Efficiently Gathering Information in Costly Domains

Shulamit Reches[1], Ya'akov (Kobi) Gal[2], and Sarit Kraus[3]

[1]Department of Applied Mathematics, Jerusalem College of Technology, Israel
[2]Department of Information Systems Engineering, Ben-Gurion University of the Negev, Israel
[3]Computer Science Department, Bar Ilan University, Israel

Abstract

This paper proposes a novel technique for allocating information gathering actions in settings where agents need to choose among several alternatives, each of which provides a stochastic outcome to the agent. Samples of these outcomes are available to agents prior to making decisions and obtaining further samples is associated with a cost. The paper formalizes the task of choosing the optimal sequence of information gathering actions in such settings and establishes it to be NP-Hard. It suggests a novel estimation technique for the optimal number of samples to obtain for each of the alternatives. The approach takes into account the trade-offs associated with using prior samples to choose the best alternative and paying to obtain additional samples. This technique is evaluated empirically in several different settings using real data. Results show that our approach was able to significantly outperform alternative algorithms from the literature for allocating information gathering actions in similar types of settings. These results demonstrate the efficacy of our approach as an efficient, tractable technique for deciding how to acquire information when agents make decisions under uncertain conditions.

1 Introduction

In many settings characterized by uncertainty agents can engage in information gathering actions before making decisions. For example, consider an e-commerce application in which a buyer needs to choose between several suppliers of a product or service (in absence of a built-in reputation system). The buyer does not know the quality of each of the suppliers in advance, but can spend time and resources to collect information about them. This information provides a "noisy signal" about the quality of the suppliers. An example of another setting— which comprises part of our empirical methodology— involves choosing one of several heuristic algorithms for solving an optimization problem. Each algorithm yields a solution that varies in computation time when applied to the problem. The agent needs to decide in advance which algorithm to choose in order to minimize the amount of expected computation time.

The focus of this paper is on settings where an agent needs to choose an alternative among a set of candidates with unknown outcomes. The agent can obtain samples of information about the different alternative candidates at a cost, prior to choosing one of them. A key facet of the settings we consider is that the agent needs to decide in advance about the amount of information to acquire about each alternative and cannot change this decision once it has chosen one of the candidates. This constraint occurs in many real-world scenarios, such as choosing the number of credit ratings to purchase about a customer before approving a requested loan, or the amount of time to spend obtaining information from referees about a potential job candidate. To succeed in such settings, it is necessary to reason about the trade-off between paying to acquire additional information about the alternative candidates and choosing the candidate that is deemed optimal based on the current available information.

The paper formalizes the task of information gathering under uncertainty as a stochastic optimization problem termed Optimal Allocation of Relevant Information (OARI) with the following characteristics: An agent must choose in advance how much information to obtain about each of a set of possible candidates prior to choosing one of them. Each of these candidates is associated with a reward sampled from a distribution that is not known to the agent. The agent is given a number of prior samples about each candidate that are drawn from its respective distribution. Obtaining additional information about each candidate provides an additional sample of its reward but is associated with a cost. The goal of the agent is to find the allocation of information gathering actions among the different alternatives that maximizes the agent's total reward while taking into account the cost of obtaining the additional information and the expected reward that is associated with the chosen alternative.

The paper establishes OARI to be an NP-Hard problem and presents a novel estimation technique for solving it called EURIKA (Estimating the Utility of Restricted Information among Alternatives). EURIKA estimates the agent's utility function by approximating the probability that it will prefer each of the candidates to all other candidates given the acquired information. It derives the optimal number of information gathering actions in polynomial time. EUREKA assumes the existence of a probability distribution over the possible alternatives, but makes no other assumptions about the domain.

The applicability of EURIKA was shown empirically by using it to make decisions on real-world data. Specifically, we evaluated EURIKA on a variety of settings that varied the type of task to optimize, the data obtained from the information gathering actions, and agents' utility functions. One of the settings required the agent to choose between various heuristic algorithms for solving 3-SAT problems while optimizing the number of problems solved and the amount of computation time. The candidate algorithms included existing 3-SAT approaches from the literature as well as the best-performing entries submitted by researchers to a SAT solver competition. Another setting involved choosing the best lecturer in order to maximize students' enrollment in a course, given that students' evaluations about lecturers can be obtained at a cost. The data for this domain was taken from real course enrollment data and evaluations submitted by college students.

In all of these domains, the performance of an agent using EURIKA was compared to alternative solutions from the literature as well as a baseline approach that never purchased any additional information. The results show that the agent using EURIKA was able to outperform both of these approaches in all of the domains. In particular, it was able to find the best alternative more often than the alternative approaches, and more efficiently, in that it acquired less or equal amounts of information to find the best alternative.

This paper revises and extends earlier work [Reches et al., 2007] and makes the following contributions. First, it formally defines the problem of optimal allocation of information gathering actions (OARI) and establishes it to be an NP-Hard problem. Second, it presents a novel heuristic for solving the OARI problem analytically by computing the estimated benefit for different information gathering actions while taking into account their associated costs. Third, it proves the efficacy of the technique empirically by applying it to several domains that include real-world data.

2 Related work

The OARI problem is related to several approaches for repeated decision-making under imperfect information. Azoulay-Schwartz and Kraus [2002] suggested a theoretical approach for optimizing the amount of information that is required in order to decide between two alternatives. A naive application of this model for multiple alternatives requires to examine all possible alternative pairs, which is infeasible for large settings, such as the ones considered in this paper. Our work extends their model to choosing the best out of multiple (more than two) alternatives using a tractable, analytical approach, and evaluates the model using real data. Talman et al. [2005] presented a model which decides the amount of information the agent should obtain based on a set of prior samples of each alternative. This technique used a fixed number of samples to distribute among the various alternatives in a way that is proportional to the quality of the prior samples. Our empirical work shows that EURIKA significantly outperforms the FNE model in all domains we considered.

In the *Max K-Armed Bandit* problem, an agent allocates trials to slot machines, each yielding a payoff from a fixed (but unknown) distribution [Cicirello and Smith, 2005, Streeter and Smith, 2006]. The objective is to allocate trials among the K arms to maximize the expected best single sample reward.[1] This problem is analogous to the OARI problem in that each trial provides an information sample about one of the alternatives. However, we do not assume that the number of trials is determined in advance, but optimize this number given the uncertainty over the different alternatives. Second, in contrast to the K-Armed Bandit problem, we allow the agent to have prior knowledge about each alternative. As we show in the empirical section, this knowledge may lead the agent to decide not to allocate additional trials because they are not expected to change its choice.

[1] This problem is distinct from the traditional K-Armed Bandit problem [Berry and Fristedt, 1985] in which the agent maximizes the cumulative, not maximal, payoffs.

Our work relates to the value of information problem which has received much attention in the Artificial Intelligence literature [Radovilsky and Shimony, 2008, Chick et al., 2010, Heckerman et al., 1993]. Within this body of work, there are several approaches that are relevant to our study. Guestrin and Krause [2009] and Krause and Guestrin [2005, 2007] suggest several models for selecting which variables to sample in a Bayesian network to minimize uncertainty in the network. Bilgic and Getoor [2007] use graphical models to minimize the cost of the acquisition of information for the purpose of making predictions about variables of interest to the agent. Heckerman et al. [1993] provide an approximation algorithm for computing the next piece of evidence to choose to observe given the results of all possible samples of the variables. They provide an approximate algorithm that is limited to specific classes of dependencies between variables, and the agent makes a binary decision whether to sample each variable. In contrast, this paper is concerned with cases in which alternatives are independent of each other and an agent can decide how may information samples to purchase about each alternative, rather than once. Such settings characterize many real world information gathering problems, as we demonstrate in the Empirical Methodology section.

Our work differs from sequential models that choose which information source to query after each decision based on the information that was observed so far. Notable examples of these works include Grass and Zilberstein [2000], who proposed a decision theoretic approach for planing and executing information-gathering actions over time, and Madani et al. [2004], who presented a model in which a learner has to identify which of a given set of possible classifiers has the highest expected accuracy. In contrast to our work, both of these works assume that obtaining the value of the various alternatives is not associated with a cost. Thus they don't consider the trade-off that arises when deciding to acquire new information or to make a choice based on the available information. Tseng and Gmytrasiewicz [2002] developed an information-gathering system that suggests to a user how to best retrieve information related to the user's decisions. Madigan and Almond [1996] propose a myopic model of value of information that samples variables iteratively. All of these techniques do not reason about the effect of choosing one information source over another on an agent's utility. In our setting, the agent chooses the amount of information to acquire in advance, and cannot choose a different candidate once it has made its decisions. Our work is further distinct in that we evaluate our approach empirically, showing that it generalizes to several domains.

Lastly, Conitzer and Sandholm [1998] formalized several "meta-reasoning" problems in which agents collect information prior to making decisions. One of these problems involves an agent that optimizes which set of anytime algorithms to use for different problem instances. The OARI problem is distinct from this problem in that the allocation of information is measured in integer numbers (i.e., units of information) rather than real values (i.e., time). We show that the OARI problem is at least as hard as this problem. In addition, we provide a tractable solution algorithm for solving the OARI problem in practice and demonstrate the efficacy of the algorithm on real data.

3 An Optimal Allocation of Information Problem

We define the problem of optimally allocating information gathering actions as follows: A risk neutral agent has to choose an alternative from a set of K independent alternatives denoted $A = \{a_1, \ldots, a_K\}$. The reward for each alternative a_i, denoted R_i, is normally distributed $R_i \sim N(\mu_i, \sigma_i^2)$, with an unknown mean μ_i and variance σ_i^2. The mean of the reward μ_i is normally distributed $\mu_i \sim N(\zeta, \tau)$ for each $i \in \{1, \ldots, K\}$, with known mean ζ and standard deviation τ. A sample r_i is a set of $n_i \geq 0$ draws of the reward R_i. The average of each sample r_i is denoted $\overline{r_i}$.

We assume that an agent has collected prior information about the alternatives consisting of samples $D = \{(r_1, n_1), \ldots, (r_k, n_k)\}$. The agent can decide to obtain an additional sample r'_i consisting of n'_i draws of the reward associated with alternative a_i. The additional samples are denoted $D' = \{(r'_1, n'_1), \ldots, (r'_k, n'_k)\}$.

Obtaining this information is associated with a cost and the number of possible samples that the agent can obtain is bounded by an integer $M > 0$, such that $\sum_{i=1}^{K} n'_i \leq M$. The goal of the agent is to find the optimal allocation of samples for maximizing its reward given its chosen alternative and the information cost. Table 1 presents the notations we use for the description of the model.

The *benefit* to the agent is the difference between its utility from acquiring additional samples D', and solely using its prior information D.[2] We denote

[2]This benefit depends on the agent's utility function which can be defined separately

Parameter	Description
$A = \{a_1, ..., a_K\}$	Set A is a set of the K alternatives.
n_i	The number of prior units of information about alternative a_i.
R_i	Unknown reward of alternative a_i
$\overline{r_i}$	The mean reward of n_i prior information units about alternative a_i.
n'_i	The number of units of information acquired about alternative a_i.
$\overline{r'_i}$	The mean reward of n'_i information units about a_i
μ_i	The mean of the reward of alternative a_i
σ_i	The standard deviation of the reward of alternative a_i
ζ	The mean of the random variable μ_i
τ	The variance of the random variable μ_i
$Cost(a_i)$	The cost of one unit of information about alternative a_i.
$D = \{(\overline{r_1}, n_1), \ldots, (\overline{r_k}, n_K)\}$	Prior information about alternatives in A.
$D' = \{(\overline{r'_1}, n'_1), \ldots, (\overline{r'_k}, n'_K)\}$	Additional information that is acquired about alternatives in A.

Table 1: Summary of notation used in the paper

the benefit as a function $B(A, D, n'_1, \ldots, n'_K, \zeta, \tau, \sigma_1, \ldots, \sigma_K)$, that inputs a set of alternatives A, the distribution parameters ζ, τ, σ_i for $1 \leq i \leq K$, the prior information D about each alternative and the allocation $n'_1, \ldots, n'_K$ of the additional information units about each alternative $a_i \in A$. The function returns the benefit from this allocation. For the remainder of this paper, we will use an abbreviated notation, $B(n'_1, \ldots, n'_K)$.

The function $Cost : A \rightarrow R$ denotes the cost of obtaining one unit of information about alternative $a_i \in A$. The *total profit* to the agent is the difference between the benefit from obtaining the additional information and its cost. This profit is a function $T(A, D, n'_1, \ldots, n'_K, \zeta, \tau, \sigma_1, \ldots, \sigma_K, Cost)$ which receives a set of alternatives A, the prior information D about each alternative, the distribution parameters ζ, τ, σ_i for $1 \leq i \leq K$, the $Cost$ function and an allocation $n'_1, \ldots, n'_K$ of the additional information units about each alternative $a_i \in A$. It computes the total profit to the agent from obtaining $(n'_1, \ldots, n'_K)$ additional information units, while taking the costs into consideration. For the remainder of this paper we will use the

for each domain, as we show in Section 4.

abbreviated notation $T(n'_1, \ldots, n'_K)$ to denote this function.

$$T(n'_1, \ldots, n'_K) = B(n'_1, \ldots, n'_K) - \sum_{i=1}^{K} n'_i \cdot Cost(a_i) \tag{1}$$

We can now formally define the *Optimal Allocation of Request Information (OARI)* problem as follows:

Definition 1. (Optimal Allocation of Request Information (OARI)) *Given a set of alternatives A, prior information D, the parameters ζ, τ, σ_i for each alternative $1 \leq i \leq K$, the bound on the number of information units M, and the $Cost$ function. The OARI problem requires to find a vector $(n^*_1, \ldots, n^*_K) \in N^K, \sum_{i=1}^{K} n^*_i \leq M$ that maximizes the total profit $T(n'_1, \ldots, n'_K)$ of the agent.*

This problem can be formulated as a decision problem as follows: Given a set of alternatives A, prior information D, the parameters ζ, τ, σ_i for each alternative $1 \leq i \leq K$, a bound on the number of information units M, and a threshold L as follows: Answer "yes" if there exists a vector $(n^*_1, \ldots, n^*_K), \sum_{i=1}^{K} n^*_K \leq M$ such that $T(n^*_1, \ldots, n^*_K) \geq L$.

Theorem 1. *OARI is an NP-hard problem.*

The proof of this theorem, via a reduction from the Knapsack problem [Garey and Johnson, 1979], is given in the Appendix.

3.1 The EURIKA Model

This section presents a model for solving the OARI problem, termed EURIKA (Estimating Utility of Restricted Information among K Alternatives). The model reasons about the trade off between exploration and exploitation when choosing among multiple alternatives. It outputs the allocation of information gathering actions among the different alternatives, taking into account the prior sample of the rewards, the additional information that is obtained, and the cost of obtaining this information.

The agent chooses its default alternative based solely on its prior information. Acquiring additional information about any of the alternatives worthwhile to the agent only if it leads the agent to change its choice. Suppose the agent has already collected n'_i units of information about a_i and that the mean reward associated with this sample, denoted $\overline{r'_i}$, is known (we will drop this assumption later). Now, how should the agent make a decision about whether it prefers alternative a_i to a_j? If the agent only considers the available information, its decision solely depends on the probability that the weighted mean reward of the information obtained about a_i is greater than that of the information obtained about a_j. In addition, the agent may also consider prior information about the distribution of these rewards. The following defines the probability that the agent will prefer alternative a_i over any alternative a_j after obtaining n'_j information units about a_j.

Definition 2. *The term $PC_i(n'_j \mid n'_i, \overline{r'_i})$ is the probability that the agent will prefer alternative a_i to alternative a_j as a result of n'_i and n'_j additional information units about a_i and a_j, given the sample r'_i.*[3]

Next, we will define the probability that a_i is the *best* alternative, that is, the agent chooses alternative a_i over all other alternatives as a result of obtaining additional information units the other alternatives.

Definition 3. *The term $PB_i(n'_1, \ldots, n'_{i-1}, n'_{i+1}, \ldots, n'_K \mid n'_i, \overline{r'_i})$ is the probability that the agent will prefer alternative a_i to all other alternatives as a result of obtaining $(n'_1, \ldots, n'_{i-1}, n'_{i+1}, \ldots, n'_K)$ additional information units.*
[4]

The following proposition states that the probability $PB_i(n'_1, \ldots, n'_{i-1}, n'_{i+1}, \ldots, n'_K \mid n'_i, \overline{r'_i})$ can be computed as the product of the probabilities that the agent prefers a_i to each other alternative a_j given the sample r'_i.

Proposition 1.

$$PB_i(n'_1, \ldots, n'_{i-1}, n'_{i+1}, \ldots, n'_K \mid n'_i, \overline{r'_i}) = \prod_{j \neq i} PC_i(n'_j \mid n'_i, \overline{r'_i}) \tag{2}$$

The proof is immediate, as the probability that the agent prefers a_i to any alternative a_j is independent of a_i when the sample mean $\overline{a_i}$ is known. Now, because the true mean sample reward $\overline{r'_i}$ is unknown, we sum over each

[3]Note that this probability also depends on the parameters n_i, n_j, μ_i, μ_j, and that the notation $PC_i(n'_j \mid n'_i, \overline{r'_i})$ is thus an abbreviation with reduced parameters of the term $PC_i(n_i, n_j, n'_i, n'_j, \overline{r_i}, \overline{r_j}, \mu_i, \mu_j, \overline{r'_i})$

[4]The notation $PB_i(n'_1, \ldots, n'_{i-1}, n'_{i+1}, \ldots, n'_K \mid n'_i, \overline{r'_i})$ is an abbreviation with reduced parameters of the term $PB_i(D, n'_1, \ldots, n'_K, \sigma_1, \ldots, \sigma_K, \mu_1, \ldots, \mu_K)$

of its possible values, and obtain the term $PB_i(n'_1, \ldots, n'_K)$, which is the probability that the alternative a_i is preferred to all other alternatives.

$$PB_i(n'_1, \ldots, n'_K) = \int_{\overline{r'_i}} \prod_{j \neq i} PB_i(n'_1, \ldots, n'_{i-1}, n'_{i+1}, \ldots, n'_K \mid n'_i, \overline{r'_i}) \cdot P(\overline{r'_i})\, d\overline{r'_i} \tag{3}$$

Here, the term $P(\overline{r'_i})$ is the probability that the mean reward from obtaining n'_i units of information of alternative a_i is $\overline{r_i}$.

3.2 Computing the Expected Benefit of Acquiring Information

In this section we show how to compute the benefit $B(n'_1, \ldots, n'_K)$ from obtaining the sample $(n'_1, \ldots, n'_K)$. Without loss of generality, suppose alternative a_1 is currently the best alternative given the prior information D. We distinguish between the following two cases:

1. The agent does not obtain additional information. The expected reward in this case is the mean reward μ_1 of the current best alternative a_1.

2. The agent decides to obtain $(n'_1, \ldots, n'_K)$ additional information units about alternatives $a_1, \ldots, a_k$. In this case there are two possibilities:

   - The agent decides not to change its initial decision a_1 based on the $(n'_1, \ldots, n'_K)$ additional information. The expected reward in this case is $PB_1(n'_1, \ldots, n'_K) \cdot \mu_1$.

   - The agent prefers some alternative $a_i, i \neq 1$ to a_1 based on the $(n'_1, \ldots, n'_K)$ additional information. The expected reward in this case is $\sum_{i=2}^{K} PB_i(n'_1, \ldots, n'_K) \cdot \mu_i$.

We can now compute the benefit from obtaining $(n'_1, \ldots, n'_K)$ additional samples as the difference between the expected reward from obtaining and not obtaining this information. This benefit is denoted $B(n'_1, \ldots, n'_K \mid \mu_1, \ldots, \mu_K)$ and computed as

$$B(n'_1, \ldots, n'_K \mid \mu_1, \ldots, \mu_K) = PB_1(n'_1, \ldots, n'_K) \cdot \mu_1 + \sum_{i=2}^{K} PB_i(n'_1, \ldots, n'_K) \cdot \mu_i - \mu_1 \tag{4}$$

Because we have that

$$PB_1(n'_1, \ldots, n'_K) = 1 - \sum_{i=2}^{k} PB_i(n'_1, \ldots, n'_K) \tag{5}$$

we can write the expected profit as

$$B(n'_1, \ldots, n'_K \mid \mu_1, \ldots, \mu_K) = \sum_{i=2}^{K} PB_i(n'_1, \ldots, n'_K) \cdot (\mu_i - \mu_1) \tag{6}$$

The above equation depends on the reward means $\{\mu_1, \ldots, \mu_K\}$, which are unknown. Therefore we need to integrate over their possible values. We use the posterior distribution $P(\mu_i \mid D, D')$ to combine the prior information D, the acquired samples D', and parameters $\zeta, \tau, \sigma_1, \ldots, \sigma_K$. The posterior distribution over μ_i can be computed in closed form. Because μ_i is a conjugate prior to the normal distribution, its posterior is a normal distribution with mean $\frac{\sigma_i^2\zeta + n_i\tau^2\overline{r_i}}{\sigma_i^2 + n_i\tau^2}$ and variance $\frac{\sigma_i^2\tau^2}{\sigma_i^2 + n_i\tau^2}$. Considering all possible values of μ_i, we attain the following proposition.

Proposition 2.

$$\begin{aligned} B(n'_1, \ldots, n'_K) = \int_{\mu_1} \cdots \int_{\mu_K} \sum_{i=2}^{K} PB_i(n'_1, \ldots, n'_K) \cdot (\mu_i - \mu_1) \cdot \\ \prod_{i=1}^{K} Pr(\mu_i \mid D, D') d\mu_1 \ldots d\mu_K \end{aligned} \tag{7}$$

The agent's expected benefit gained from obtaining $(n'_1, \ldots, n'_K)$ additional units of information, while considering the various costs involved in obtaining $\sum_{i=1}^{K} n'_i$ units of information is described in Equation 1, which we restate here for convenience.

$$T(n'_1, \ldots, n'_K) = B(n'_1, \ldots, n'_K) - \sum_{i=1}^{K} n'_i \cdot Cost(a_i)$$

The solution to the OARI problem is a vector $(n^*_1, \ldots, n^*_K)$ that maximizes the above function, such that $\sum_1^K n^*_i < M$. This computation is exponential in the number of possible alternatives to consider.[5] An alternative approach

[5] A brute-force implementation of this computation on the domains we consider in our empirical methodology took three days of computation on a commodity desktop.

is to maximize the function analytically, using several approximation which we describe in the next section.

We assume that the agent bases its decision solely on its observations. These include the prior information and the acquired samples about the various alternatives. This means that for any two alternatives a_i and a_j and samples r'_i and r'_j, the agent prefers a_i over a_j if the following holds:

$$\frac{n_i\overline{r_i} + n'_i\overline{r'_i}}{n_i + n'_i} > \frac{n_j\overline{r_j} + n'_j\overline{r'_j}}{n_j + n'_j} \tag{8}$$

The following proposition states that the probability that the agent changes its alternative can be computed using the normal distribution.

Proposition 3. *Given $n_i, n_j, \overline{r_i}, \overline{r_j}, n'_i, n'_j, \sigma_i, \sigma_j$ and $\overline{r'}_i$ the value of $PC_i(n'_i, n'_j, \overline{r'_i})$, is as follows:*

$$PC(n'_i, n'_j, \overline{r'_i}) = Pr(Z < Z_\alpha(n'_i, n'_j, \overline{r'_i})) \tag{9}$$

Here, Z *is a random variable, with a standard normal distribution, $Pr(Z < Z_\alpha(n'_i, n'_j, \overline{r'_i}))$ is the probability that the random variable Z will have a value less than $Z_\alpha(n'_i, n'_j, \overline{r'_i})$, and*

$$Z_\alpha(n'_i, n'_j, \overline{r'_i}) = \frac{\sqrt{n_j}((n_j + n'_j)(n_i\overline{r_i} + n'_i, r'_i) - (n_i + n'_i)(n_j\overline{r_i} - \mu_j, n'_j))}{n'_j(n_i + n'_i)\sigma_j}$$

The proof is in the Appendix. Thus we can write

$$PC_i(n'_j \mid n'_i, \overline{r'_i}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Z_\alpha(n'_i, n'_j, r'_i)} e^{\frac{-t^2}{2}} \, dt \tag{10}$$

3.3 Approximations

In order to compute $PC_i(n'_j \mid n'_i, \overline{r'_i})$ as a function of the parameters n'_i, n'_j we use the following approximation. The following proposition uses a summation of polynomials to compute the integration in Equation 10.

Proposition 4. *The function*

$$P\_approx(x) = \begin{cases} 0.5 - \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{n} \frac{x^{2k+1}(-1)^k}{2^k(2k+1)k!} & |x| < d \\ 0 & x \geq d \\ 1 & x \leq -d \end{cases}$$

is an approximation of the function $F(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-t^2}{2}}\ dt$*. Here, d is a positive integer that binds the error* $R_n = |F(x) - P\_approx(x)|$ *as follows. If* $|x| < d$*, then* $R_n \leq \frac{d^{n+1}}{(n+1)!}$*. Otherwise,* $R_n \leq 0.5 - \frac{1}{\sqrt{2\pi}} \int_0^{|d|} e^{\frac{-t^2}{2}} dt$*.*

The proof is in the Appendix. According to Proposition 4, we can use P_approx to approximate the density of a standard normal probability function $F(x)$ within the bounds of $|d|$ as a piecewise integrable function. The size of the approximation error depends on d and n. By Equation 10 we get that $1 - F(Z_\alpha(n'_i, n'_j)) = PC_i(n'_j \mid n'_i, \overline{r'_i})$. We use Proposition 4, where $x = Z_\alpha(n'_i, n'_j, r'_i)$, to approximate $PC_i(n'_j \mid n'_i, \overline{r'_i})$.

We now place $PC_i(n'_j \mid n'_i, \overline{r'_i})$ in Equation 2 to compute $PB_i(n'_1, \ldots, n'_{i-1}, n'_{i+1}, \ldots, n'_K \mid n'_i, \overline{r'_i})$ (the probability that a_i is the best alternative); place $PB_i(n'_1, \ldots, n'_{i-1}, n'_{i+1}, \ldots, n'_K \mid n'_i, \overline{r'_i})$ in Equation 7 to compute B (the expected profit), and place B in Equation 1 to compute T (the expected benefit). Finally, we can find the vector $(n^*_1, \ldots, n^*_K)$ that maximizes T numerically using the Simplex algorithm [Nelder and Mead, 1965].

4 Experimental Design and Analysis

In this section, we provide an extensive evaluation of the EURIKA model in settings that include synthetic as well as ecologically realistic data. These settings differ in the way the samples determine how agents incur utilities or costs. Therefore we adapted separate B and T functions for each of the settings.

Two of the domains involve choosing algorithms for solving 3-SAT formulas. The candidate algorithms consisted of heuristic approaches from the SAT literature as well as algorithms submitted by researchers to a competition for solving SAT formulas. A third domain involves choosing lecturers for courses from among different student evaluations. We describe each of the domains, and show how to adapt the OARI model to the domain. We then show how we use the EURIKA model to find the optimal number of samples to obtain for each alternative. We compare the performance of the EURIKA model to several candidate models. Although the OARI formalism assumes that populations are normally distributed, our empirical results show that our approach can also be applied towards populations that may not adhere to this assumption.

4.1 The SAT Simulation domain

In the SAT Simulation domain, an agent is given a set of 3-SAT formulas to satisfy. Each alternative a_i represents a heuristic algorithm for solving 3-SAT formulas. Changing the assignment of an attribute in a formula, referred to as a "flip" operation, costs one unit of computation time. Each candidate algorithm a_i solves a 3-SAT formula using an expected number of μ_i flip operations. In this domain, we use μ_i to refer to costs rather than rewards as originally formalized. A sample of n'_i applications of a_i generates $n'_i \cdot \mu_i$ expected flips and solves n'_i 3-SAT formulas. There are two possible configurations in this domain: In the first, the agent must minimize the amount of computation time to solve a given set of formulas. In the second, the agent must maximize the number of formulas it solves for a fixed amount of computation time.

4.1.1 Minimal Time (MT) Configuration

In this configuration, the objective is to solve N formulas using the least amount of computation time. Suppose that, without loss of generality, algorithm a_1 is the best alternative given the prior information D. If the agent chooses this algorithm, the expected number of flip operations for solving N formulas is $N \cdot \mu_1$. Now, obtaining $(n'_1, \ldots, n'_K)$ information units of the various algorithms solves $\sum_{i=1}^{K} n'_i$ 3-SAT formulas at an expected cost of $\sum_{i=1}^{K} n'_i \cdot \mu_i$ flip operations. Suppose the agent decides to obtain additional information. In this case there are two possibilities:

- The agent will choose to continue to use algorithm a_1 to solve the remaining $(N - \sum_{i=1}^{K} n'_i)$ flip operations. In this case the expected number of flip operations is

$$PB_1(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_1$$

- The agent will choose alternative $a_j, j \neq 1$ to solve the remaining $(N - \sum_{i=1}^{K} n'_i)$ flip operations. In this case the expected number of flip operations is

$$\sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_j$$

The expected number of flip operations for obtaining $(n'_1, \ldots, n'_K)$ information units, denoted $EF(n'_1, \ldots, n'_K)$ sums over these two cases

$$EF(n'_1, \ldots, n'_K) = \sum_{j=1}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_j \tag{11}$$

The expected benefit for obtaining $(n'_1, \ldots, n'_K)$ information units is the difference between the expected number of flips before and after obtaining the additional information.

$$B(n'_1, \ldots, n'_K) = (N \cdot \mu_1) - EF(n'_1, \ldots, n'_K) \tag{12}$$

The cost function in this domain assigns one unit of computation time to each flip operation. Therefore the expected profit for obtaining $(n'_1, \ldots, n'_K)$ information units, defined in Equation 1, is computed as

$$\begin{aligned} T(n'_1, \ldots, n'_K) &= B(n'_1, \ldots, n'_K) - \sum_{i=1}^{K} n'_i \cdot \mu_i \\ &= (N \cdot \mu_1) - EF(n'_1, \ldots, n'_K) - \sum_{i=1}^{K} n'_i \cdot \mu_i \\ &= (N \cdot \mu_1) - \sum_{j=1}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_j - \sum_{i=1}^{K} n'_i \cdot \mu_i \\ &= (N \cdot \mu_1) - (PB_1(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_1 + \\ &\sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_j) - \sum_{i=1}^{K} n'_i \cdot \mu_i \end{aligned} \tag{13}$$

Since $PB_1(n'_1,\ldots,n'_K) = 1 - \sum_{j=2}^{K} PB_j(n'_1,\ldots,n'_K)$ the above equation can be written as

$$
\begin{aligned}
T(n'_1,\ldots,n'_K) = (N\cdot\mu_1) - \Big((1 - \sum_{j=2}^{K} PB_j(n'_1,\ldots,n'_K)\cdot(N-\sum_{i=1}^{K} n'_i)\cdot\mu_1 + \\
\sum_{j=2}^{K} PB_j(n'_1,\ldots,n'_K)\cdot(N-\sum_{i=1}^{K} n'_i)\cdot\mu_j\Big) - \sum_{i=1}^{K} n'_i\cdot\mu_i \\
= -\sum_{i=1}^{K} n'_i\cdot(\mu_i-\mu_1) + (N-\sum_{i=1}^{K} n'_i) \\
\cdot\sum_{j=2}^{K} PB_j(n'_1,\ldots,n'_K)\cdot(\mu_1-\mu_j)
\end{aligned}
\tag{14}
$$

The OARI problem in this configuration is to find the optimal allocation of heuristic algorithms $(n^*_1,\ldots,n^*_K)$ that maximize Equation 14.

4.1.2 Maximal Number of Formulas (MF) Configuration

In this configuration, the objective is to solve as many formulas as possible using T flips. We formulate the EURIKA problem for this setting as follows. Suppose that, without loss of generality, algorithm a_1 is the best alternative given the prior information D. If the agent chooses this algorithm without obtaining additional samples, the expected number of formulas it would solve is $\frac{T}{\mu_1}$. As before, obtaining $(n'_1,\ldots,n'_K)$ information units of the various algorithms solves $\sum_{i=1}^{K} n'_i$ 3-SAT formulas and performs $\sum_{i=1}^{K} n'_i\cdot\mu_i$ expected flip operations. At this point there are $T-\sum_{i=1}^{K} n'_i\mu_i$ flips remaining. If the agent decides to obtain this information then there are two possibilities:

- The agent will choose to continue to use algorithm a_1 with a probability of $PB_1(n'_1,\ldots,n'_K)$. In this case the expected number of formulas solved, is
$$PB_1(n'_1,\ldots,n'_K)\cdot\frac{T-\sum_{i=1}^{K} n'_i\mu_i}{\mu_1}$$
- The agent will choose alternative $a_j, j\neq 1$ to use for the remaining $T-\sum_{i=1}^{K} n'_i\mu_i$ flip operations. In this case the expected number of

formulas solved is

$$\sum_{j=2}^{K} PB_j(n'_1,\ldots,n'_K)\cdot\frac{T-\sum_{i=1}^{K} n'_i\mu_i}{\mu_j}$$

The expected number of flip operations for obtaining $(n'_1,\ldots,n'_K)$ information units, denoted $EN(n'_1,\ldots,n'_K)$ is the sum of these two cases:

$$EN(n'_1,\ldots,n'_K)=\sum_{j=1}^{K} PB_j(n'_1,\ldots,n'_K)\cdot\frac{T-\sum_{i=1}^{K} n'_i\mu_i}{\mu_j} \tag{15}$$

The expected benefit to the agent from obtaining $(n'_1,\ldots,n'_K)$ information units is the difference between the expected number of formulas solved after and before obtaining the additional information.

$$B(n'_1,\ldots,n'_K)=EN(n'_1,\ldots,n'_K)-\frac{T}{\mu_1} \tag{16}$$

The expected profit to this cost configuration, given in Equation 1 is computed as

$$\begin{aligned}
T(n'_1,\ldots,n'_K) &= B(n'_1,\ldots,n'_K)+\sum_{i=1}^{K} n'_i\\
&= EN(n'_1,\ldots,n'_K)-\frac{T}{\mu_1}+\sum_{i=1}^{K} n'_i\\
&= \sum_{j=1}^{K} PB_j(n'_1,\ldots,n'_K)\cdot\frac{T-\sum_{i=1}^{K} n'_i\mu_i}{\mu_j}-\frac{T}{\mu_1}+\sum_{i=1}^{K} n'_i \qquad (17)\\
&= PB_1(n'_1,\ldots,n'_K)\cdot\frac{T-\sum_{i=1}^{K} n'_i\mu_i}{\mu_1}+\\
&\sum_{j=2}^{K} PB_j(n'_1,\ldots,n'_K)\cdot\frac{T-\sum_{i=1}^{K} n'_i\mu_i}{\mu_j}-\frac{T}{\mu_1}+\sum_{i=1}^{K} n'_i
\end{aligned}$$

Since $PB_1(n'_1, \ldots, n'_K) = 1 - \sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K)$ the above equation equals to

$$
\begin{aligned}
T(n'_1, \ldots, n'_K) = (1 - \sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K)) \cdot \frac{T - \sum_{i=1}^{K} n'_i \mu_i}{\mu_1} + \\
\sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot \frac{T - \sum_{i=1}^{K} n'_i \mu_i}{\mu_j} - \frac{T}{\mu_1} + \sum_{i=1}^{K} n'_i \\
= \sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (T - \sum_{i=1}^{K} n'_i \mu_i) \cdot (\frac{1}{\mu_j} - \frac{1}{\mu_1}) + \sum_{i=1}^{K} n'_i \cdot (1 - \frac{\mu_i}{\mu_1})
\end{aligned} \quad (18)
$$

The OARI problem in this configuration is to find the optimal allocation of heuristic algorithms $(n^*_1, \ldots, n^*_K)$ that maximizes Equation 18.

4.2 The SAT Competition domain

This domain uses real automatic 3-SAT solvers submitted to the Ninth International Conference on Theory and Applications of Satisfiability Testing Conference.[6] Performance in the competition was based on a score that takes into account two factors:

- the number of instances solved within a given run-time limit.
- the total time needed to solve all instances.

We detail how to tailor the OARI problem for this domain. We set N to equal the total number of formulas in the competition. As before, a_i is a possible solution algorithm, and n'_i is the number of 3-SAT equations that are solved using algorithm a_i. The mean score in the competition associated with algorithm a_i is represented as μ_i. Thus, obtaining $(n'_1, \ldots, n'_K)$ samples of the various algorithms solves $\sum_{i=1}^{K} n'_i$ 3-SAT formulas and provides an expected score of $\sum_{i=1}^{K} \mu_i \cdot n'_i$ points. There are two possibilities.

- The agent will choose to continue to use algorithm a_1 to solve the remaining $(N - \sum_{i=1}^{K} n'_i)$ formulas. In this case the expected score is

$$PB_1(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i)\mu_1$$

[6] `http://fmv.jku.at/sat-race-2006/results.html`

- The agent will choose alternative $a_j, j \neq 1$ to use for the remaining $(N - \sum_{i=1}^{K} n'_i)$ formulas. In this case the expected score is

$$\sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i)\mu_j$$

The expected score for obtaining $(n'_1, \ldots, n'_K)$ information units, denoted $ES(n'_1, \ldots, n'_K)$ sums over these two cases

$$ES(n'_1, \ldots, n'_K) = \sum_{j=1}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_j \tag{19}$$

The expected benefit for obtaining the sample $(n'_1, \ldots, n'_K)$ is the difference in score from obtaining and not obtaining the additional information.

$$B(n'_1, \ldots, n'_K) = ES(n'_1, \ldots, n'_K) - (N \cdot \mu_1) \tag{20}$$

The expected profit to this cost configuration, given in Equation 1, is computed as

$$\begin{aligned}
T(n'_1, \ldots, n'_K) &= B(n'_1, \ldots, n'_K) + \sum_{j=1}^{K} n'_i \cdot \mu_i \\
&= \sum_{j=1}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_j - (N \cdot \mu_1) + \sum_{j=1}^{K} n'_i \cdot \mu_i \\
&= PB_1(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_1 + \\
&\sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_j - (N \cdot \mu_1) + \sum_{j=1}^{K} n'_i \cdot \mu_i \\
&= (1 - \sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K)) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_1 + \\
&\sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot \mu_j - (N \cdot \mu_1) + \sum_{j=1}^{K} n'_i \cdot \mu_i \\
&= \sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (N - \sum_{i=1}^{K} n'_i) \cdot (\mu_j - \mu_1) - \sum_{i=1}^{K} n'_i \cdot (\mu_1 - \mu_i)
\end{aligned} \tag{21}$$

4.3 The Professor Evaluation domain

In this domain, the objective is to choose the lecturer with the highest enrollment potential of K possible lecturers. Each alternative a_i represents a sample from a database of student satisfaction scores. This domain includes real data from a survey filled out by students at the Jerusalem College of Technology. The enrollment potential for courses depends on the satisfaction rating associated with the lecturer. The agent can query the database, for a cost, about professors' satisfaction ratings. The agent's objective is to choose the lecturer with the highest enrollment potential while taking into account the cost of obtaining information about students' ratings.

We formulate the OARI problem for this setting. Every lecturer a_i is associated with a mean satisfaction score that is represented by μ_i. The profit to the college is $\mu_i \cdot v$, where v is a positive constant, representing the fact that popular lecturers are more likely to draw higher course enrollments, leading to higher profits. Suppose that, without loss of generality, the agent chooses lecturer a_1 based on the prior information D. In this case, the expected benefit is $\mu_1 \cdot v$. Suppose that the agent has obtained $(n'_1, \ldots, n'_K)$ samples of the various lecturers. As before, there are two possibilities.

- The agent will continue to choose lecturer a_1. In this case the expected benefit is $PB_1(n'_1, \ldots, n'_K) \cdot \mu_1 \cdot v$.
- The agent will choose a different lecturer $a_j, j \neq i$ with probability $PB_j(n'_1, \ldots, n'_K)$. In this case the expected benefit is

$$\sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot \mu_j \cdot v$$

The expected reward of the sample $(n'_1, \ldots, n'_K)$ is the sum of these two cases:

$$(PB_1(n'_1, \ldots, n'_K) \cdot \mu_1 \cdot v) + (\sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot \mu_j \cdot v) \tag{22}$$

Because

$$PB_1(n'_1, \ldots, n'_K) \cdot \mu_1 \cdot v = (1 - \sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot \mu_j) \cdot v$$

the expected benefit to the agent from obtaining the sample $(n'_1, \ldots, n'_K)$ can be written as

$$\begin{aligned} B(n'_1, \ldots, n'_K) =& (1 - \sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K)) \cdot \mu_1 \cdot v \\ &+ \sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot \mu_j \cdot v - \mu_1 \cdot v \\ &= \sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (\mu_i - \mu_1) \cdot v \end{aligned} \tag{23}$$

The cost function in this domain assigns c units to obtaining each satisfaction rating. Incorporating this cost we obtain the following:

$$\begin{aligned} T(n'_1, \ldots, n'_K) =& B(n'_1, \ldots, n'_K) - \sum_{i=1}^{K} n'_i \cdot c = \\ & \sum_{j=2}^{K} PB_j(n'_1, \ldots, n'_K) \cdot (\mu_i - \mu_1) \cdot v - \sum_{i=1}^{K} n'_i \cdot c \end{aligned} \tag{24}$$

5 Empirical Methodology

To evaluate the performance of EURIKA, we conducted a number of experiments, using each of the domains described in the previous section. We compared the EURIKA approach to the FNE model [Talman et al., 2005], as well as a baseline that solely uses the agent's prior information to choose the best alternative. Our hypothesis was that using the EURIKA technique would increase the agent's overall gain as compared to the other methods. For each of the domains, the evaluation was performed over a number of rounds. Each round proceeded as follows:

- EURIKA was used to solve the OARI problem in each of the domains, and the optimal information gathering actions $(n^*_1, \ldots n^*_K)$ were obtained.
- The alternative that is associated with the highest sample mean based on the obtained and prior information was chosen to solve the relevant problem instances in the domain.

Each round included four alternative solutions and a set of five instances of prior information about each alternative. We varied the total number of rounds between 15-40 for each domain, such that all alternatives would be considered. The parameters ζ and τ^2 were assigned the average and the standard deviation of all of the rewards in each domain. The parameter σ_i was assigned the value of the standard deviation of the prior data obtained for each domain. In practice, when computing $PB_i(n'_1, \ldots, n'_K)$, we assumed that the differences between sample means of different alternatives are independent.

For the *SAT simulation* domain, we used the same algorithms and settings used by Talman et al. [2005] to facilitate comparison. These included a Greedy-SAT algorithm and variant GSAT algorithm with Random Walk probabilities of 40%, 60% and 80%. For the MF cost configuration, the number N of formulas to solve was set to 300. For the MT cost configurations, the number of flip operations was set to 200,000 or 500,000 for each run. We executed all algorithms on the same 300 3-SAT formulas used by Talman et al. [2005]. Each formula consisted of 100 different variables and 430 clauses. Each of the formulas was guaranteed to have a valid truth assignment. The prior data D for a given round consisted of the results of applying each of the four candidate algorithms on five different 3-SAT formulas. Based on the average and standard deviation of the number of flip operations on all problems for all algorithms, we set $\zeta = 55200$ and $\tau = 22140$.

In the *SAT Competition* domain we used the 16 finalists of the SAT-Race 2006 competition. All algorithms were evaluated on the same 100 SAT formulas that were used in the finals. We followed the declared rules of the competition, in that solving each SAT problem earned the solver 1 point and additional "speed"' points. The execution time was limited to 15 minutes per formula, otherwise the solver received 0 points for that instance. The number of speed points p_s for each successful solver s was computed by $p_s = P_{max} \cdot (1 - \frac{t_s}{T})$ where t_s is the execution time solver s requires in order to solve the SAT instance, T is the execution time threshold, and P_{max} is the maximal speed score a solver can receive for solving one SAT formula. Based on the average and standard deviation of the scores achieved by the different algorithms, we set $\zeta = 0.585$ and $\tau = 0.181$. Other parameters were set to correspond to those in the actual tournament: P_{max} was set at 1, N was set at 100, and T at 15 minutes. The candidate algorithms comprised the top-scoring entries in the competition.[7] The 3-SAT formulas in the competition were taken from several benchmarks applications in industry, such as bounded model checking, and pipelined machines, as well as a set of formulas used in past competitions.

In the *Professor Evaluation* domain we used a student survey which was held at the Jerusalem College of Technology. Students were asked to rate lecturers' performance on their courses, using a scale between 1 to 10. The 16 lecturers that received students' ratings were divided into four groups of four lecturers. Each round consisted of four lecturers and a prior sample of the ratings of five students about each of the lecturers. We evaluated EURIKA over all lecturers for different values of v (250, 500, 750, 1000). We set the cost of obtaining each student rating for a particular lecturer at $c = 5$ dollars. Using our database we set $\zeta = 8.06$ and $\tau = 2.14$, which are respectively the average and the standard deviation of the grades that were given to 128 different lecturers.

The FNE model solely uses the prior information to decide on the number of samples to obtain. Alternatives with higher prior means are sampled more often than those with lower prior means. Again, we used the same procedure used by Talman et al. [2005], in which the best alternative was sampled 6 times, the second-best was sampled 4 times, the third-best was sampled 3 times, and the worst alternative was sampled twice.

Figure 1 presents the average number of flip operations per formula for the MF (left) and MT scenario (right) for the SAT simulation domain. The average number of flip operations per formula using the EURIKA model was 5825, while the average number according to the prior-best alternative model was 9273 (T-test $p < 0.002$). Using EURIKA allowed to save up to 47% flip operations per formula compared to the case in which additional information is not obtained. The EURIKA approach also used significant less flip operations per formula than the FNE model (5825 operations versus 6390 operations, T-test PV= 0.054). The average number of samples recommended by EURIKA was 16.35, which was similar to the 15 samples used by the FNE, but EURIKA allocated these samples over different algorithms.

Figure 1 also presents the average number of formulas solved in the MF scenario for the $T = 500K$ and $T = 200K$ settings. For the $T = 200K$

[7] Specifically, we chose the following entries: Minisat 2.0, Eureka 2006, Rsat and Cadence MiniSat v1.14, QCompSAT, TINISAT, Eureka 2006 and Cadence MiniSat v1.14, Rsat, QPicoSAT, zChaff 2006, and HyperSAT

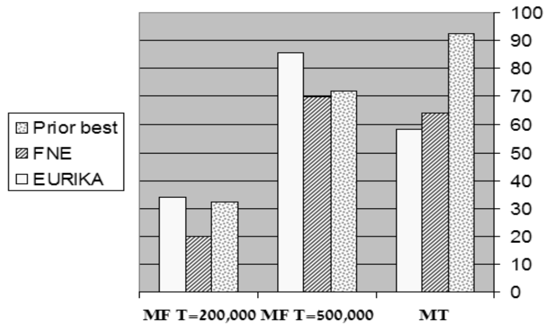

Figure 1: (left) Average # of formulas solved in the MF scenario (higher is better); (right) Average # of flips per formula (divided by 10) in the MT scenario (lower is better)

Model	GSAT	Random 60%	Random 80%	Random 40%
Prior-best	7.5%	22.5%	50.0%	20.0%
FNE model	0.7%	25.1%	67.2%	7.0%
EURIKA	**0.0%**	19.3%	**76.0%**	4.7%

Table 2: 3-SAT frequency (in percentages) of choosing each heuristic algorithm in the MT scenario according to the different approaches

setting, the EURIKA completed 34 formulas on average, versus 32 formulas solved by the FNE model, and 20 formulas solved by the prior-best alternative model. Although the difference between EURIKA and the FNE model was small, it was statistically significant. The difference in performance increased substantially for the $T = 500K$ setting. Here, EURIKA completed 86 formulas on average, versus 70 formulas using the FNE model, and 71 formulas using the prior-best alternative model (T-test $p < 0.001$). The average number of samples recommended by EURIKA for this scenario was 8.15, almost a half of the 15 samples used by the FNE. This shows that EURIKA was able to outperform the FNE model while acquiring less information that did the FNE model.

Table 2 summarizes the extent to which each heuristic algorithm was chosen (in percentages) by the various models for the MT cost configuration

	Experiment 1	Experiment 2	Experiment 3
Prior-best	79.54	66.08	64.36
FNE model	80.77	69.1	70.54
EURIKA	81.42	73.5	75.9

Table 3: Average performance for each approach in the SAT competition domain

in the SAT simulation domain. A post-hoc analysis of this domain revealed that Random 80% was the best heuristic algorithm for solving the 3-SAT formulas in the simulation, followed by the Random 40%, Random 80%, and GSAT algorithms. As shown by the table, all algorithms chose the best heuristic more often than they chose other heuristics. However, EURIKA was able to choose the best heuristic 27.2% more often than the prior-best alternative model, and 8.8% more often than the FNE model (Chi-square test, PV < 0.001). In contrast to the other approaches, EURIKA did not use GSAT at all, which the worst heuristic algorithm. Table 3 compares the performance of the various approaches in the SAT-competition domain.

As shown in the Figure, the EURIKA model significantly outperformed all the other approaches. The average number of points using EURIKA (77 points) was significantly higher than the average number of points for the prior-best alternative (70 points) and the FNE model (73.5 points).

Figure 2 compares the performance of the various models in the Professor Evaluation domain. We used several different values for v, and in all of these EURIKA significantly outperformed the prior-best method and the FNE model (T-test $p < 0.001$). On average, the EURIKA model achieved 54,638 points while the prior-best model achieved 52,136 points and the FNE model achieved 53,090 points, (T-test, PV < 0.001).

Table 4 concludes this section with two examples of the way EURIKA informed the information gathering actions in the 3-SAT simulation domain. Each example is drawn from one of the evaluation rounds, in which Algorithm 1,2,3, and 4 correspond to different candidate algorithms. In the first example, Algorithm 4 had the lowest average cost when considering the prior information, but with high standard deviation. Therefore EURIKA recommended additional samples. The new information included 35 samples of Algorithm 1, zero samples of Algorithms 2 and 3, and four samples of Algorithm 4. In this example, the prior cost of using Algorithm 2 and its

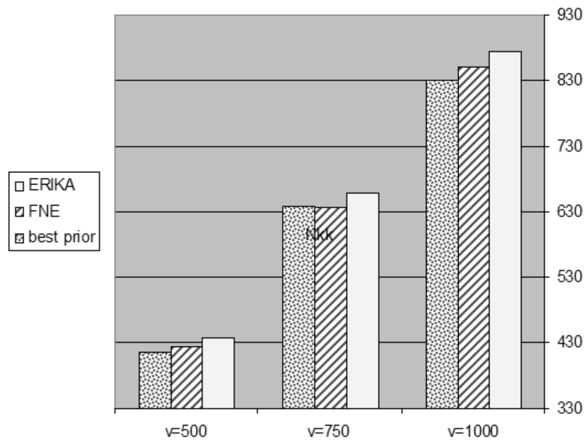

Figure 2: Average Performance (in hundreds of dollars) in the Professor Evaluation domain.

	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4
Average cost	25751.40	22708.80	26332.60	14201.60
Standard Deviation (σ_i)	42,039.39	30,161.97	13,926.25	21,474.92
Additional samples	35	0	0	4
Average cost	6,747.40	1,143.20	9,273.60	8,503.20
Standard Deviation (σ_i)	10,034.72	273.99	13,879.72	9,530.35
Additional samples	0	0	0	0

Table 4: Examples of Performance the 3-SAT Simulation Domain

standard deviation is considerably lower than using the other algorithms. In this clear-cut situation, EURIKA did not recommended to obtain any additional information about the various alternatives.

6 Conclusion and Future Work

This paper formalized the problem of obtaining information gathering actions about stochastic processes with unknown outcomes that affect agents' utilities. Agents can obtain information at a cost about the different alternative processes. The paper established this problem to be NP-Hard, and provided a tractable, analytical solution to the problem by approximating the expected benefit from obtaining information about each alternative, while taking into account the associated costs. The solution to the problem is based on estimating the agent's expected benefit from gaining additional units of information about the alternative processes using statistical measures. The robustness of our technique is demonstrated empirically by deploying it in settings that varied the type of task to optimize, the nature of information gathering actions, and the measure of performance. These settings included "ecologically realistic" data that was obtained from the real world. Although our theoretical model assumes that populations are normally distributed, our empirical results show that in practice, our approach can also be applied towards populations that may not adhere to this assumption. In future work we will augment the domain for situations in which agents' rewards are biased as well as situations in which distributions over rewards are unknown. We will also consider situations which include other decision-makers, requiring agents to consider the effect of their information gathering actions on each other's utilities.

References

R. Azoulay-Schwartz and S. Kraus. Acquiring an optimal amount of information for choosing from alternatives. *Cooperative Information Agents VI*, pages 123–137, 2002.

D.A. Berry and B. Fristedt. *Bandit problems: sequential allocation of experiments*. Chapman and Hall London, 1985.

M. Bilgic and L. Getoor. Voila: Efficient feature-value acquisition for classification. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2007.

S.E. Chick, J. Branke, and C. Schmidt. Sequential sampling to myopically maximize the expected value of information. *INFORMS Journal on Computing*, 22(1):71–80, 2010.

V. Cicirello and S. F. Smith. The max K-armed bandit: A new model of exploration applied to search heuristic selection. In *National Conference on Artificial Intelligence (AAAI)*, pages 1355–1361, 2005.

V. Conitzer and T. Sandholm. Definition and complexity of some basic metareasoning problems. In *International Joint Conference of Artificial Intelligence (IJCAI)*, pages 208–213, 1998.

R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman San Francisco, CA, 1979.

J. Grass and S. Zilberstein. A value-driven system for autonomous information gathering. *Journal of Intelligent Information Systems*, 14(1):5–27, 2000. ISSN 0925-9902.

C. Guestrin and Krause. Optimal value of information in graphical models. *Journal of Artificial Intelligence Research*, 35:557–591, 2009.

D. Heckerman, E. Horvitz, and B. Middleton. An approximate nonmyopic computation for value of information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(3):292–298, 1993.

A. Krause and C. Guestrin. Optimal nonmyopic value of information in graphical models-efficient algorithms and theoretical limits. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

A. Krause and C. Guestrin. Near-optimal observation selection using submodular functions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2007.

O. Madani, D.J. Lizotte, and R. Greiner. Active model selection. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, pages 357–365, 2004.

D. Madigan and R.G. Almond. *On test selection strategies for belief networks*, chapter Learning from Data: AI and Statistics IV, pages 89–98. Springer-Verlag, 1996.

J.A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

Y. Radovilsky and S.E. Shimony. Observation subset selection as local compilation of performance profiles. In *Proc. of the 24th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-08)*, pages 460–467, 2008.

S. Reches, S. Talman, and S. Kraus. A statistical decision-making model for choosing among multiple alternatives. In *International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, 2007.

J. Streeter and S. Smith. An asymptotically optimal algorithm for the max k-armed bandit problem. In *National Conference On Artificial Intelligence (AAAI)*, 2006.

S. Talman, R. Toester, and S. Kraus. Choosing between heuristics and strategies - an enhanced model. In *International Joint Conference of Artificial Intelligence (IJCAI)*, pages 324–330, 2005.

B. Thomas and R. L. Finney. *Calculus and Analytic Geometry.* Addison Wesley, 1996.

C.C. Tseng and P.J. Gmytrasiewicz. Time sensitive sequential myopic information gathering. In *Proceedings of the 35th Annual Hawaii International Conference on Systems Science*, page 7, 2002.

7 Appendix

Proof of Thoerem 1.

Proof. We present a reduction from the Knapsack problem [Garey and Johnson, 1979]. An instance of the Knapsack problem is given by a constraint $C > 0$, a target value $V > 0$ and a set of n items $\{1, \ldots, n\}$ when each item i has a positive integer value v_i and a positive integer weight w_i. The aim is to answer "yes" if a vector $(u_1, \ldots, u_n) \in N^n$ exists such that $\sum_{i=1}^{n} u_i \cdot v_i \geq V$ under the condition that $\sum_{i=1}^{n} u_i \cdot w_i \leq C$. We create an instance of *OARI* as follows.

- For each item i we create an alternative a_i. The number of alternatives K equals to the number of items n.
- For each variable u_i we create a variable n'_i.
- We set $M = C$.
- We set $L = V$.

- We define the function $Cost$ as follows:

$$Cost(a_i) = \begin{cases} 0 & \text{if } \sum_{j=1}^{n} n'_j \cdot w_j \leq C \\ v_i & \text{otherwise} \end{cases}$$

Suppose we already know that the expected profit from obtaining n'_i units of information about alternative a_i is equal to $n'_i \cdot v_i$. As a result $B(n'_1, \ldots, n'_n) = \sum_{j=1}^{n} n'_j \cdot v_j$. Since $T(n'_1, \ldots, n'_n) = B(n'_1, \ldots, n'_n) - \sum_{i=1}^{n} n'_i \cdot Cost(a_i)$ we obtain:

$$T(n'_1, \ldots, n'_n) = \begin{cases} \sum_{i=1}^{n} n'_i \cdot v_i & \text{if } \sum_{i=1}^{n} n'_i \cdot w_i \leq C \\ 0 & \text{otherwise} \end{cases}$$

We now prove that a vector $\overline{v} \in N^n$ solves the $OARI$ if and only if it solves the Knapsack problem.

($\Rightarrow$) Suppose there is a solution for the $OARI$ instance, that is a vector $(n'_1, \ldots, n'_n) \in N^n$ such that $\sum_{i}^{n} n'_i \leq C$ and $T(n'_1, \ldots, n'_n) \geq V$. Since $V > 0$, we have the following:

- $T(n'_1, \ldots, n'_n) \neq 0$ and thus $\sum_{i=1}^{n} n'_i \cdot w_i \leq C$
- $T(n'_1, \ldots, n'_n) = \sum_{i=1}^{n} n'_i \cdot v_i$.

As a result, $\sum_{i=1}^{n} n'_i \cdot v_i \geq V$ and thus the vector $(n'_1, \ldots, n'_n)$ is a solution to the KNAPSACK instance.

($\Leftarrow$) Suppose there is a solution to the KNAPSACK instance, that is, a vector $(u_1, \ldots, u_n) \in N^n$, such that $\sum_{i=1}^{n} u_i \cdot w_i \leq C$ and $\sum_{i=1}^{n} u_i \cdot v_i \geq V$. Since $\sum_{i=1}^{n} u_i \cdot w_i \leq C$, $T(u_1, \ldots, u_n) = \sum_{i=1}^{n} u_i \cdot v_i$ then $T(u_1, \ldots, u_n) \geq V$. In addition, since w_i is a positive integer for $1 \leq i \leq n$, $\sum_{i=1}^{n} u_i \leq \sum_{i=1}^{n} u_i \cdot w_i \leq C$, and thus $(u_1, \ldots, u_n)$ solves the $OARI$ problem. Therefore the OARI problem is NP-Hard.[8] □

Proof of Proposition 3.

Proof. After obtaining the $\overline{r'_i}, \overline{r'_j}$ additional information about alternatives a_i, a_j the agent will choose alternative a_i if : $\frac{n_i \overline{r_i} + n'_i \overline{r'_i}}{n_i + n'_i} > \frac{n_j \overline{r_j} + n'_j \overline{r'_j}}{n_j + n'_j}$ iff $\overline{r'_j} <$

[8] Given information gathering actions $(n'_1, \ldots, n'_K)$, if the expected benefit $T(n'_1, \ldots, n'_K)$ can be calculated in polynomial time, then the $OARI$ problem is NP-complete.

$\frac{(n_j+n'_j)(n_i\overline{r_i}+n'_i\overline{r'_i})}{n'_j(n_i+n'_i)} - \frac{n_j\overline{r_j}}{n'_j}$ Since $r'_j \sim N(\mu_j, \sigma^2)$, $\overline{r'_j} \sim N(\mu_j, \frac{\sigma^2}{n})$ and thus $Pr(\overline{r'_j} < \frac{(n_j+n'_j)(n_i\overline{r_i}+n'_i\overline{r'_i})}{n'_j(n_i+n'_i)} - \frac{n_j\overline{r_j}}{n'_j})$ is equal to the probability $Pr(Z < Z_\alpha(n'_i, n'_j, \overline{r'_i}))$ where $Z_\alpha(n'_i, n'_j, \overline{r'_i}) = \frac{\sqrt{n_j}((n_j+n'_j)(n_i\overline{r_i}+n'_ir'_i)-(n_i+n'_i)(n_j\overline{r_i}-\mu_jn'_j))}{n'_j(n_i+n'_i)\sigma_j}$ □

Proof of Proposition 4.

Proof. Using the Maclaurin series expansion of the function e^x Thomas and Finney [1996] we obtain:

$$e^x \approx \sum_{k=0}^{\infty} \frac{x^k}{k!} \tag{25}$$

Therefore:

$$e^{\frac{-x^2}{2}} \approx \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{2^k k!}. \tag{26}$$

Since $\frac{1}{\sqrt{2\pi}} \int_0^\infty e^{\frac{-t^2}{2}} dt = 0.5$, we attain:

$$\frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-t^2}{2}} dt = 0.5 - \frac{1}{\sqrt{2\pi}} \int_0^x e^{\frac{-t^2}{2}} dt =$$

$$0.5 - \frac{1}{\sqrt{2\pi}} \int_0^x \Sigma_{k=0}^{\infty} \frac{(-1)^k t^{2k}}{k! 2^k} dt + R_n$$

$$= 0.5 - \frac{1}{2\pi} \Sigma_{k=0}^{n} \frac{(-1)^k x^{2k+1}}{2^k(2k+1)k!} + R_n$$

According to Lagrange Reminder theorem (see Thomas and Finney [1996]) we find that when $|x| < d$:

$$|R_n| < |\frac{x^{n+1}}{(n+1)!}| < \frac{d^{n+1}}{(n+1)!} \tag{27}$$

As a result $R_n \longrightarrow 0$,when $n \longrightarrow \infty$, and thus $P\_approx(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-t^2}{2}} dt$ when $n \longrightarrow \infty$.

In addition, since $F(x)$ is the integration over a density function, $F(x) \longrightarrow 1$ when $x \longrightarrow -\infty$, and $F(x) \longrightarrow 0$ when $x \longrightarrow \infty$. Thus, the error when $|x| \geq d$ holds $R_n \leq 0.5 - \frac{1}{\sqrt{2\pi}} \int_0^{|d|} e^{\frac{-t^2}{2}} dt$ □