Thompson sampling for contextual bandits with linear payoffs," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 127–135.

[33] S. Glimsdal and O.-C. Granmo, "Gaussian process based optimistic knapsack sampling with applications to stochastic resource allocation," in *Proceedings of the 24th Midwest Artificial Intelligence and Cognitive Science Conference 2013*. CEUR Workshop Proceedings, 2013, pp. 43–50.

[34] O.-C. Granmo and S. Glimsdal, "Accelerated bayesian learning for decentralized two-armed bandit based decision making with applications to the goore game," *Applied intelligence*, vol. 38, no. 4, pp. 479–488, 2013.

[35] L. Jiao, X. Zhang, B. J. Oommen, and O.-C. Granmo, "Optimizing channel selection for cognitive radio networks using a distributed bayesian learning automata-based approach," *Applied Intelligence*, vol. 44, no. 2, pp. 307–321, 2016.

[36] D. Tolpin and F. Wood, "Maximum a posteriori estimation by search in probabilistic programs," in *Eighth Annual Symposium on Combinatorial Search*, 2015.

[37] B. Selman, H. A. Kautz, and B. Cohen, "Noise strategies for improving local search," in *AAAI*, vol. 94, 1994, pp. 337–343.

[38] P. Larrañaga, H. Karshenas, C. Bielza, and R. Santana, "A review on evolutionary algorithms in bayesian network learning and inference tasks," *Information Sciences*, vol. 233, pp. 109–125, 2013.

[39] T. Soh, M. Banbara, and N. Tamura, "Proposal and evaluation of hybrid encoding of csp to sat integrating order and log encodings," *International Journal on Artificial Intelligence Tools*, vol. 26, no. 01, p. 1760005, 2017.

[40] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. Springer, 2011, pp. 760–766.

[41] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.

[42] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

Appendix B

Thompson Sampling Guided Stochastic Searching on the Line for Deceptive Environments with Applications to Root-Finding Problems

Journal of Machine Learning Research 20 (2019) 1-24 Submitted 5/18; Revised 12/18; Published 03/19

Thompson Sampling Guided Stochastic Searching on the Line for Deceptive Environments with Applications to Root-Finding Problems*

Sondre Glimsdal SONDRE.GLIMSDAL@UIA.NO
Centre for Artificial Intelligence Research (CAIR)
University of Agder Postboks 422, 4604 Kristiansand, Norway

Ole-Christoffer Granmo OLE.GRANMO@UIA.NO
Centre for Artificial Intelligence Research (CAIR)
University of Agder Postboks 422, 4604 Kristiansand, Norway

Editor: Avi Pfeffer

Abstract

The multi-armed bandit problem forms the foundation for solving a wide range of online stochastic optimization problems through a simple, yet effective mechanism. One simply casts the problem as a gambler who repeatedly pulls one out of N slot machine arms, eliciting random rewards. Learning of reward probabilities is then combined with reward maximization, by carefully balancing reward exploration against reward exploitation. In this paper, we address a particularly intriguing variant of the multi-armed bandit problem, referred to as the *Stochastic Point Location (SPL)* problem. The gambler is here only told whether the optimal arm (point) lies to the "left" or to the "right" of the arm pulled, with the feedback being erroneous with probability $1 - \pi$. This formulation thus targets optimization in continuous action spaces with both *informative* and *deceptive* feedback. To tackle this class of problems, we formulate a compact and scalable Bayesian representation of the solution space that simultaneously captures both the location of the optimal arm as well as the probability of receiving correct feedback. We further introduce the accompanying Thompson Sampling guided Stochastic Point Location (TS-SPL) scheme for balancing exploration against exploitation. By learning π, TS-SPL also supports *deceptive* environments that are lying about the direction of the optimal arm. This, in turn, allows us to address the fundamental Stochastic Root Finding (SRF) problem. Empirical results demonstrate that our scheme deals with both deceptive and informative environments, significantly outperforming competing algorithms both for SRF and SPL.

Keywords: thompson sampling, searching on the line, probabilistic bisection search, deceptive environment, stochastic point location

1. Introduction

Research on the *Stochastic Point Location* (SPL) problem (Oommen, 1997) has delivered increasingly efficient schemes for locating the optimal point on a line. In all brevity, the optimal point must be found based on iteratively proposing candidate points, with each candidate revealing whether the optimal point lies to the candidate's left or to its right. The provided directions can be erroneous, and the goal is to locate the optimal point with

*. A preliminary version of some of the results of this paper appears in the Proceedings of AIAI'15.

as few non-optimal candidate proposals as possible. The SPL problem can also be cast as an agent that moves on a line, attempting to locate a particular location λ^*. The agent communicates with a teacher that notifies the agent whether its current location λ is greater or lower than λ^*. However, the teacher is of a stochastic nature and feeds the agent erroneous feedback with probability $1 - \pi$.

Despite the simplicity of the SPL problem, SPL schemes have provided novel solutions for a wide range of problems. Intriguing applications include the estimation of non-stationary binomial distributions (Yazidi et al., 2012b), communication network routing (Oommen et al., 2007), and meta-optimization (Oommen et al., 2009). Furthermore, recent research that addresses the related *Stochastic Root-Finding* (SRF) problem provides promising solutions for parameter estimation, transportation system optimization, as well as supply chain optimization (Chen and Schmeiser, 2001; Pasupathy and Kim, 2011).

State-of-the-art. Adaptive Step Searching (ASS) (Tao et al., 2013) is currently the leading approach to solving SPL problems, although it is outperformed by Hierarchical Stochastic Searching on the Line (HSSL) (Yazidi et al., 2012a) in highly volatile non-stationary environments (Tao et al., 2013). Optimal Computing Budget Allocation (OCBA) has also been applied to SPL (Zhang et al., 2015) and provides stable solutions while converging slightly slower than ASS. Unfortunately, these state-of-the-art schemes fail when noise increases beyond a certain degree, which happens when the majority of obtained directions mislead rather than guide. Indeed, by naively following the directions provided under such circumstances, one is systematically led away from the optimal point. We refer to these kinds of problem environments as *deceptive* environments, as opposed to *informative* ones, which are explained in more detail below.

To the best of the authors' knowledge, the pioneering CPL-AdS (Oommen et al., 2003) scheme was the first known approach handling deceptive SPL environments. CPL-AdS relies on two consecutive phases. In the first phase, a sequence of intelligently selected questions is used to classify the environment as either informative or deceptive. By spending a sufficient amount of time in this phase, the classification can be made arbitrarily accurate. In the second phase, a regular SPL scheme is applied, except that the directions obtained are reversed if the problem environment was classified as deceptive in the first phase. This means that the scheme may have to remain in the first phase for an extensive amount of time to ensure that the problem environment is correctly classified, otherwise, one risks being systematically misled in the second phase. These properties largely render CPL-AdS inappropriate for online or anytime problem solving.

Recently, HSSL has been extended by Zhang et al. to cover both informative and deceptive environments, using a Symmetric HSSL (SHSSL) scheme (Zhang et al., 2016). This scheme essentially runs two HSSL schemes in conjunction: one regular, which handles informative environments, and one that inverts all feedback from the environment, to handle deceptive environments. The hierarchy navigation capabilities of HSSL are then exploited to allow SHSSL to switch between the two HSSLs, depending on the nature of the environment. However, a significant limitation of HSSL, namely, that π must be larger than the conjugate of the golden ratio, carries over to SHSSL. Indeed, SHSSL fails to converge for $\pi \in [0.382, 0.682]$, which amounts to approximately 30% of the feasible values for π. This is in contrast to the approach we propose in this paper, as well as to CPL-AdS (Oommen

et al., 2003), since both of these schemes operate along the whole range of π (apart from $\pi = 0.5$).

To cast further light on the challenges lined out above, we here introduce the *N-Door Puzzle* as a framework for modeling deception. We also propose an accompanying novel solution scheme — *Thompson Sampling guided Stochastic Point Location* (TS-SPL). The TS-SPL scheme handles both SPL and SRF problems, and is capable of *simultaneously* solving the problem as well as determining whether we are dealing with an informative or a deceptive environment. As we shall see, not only does this scheme handle an arbitrary level of noise, but it also outperforms current state-of-the-art techniques in both informative and deceptive environments.

The N-Door Puzzle. In the book "To Mock a Mockingbird" (Smullyan, 1988) the following puzzle is formulated: *"Someone was sentenced to death, but since the king loves riddles, he threw this guy into a room with two doors. One leading to death, one leading to freedom. There are two guards, each one guarding one door. One of the guards is a perfect liar, the other one will always tell the truth. The man is allowed to ask one guard a single yes-no question and then has to decide, which door to take. What single question can he ask to guarantee his freedom?"* To avoid spoiling the puzzle for the reader, we omit the solution here and note that asking a double negative question will often be the correct course of action for these types of puzzles.

The above puzzle can be generalized by increasing the number of doors. Instead of deciding between merely two doors, the prisoner now faces N doors, with a guard posted between each pair of doors. Only a single door leads to freedom, the remaining doors lead to death. Every day at sunrise, the prisoner is allowed to ask one of the guards whether the door leading to freedom is to the left or to the right of the guard. However, only a fixed proportion of the guards answers truthfully, the rest are compulsive liars. Further, the guards are randomly assigned a position at each sunrise, and thus, knowing who lies and who tells the truth is impossible. As an additional complication, depending on the mood of the king, the prisoner may be ordered to walk through a door of his choice at an arbitrary day. Therefore, to save his life, it is imperative that the prisoner determines as quickly as possible which door leads to freedom.

Specifically, let $\pi = \frac{\#\text{truthful guards}}{\#\text{guards}}$ be the fraction of truthful guards. Since the guards are randomly assigned a position each day, the probability of obtaining a truthful answer is governed by π. If $\pi < 0.5$ then the majority of the guards are compulsive liars, and the guards as an entity can be characterized as being *deceptive*. Conversely, if $\pi > 0.5$ then the majority of the guards are truthful and the guards can be seen as *informative*. For completeness, we mention that the puzzle is unsolvable for the case where π is exactly equal to $\frac{1}{2}$, since it is then impossible to obtain any information on neither the nature of the doors nor the guards.

Thompson Sampling. The Thompson Sampling (TS) principle was introduced by Thompson already in 1933 (Thompson, 1933) and now forms the basis for several state-of-the-art approaches to the Multi-Armed Bandit (MAB) problem — a fundamental sequential resource allocation problem that has challenged researchers for decades. At each time step in the MAB problem, one is offered to pull one out of N bandit arms, which in turn triggers a stochastic reward. Each arm has an underlying probability of providing a reward, however,

these probabilities are unknown to the decision maker. The challenge is thus to decide which of the arms to pull at each time step, to maximize the expected total number of rewards obtained (Bubeck and Cesa-Bianchi, 2012).

In all brevity, TS seeks to achieve the above goal by quickly shifting from exploring reward probabilities to maximizing the number of rewards obtained. This is achieved by recursively estimating the underlying reward probability of each arm, using Bayesian filtering of the rewards obtained so far. TS then simply selects the next arm to pull based on the Bayesian estimates of the reward probabilities (one reward probability density function per arm).

The arm selection strategy of TS is rather straightforward, yet surprisingly efficient. To determine which arm to pull, a single candidate reward probability is sampled from the probability density function of each arm. *The arm with the highest sample value is the one pulled next.* The outcome of pulling this arm is in turn used to perform the next Bayesian update of the arm's reward probability estimate. It is this simple scheme that makes TS select arms with frequency proportional to the posterior probability of being optimal, leading to quick convergence towards always selecting the optimal arm.

TS has turned out to be among the top performers for traditional MAB problems (Granmo, 2010; Chapelle and Li, 2011), supported by theoretical regret bounds (Agrawal and Goyal, 2012, 2013a; Dong and Van Roy, 2018). It has also been successfully applied to contextual MAB problems (Agrawal and Goyal, 2013b), constrained Gaussian process optimization (Glimsdal and Granmo, 2013), distributed quality of service control in wireless networks (Granmo and Glimsdal, 2013), cognitive radio optimization (Jiao et al., 2016), as well as a foundation for solving the maximum a posteriori estimation problem (Tolpin and Wood, 2015).

Pure Exploration Bandits. Throughout this paper we assume that each SPL problem potentially takes part in a larger system consisting of multiple SPL problems, and not necessarily operating in isolation. From existing applications in the literature, such as web crawler load balancing (Granmo et al., 2007), it is clear that the value of an SPL scheme does hinge upon its ability to cooperate and interact with other decision makers. Such cooperation demands predictable behaviour from the individual decision makers, as well as coordinated balancing of exploring new solution candidates against maintaining good solution candidates. Without such an ability, the system as a whole will not be able to systematically move towards the more promising areas of the search space, gradually focusing in on an optimal configuration. Therefore, in this paper we omit a direct comparison with schemes that rely on a "fixed sampling *then* decide" approach, such as unimodal bandits (Jia and Mannor, 2011). For the same reason, we will not investigate purely exploitative bandits (Even-Dar et al., 2006; Jamieson et al., 2014; Audibert and Bubeck, 2010; Gabillon et al., 2011; Karnin et al., 2013), bandits that have a predefined finite time horizon and whose performance is only measured at the end of that horizon. Such algorithms are free to explore without any negative impact, and this allows them to outperform traditional exploitation-exploration bandits such as TS and UCB in scenarios where exploitation is not required.[1]

1. There also exists a wide spectrum of techniques and schemes in the literature on the topic of searching with noise. See for instance (Pelc, 2002) for a comprehensive survey. These are unable to handle unknown and deceptive environments, with stochastic directional feedback, and are therefore not directly

Paper Contributions. In this paper, we introduce a novel scheme for solving the SPL problem, namely, TS-SPL. At the core of TS-SPL, we find a compact and scalable Bayesian representation of the SPL solution space. This Bayesian representation simultaneously captures both the location of the optimal point (bandit arm) as well as the probability of receiving correct feedback. We further introduce an accompanying scheme for balancing exploration against exploitation, based on TS. By learning π, TS-SPL also supports *deceptive* environments that are lying about the direction of the optimal arm. This, in turn, allows us to solve the fundamental SRF problem. More specifically, the contributions of the paper can be summarized as follows:

1. We introduce a novel TS-SPL scheme that represents the solution space of N-Door Puzzles, and thus SPL problems, in terms of a Bayesian model. As opposed to competing solutions that merely maintain and refine a single candidate solution, our Bayesian model encompasses the complete space of candidate solutions at every time instant.

2. We formulate a compact and scalable Bayesian representation of the solution space that simultaneously captures both the location of the optimal point (arm), as well as the probability of receiving correct feedback. This Bayesian representation of the problem opens up for efficient exploration and exploitation of the solution space with TS.

3. We link TS-SPL to so-called stochastic bisection search; and unify accompanying methods under the umbrella of TS.

4. Similarly, we enhance the Soft Generalized Binary Search (SGBS), Probabilistic Bisection Search (PBS) and Burnashev-Zigangirov Algorithm (BZ) by introducing novel parameter free solutions that take advantage of our Bayesian model of the N-door puzzle and the SPL problem. This approach eliminates previous reliance on knowing the exact degree of noise affecting the system to be optimized.

5. We provide the first unified empirical comparison of the key state-of-the-art SPL- and SRF solvers.

6. We finally demonstrate the empirical performance of TS-SPL for both SPL and SRF problems. TS-SPL outperforms the state-of-the-art algorithms in both informative and deceptive environments, except for the SGBS and BZ schemes with correctly specified observation noise.

Paper Outline. The paper is organized as follows. In Section 2, we present our scheme for TS guided SPL (TS-SPL). We first introduce the Bayesian model of the N-door puzzle. Based on the Bayesian model, we then formulate our TS-based scheme that balances solution space exploration against reward maximization. We further extend selected state-of-the-art solution schemes with our Bayesian N-door puzzle model. This extension removes the need for knowing the observation noise beforehand. In Section 3, we provide extensive empirical results comparing TS-SPL with state-of-the-art schemes for both SPL and SRF. We conclude in Section 4 and point to promising directions for further work.

comparable to SPL solution schemes. We have therefore not included this class of techniques in the present paper.

2. Thompson Sampling Guided Stochastic Point Location

In this section, we introduce the TS-SPL scheme. The scheme can be summarized as follows. At the core of TS-SPL we find a Bayesian model of the N-Door Puzzle. Formally, we represent an instance of the N-door puzzle as a tuple $(\lambda^*, \pi^*) \in D \times T$, where $D = \{d_1, \ldots, d_N\}$ is the set of doors and $T \in [0, 1]$ is the truthfulness of the guards. Let (λ^*, π^*) be the particular N-door puzzle faced. A novel aspect of TS-SPL is that instead of maintaining a single or a limited set of candidate solutions, we instead maintain a posterior distribution over the whole solution space, $(\lambda, \pi) \in D \times T$. This distribution is conditioned on the feedback already obtained up to time step n, allowing us to single in on (λ^*, π^*) as the number of time steps increases, ultimately converging to (λ^*, π^*).

Assuming no prior information, we assign a uniform distribution over $D \times T$, i.e., all puzzle instances are equally probable. By gradually refining the posterior distribution over $D \times T$, we can select guards to question in a goal-oriented manner. In all brevity, we sample a solution candidate (λ^c, π^c) from $D \times T$, selecting the guard to the left or to right of λ^c. The answer of the selected guard is then used to update our posterior distribution. By repeating this procedure, the expected probability of the underlying N-door puzzle instance (λ^*, π^*) increases monotonically, reducing the probability of other puzzle instances. In effect, given enough iterations, TS-SPL will correctly identify the door leading to freedom as the posterior probability of (λ^*, π^*) approaches unity.

2.1 Bayesian Model of the N-Door Puzzle

The main purpose of the Bayesian model is to facilitate the efficient calculation of a posterior distribution over the possible N-door puzzle instances, $D \times T$. Since the prisoner does not initially know which problem instance he is facing, and since the observations are stochastic, we cast D and T as two random variables. We further assume that D and T are independent of each other. Furthermore, the information we obtain from questioning the guards is represented as a set of random variables $Q = \{Q_1, \ldots, Q_n\}$, with each random variable Q_k representing the answer from question k. Finally, we assume that the outcomes of the individual questions $Q_k \in Q$ are independent when conditioned on D and P. For each question Q_k, we can then compute the probability of the answer ("left" or "right") that we received from the guard, as summarized in Table 1.

Guard to the left of door to freedom	$P(\text{left} \mid \text{guard, door}, t) = t$
	$P(\text{right} \mid \text{guard, door}, t) = 1 - t$
Guard to the right of door to freedom:	$P(\text{left} \mid \text{guard, door}, t) = 1 - t$
	$P(\text{right} \mid \text{guard, door}, t) = t$

Table 1: Conditional door probabilities

As an example, let us assume that the truthfulness of the guards is $t = 0.75$. If for instance the guard to the left of door d_4 replies that the door leading to freedom lies to his left, we can infer that all doors to the left have the likelihood of $t = 0.75$ of leading to freedom, and all the doors to the right have the likelihood $1 - t = 0.25$ of leading to freedom.

Applying Bayes Theorem to $P(Q|d,t)$, defined in Table 1, we are able to derive closed-form expressions for the posterior distributions of both D and T. The derivation of $P(d|Q)$, $d \in D$, follows [the derivation of $P(t|Q)$, $t \in T$, is analogous, and is left out here for the sake of brevity]:

$$P(d|Q) = \sum_{t \in T} P(d,t|Q) \tag{1}$$

$$\propto \sum_{t \in T} P(Q|d,t)P(d,t) \tag{2}$$

$$= \sum_{t \in T} P(Q|d,t)P(d)P(t) \tag{3}$$

$$= \sum_{t \in T} \hat{Q}Q^{+}P(d)P(t) \tag{4}$$

Above, $\hat{Q} = \prod_{k=1}^{n-1} P(Q_k|d,t)$ and $Q^{+} = P(Q_n|d,t)$. Further, (2) follows directly from Bayes Theorem. We obtain (3) as a result of the independence of D and T, and (4) from the independence between the questions in Q. This leads to the following two equations for updating our knowledge about both the door probabilities (5) and the truthfulness of the guards (6).

$$P(d|Q) \propto \sum_{t \in T} \hat{Q}Q^{+}P(d)P(t) \tag{5}$$

$$P(t|Q) \propto \sum_{d \in D} \hat{Q}Q^{+}P(d)P(t) \tag{6}$$

2.2 Guard Selection

We have now formally determined how we can transform information from the guards into a probability distribution over which door leads to freedom. However, as mentioned previously, we also face a trade-off between exploring different doors and zeroing in on the best door found so far. To handle this trade-off, we model the door selection as a so-called Global Information MAB (GI-MAB) (Atan et al., 2015).

To decide which door should be selected at each iteration, we solve the GI-MAB by utilizing the principle of TS. Here, the selection process is simply to select a random door proportional to the probability that this door is the one that leads to freedom. Once the door has been selected, we need to decide which of the guards to query: the guard to the left or to the right of the door selected. We do this by randomly selecting one of the guards, again proportional to the sum of the probabilities of the doors next to each guard. Let us assume for instance that we have three doors $d_k, 1 \leq k \leq 3$ with the probabilities of leading to freedom: $P(d_1) = 0.1, P(d_2) = 0.2, P(d_3) = 0.7$. Then, according to the TS principle, these are also the probabilities we use to sample a particular door. Note that since the answer obtained from each guard affects the complete probability distribution over D (the probability associated with every door is updated), we have a GI-MAB as opposed to a traditional MAB.

2.3 Improving State-of-the-Art Schemes with the Bayesian Model of the N-Door Puzzle

A main advantage of TS-SPL compared to similar schemes is the utilization of the Bayesian model that enables TS-SPL to operate without knowing the problem parameters in advance. Due to TS-SPL's close connection to the Probabilistic Bisection Search (PBS) (Horstein, 1963), Noisy Generalized Binary Search (NGBS) (Nowak, 2008) and the BZ algorithm (Burnashev and Zigangirov, 1974), we will here use our Bayesian TS-SPL model to also make these other schemes parameter free.

Probabilistic Bisection Search

The goal of PBS[2] (Waeber et al., 2013; Nowak, 2008) is to locate an unknown point $X^* \in [0, 1]$. To acquire intelligence on the location of X^*, one queries an oracle of the relation between a point x and X^*. The oracle responds by informing whether x is on the left or the right side of X^*. If we assume that the oracle is always telling the truth, then the well-known deterministic bisection search, which halves the search space with each query, can be employed to efficiently find X^*. However, in PBS we assume that the Oracle provides correct answers with probability $p \in (0.5, 1.0]$ and erroneous ones with probability $1 - p$.

The PBS can be traced back to Horstein (Horstein, 1963). In PBS a probability distribution is mapped over the search space and is gradually updated using a Bayesian methodology under the assumption that the environment noise p is known a priori. The search space is then continuously explored using the median of the posterior distribution as the point of interest. It has been shown that PBS has a geometric rate of convergence under the latter assumptions (Waeber et al., 2013).

As the noise p is assumed to be given, one can simply invoke (7) to calculate the posterior distribution:

$$P(d \mid Q) \propto P(Q \mid d)\, P(d). \tag{7}$$

Here, $P(Q \mid d)$ is the conditional probability of obtaining answer Q (point to the right). That is, for every location d to the left of X^*, the probability that the oracle directs the decision maker to the right is $P(Q \mid d) = p$. And conversely, $P(Q \mid d) = 1 - p$ for d to the right of X^*.

To explicitly represent PBS' dependence on knowing p beforehand, we can cast (7) in terms of (5) and (6). The resulting model is identical to TS-SPL, with the major difference that PBS employs the median to explore the search space. We denote this new and improved scheme PBS-M.

We also observe that due to its simple nature, PBS is particularly well-suited for parallel computing environments (Pallone et al., 2014), as opposed to more traditional stochastic approximation methods (Kushner and Yin, 1987).

PowerTest-Probabilistic Bisection Search

In a recent paper, Frazier et al. (Frazier et al., 2019) demonstrated an alternative approach to removing the dependency of PBS on knowing the fixed noise probability p. Instead of

2. In this context this scheme also covers the Stochastic *Binary* Search

applying a Bayesian prior over p, as done in TS-SPL, they introduce a frequency-based approach, referred to as PowerTest-PBS (PT-PBS). PT-PBS is based on repeatedly sampling the underlying function $g(x)$ until a pre-specified confidence α is archived on a hypothesis test over the sign of the feedback of $g(x)$. They further demonstrated that the asymptotic convergence of PT-PBS is similar to that of Stochastic Approximation (SA).

Generalized Binary Search

The Generalized Binary Search (GBS) problem can be formulated as follows (Nowak, 2008, 2011). Consider a collection of unique binary-valued functions H defined on a domain X. Each $h \in H$ is defined as a mapping from X to $\{-1, 1\}$. Assume that there exists an optimal function $h^* \in H$ that produces the correct binary labeling for each $x \in X$. For each query $x \in X$, the value of $h^*(x)$ is observed, possibly corrupted by independent binary noise. The objective is then to determine the function h^* using as few queries as possible. In this paper, we restrict H to the class of threshold binary functions with the effect of turning the GBS into an informative N-door puzzle.

If the feedback is noiseless then the problem simplifies to the combinatorial problem of finding an optimal decision tree in the H space, a problem that Hyafil and Rivest showed to be NP-complete (Hyafil and Rivest, 1976; Nowak, 2011).

The Soft-Decision Generalized Binary Search (SDGB-Search) (Nowak, 2008, 2011) is the *state-of-art* algorithm for finding $h^*(x) \in H$ when the probability of binary noise is less than 1/2, that is, for informative environments.

Similar to TS-SPL, SDBG-search employs a probabilistic model that for time step n assigns a probability $p_n(h)$ to each $h \in H$. However, for each time-step, it decides which $x \in X$ is queried next based on a deterministic heuristic:

$$\arg\min_{x \in X} \sum_{h \in H} |p(h)h(x)| \tag{8}$$

SDGB uses the following equation to determine and update $p_n(h)$ at each time step:

$$p_{i+1}(h) \propto p_i(h)\beta^{(1-z_i(h))/2}(1-\beta)^{(1+z_i(h))/2}. \tag{9}$$

Here, $z_i(h) = h(x_i)y_i$ and $y_i \in \{-1, 1\}$ are the responses from $h^*(x_i)$. Simplifying (9), we observe that $z_i(h)$ represents an *AND* operator that takes on the value 1 if $h(x_i)$ is equal to $h^*(x_i)$ and -1 otherwise. Furthermore, we note that since $z_h(i) \in \{-1, 1\}$, then one of $1 - z_i(h)$ and $1 + z_i(h)$ will have to take the value 2, while the other takes the value 0.

By applying the transformation $\pi = 1 - \beta$, we can rewrite (9) as:

$$p_{i+1}(h) \propto= \begin{cases} p_i(h) \times \pi & \text{if } y_i = h^*(x_i) \\ p_i(h) \times (1-\pi) & \text{else} \end{cases} \tag{10}$$

.

This update scheme is identical to the one in PBS and thus suffers from the same limitation (noise probability is assumed to be known a priori). In the same manner as we enhanced PBS by employing our Bayesian TS-SPL scheme, we can make SDGB parameter-free using (5,6). In the following, we will denote this improved version of SDGB as SDGB-M.

Burnashev-Zigangirov Algorithm

The Burnashev-Zigangirov (BZ) algorithm (Burnashev and Zigangirov, 1974) is one of the most widely used algorithms for solving the discrete PBS problem and has in particular been used in active learning (Singh et al., 2006; Castro and Nowak, 2006). The BZ algorithm searches for a point θ^* that is located on a line. This line is discretized into m bins and we are only allowed to query the borders of the bins for the direction of θ^*. The BZ algorithm suffers from the same practical limitation as PBS and SDGB, namely a dependency on knowing the exact noise level.

We will now show how the BZ algorithm can be improved in a similar fashion as PBS and SDGB, leveraging our Bayesian model. Let $a_i(j)$ denote the probability of θ^* residing in bin I_i at time-step j. The probability mass function (pmf) of all the bins is therefore $\mathbf{a}(j) = \{a_1(j), a_2(j), \ldots, a_m(j)\}$ with its cumulative density function (cdf) denoted as $\mathbf{A}(j)$.

To decide which point to investigate next (i.e., decide a value for X_{j+1}), the BZ algorithm selects one of the two closest points to the median of $\mathbf{a}(j)$. We denote this point $k = k(j+1)$. The binary response variable $Y_{j+1} = \mathbb{1}\{X_{(j+1)} \geq \theta^*\}$ is observed with probability $1-\alpha$, whereas $Y_{j+1} = \mathbb{1}\{X_{(j+1)} < \theta^*\}$ is observed with probability α (the noise probability).

To update the probability distribution over $\mathbf{a}(j)$, we introduce $\beta = 1-\alpha$ and $\tau = 2A(k(j+1)) - 1$. For $i \leq k$, we then have

$$a_i(j+1) = a_i(j) \begin{cases} \frac{2\alpha}{1-\tau(\beta-\alpha)} & \text{if } Y_{j+1} = 0 \\ \frac{2\beta}{1+\tau(\beta-\alpha)} & \text{if } Y_{j+1} = 1 \end{cases}$$

and for $i > k$ we have:

$$a_i(j+1) = a_i(j) \begin{cases} \frac{2\beta}{1-\tau(\beta-\alpha)} & \text{if } Y_{j+1} = 0 \\ \frac{2\alpha}{1+\tau(\beta-\alpha)} & \text{if } Y_{j+1} = 1 \end{cases}$$

To make the BZ algorithm parameter-free, we first note that for any given noise $t \in T$, we have that $\beta = t$, $\alpha = 1-t$, $\beta - \alpha = 2t-1$, and $\tau = A_k(j) - (1 - A_k(j))$. After some simple algebraic manipulations, it turns out that the updating scheme of the BZ algorithm is identical to PBS except that:

1. The BZ algorithm calculates the normalizing factor as a part of the updating rule instead of using the likelihood value, and then later normalizes as PBS does.

2. The BZ algorithm samples on the interval edges while PBS samples the midpoints of each interval.

To obtain an enhanced parameter-free version of the BZ algorithm, we simply replace α as a pre-determined constant with a prior distribution that we marginalize out using (5) and (6). We denote the resulting scheme BZ-M.

3. Empirical Results

In this section, we evaluate the performance of TS-SPL empirically, in comparison with competing schemes. We investigate both the effect that the various parameter settings

have on the behavior, as well as the capability of TS-SPL to handle different applications, including SPL and SRF problems. Unless otherwise noted, the empirical results report the average of 10 000 independent trials.

For some of the applications we investigate here, we do not find any existing scheme that handles deceptive environments. Instead, the schemes we identified assume that feedback is on average informative. To make the comparison fair, we thus introduce TS-SPL-INF, which is configured with the precondition that the feedback is informative. This modification also serves to exemplify one of the advantages of our Bayesian approach — we can tailor the the prior distribution of the noisy probability for the task at hand. Note that this informed prior is equivalent to the priors we use for the other probability theory based schemes we introduced in this paper, namely, PBS-M and SDGB-M.

Further note that we apply a fixed set of parameter values across the whole suite of experiments, set to optimize overall performance. For SHSSL (Zhang et al., 2016) and HSSL (Yazidi et al., 2012a) we used a tree branching factor of $D = 8$, and for ASS (Tao et al., 2013) we set $N_{\max} = 256$ and $N_{\min} = 1$. For OCBA (Zhang et al., 2016), we set $n_0 = 15$ and $\theta = 1/256$. We additionally set the confidence γ of PT-PBS (Frazier et al., 2019) to 0.55 based on a comprehensive brute force search for the best value. The prior used for TS-SPL is uniform over the unit interval and is discretized as $|D| = 201$ and $|T| = 101$. For the informative schemes TS-SPL-INF, PGA-M, SGDB-M, BZ-M, we use the same prior for the doors as for TS-SPL, $|D| = 201$ however, we use a uniform prior over the interval $(0.5, 1]$ for truthfulness, with $|T| = 51$.

We will in the following subsections investigate (1) the effect of different priors on TS-SPL; (2) TS-SPL's ability to identify the nature of the underlying stochastic environment; (3) the ability to solve the SPL problem; and (4) performance on SRF problems – a particularly intriguing class of deceptive environments that arises naturally as a result of the properties of stochastic root finding.

3.1 Sensitivity to Discretization and Distribution of Prior

Although TS-SPL is a parameter free scheme, it depends on defining $D \times T$, the set of all possible N-Door Puzzles, and then formulating a prior distribution over this space. We here investigate to what degree the performance of TS-SPL is affected by the degree of discretization, that is, the cardinality of $D \times T$.

To measure performance, we count how many time steps passes before 95% of the probability mass is contained within the target interval I, that is, $P(I|\text{Observed History}) \geq 0.95$. We refer to this event as convergence of the learning process.

From Table 2, we observe that the cardinality of D, in fact, does affect the performance of TS-SPL. As $|D|$ increases, so does the time it takes before TS-SPL converges. However, it is evident that the relationship between convergence time and $|D|$ is non-linear. Indeed, the increase in convergence time is insignificant even when doubling the number of doors from 3200 to 6400. The behaviour reported in the table thus indicates a logarithmic relation between $|D|$ and convergence time.

To see how the cardinality $|T|$ of T affects performance, we increase $|T|$ from 50 to 3200, fixing $|D|$ to 100. From Table 3, it is clear that $|T|$ does not significantly affect performance.

$\|D\|$:	100	200	400	800	1600	3200	6400
Convergence Steps:	31.4	36.0	38.9	39.4	39.3	40.2	40.9

Table 2: Convergence steps for TS-SPL solving the N-Door Puzzle with $\lambda^* = 0.15$, $I = \{0.15 \pm 0.01\}$, $T = \{0.8\}$, and $\pi = 0.8$.

$\|T\|$:	50	100	200	400	800	1600	3200
Convergence Steps:	51.6	50.8	48.4	52.1	51.0	52.4	52.1

Table 3: Convergence steps for TS-SPL solving the N-Door Puzzle with $\lambda^* = 0.15$, $I = \{0.15 \pm 0.01\}$, $|D| = 101$, and $\pi = 0.8$.

Another advantage of our Bayesian scheme is the ability to incorporate prior information to guide the algorithm. On the other hand, specifying an incorrect prior can potentially deteriorate performance instead of enhancing it. In Table 4, we provide performance results from employing an informed prior over T and D. With the correct underlying values $\lambda^* = \pi = 0.85$, we specify three types of priors: Correct $\propto N(\mu = 0.85, \sigma = 0.3)$, Incorrect $\propto N(\mu = 0.15, \sigma = 0.3)$ and Flat (all solutions equally probable), denoted C, I, and F, respectively. We can see the effect of these different priors in Table 4. In brief, having a correct prior over the doors contributes more to convergence time than having a correct prior over the truthfulness of the guards. The disadvantage of setting an incorrectly biased prior is also evident, as the flat prior performs better than any combination involving a incorrectly biased prior.

3.2 Tracking the Truthfulness of the Environment

An interesting property of TS-SPL is its ability to provide a distribution over the truthfulness π of the environment. This can be a significant advantage because information on π can be leveraged in various ways. As an example, information on π can be used in the case of repeated trials, where the information from previous trials can be used as a prior in subsequent trials, greatly increasing convergence speed (cf. Section 3.1). Figure 1 plots the probability of each level of noise as the TS-SPL progresses with noise probability $\pi = 0.15$ (a highly deceptive environment). As seen, TS-SPL is capable of quickly estimating π accurately.

3.3 Stochastic Point Location

The N-Door Puzzle, as outlined in the introduction, is dependent on two variables λ^* and π^*, with λ^* specifying the door leading to freedom and π^* the truthfulness of the guards. Since the N-Door Puzzle does not pose any spatial requirements on the placements of the doors we can generate a mapping from the N-Door Puzzle to the SPL problem by uniformly placing the doors over the unit interval.

We here use so-called regret to measure performance because not all of the schemes evaluated in this section are Bayesian. Regret is further typical for evaluating multi-armed bandit algorithms Regret can be stated as the cumulative penalty from selecting sub-optimal

Door	Truthfulness	Convergence
F	F	36.4
F	C	35.7
F	I	41.2
C	F	30.2
C	C	30.0
C	I	40.5
I	F	46.4
I	C	45.2
I	I	113.1

Table 4: Convergence steps for TS-SPL solving the N-Door Puzzle with different priors: C - Correct Prior, F - Flat Prior, I - Incorrect prior. Here, $\lambda^* = 0.85$, $I = \{0.15 \pm 0.01\}$, $|T| = |D| = 101$, and $\pi = 0.85$.

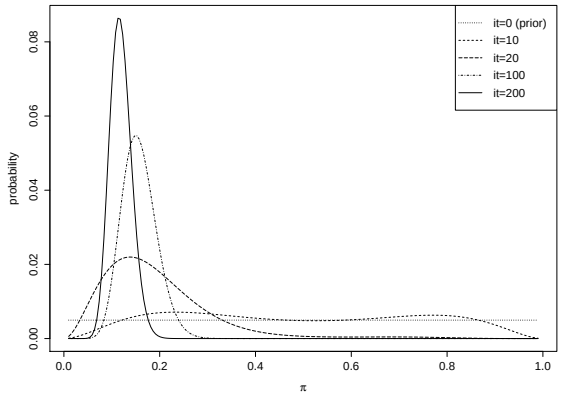

Figure 1: TS-SPL maintains a posterior distribution over π. Here, the true underlying value of π is 0.15. The figure shows the posterior distribution of π after various number of iterations during a single run of TS-SPL. As seen, TS-SPL obtains a sharply peaked posterior over π after only 20 iterations.

actions. In the case of SPL we define regret as the (unsigned) distance between the selected point x and the optimal point λ^*.

3.3.1 Informative SPL

We first evaluate the performance of TS-SPL and TS-SPL-INF on an informative SPL problem, in comparison with algorithms designed to handle informative environments. To the best of our knowledge this is the first time both the family of PBS based schemes and the family of SPL based schemes are compared.

The performance of the different schemes is summarized in Table 5. One significant observation is the performance difference between the Learning Automata (LA) based schemes (HSSL and SHSSL) and the Bayesian schemes. It is clear that performance-wise, Bayesian schemes significantly outperform the LA based schemes. However, it should be noted that the LA based schemes require less memory and run faster than the Bayesian ones due to their simplicity.

As seen in Table 5, the distance $|\frac{1}{2} - \lambda^*|$ is an important metric for how hard a particular SPL problem is to solve. This can be explained by the fact that most schemes start exploring from the center. Thus, if λ^* is far from the center, such a scheme needs more evidence before it starts exploring the peripheral regions of the search space. This is particularly apparent for PBS-M as its performance peaks in the case where $\lambda^* = 0.25$, even when faced with significant noise ($\pi = 0.65$).

Since PBS-M pursues the median of the probability distribution, we can say that PBS-M is conservative in its exploration. This is because it takes significant more evidence to move the point of exploration compared to TS-SPL. TS-SPL, on the other hand, has a tendency to explore too much. Indeed, as noted by Lattimore (Lattimore, 2015), using TS for exploration can lead to over-exploration when facing high variance distributions. In the low noise scenarios, however, NGBS-M is the most efficient scheme, exploring deterministically. For PT-PBS, we observe throughout all the experiments significantly worse performance than PBS-M. We thus omit the PT-PBS results for the remaining experiments, focusing on PBS-M instead.

Finally, from Table 6 we observe that TS-SPL-INF exhibits the lowest standard deviation overall, and is consequentially the scheme that consistently perform closest to its expected regret for every trial. This is in sharp contrast to PBS-M who outperform TS-SPL when it comes to average regret, but is unable to do so consistently. NGBS-M also displays significant variance in high noise scenarios.

3.3.2 SPL in Deceptive Environments

With the underlying π taking on values in the interval $[0, 1]$, we test TS-SPL, CPL-AdS (Oommen et al., 2003) and SHSSL (Zhang et al., 2016) for speed of convergence as well as accumulation of regret. However, since CPL-AdS operates in a two-phase manner, direct comparison with TS-SPL and SHSSL is inappropriate (the latter schemes also operate online). Oommen et al. states that this decision phase needs approximately 200 time steps (Oommen et al., 2003), and by this time TS-SPL is already close to converging to the actual solution. Table 7 further explores this difference. As seen, TS-SPL outperforms CPL-AdS by several orders of magnitude, also outperforming SHSSL.

	Avg Regret $\lambda^* = 0.25$	Avg Regret $\lambda^* = 0.85$	Avg Regret $\lambda^* = 0.95$
TS-SPL	29.2 / 9.8 / 5.1	36.4 / 12.9 / 6.2	57.3 / 20.3 / 10.1
TS-SPL-INF	22.2 / 7.3 / 3.7	**22.5** / 7.7 / 3.8	**23.9** / 8.7 / 4.3
PBS-M	**9.8** / 4.0 / 2.6	32.7 / 14.2 / 8.5	52.1 / 29.6 / 16.9
PT-PBS	245.4 / 247.5 / 247.4	548.2 / 751.5 / 785.2	551.4 / 815.0 / 828.4
BZ-M	23.5 / 5.9 / 2.2	27.5 / 6.3 / 2.5	35.1 / 9.6 / 3.4
NGBS-M	36.9 / **3.5** / **1.0**	48.9 / **4.5** / **1.5**	68.5 / **7.1** / **2.3**
ASS	45.8 / 17.0 / 6.7	30.4 / 8.9 / 3.6	38.8 / 11.7 / 3.9
OCBA	70.8 / 47.4 / 35.2	89.9 / 55.8 / 37.1	112.1 / 78.4 / 48.8
HSSL	117.3 / 23.1 / 8.2	111.7 / 16.7 / 4.8	131.5 / 19.1 / 5.3
SHSSL	152.2 / 32.6 / 11.8	151.8 / 23.5 / 6.5	175.1 / 26.1 / 7.3

Table 5: Average regret for the different schemes in an informative SPL. The result is reported in the format $a/b/c$, where a is the average regret for $\pi = 0.65$, b for $\pi = 0.75$, and c for $\pi = 0.85$. The number of time steps per trial is 1000.

	Std. dev. $\lambda^* = 0.25$	Std. dev. $\lambda^* = 0.85$	Std. dev. $\lambda^* = 0.95$
TS-SPL	16.8 / 5.9 / 2.6	20.5 / 6.5 / 3.1	30.9 / 10.3 / 4.6
TS-SPL-INF	**13.8** / **4.2** / 2.0	**14.2** / **4.4** / 2.5	**15.7** / **5.7** / 2.4
PBS-M	15.2 / 10.3 / 10.2	69.1 / 40.9 / 31.5	94.2 / 71.1 / 56.6
PT-PBS	20.1 / 19.0 / 13.6	145.6 / 31.0 / 22.8	192.1 / 60.5 / 55.8
BZ-M	30.4 / 8.9 / 3.1	40.8 / 9.8 / 4.9	48.8 / 15.3 / 5.3
NGBS-M	68.5 / 8.9 / **0.9**	83.7 / 13.6 / **1.4**	108.7 / 19.4 / **1.6**
ASS	51.6 / 22.4 / 10.1	47.8 / 15.7 / 5.4	62.3 / 23.1 / 4.6
OCBA	46.2 / 27.6 / 19.4	63.9 / 43.9 / 25.6	76.1 / 64.6 / 41.9
HSSL	71.7 / 16.1 / 4.6	83.6 / 16.1 / 4.2	94.8 / 19.4 / 4.5
SHSSL	89.7 / 23.5 / 6.2	108.5 / 23.4 / 5.8	126.5 / 27.7 / 6.4

Table 6: Standard deviation for the different schemes in an informative SPL. The result is reported in the format $a/b/c$, where a is the standard deviation for $\pi = 0.65$, b for $\pi = 0.75$, and c for $\pi = 0.85$. The number of time steps per trial is 1000.

	$\pi = 0.85$	$\pi = 0.15$
TS-SPL ($\lambda^* = 0.85$)	6.2	**6.2**
CPL-AdS ($\lambda^* = 0.85$)	501.6 / 354.9	842.8/502.3
PBS-M ($\lambda^* = 0.85$)	31.5	77.5
BZ-M ($\lambda^* = 0.85$)	4.9	352.5
NGBS-M ($\lambda^* = 0.85$)	**1.4**	191.2
SHSSL ($\lambda^* = 0.85$)	6.5	6.5

Table 7: Cumulative regret for the deceptive SPL problem after $N = 1000$ time steps. For CPL-AdS, we report both the total accumulated regret, as well as regret obtained after the nature of the environment has been decided.

Another interesting observation is that the performance of TS-SPL is symmetric around 0.5. Further note that SHSSL fails to converge for $\pi \in [0.382, 0.682]$, as stated earlier. Hence, SHSSL is effectively operating with a 30% smaller search space for π than both TS-SPL and CPL-AdS.

After modifying PBS, NGBS and BZ to support a Bayesian model of truthfulness, we can use the same prior that we apply in TS-SPL also for these schemes, leading to PBS-M, NGBS-M and BZ-M. The effect of this enhancement to existing schemes is summarized in Table 7. As clearly seen, the query selection method for these schemes is not suited to handle deceptive environments.

3.4 Stochastic Root-Finding

The deterministic root finding problem concerns locating a root x^* of a function $g(x)$, defined over an interval (a, b) [i.e., finding x^*, $g(x^*) = 0$]. We assume that $g(x)$ is unknown, however, an oracle returns the value of $g(x)$ at any point x queried. The problem is then how to determine the root x^* using as few queries as possible. One approach to solving the deterministic root finding problem is the Bisection Method. This approach halves the search space in each iteration by continually querying the oracle using the midpoint of the remaining search space.

If the response from the oracle is noisy, we obtain the SRF problem (Pasupathy and Kim, 2011). We define the SRF problem as follows. For any $x \in (0, 1)$, the oracle generates a sample $Y(x) = g(x) + w_{\text{noise}}$, where w_{noise} is a random variable with mean zero. Let $S(x)$ denote the sign of $Y(x)$: $S(x) = \text{sgn}[Y(x)]$. Notice that the noise w_{noise} may render $\text{sgn}[Y(x)]$ different from $\text{sgn}[g(x)]$. Thus, with noisy feedback, the Bisection Method may discard the wrong half of the search space. The challenge is then how to select a sequence of queries $x_1, x_2, \ldots, x_n$ to gather information on x^*, so that the final query x_n is close to x^*, $|x_n - x^*| < \epsilon$, despite the noise (Waeber et al., 2011). Note that in the SRF problems we investigate here, $S(x)$ returns $\text{sgn}[g(x)]$ with probability π and $-\text{sgn}[g(x)]$ with probability $1 - \pi$.

The traditional approach to solving SRF problems is to apply a variant of Stochastic Approximation (SA) (Robbins and Monro, 1951; Kiefer and Wolfowitz, 1952). Implementation-

wise SA methods[3] extend or modify the iterative Newton-Raphson algorithm to handle noise:

$$x_{n+1} = x_n - a_n Y_n(x_n)$$

where $\{a_n\}$ is a sequence of step lengths that decreases as n increases.

Approaches for applying SA to the SRF problem has been extensively studied in the literature. It is outside the scope of this article to give a full literature review, however, interested readers are referred to surveys in recent studies (Lai, 2003; Asmussen and Glynn, 2007; Pasupathy and Kim, 2011). As there exists a myriad of different SA approaches, we have selected one of the more fundamental ones to form a basis for contrasting the different schemes.

Note that the SA approach we use here requires that $g(x)$ is monotonic. This can be explained as follows. The main difference between SRF and SPL is that, unlike SPL, the SRF oracle does not directly provide feedback on the direction of the root x^* from the query location x. However, for monotonic $g(x)$, one can obtain this direction from the sign of $Y(x)$, $S(x) \in \{-1, 1\}$ and from the derivative $g'(x)$ of $g(x)$. If $g(x)$ is increasing and $S(x) = 1$, then the direction derived from the oracle is "to the left of x" [and "to the right" if $S(x) = -1$]. Conversely, if $g(x)$ is decreasing and $S(x) = 1$, then the direction obtained is "to the right of x" [and "to the left" if $S(x) = -1$].

TS-SPL, on the other hand, does not need to know the derivative of $g(x)$. Indeed, $g(x)$ does not even need to be monotonic. Instead, TS-SPL merely requires an arbitrary mapping of the sign $S(x)$ to a direction. One could, for instance, define positive to mean "left", $S(x) = 1 \Rightarrow$ *left*, and negative to mean "right", $S(x) = -1 \Rightarrow$ *right*. If it turns out that the opposite is the case, TS-SPL will recognize that the feedback is deceptive and still solve the problem. An informative scheme, on the other hand, will be misled in such a deceptive environment.

Informative SPL schemes can also be used for SRF problems. However, then we need an initial sampling step that decides the nature of the function $g(x)$. Learning whether the function $g(x)$ starts above zero and falls below zero, or vice versa, can be done by repeatedly querying a single point on the edge of the interval $(0, 1)$, obtaining multiple samples from either $S(x)$. In brief, by estimating $E[S(x)]$, we can decide the nature of $g(x)$. To gain insight into how many repeated samples are sufficient for estimating $E[S(x)]$ accurately, we employ the two sided Hoeffding's inequality $P(|\bar{X} - E[\bar{X}]| \geq \delta) \leq 2e^{-2n\delta^2}$. Here, $\bar{X}$ is the average of n queries at x and δ is a value such that $|\pi - \frac{1}{2}| \geq \delta$. Setting the rhs. equal to p and solving for n, we obtain $n \geq -\frac{\log(p/2)}{2\delta^2}$. Plugging in for $\delta = 0.05$ and $p = 0.99$ we obtain $\lceil n \rceil = 62$. Thus we are 99% sure of our estimate of $S(x)$, given that $|\pi - 0.5| \geq 0.05$.

The functions that we use to measure performance and compare schemes are illustrated in Figure 2. From Table 8, 9 and 10 it is clear that TS-SPL is the most efficient root solver among state-of-the-art schemes. We believe this largely comes from the fact that it simultaneously learns the nature of the oracle (informative or deceptive), as well as trying to locate the root x^*. In addition, there is the risk that the sampling procedure that the other schemes apply to determine which direction $g(x)$ is increasing may conclude with the wrong answer. If this happens, none of the schemes depending on the sampling will

3. The form of SA shown here is also referred to as Classical Stochastic Approximation (CSA) as it closely resembles the original form proposed by Robbins and Monro (Pasupathy and Schmeiser, 2010).

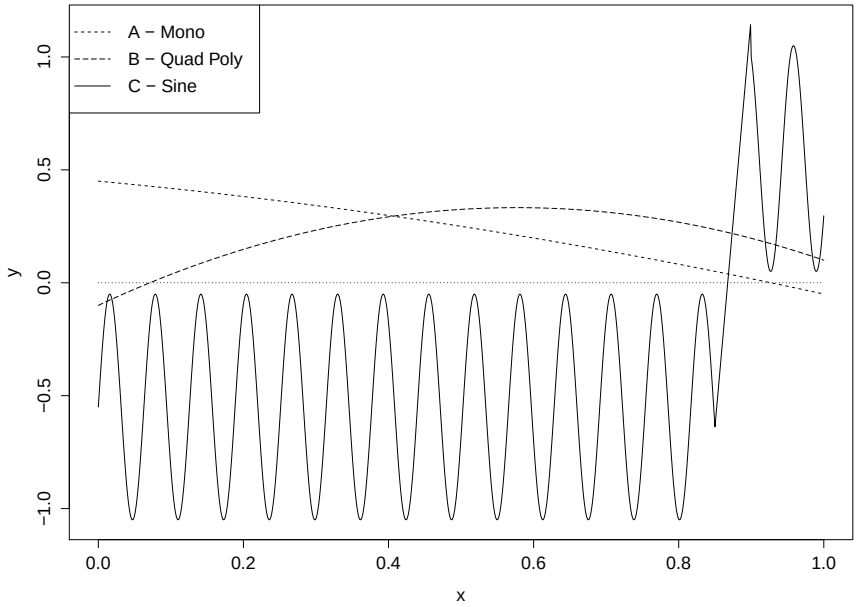

Figure 2: The three functions A, B and C for benchmarking stochastic root finding schemes.

Func A	Avg Regret. $\pi = 0.65$	Avg Regret. $\pi = 0.75$	Avg Regret $\pi = 0.85$
TS-SPL	46.0	17.1	8.6
TS-SPL-INF	55.2 (24.3)	39.1 (8.1)	35.0 (4.1)
PBS-M	55.1 (24.2)	40.3 (9.3)	35.2 (4.2)
NGBS-M	53.4 (22.4)	35.3 (4.3)	32.9 (1.9)
BZ-M	63.8 (32.8)	39.9 (8.9)	33.8 (2.8)
SA	**32.4**	**14.8**	**5.4**
ASS	80.4 (49.4)	38.7 (7.7)	33.5 (2.5)
HSSL	60.4 (29.4)	45.5 (14.6)	35.8 (4.8)
SHSSL	62.8	20.2	6.7
CPL-AdS	162.1 (107.9)	146.3 (97.4)	135.3 (90.1)

Table 8: Average residuals for the different schemes when finding the root of the monotonic function A under various noise levels. The root is $x^* = 0.07104$. The results are given in the format "average residuals (average residuals after sampling)" for each scheme. For CPL-AdS the sampling period is the estimation period (epoch 0) as defined by the scheme. The number of iterations per trial is 250.

Func B	Avg Regret. $\pi = 0.65$	Avg Regret. $\pi = 0.75$	Avg Regret $\pi = 0.85$
TS-SPL	47.1	**17.8**	**8.5**
TS-SPL-INF	53.8 (22.9)	39.5 (8.5)	35.1 (4.1)
PBS-M	**41.1** (10.2)	35.3 (4.31)	33.3 (2.3)
SGBS-M	50.6 (19.7)	35.7 (4.7)	33.0 (2.0)
BZ-M	60.3 (29.4)	39.6 (8.7)	33.6 (2.6)
SA	175.1	204.5	223.3
ASS	81.6 (50.6)	39.5 (8.7)	40.2 (9.0)
HSSL	85.3 (54.4)	50.6 (19.6)	39.0 (8.0)
SHSSL	75.4	30.8	12.7
CPL-AdS	117.9 (109.3)	116.7 (107.1)	144.9 (96.5)

Table 9: Average residuals for the different schemes when finding the root of the quadric function B under various noise levels. The root is $x^* = 0.9270$. The results are given in the format "average residuals (average residuals after sampling)" for each scheme. For CPL-AdS the sampling period is the estimation period (epoch 0) as defined by the scheme. The number of iterations per trial is 250.

Func C	Avg Regret. $\pi = 0.65$	Avg Regret. $\pi = 0.75$	Avg Regret $\pi = 0.85$
TS-SPL	**36.9**	**13.7**	**6.4**
TS-SPL-INF	52.8 (21.9)	38.8 (7.8)	34.9 (3.9)
PBS-M	49.6 (18.7)	39.2 (8.3)	34.3 (3.3)
SGBS-M	47.2 (16.2)	34.6 (3.6)	32.5 (1.6)
BZ-M	58.8 (27.9)	38.4 (7.4)	33.7 (2.7)
SA	149.0	178.0	185.0
ASS	54.3 (23.4)	39 (8.0)	33.5 (2.5)
HSSL	75.2 (44.3)	44.3 (13.4)	35.6 (4.6)
SHSSL	56.5	18.4	**6.4**
CPL-AdS	153.0 (101.6)	156.2 (103.7)	165.0 (109.6)

Table 10: Average residuals for the different schemes when finding the root of the sinusoidal function C under various noise levels. The root is $x^* = 0.8675$. The results are given in the format "average residuals (average residuals after sampling)" for each scheme. For CPL-AdS the sampling period is the estimation period (epoch 0) as defined by the scheme. The number of iterations per trial is 250.

converge towards the root x^*. A perhaps even stronger advantage of TS-SPL is that it can be applied to a wide range of functions without regards to the presence of local extrema. SA, on the other, only performs well for monotonic functions as exemplified in Table 8.

4. Conclusions and Further Work

In this paper, we investigated a novel reinforcement learning problem derived from the so-called "N-Door Puzzle". This puzzle has the fascinating property that it involves stochastic *compulsive liars*. Feedback is erroneous on average, systematically misleading the decision maker. This renders traditional reinforcement learning (RL) based approaches ineffective due to their dependency on *"on average"* correct feedback.

To solve the problem of deceptive feedback, we recast the problem as a challenging variant of the multi-armed bandit problem, referred to as the *Stochastic Point Location* (SPL) problem. In SPL, the decision maker is only told whether the optimal point on a line lies to the "left" or to the "right" of a current guess, with the feedback being erroneous with probability $1-\pi$. Solving this problem opens up for optimization in continuous action spaces with both *informative* and *deceptive* feedback.

Our solution to the above problem, introduced in the present paper, is based on a novel Bayesian representation of the solution space that is both compact and scalable. This model simultaneously captures both the location of the optimal point, as well as the probability of receiving correct feedback π. We further introduced an accompanying Thompson Sampling (TS) guided Stochastic Point Location (TS-SPL) scheme for balancing exploration against exploitation. By learning π, TS-SPL supports deceptive environments that are lying about the direction of the optimal point.

We used TS-SPL to solve the Stochastic Point Location (SPL) problem and outperformed all of the Learning Automata driven methods. However, by enhancing the Soft Generalized Binary Search (SGBS) scheme with our Bayesian representation of the solution space, SGBS was able to outperform TS-SPL under informative feedback. For deceptive SPL problems, TS-SPL outperformed all of the existing state-of-art schemes by several orders of magnitude, even when the latter schemes were supported by our Bayesian model.

We also applied TS-SPL to the *Stochastic Root Finding* (SRF) problem. We further demonstrated that SRF can be seen as a deceptive problem, allowing TS-SPL to outperform existing dedicated state-of-art SRF schemes by an order of magnitude. Thus, TS-SPL can be considered state-of-the-art for both deceptive SPL and for SRF, while yielding comparable results to the top performing schemes in the case of informative SPLs.

Despite the above performance gains, TS-SPL is based on Thompson Sampling, which is known to have a tendency to over-explore high variance reward distributions (Lattimore, 2015). In future work, it is therefore interesting to investigate mechanisms that eliminate or reduce this tendency, to further increase convergence speed.

Another important avenue for future work is the establishment of theoretical results, including proof of convergence, to corroborate the purely empirical findings presented in this paper. We suggest that a promising starting point for such an endeavour would be to combine the theoretical properties of TS (Agrawal and Goyal, 2012, 2013a; Dong and Van Roy, 2018) with the theoretical results of PBS (Waeber et al., 2013; Frazier et al., 2019), as they are closely related.

References

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Proceedings of the 16th Conference on Artificial Intelligence and Statistics*, pages 99–107, 2013a.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 127–135, 2013b.

Soren Asmussen and Peter W Glynn. *Stochastic simulation: Algorithms and Analysis*, volume 57. Springer Science & Business Media, 2007.

Onur Atan, Cem Tekin, and Mihaela van der Schaar. Global multi-armed bandits with hölder continuity. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pages 28–36, 2015.

Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *Conference on Learning Theory*, pages 13–p, 2010.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.

Marat Valievich Burnashev and Kamil'Shamil'evich Zigangirov. An interval estimation problem for controlled observations. *Problemy Peredachi Informatsii*, 10(3):51–61, 1974.

Rui M Castro and Robert D Nowak. Upper and lower error bounds for active learning. In *In Proceedings of the 44th Conference on Communication, Control and Computing*, volume 2, page 1, 2006.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Proceedings of the Advances in Neural Information Processing Systems 24*, pages 2249–2257. Curran Associates, Inc., 2011.

Huifen Chen and Bruce W Schmeiser. Stochastic root finding via retrospective approximation. *IIE Transactions*, 33(3):259–275, 2001.

Shi Dong and Benjamin Van Roy. An information-theoretic analysis for thompson sampling with many actions. In *Proceedings of the Advances in Neural Information Processing Systems 31*, pages 4157–4165, 2018.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7(Jun):1079–1105, 2006.

Peter I Frazier, Shane G Henderson, and Rolf Waeber. Probabilistic bisection converges almost as quickly as stochastic approximation. *Mathematics of Operations Research*, 2019.

Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Sébastien Bubeck. Multi-bandit best arm identification. In *Proceedings of the Advances in Neural Information Processing Systems 24*, pages 2222–2230, 2011.

Sondre Glimsdal and Ole-Christoffer Granmo. Gaussian process based optimistic knapsack sampling with applications to stochastic resource allocation. In *Proceedings of the 24th Midwest Artificial Intelligence and Cognitive Science Conference 2013*, pages 43–50. CEUR Workshop Proceedings, 2013.

Ole-Christoffer Granmo. Solving two-armed bernoulli bandit problems using a bayesian learning automaton. *International Journal of Intelligent Computing and Cybernetics*, 3 (2):207–234, 2010.

Ole-Christoffer Granmo and Sondre Glimsdal. Accelerated bayesian learning for decentralized two-armed bandit based decision making with applications to the goore game. *Applied intelligence*, 38(4):479–488, 2013.

Ole-Christoffer Granmo, B John Oommen, Svein Arild Myrer, and Morten Goodwin 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*, 37(1):166–175, 2007.

Michael Horstein. Sequential transmission using noiseless feedback. *IEEE Transactions on Information Theory*, 9(3):136–143, 1963.

Laurent Hyafil and Ronald L Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1):15–17, 1976.

Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, volume 35, pages 423–439, 2014.

Y Yu Jia and Shie Mannor. Unimodal bandits. In *International Conference on Machine Learning*, pages 41–48, 2011.

Lei Jiao, Xuan Zhang, B. John Oommen, and Ole-Christoffer Granmo. Optimizing channel selection for cognitive radio networks using a distributed bayesian learning automata-based approach. *Applied Intelligence*, 44(2):307–321, 2016. ISSN 1573-7497.

Zohar Shay Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. *International Conference on Machine Learning*, 28:1238–1246, 2013.

Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

HJ Kushner and G Yin. Stochastic approximation algorithms for parallel and distributed processing. *Stochastics: An International Journal of Probability and Stochastic Processes*, 22(3-4):219–250, 1987.

Tze Leung Lai. Stochastic approximation. *Annals of Statistics*, pages 391–406, 2003.

Tor Lattimore. Optimally confident ucb: Improved regret for finite-armed bandits. *arXiv preprint arXiv:1507.07880*, 2015.

Robert Nowak. Generalized binary search. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 568–574. IEEE, 2008.

Robert D Nowak. The geometry of generalized binary search. *Information Theory, IEEE Transactions on*, 57(12):7893–7906, 2011.

B John Oommen. Stochastic Searching on the Line and its Applications to Parameter Learning in Nonlinear Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 27(4):733–739, 1997.

B John Oommen, Govindachari Raghunath, and Benjamin Kuipers. On how to learn from a stochastic teacher or a stochastic compulsive liar of unknown identity. In *AI 2003: Advances in Artificial Intelligence*, pages 24–40. Springer, 2003.

B John Oommen, Sudip Misra, and Ole-Christoffer Granmo. Routing bandwidth-guaranteed paths in mpls traffic engineering: A multiple race track learning approach. *IEEE Transactions on Computers*, 56(7):959–976, 2007.

B John Oommen, Ole-Christoffer Granmo, and Zuoyuan Liang. A novel multidimensional scaling technique for mapping word-of-mouth discussions. In *Opportunities and Challenges for Next-Generation Applied Intelligence*, pages 317–322. Springer, 2009.

Stephen Pallone, Peter I Frazier, and Shane G Henderson. Multisection: Parallelized bisection. In *Simulation Conference, Winter*, pages 3773–3784. IEEE, 2014.

Raghu Pasupathy and Sujin Kim. The stochastic root-finding problem: Overview, solutions, and open questions. *ACM Transactions on Modeling and Computer Simulation*, 21(3): 19, 2011.

Raghu Pasupathy and Bruce W Schmeiser. Root finding via darts – dynamic adaptive random target shooting. In *Simulation Conference (WSC), Proceedings of the 2010 Winter*, pages 1255–1262. IEEE, 2010.

Andrzej Pelc. Searching games with errors – fifty years of coping with liars. *Theoretical Computer Science*, 270(1):71–109, 2002.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Aarti Singh, Robert Nowak, and Parmesh Ramanathan. Active learning for adaptive mobile sensing networks. In *In Proceedings of the 5th international conference on Information processing in sensor networks*, pages 60–68. ACM, 2006.

Raymond Smullyan. *To Mock a Mockingbird and Other Logic Puzzles: Including an Amazing Adventure in Combinatory Logic*. Knopf, 1988. ISBN 0-19-280142-2.

Tongtong Tao, Hao Ge, Guixian Cai, and Shenghong Li. Adaptive step searching for solving stochastic point location problem. In *Intelligent Computing Theories*, pages 192–198. Springer, 2013.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

David Tolpin and Frank Wood. Maximum a posteriori estimation by search in probabilistic programs. In *Proceedings of the 8th Annual Symposium on Combinatorial Search*, 2015.

Rolf Waeber, Peter Frazier, and Shane G Henderson. A bayesian approach to stochastic root finding. In *Proceedings of the 2011 Winter Simulation Conference*, pages 4033–4045. IEEE, 2011.

Rolf Waeber, Peter I Frazier, and Shane G Henderson. Bisection search with noisy responses. *SIAM Journal on Control and Optimization*, 51(3):2261–2279, 2013.

Anis Yazidi, Ole-Christoffer Granmo, B John Oommen, and Morten Goodwin. A hierarchical learning scheme for solving the stochastic point location problem. In *Advanced Research in Applied Artificial Intelligence*, pages 774–783. Springer, 2012a.

Anis Yazidi, B John Oommen, and Ole-Christoffer Granmo. A novel stochastic discretized weak estimator operating in non-stationary environments. In *Proceedings of the International Conference on Computing, Networking and Communications*, pages 364–370. IEEE, 2012b.

Junqi Zhang, Liang Zhang, and Mengchu Zhou. Solving stationary and stochastic point location problem with optimal computing budget allocation. In *Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 145–150, Oct 2015. doi: 10.1109/SMC.2015.38.

Junqi Zhang, Yuheng Wang, Cheng Wang, and Mengchu Zhou. Symmetrical hierarchical stochastic searching on the line in informative and deceptive environments. *IEEE Transactions on Cybernetics*, PP(99):1–10, 2016. ISSN 2168-2267. doi: 10.1109/TCYB.2016.2521859.

Appendix C

Thompson Sampling Guided Stochastic Searching on the Line for Non-stationary Adversarial Learning

Appendix D

Accelerated Bayesian learning for decentralized two-armed bandit based decision making with applications to the Goore Game

Appendix E

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

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, σ_f^2 is the signal variance and σ_n^2 is white noise (note that δ_{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, 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 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.

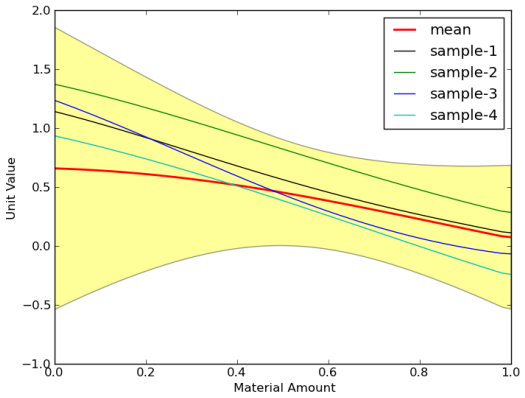

Figure 1: Gaussian Process based representation of material unit value

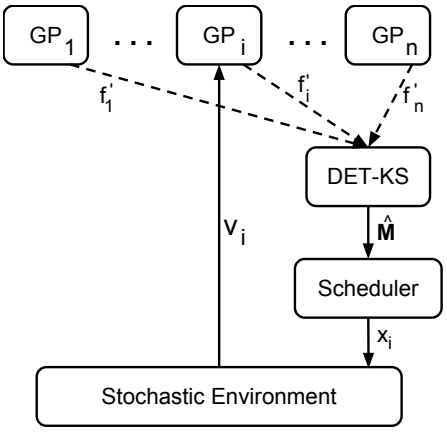

Figure 2: GPOKS Architectural Overview

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 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

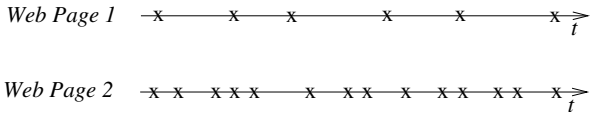

Figure 6: Web resource changes occurring over time. An 'x' on the time-lines denotes that the respective web resource has changed.

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

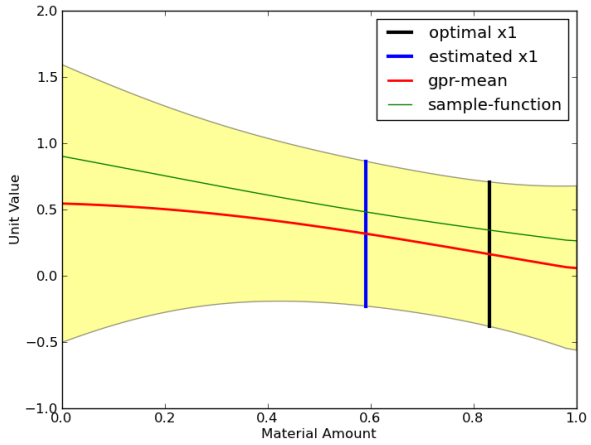

Figure 3: Estimate of material unit value $f'_1(x_1)$ after 7 observations, with optimal and estimated material amounts x_1.

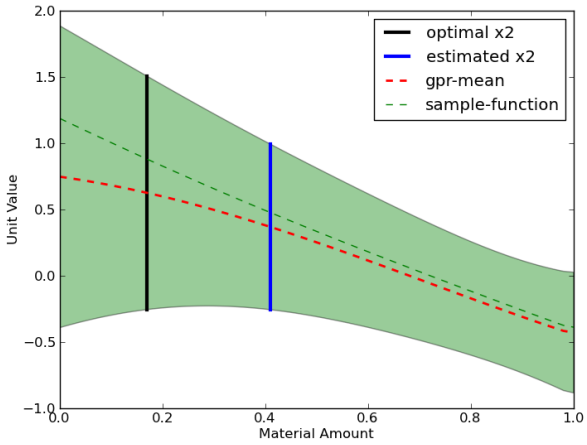

Figure 4: Estimate of material unit value $f'_2(x_2)$ after 7 observations, with optimal and estimated material amounts x_2.

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

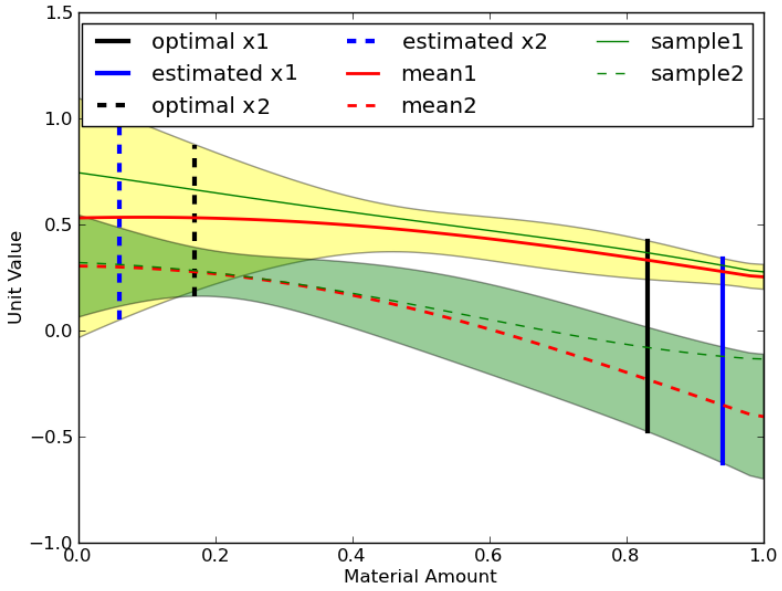

Figure 5: Estimate of material unit values $f_1'(x_1)$ and $f_2'(x_2)$ after 193 observations, with optimal and estimated material amounts x_1 and x_2.

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_σ), 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

Appendix F

Travel Time Estimation