

How to Probe for an Extreme Value*

Ashish Goel[†]

Sudipto Guha[‡]

Kamesh Munagala[§]

Abstract

In several systems applications, parameters such as load are known only with some associated uncertainty, which is specified, or modeled, as a distribution over values. The performance of the system optimization and monitoring schemes can be improved by spending resources such as time or bandwidth in *observing* or *resolving* the values of these parameters. In a resource-constrained situation, deciding which parameters to observe in order to best optimize the expected system performance (or in general, optimize the expected value of a certain objective function) itself becomes an interesting optimization problem.

In this paper, we initiate the study of such problems that we term “model-driven optimization”. In particular, we study the problem of optimizing the minimum value in the presence of observable distributions. We show that this problem is NP-HARD, and present greedy algorithms with good performance bounds. The proof of the performance bounds are via novel sub-modularity arguments and connections to covering integer programs.

*A preliminary version of this paper [Goel *et al.* 2007] appeared in the Proceedings of the 25th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), 2006.

[†]Departments of Management Science and Engineering and (by courtesy) Computer Science, Stanford University. Research supported by an NSF CAREER award and an Alfred P. Sloan faculty fellowship. Email: ashishg@stanford.edu.

[‡]Department of Computer and Information Sciences, University of Pennsylvania, Philadelphia PA 19104-6389. Supported in part by an Alfred P. Sloan Research Fellowship, by an NSF CAREER award, and by NSF award CCF-0430376. Email: sudipto@cis.upenn.edu.

[§]Department of Computer Science, Duke University, Durham NC 27708-0129. Research supported by NSF via a CAREER award and grant CNS-0540347. Email: kamesh@cs.duke.edu.

1 Introduction

Optimization problems arising in databases, streaming, cluster computing, and sensor network applications often involve parameters and inputs whose values are known only with some uncertainty. In many of these situations, the optimization can be significantly improved by *resolving* the uncertainty in the input before performing the optimization. For instance, a query optimizer often has the ability to observe characteristics in the actual data set, like selectivities, either via random sampling or by performing inexpensive filters [Babu *et al.* 2005; Babcock and Chaudhuri 2005]. As another example, a system like Eddies [Avnur and Hellerstein 2000] finds the best among several competing plans which are run simultaneously. Each plan’s running time is a distribution which is observed by executing the plan for a short amount of time. In all such examples, the process of resolving the uncertainty also consumes resources, e.g., time, network bandwidth, space, etc., which compete with finding the solution of the original problem.

Therefore, judiciously choosing which variables to observe, itself becomes an important problem in this context. In this paper, we study the optimization problem of finding the minimum, when the inputs are random variables and the values of one or more inputs can be observed and resolved by paying some cost. We show that even for this simplest of optimization problems, this choice becomes non-trivial and intractable, motivating the development of algorithmic techniques to address them.

We initiate the study of the following abstractly defined class of problems, that we term “model-driven optimization”:

Problem 1. *We are given the distributions of non-negative independent random variables $\{X_i\}_{i=1}^n$. Further, these random variables are observable: we can find the value of X_i by spending cost c_i . Given a budget C and an objective function f , can we choose a subset S of random variables to observe whose total observation cost is at most C , and optimize the expected value of the function $f(S)$? Note that the function f is evaluated after the observations, and the expectation is over the outcome of the observations.*

In this paper, we focus on the function $f(S) = \min_{i \in S} X_i$, so that the goal is to choose a subset S whose observation cost is at most C , so that $\mathbf{E}[\min_{i \in S} X_i]$ is minimized. We define this problem as the MINIMUM ELEMENT problem. In Section 1.3, we present a brief survey of results [Guha and Munagala 2007; Guha and Munagala 2008; Guha *et al.* 2008] on other objective functions f .

As a motivating example, in the context of traditional and P2P networks, “multi-homing” schemes are becoming a common method for choosing communication routes and server assignments. These schemes [Akella *et al.* 2003; Gummadi *et al.* 2004] probe and observe the current state of multiple routes before deciding the optimum (minimum latency) route to use for a specific connection. The distribution of the latency of each route is available *a priori*. The number of probes needs to be bounded since flooding the network is undesirable. Therefore the goal is to minimize the latency of the route found by a bounded number of probes. The mapping to our framework in this case is as follows: X_i are the distributions of route latencies. The probing cost c_i is a function of the delay and load incurred in detecting latency of route i . The budget C is the total load and processing time that can be incurred by the route choosing algorithm. Finally, $f = \min_{i \in S} X_i$, where S is the set of routes probed. This corresponds to the goal of choosing that set of routes to probe which minimizes the expected value of the smallest latency detected.

Note that if the minimum value among the probed set is more than the expected value of a variable that has not been probed, we would prefer to use that variable (the “backup”) as opposed to one of the probed values. We will not take this optimization into account while analyzing our algorithm. Refer Section 6 for the reason. We now show the benefit of probing with an example.

Example 1.1. *If all variables are Bernoulli $B(1, p)$ (with any p), the estimate of the **minimum** is p if only one probe is allowed, but is $p^n \ll p$ if all nodes are probed. Probing can therefore yield an estimate which is **exponentially** smaller, which means that if there is a low utilization, we will very likely find it.*

Note that the optimization chooses the final variable *after* the results of the probes are known. Therefore, the overall optimization is to choose S with cost at most C , so that $\mathbf{E}[\min_{i \in S} X_i]$ is minimized. This is very different from optimizing $\min_{i \in S} \mathbf{E}[X_i]$, which is obtained when we choose the final variable *before* the results of the observations are known. The latter version is of course trivial to solve.

Consider the simple case when the probing costs of all nodes are equal. Let m denote the constraint on the number of variables that can be probed. It would appear that the optimal strategy would be to choose the m nodes with the smallest expected values. The example below shows that this need not be the case. Note further that our problem is not the same as minimizing residual entropy [Krause and Guestrin 2005] which minimizes uncertainty of the joint distribution – we are concerned with minimizing the uncertainty of an optimization that depends on the joint distribution, the minimum value. Minimizing residual entropy will most often involve probing a different set of variables than those required for estimating best the specific function at hand (see example below); therefore, the problems are orthogonal.

Example 1.2. *There are 3 distributions X_1, X_2 and X_3 and $m = 2$. Distribution X_1 is 0 with probability $\frac{1}{2}$ and 1 with probability $\frac{1}{2}$. Distribution X_2 is 1 with probability $\frac{1}{2}$ and 2 with probability $\frac{1}{2}$. Distribution X_3 is 0 with probability $\frac{1}{5}$ and 2 with probability $\frac{4}{5}$. Clearly, $\mathbf{E}[X_1] < \mathbf{E}[X_2] < \mathbf{E}[X_3]$. However, probing X_1, X_2 yields an expected minimum value of 0.5, while probing X_1, X_3 yields an expected minimum value of 0.4. Minimizing residual entropy [Krause and Guestrin 2005] would also choose the sub-optimal set $\{X_1, X_2\}$.*

The simple strategy does not take into account the *shape* of the distributions. In fact, the MINIMUM-ELEMENT problem becomes NP-HARD for arbitrary distributions even with uniform costs. In fact, we show a stronger hardness result: it is NP-HARD to approximate the minimum value up to polynomial factors without exceeding the observation budget C . Hence the natural model to study this problem is from the perspective of *resource augmentation*: can we guarantee that we achieve the same solution as the optimum, while we pay a larger observation cost?

1.1 Results

We introduce the problem of model driven optimization in the presence of observations, and particularly consider the MINIMUM-ELEMENT problem. We present natural algorithms that are easy to implement and provide strong evidence that these algorithms are likely to be the best possible. We note that naive greedy algorithms do not work and extra algorithmic techniques are required to augment the greedy algorithm.

Our first result shows a deep connection between the MINIMUM-ELEMENT problem and a certain type of covering integer program. We use this connection to show that it is NP-HARD to approximate the objective up to any polynomial factor without augmenting the cost budget. Consequently, we design algorithms that approximate the cost while achieving nearly optimal objective value. The algorithms we design yield a $(1 + \epsilon)$ approximation to the optimal value by increasing the cost budget by a factor that looks like $O(\eta + \gamma_\epsilon)$, where $\gamma_\epsilon = \log \frac{1}{\log(1+\epsilon)}$ is a constant independent of the input. Our results show different values of η for different types of distributions.

In the most general case, we show that the function $f(S) = \mathbf{E}[\min_{i \in S} X_i]$ is *log-concave*, i.e., $\log 1/f(S)$ is sub-modular. (Simply showing $f(S)$ is submodular, yields a approximation ratio polynomial in m .) Consequently a greedy algorithm gives $\eta = \log \log \frac{\min_{i \in S^*} \mathbf{E}[X_i]}{\mathbf{E}[\min_{i \in S^*} X_i]}$ where S^* is the optimal solution.

We then show approximation algorithms whose bounds depend on entirely different parameters of the problem, as well as improved results for special cases. These are summarized below.

1. If the distributions are discrete over a domain of m arbitrary values, then $\eta = \log m$. This uses the connection with covering programs mentioned above, and proceeds by using the approximately optimal solution to a covering integer program as the start solution to the greedy scheme.

2. If the distributions are over the domain $\{0, 1, 2, \dots, M\}$, then $\eta = \log \log M$. This uses a scaling argument combined with the covering program.
3. For arbitrary log-concave distributions, $\eta = O(1)$. To show this, we develop a truncation result for log-concave distributions. We show that a modified greedy algorithm that tries different truncated distributions in turn yields the desired result. Log-concave distributions capture uniform, Gaussian, and Beta densities among others [6].

In terms of techniques, we combine an involved sub-modularity argument along with the analysis of the best *fractional* solutions of covering problems. Although the analyses are complicated, the algorithms are natural.

1.2 Related Work

The notion of refining uncertainty has been considered in an *adversarial setting* by several researchers [Olston 2003; Feder *et al.* 2003; Khanna and Tan 2001; Charikar *et al.* 2002]. In the adversarial model, the only prior information about an input is the lower and upper bounds on its value. The goal is to minimize the observations needed to estimate some function over these inputs *exactly*, and often negative results arise. The use of lower and upper bounds do not exploit the full power of models/samples/stochasticity of the data, i.e., the *distributions* of inputs. However to use the distributional information we must optimize the *expected value* of the function, which is also referred to as stochastic optimization.

More recently, significant attention has been devoted towards developing and using models and estimates of data based on prior knowledge, e.g., [Chu *et al.* 1999; Chu *et al.* 2002; Deshpande *et al.* 2004; Babcock and Olston 2003; Babu *et al.* 2005; Babcock and Chaudhuri 2005; Silberstein *et al.* 2006] among many others. Our work complements the body of research on maintenance of samples and estimates, and we show that judicious probing may yield exponentially better estimates.

Another line of work [Dean *et al.* 2004] considers knapsack problem in the model that the job sizes are revealed only after the job has *irrevocably* placed in the knapsack. In the settings we described, this would imply that the decision to refine our estimate, i.e., probing, is equivalent to selecting the item in the final solution. This effectively disallows probing. In our model the choice of which variables to pack in the knapsack would be made *after* the observations. There also has been ongoing research in multi-stage stochastic optimization [Kleinberg *et al.* 2000; Goel and Indyk 1999; Immorlica *et al.* 2004; Gupta *et al.* 2004; Gupta, Ravi, and Sinha 2004; Shmoys and Swamy 2004], however most of this literature also involves making irrevocable commitments.

1.3 Subsequent Results

The model-driven optimization framework can be defined for other objective functions f . Though this paper presents the best results known for MINIMUM-ELEMENT other work has considered different objective functions. First, when f is a single constraint packing problem such as knapsack with random profits which are observable, Guha and Munagala [Guha and Munagala 2007] show a 8 approximation based on rounding the solution of a natural linear program. The approximation ratio holds even when the optimal solution is allowed adaptive (*i.e.*, can be based on the results of previous observations). It further holds even when the hidden quantity is a distribution (instead of a single value) and a prior on this distribution is specified as input. A special case of this result is when $f(S) = \max_{i \in S} X_i$, or the MAXIMUM ELEMENT problem. Since the MINIMUM and MAXIMUM element problems are equivalent from the point of view of exact solution, the NP-HARDNESS result we present below also shows that MAXIMUM ELEMENT is NP-HARD. However, from the point of view of approximation, the techniques and results for the MINIMUM and MAXIMUM element problems are very different.

In [Guha *et al.* 2008], Guha, Munagala, and Sarkar consider the Lagrangean version of the MAXIMUM ELEMENT problem, where the observations are adaptive and the goal is to maximize the expected difference between the maximum value and the observation cost. Note that there is no budget on this cost, instead it is part of the objective function. If the maximum value needs to be chosen from the observed distributions, this problem has an optimal solution. The same trivially holds for optimizing the sum of the minimum value and the observation cost. When an unobserved distribution can also be chosen as the maximum value, the Lagrangean version is shown to have a 1.25 approximation via a greedy algorithm that is analyzed by a non-trivial structural characterization of the optimal solution.

Finally, when f is geometric problem, such as K -median clustering or spanning tree construction, on independent (and observable) point clouds, Guha and Munagala [Guha and Munagala 2007] show constant factor approximation algorithms when the observations are adaptive. These algorithms are based on converting the observation problem into an *outlier* version of the problem. Similar results are also shown for the average completion time scheduling problem.

2 Minimum–Element: Preliminaries

We are given n independent non-negative random variables X_1, X_2, \dots, X_n . Assume that X_i has observation cost c_i . There is a budget C on the total cost. The goal is to choose a subset S of distributions with total cost at most C which minimizes $\mathbf{E}[\min_{i \in S} X_i]$. Without loss of generality, we can assume that $c_i \leq C$ for all i . We further assume the distributions are specified as discrete distributions over m values, $0 \leq a_1 \leq a_2 \leq \dots \leq a_m \leq M$. Let $p_{ij} = \Pr[X_i \geq a_j]$. The input is specified as the p_{ij} and a_j values. Note that m is not very large since frequently the distribution is learned from a histogram/quantile.

Some Notation: Let $a_0 = 0$ and $a_{m+1} = M$. For $j = 0, 1, \dots, m$, let $l_j = a_{j+1} - a_j$. We call $I_j = [a_j, a_{j+1}]$ the j^{th} interval. This is illustrated in Figure 1. Recall that $p_{ij} = \Pr[X_i \geq a_j]$. We have $\mathbf{E}[X_i] = \sum_{j=0}^{m-1} p_{ij} l_j$. We define $f(S) = \mathbf{E}[\min_{i \in S} X_i]$ for each subset S of variables. All logarithms are to base e . Let $f(\Phi) = M$.

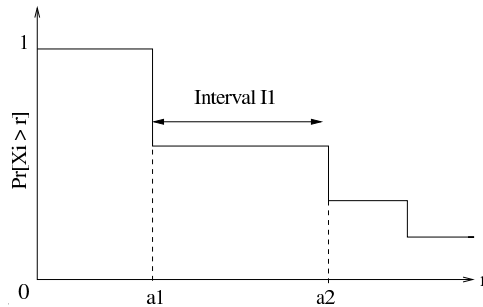


Figure 1: Notation used in MINIMUM–ELEMENT.

3 NP–Hardness

We begin with a hardness result: It is NP–HARD to obtain a $\text{poly}(m)$ approximation on the objective for MINIMUM–ELEMENT while respecting the cost budget, even for uniform costs. We therefore focus on approximation algorithms for this problem which achieve the optimal objective while augmenting the cost budget. Thus the approximation results are on the *cost* in this paper.

Definition 3.1. A *Covering Integer Program (CIP)* over n variables x_1, x_2, \dots, x_n (indexed by i) and m constraints (indexed by j) has the form

$$\begin{aligned} \min \sum_i c_i x_i \\ \text{subject to: } \quad A\vec{x} \geq \vec{b} \\ \vec{x} \in \{0, 1\}^n \end{aligned}$$

where $c_i \in \mathbb{R}^+$ and $A \in \mathbf{Z}^{m \times n}$, i.e., the elements A_{ji} of the constraint matrix are non-negative integers. This is a generalization of SET-COVER where the matrix A is $\{0, 1\}$ and $c_i = 1$. A CIP is defined to be column-monotone if $A_{ji} \leq A_{(j+1)i}$ for all i and for all $j < m$. Without loss of generality, we can assume that $b_j \in \mathbf{Z}^+$ and $b_{j+1} \geq b_j$.

We start with the following decision problem: Given a column-monotone CIP, determine whether the optimum objective value is less than C or more than rC ; if the optimum value lies between C and rC then either of the two answers is considered valid. We will relate the hardness of this decision problem (which we call r -GAP-CIP) to the hardness of approximating the MINIMUM-ELEMENT problem.

An (r, s) -approximation for the MINIMUM-ELEMENT problem violates the cost budget by at most r and obtains an objective function value (i.e. the expectation of the minimum) that is at most s times the optimum objective function value with the original cost budget.

Lemma 3.2. *The r -GAP-CIP problem with polynomially bounded (in n and m) coefficients A_{ji} reduces, in poly-time, to the problem of obtaining an $(r, \text{poly}(m))$ -approximation for MINIMUM-ELEMENT.*

Proof. Fix any constant k , and let $q = m^{k+1}$. Define n distributions over values $\mu_0, \mu_1, \dots, \mu_m$ where $\mu_0 = 0$ and $\mu_j - \mu_{j-1} = q^{b_j}$. Distribution X_i has observation cost c_i , and $\Pr[X_i > \mu_{j-1}] = q^{-A_{ji}}$. Further, let $\Pr[X_i > \mu_m] = 0$. Note that these definitions imply $\Pr[X_i = 0] = 1 - q^{-A_{0i}}$.

Observe that column-monotonicity is crucial for this definition to correspond to a valid probability distribution; the requirement that A_{ji} 's be polynomially bounded is crucial for the reduction to be polynomial time. Let the cost budget for the MINIMUM-ELEMENT problem be C .

First assume that the original CIP has a solution x_1, x_2, \dots, x_n with cost at most C . Let S be the set $\{i : x_i = 1\}$. For the MINIMUM-ELEMENT problem, probe the variables $X_i, i \in S$. For any j ,

$$\Pr[\min_{i \in S} X_i \geq \mu_{j-1}] = \prod_{i \in S} \Pr[X_i > \mu_{j-1}] = q^{-\sum_{i \in S} A_{ji}} \leq q^{-b_j}.$$

Therefore,

$$\mathbf{E}[\min_{i \in S} X_i] = \sum_{j=1}^m (\mu_j - \mu_{j-1}) \Pr[\min_{i \in S} X_i > \mu_{j-1}] \leq \sum_{j=1}^m q^{b_j} q^{-b_j} = m.$$

Now suppose that the original CIP has no solution of cost rC or less. Then for any index set S such that $\sum_{i \in S} c_i \leq rC$, there must be at least one constraint j such that $\sum_{i \in S} A_{ji} \leq b_j - 1$. Thus, $\Pr[\min_{i \in S} X_i > \mu_{j-1}] = q^{-\sum_{i \in S} A_{ji}} \geq q^{1-b_j}$. Now,

$$\mathbf{E}[\min_{i \in S} X_i] \geq (\mu_j - \mu_{j-1}) \Pr[\min_{i \in S} X_i > \mu_{j-1}] \geq q^{b_j} q^{1-b_j} = q.$$

Thus, the problem of distinguishing whether the optimum value of the original CIP was less than C or more than rC has been reduced to the problem of deciding whether MINIMUM-ELEMENT has an optimum objective value $\leq m$ with cost budget C or an optimum value $\geq q$ with cost budget rC .

Since $q/m = m^k$, we have obtained a polynomial time reduction from r -GAP-CIP to the problem of obtaining an (r, m^k) -approximation of the MINIMUM-ELEMENT problem. \square

Theorem 3.3. *It is NP-HARD to obtain any $\text{poly}(m)$ approximation on the objective for MINIMUM-ELEMENT while respecting the cost budget.*

Proof. We reduce from the well-known NP-HARD problem of deciding if a set cover instance has a solution of value k . The SET COVER problem is the following: given a ground set U with m elements and n sets $S_1, S_2, \dots, S_n \subseteq U$ over these elements, decide if there are k of these sets whose union is U .

Write this set cover instance as a CIP as follows. There is an row for each element and a column for each set. $A_{ji} = 1$ if element j is present in set S_i and 0 otherwise. All $b_j = 1$ and all $c_i = 1$. To make this column-monotone, set $A_{ji} \leftarrow A_{ji} + j$ for each j, i and set $b_j \leftarrow 1 + jk$. Clearly, if there is a solution to this monotone instance of value k , this solution has to be feasible for the set cover instance and is composed of k sets. Conversely, if the set cover instance has a solution with k sets, the monotone CIP has a solution of value k . Since deciding if a set cover instance has a solution using k sets is NP-HARD, solving this class of 1-GAP-CIP instances is NP-HARD. By the proof of Lemma 3.2, this implies a $(1, \text{poly}(m))$ -approximation to the MINIMUM-ELEMENT problem is NP-HARD. \square

We have only been able to prove NP-Hardness of column-monotone CIPs, and so have not been able to fully exploit the approximation preserving reduction in Lemma 3.2. A hardness of approximating column-monotone CIPs will immediately lead to a stronger hardness result for the MINIMUM-ELEMENT problem via Lemma 3.2.

```

MINIMUM-ELEMENT( $\tilde{C}$ )
/*  $\tilde{C} =$  Relaxed cost bound ( $\tilde{C} \geq C$ ). */
 $S \leftarrow \Phi$ .
While ( $\sum_{i \in S} c_i \leq \tilde{C}$ )
     $X_q \leftarrow \operatorname{argmax}_i \frac{\log f(S) - \log f(S \cup \{X_i\})}{c_i}$ .
     $S \leftarrow S \cup \{X_q\}$ 
endwhile
Output  $S$ 

```

Figure 2: Greedy Algorithm for MINIMUM-ELEMENT.

4 Greedy Algorithm

The algorithm is described in Figure 2 and takes a relaxed cost bound $\tilde{C} \geq C$ as parameter, and outputs a solution of cost \tilde{C} . As we discuss later, the parameter \tilde{C} trades-off in a provable fashion with value of the solution found. The algorithm uses the slightly unnatural function $\log f(S)$ instead of the more natural function $f(S)$. As our analysis shows, this modification provably improves our approximation bound. The analysis of this algorithm uses the theory of submodularity [Nemhauser *et al.* 1978]. *Sub-modularity* is a discrete analogue of convexity which is the basis of many greedy approximation algorithms. We formally define sub-modularity next.

Definition 4.1. *A function $g(S)$ defined on subsets $S \subseteq U$ of a universal set U is said to be sub-modular if for any two sets $A \subset B \subseteq U$ and an element $x \notin B$, we have $g(A \cup \{x\}) - g(A) \geq g(B \cup \{x\}) - g(B)$.*

The key result in this section shows that the function $\log \frac{1}{f}$ used by the greedy algorithm is sub-modular.

Lemma 4.2. *Let $f(S) = \mathbf{E}[\min_{i \in S} X_i]$. Then, the function $\log \frac{1}{f}$ is sub-modular.*

Proof. Consider two sets of variables A and $B = A \cup C$, and a variable $X \notin B$. In order to prove the theorem, we need to show that $\frac{f(A \cup \{X\})}{f(A)} \leq \frac{f(B \cup \{X\})}{f(B)}$. We first define the following terms for each $j = 0, 1, \dots, m-1$:

1. $\alpha_j = \Pr[(\min_{Y \in A} Y) \geq a_j] = \prod_{Y \in A} \Pr[Y \geq a_j]$.
2. $\beta_j = \Pr[(\min_{Y \in C} Y) \geq a_j] = \prod_{Y \in C} \Pr[Y \geq a_j]$.
3. $\gamma_j = \Pr[X \geq a_j]$.

First note that the α_j, β_j and γ_j values are non-negative and monotonically non-increasing with increasing j . Next, by the independence of the variables, we have:

$$\begin{aligned} f(A \cup \{X\}) &= \sum_{j=0}^{m-1} l_j \Pr[(X \geq a_j) \wedge (\min_{Y \in A} Y \geq a_j)] \\ &= \sum_{j=0}^{m-1} l_j \Pr[X \geq a_j] \Pr[(\min_{Y \in A} Y) \geq a_j] = \sum_{j=0}^{m-1} l_j \alpha_j \gamma_j \end{aligned}$$

Similarly, $f(B) = \sum_{j=0}^{m-1} l_j \alpha_j \beta_j$ and $f(B \cup \{X\}) = \sum_{j=0}^{m-1} l_j \alpha_j \beta_j \gamma_j$.

Using the above, it follows that:

$$\frac{f(A \cup \{X\})}{f(A)} = \frac{\sum_j l_j \alpha_j \gamma_j}{\sum_j l_j \alpha_j} \quad \text{and} \quad \frac{f(B \cup \{X\})}{f(B)} = \frac{\sum_j l_j \alpha_j \beta_j \gamma_j}{\sum_j l_j \alpha_j \beta_j}$$

Therefore, we have:

$$f(A \cup \{X\})f(B) - f(B \cup \{X\})f(A) = \sum_{j < j'} l_j l_{j'} \alpha_j \alpha_{j'} (\gamma_j - \gamma_{j'}) (\beta_{j'} - \beta_j) \leq 0$$

The final inequality follows due to the monotonicity of both the γ_j and β_j values. The above implies $\frac{f(A \cup \{X\})}{f(A)} \leq \frac{f(B \cup \{X\})}{f(B)}$, which shows $\log \frac{1}{f}$ is a sub-modular function. \square

The connection between sub-modular functions and the greedy algorithm is captured by the following theorem:

Theorem 4.3 ((Nemhauser *et al.* 1978)). *Given a non-decreasing submodular function $g(\cdot)$ on a universal set U , where each element $i \in U$ has a cost c_i , and given a cost bound $C \geq \max_i c_i$, let $S^* = \operatorname{argmax}\{g(S) \mid \sum_{i \in S} c_i \leq C\}$. Consider the greedy algorithm that, having chosen a set T of elements, chooses the next one element i that maximizes the ratio $\frac{g(T \cup \{i\}) - g(T)}{c_i}$. Let $g(\Phi)$ denote the initial solution. Then, for any ϵ , the greedy algorithm using cost budget $C \log \frac{g(S^*) - g(\Phi)}{\epsilon}$ finds a set T_ϵ such that $g(T_\epsilon) \geq g(S^*) - \epsilon$.*

Intuitively, sub-modularity ensures that current greedy choice has cost per unit increase in value of $g(\cdot)$ at most the corresponding value for the optimal solution. For MINIMUM-ELEMENT, let S^* denote the optimal solution using cost C .

Theorem 4.4. *Let $V = \mathbf{E}[\min_{i=1}^n X_i]$. The greedy algorithm for MINIMUM ELEMENT achieves a $(1 + \epsilon)$ approximation to $f(S^*)$ with cost $\tilde{C} = C (\log \log \frac{M}{V} + \gamma_\epsilon)$, where $\gamma_\epsilon = \log \frac{1}{\log(1+\epsilon)}$ is independent of the input parameters.*

Proof. In the above theorem, set $g = \log \frac{1}{f}$. Since $f(\Phi) = M$, we have $g(\Phi) = \log \frac{1}{M}$. Let S denote the greedy set. Suppose $g(S) \geq g(S^*) - \log(1 + \epsilon)$. Then, $f(S) \leq (1 + \epsilon)f(S^*)$.

Therefore, in the above theorem, we have $T_{\log(1+\epsilon)} = S$, which implies the cost is $\tilde{C} \leq C(\log \log \frac{f(\Phi)}{f(S^*)} - \log \log(1 + \epsilon))$. Since $f(\Phi) = M$ and $f(S^*) \geq V$, we have $\tilde{C} \leq C(\log \log \frac{M}{V} + \gamma_\epsilon)$. \square

Note: If instead of $\log \frac{1}{f}$, we had used $-f$ (which is submodular) as suggested by a naive greedy algorithm then we would have needed a significantly worse cost of $C(\log \frac{M}{V} + \log \frac{1}{\epsilon})$ to achieve a $(1 + \epsilon)$ approximation to $f(S^*)$. Thus the improved analysis (and algorithm) was necessary. We next prove the following lower bound.

Theorem 4.5. *The analysis of the greedy algorithm is tight on the cost.*

Proof. There are $K = \log \log M$ intervals of form $I_i = [2^{2^i}, 2^{2^{i+1}}]$ for $i = 1, 2, \dots, K$. There are K distributions X_1, X_2, \dots, X_K . Distribution X_i ($i = 1, 2, \dots, K$) takes value 2^{2^i} with probability $(1 - 2^{-2^{i+1}+1})$ and takes value 2^{2^r} w.p. $2^{-2^{i+1}+1}$. Here, $r = K + 1$, and $K = \omega(1)$. Note that $2^{2^r} = 2^{2^{1+\log \log M}} = M^2$ is the range of the distributions.

There are two special distributions Y_1 and Y_2 such that $\Pr[Y_j > 2^{2^i}] = 2^{-2^i+1}$, for $j = 1, 2$, and $i = 1, 2, \dots, K$. We also have $\Pr[Y_j \geq 0] = 1$ and $\Pr[Y_j > 2^{2^r}] = 0$. All distributions have unit cost. Let $C = 2$. The optimal solution chooses the distributions Y_1 and Y_2 . We have:

$$\begin{aligned} \mathbf{E}[\min(Y_1, Y_2)] &= 4 + \sum_{i=1}^K (2^{-2^i+1})^2 (2^{2^{i+1}} - 2^{2^i}) \\ &\leq 4 \left(1 + \sum_{i=1}^K 2^{-2^i+1} 2^{2^{i+1}} \right) \\ &\leq 4(1 + K) = O(K) \end{aligned}$$

We claim that GREEDY first chooses X_K . If GREEDY chose any other distribution Z , then the expectation can be lower bounded using just the contribution from the last interval I_K :

$$\mathbf{E}[Z] \geq (2^{2^r} - 2^{2^{r-1}}) \Pr[Z > 2^{2^{r-1}}] \geq (2^{2^r} - 2^{2^{r-1}}) 2^{-2^{r-1}+1} = 2(2^{2^{r-1}} - 1)$$

Further, noting that $K = r - 1$ and that $r = \omega(1)$, we have:

$$\mathbf{E}[X_K] = 2^{2^{r-1}} (1 - 2^{-2^r+1}) + 2^{2^r} 2^{-2^r+1} = 2^{2^{r-1}} + 2 + 2 \cdot 2^{-2^{r-1}} < 2(2^{2^{r-1}} - 1)$$

This implies GREEDY chooses X_K first. At this point, the contribution from the interval I_K to GREEDY is $(2^{2^r} - 2^{2^{r-1}}) 2^{-2^r+1} \leq 2$. However, $\mathbf{E}[X_K] \geq 2^{2^K} = \omega(K)$.

We now repeat the whole argument using interval I_{K-1} , and show that GREEDY chooses X_{K-1} next. We lower bound the contribution of $\mathbf{E}[\min(Z, X_K)]$ for every distribution $Z \neq X_K, X_{K-1}$ on the interval I_{K-1} , and argue that this is at least $\mathbf{E}[\min(X_{K-1}, X_K)]$. However, $\mathbf{E}[\min(X_{K-1}, X_K)] \geq 2^{2^{K-1}} = \omega(K)$. Continuing the same argument on I_{K-2}, I_{K-3}, \dots , it can now be shown that GREEDY chooses $X_K, X_{K-1}, \dots, X_{\log \log K}$ in order to achieve objective value $\Theta(K)$. In other words, greedy spends cost $\Omega(K)$ in order to achieve minimum value of $\Theta(K)$. The range M^2 of the distributions satisfies $K = \log \log M$, and further, $V = \mathbf{E}[\min(\min_{i=1}^K X_i, Y_1, Y_2)] = \Omega(1)$. Therefore, GREEDY needs to exceed the cost budget by a factor of $\Omega(\log \log \frac{M}{V})$ in order to achieve any constant factor approximation to the minimum value. \square

5 Improved Approximation Algorithms

The GREEDY algorithm has approximation ratio which depends (inversely) on the value of the optimal solution on that instance (a lower bound on which is $V = \mathbf{E}[\min_{i=1}^n X_i]$, which could be exponentially small). The approximation ratio increases as the value of the optimal solution reduces. However, the algorithm is desirable since the dependence is of the form $\log \log V$.

We now present an algorithm CIP-GREEDY whose approximation ratio (on the cost) of $O(\log m + \gamma_\epsilon)$ depends just on m , the number of possible values that can be taken by the input discrete distributions. (Recall the notation from Theorem 4.4.) This algorithm is based on computing a good start solution for GREEDY using linear program rounding. This also has the advantage of yielding an algorithm for the case when the input distributions are integer valued over a small range – the approximation ratio in this case depends only on the range and not on the value, V . We finally show that if each distribution is log-concave, the approximation on the cost is $O(\gamma_\epsilon)$ in order to obtain a $(1 + \epsilon)$ approximation on the minimum value. The approximation factor therefore becomes *independent* of the parameters of the distribution.

5.1 CIP-GREEDY Algorithm

We first consider the case where the X_i are discrete distributions over m values. We present a $O(\log m)$ approximation (on the cost) by combining the greedy algorithm with column monotone CIPs. Let the number of distinct values taken by the discrete distributions be m , corresponding to the intervals I_1, I_2, \dots, I_m . Let l_j denote the length of the interval I_j . Let X^* denote the value of the optimal solution to MINIMUM-ELEMENT. Let $a_{ji} = \log \frac{1}{p_{ij}}$. The CIP-GREEDY algorithm runs in the following three steps.

Step 1: The first step of the algorithm defines the following integer program:

$$\begin{aligned} & \text{Minimize } z \\ & \sum_{i=1}^n y_i a_{ji} \geq \log l_j - z \quad \forall j = \{1, 2, \dots, m\} \\ & \sum_{i=1}^n c_i y_i \leq C \\ & y_i \in \{0, 1\} \quad \forall i \in \{1, 2, \dots, n\} \end{aligned}$$

Claim 5.1. $z = \log X^*$ is feasible for the above IP.

Proof. Consider the optimal solution S to MINIMUM-ELEMENT and set $y_i = 1$ if $i \in S$ and 0 otherwise. Clearly, we have $\sum_i c_i y_i \leq C$. Furthermore, the IP constraints together with the definition of X^* implies the following:

$$X^* = \mathbf{E}[\min_{i \in S} X_i] = \sum_{j=1}^m l_j \prod_{i \in S} p_{ij} = \sum_{j=1}^m l_j \left(\prod_i p_{ij}^{y_i} \right) \geq \max_j \left(l_j \left(\prod_i p_{ij}^{y_i} \right) \right)$$

Taking logs, we have

$$\log X^* \geq \max_j \left(\log l_j - \sum_i y_i a_{ji} \right)$$

Rearranging, this implies:

$$\sum_{i=1}^n y_i a_{ji} \geq \log l_j - \log X^* \quad \forall j = \{1, 2, \dots, m\}$$

Therefore $z = \log X^*$ satisfies all the constraints of the LP. \square

The next claim follows using the same proof argument as the previous claim.

Claim 5.2. *Consider any feasible solution to the above IP with cost budget C' and objective z . Then the subset S corresponding to i s.t. $y_i = 1$ satisfies: $\sum_{i \in S} c_i \leq C'$, and $\mathbf{E}[\min_{i \in S} X_i] \leq me^z$.*

Proof. Clearly, we have $\sum_i c_i y_i = \sum_{i \in S} c_i \leq C'$. Further, we have:

$$z \geq \log l_j - \sum_{i=1}^n y_i a_{ji} \Rightarrow l_j \prod_i p_{ij}^{y_i} \leq e^z \quad \forall j = 1, 2, \dots, m$$

We finally have:

$$\mathbf{E}[\min_{i \in S} X_i] = \sum_{j=1}^m l_j \prod_{i \in S} p_{ij} = \sum_{j=1}^m l_j \prod_i p_{ij}^{y_i} \leq me^z$$

□

Step 2: Solve the linear relaxation of the IP presented in Step (1), with cost budget C . Suppose the optimal solution has value z^* . From Claim 5.1, $z^* \leq \log X^*$. Re-write the LP relaxation as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_i c_i y_i \\ \sum_{i=1}^n y_i a_{ji} \geq & \log l_j - z^* \quad \forall j = \{1, 2, \dots, m\} \\ y_i \in & [0, 1] \quad \forall i \in \{1, 2, \dots, n\} \end{aligned}$$

It is clear by the choice of z^* that there exists a solution to the above LP with objective value at most C . Note further that the above is the LP relaxation of a covering integer program (CIP). This CIP is COLUMN-MONOTONE, but we do not use that fact below. We now use the following proposition proved in [Carr *et al.* 2000; Kolliopoulos and Young 2001].

Proposition 5.3. *Given a CIP with m constraints whose LP relaxation has a feasible solution, we can find an integral solution (all y_i either 0 or 1) in poly-time with approximation ratio $O(\log m)$ against the LP solution.*

The above implies that there is a solution to the above LP where all y_i are either 0 or 1; that respects all the constraints; and that has objective value (or cost) $O(C \log m)$. Using Claim 5.2 with $C' = O(C \log m)$ and $z \leq \log X^*$, this integer solution corresponds to a subset S^0 such that $\mathbf{E}[\min_{i \in S^0} X_i] \leq mX^*$, and $\sum_{i \in S^0} c_i = O(C \log m)$.

Step 3: Run the greedy algorithm in Figure 2 with initial solution S^0 and with cost budget $\tilde{C} = C(\log \log m + \gamma_\epsilon)$, where $\gamma_\epsilon = \log \frac{1}{\log(1+\epsilon)}$. Let the final solution be S^f .

Claim 5.4. $\mathbf{E}[\min_{i \in S^f} X_i] \leq (1 + \epsilon)X^*$.

Proof. As in the proof of Theorem 4.4, we start with the statement of Theorem 4.3. Use $g(S) = -\log \mathbf{E}[\min_{i \in S} X_i]$. Since the greedy algorithm starts with S^0 , we have $g(\Phi) = g(S^0) = -\log(mX^*)$. Further, $g(S^*) = -\log X^*$. Our choice of cost budget implies $S^f = T_{\log(1+\epsilon)}$, so that $g(S^f) \geq -\log((1 + \epsilon)X^*)$. Therefore, $\mathbf{E}[\min_{i \in S^f} X_i] \leq (1 + \epsilon)X^*$. □

We finally have the following theorem:

Theorem 5.5. *For discrete distributions on m values, the CIP-GREEDY algorithm achieves a $O(\log m + \gamma_\epsilon)$ approximation on the cost in order to find a solution of value at most $(1 + \epsilon)X^*$.*

5.2 Algorithm for Polynomial Input Domain

We now improve the above result when the domain of the distributions is the set $\{0, 1, \dots, M-1\}$. Assume w.l.o.g. that M is a power of 2. We first group the intervals so that the lengths are increasing in powers of 2, and the first interval has length 1. For each distribution X_i , construct a new distribution \tilde{X}_i over the domain $R = \{0, 1, 2, 4, \dots, 2^{\log M}\}$ as follows: Set $\tilde{p}_{i0} = p_{i0}$, and for $k \in R \setminus \{0\}$, set $\tilde{p}_{ik} = \sum_{j=\lceil k/2 \rceil}^k p_{ij}$. Let X^* denote the value of MINIMUM-ELEMENT for the distributions X_i and let Y^* denote the corresponding value for the distributions \tilde{X}_i . It is easy to show that $X^* \leq Y^* \leq 2X^*$.

The algorithm is as follows: Run Steps (1) and (2) of the CIP-GREEDY algorithm using the distributions \tilde{X}_i . Setting $m = \log M$, it is easy to see that this produces a solution S^0 so that $\mathbf{E}[\min_{i \in S^0} X_i] \leq 2X^* \log M$ using cost $O(C \log \log M)$. Now run the greedy algorithm in Figure 2 using starting solution S^0 , and using the original distributions X_i , with budget set to $\tilde{C} = C(\log \log \log(2M) + \gamma_\epsilon)$, where $\gamma_\epsilon = \log \frac{1}{\log(1+\epsilon)}$. Using the same proof as Theorem 4.4, this yields a solution of value $(1 + \epsilon)X^*$. We therefore have the following theorem:

Theorem 5.6. *In the MINIMUM-ELEMENT problem, when the domain of values is $\{0, 1, \dots, M-1\}$, then the approximation ratio (on the cost) of the modified CIP-GREEDY is $O(\log \log M + \gamma_\epsilon)$, and this achieves a solution of value $(1 + \epsilon)X^*$.*

Remarks. An interesting open question is to improve the above approximation factor to $O(1)$ on the cost. The super-constant approximation ratio is in sharp contrast with the lack of even a NP-HARDNESS result for the case where M is polynomially bounded. (Note that the NP-HARDNESS proof required exponentially large values in the domain.)

Next, note that in both the cases discussed in this section, the bottleneck in the approximation ratio is in solving the covering program. In particular, replacing the GREEDY algorithm in each case by the *naive* greedy algorithm that adds the variable $\operatorname{argmax}_i \frac{f(S) - f(S \cup \{X_i\})}{c_i}$ at each step, would also yield the same approximation ratio. This strongly suggests that the m -constraint COLUMN-MONOTONE CIPs that arise in these settings have a better approximation ratio than $O(\log m)$. Note that we have only been able to show NP-HARDNESS for these problems, and the hardness of approximation proofs for general CIPs do not carry over to COLUMN-MONOTONE CIPs.

5.3 Log-concave Distributions

We now show that a slightly modified greedy algorithm actually performs much better when each distribution X_i is a log-concave distribution. Let $F_i(r) = \Pr[X_i \geq r]$. A distribution is said to be *log-concave* if $\log F_i(r)$ is a concave function of r . We note that several common distributions such as uniform, Gaussian, and Beta densities¹ are known to be log-concave [6].

The problem with using the greedy algorithm directly is that initially, the algorithm could make a sequence of wrong choices which are costly to rectify. We show that if the distributions are truncated near the optimal solution value, such a problem cannot arise, and therefore the greedy algorithm performs much better. To get around the issue of not knowing the optimal solution value, we try all possible truncations in powers of 2. In order to describe the new algorithm, we first define the *truncation* of a distribution:

Definition 5.7. X^t is a truncation of X at point t if $\Pr[X^t = t] = \Pr[X \geq t]$, and $\Pr[X^t = r] = \Pr[X = r]$ for $r < t$.

The algorithm is presented in Figure 3. It clearly has polynomial running time since it tries values of t in powers of 2, so that the number of tries is polynomial in the bit complexity of the input. As before, let S^* denote the optimal solution and $X^* = \mathbf{E}[\min_{i \in S^*} X_i]$.

¹Beta(a, b) is log-concave when $a, b \geq 1$.

MODIFIED GREEDY (C, ϵ) */* $\epsilon \leq 0.2$ */*

$\tilde{C} \leftarrow 2C\gamma_\epsilon$
 $t \leftarrow \min_i \mathbf{E}[X_i]/\epsilon$
While $t \geq \mathbf{E}[\min_i X_i]$ **do**:
 $S^t \leftarrow$ GREEDY solution on X_1^t, \dots, X_n^t with cost bound \tilde{C} .
 $t \leftarrow t/2$.
endwhile
Output S^t with minimum $\mathbf{E}[\min_{i \in S^t} X_i]$.

Figure 3: Modified Greedy Algorithm for Log-Concave Distributions.

We will use the following lemma about the log-concave densities [6].

Lemma 5.8. *Consider any distribution X with $F(r) = \Pr[X \geq r]$. If $\log F(r)$ is concave, then*

$$g(r) = \frac{\int_{x=r}^{\infty} F(x)dx}{F(r)}$$

is monotonically non-decreasing.

The crux of the proof is the following lemma, which in effect states that truncation around X^*/ϵ preserves the expected minimum of all solutions whose original minimum was close to X^* .

Lemma 5.9. *Let $t = \alpha \frac{X^*}{\epsilon}$ for $\alpha \in [1/2, 1]$. For any set S with $\mathbf{E}[\min_{i \in S} X_i] = q \geq X^*$, either $\mathbf{E}[\min_{i \in S} X_i^t] \geq 1.2X^*$ or $\mathbf{E}[\min_{i \in S} X_i^t] \geq (1 - 5\epsilon)q$.*

Proof. Consider any set S and $q = \mathbf{E}[\min_{i \in S} X_i] \geq X^*$. Let $t = \alpha \frac{X^*}{\epsilon}$ for $\alpha \in [1/2, 1]$. Define the following:

$$F_S(t) = \Pr[\min_{i \in S} X_i \geq t] = \prod_{i \in S} F_i(t)$$

Since each X_i is log-concave, we have that $\log F_i(r)$ is concave. By linearity, this implies $\log F_S(t)$ is concave as well. Therefore, from the previous lemma, we have that:

$$g_S(t) = \frac{\int_{x=t}^{\infty} F_S(x)dx}{F_S(t)}$$

is a monotonically non-increasing function of t . We split the analysis into two cases:

Case 1: If $F_S(t) \geq 2.4\epsilon$, then,

$$\mathbf{E}[\min_{i \in S} X_i^t] = \int_{x=0}^t F_S(x)dx \geq tF_S(t) \geq \alpha \frac{X^*}{\epsilon} 2.4\epsilon \geq 1.2X^*$$

where we have used $\alpha \geq 1/2$.

Case 2: If $F_S(t) \leq 2.4\epsilon$, since $g_S(0) = \mathbf{E}[\min_{i \in S} X_i] \geq g_S(t)$, we have:

$$\int_{x=t}^{\infty} F_S(x)dx \leq \mathbf{E}[\min_{i \in S} X_i] F_S(t) \leq 5\epsilon q$$

Therefore,

$$\mathbf{E}[\min_{i \in S} X_i^t] = \int_{x=0}^t F_S(x)dx = \mathbf{E}[\min_{i \in S} X_i] - \int_{x=t}^{\infty} F_S(x)dx \geq q(1 - 5\epsilon)$$

□

Theorem 5.10. Let $\gamma_\epsilon = \log \frac{1}{\log(1+\epsilon)}$. For $\epsilon \leq 0.2$, the modified GREEDY algorithm yields a solution of value $(1 + 7\epsilon)X^*$ for log-concave distributions using cost $2C\gamma_\epsilon$.

Proof. During some point in the execution of the algorithm, we will have $t = \alpha \frac{X^*}{\epsilon}$ for $\alpha \in [1/2, 1]$. Consider this value of t . For any set S with cost at most C , we have $\mathbf{E}[\min_{i \in S} X_i] \geq X^*$. Applying the previous lemma, it is clear that $\mathbf{E}[\min_{i \in S} X_i^t] \geq (1 - 5\epsilon)X^*$.

Note now that $\mathbf{E}[X_i^t] \leq X^*/\epsilon$ for all i . Therefore, using $g(S) = -\log \mathbf{E}[\min_{i \in S} X_i^t]$ (with $g(\Phi) = -\log(X^*/\epsilon)$, $g(S^*) = -\log((1 - 5\epsilon)X^*)$, and $g(S^t) = -\log((1 + \epsilon)X^*)$), and repeating the proof of Theorem 4.4 yields (for $\epsilon \leq 0.2$):

$$\tilde{C} \leq C \left(\log \log \frac{1 + 5\epsilon}{\epsilon} + \gamma_\epsilon \right) \leq 2C\gamma_\epsilon$$

This yields a set S^t such that $\mathbf{E}[\min_{i \in S^t} X_i^t] \leq (1 + \epsilon)X^* < 1.2X^*$.

Now, either $\mathbf{E}[\min_{i \in S^t} X_i] \leq X^*$, in which case we are done; or we apply the previous lemma to show that $\mathbf{E}[\min_{i \in S^t} X_i] \leq (1 + 5\epsilon)\mathbf{E}[\min_{i \in S^t} X_i^t] \leq (1 + 5\epsilon)(1 + \epsilon)X^* \leq (1 + 7\epsilon)X^*$ for $\epsilon \leq 0.2$. \square

6 The Mixed Model

So far the only variables we were allowed to use in our solution were the ones that we observed. In general, our solution can use both probed and unprobed variables: If the minimum value among the probed set is larger than the expected value of a variable that has not been probed, we would prefer to use that variable as opposed to one of the probed values.

We show that the restriction of using only the probed set does not significantly alter the problem:

Theorem 6.1. *In order to achieve the same (or better) objective for MINIMUM-ELEMENT, the solution that uses only probed variables probes at most one more variable than the solution that is allowed to use unprobed variables.*

Proof. Consider the optimal solution in the mixed model. Suppose it probes set S^* and let X^* denote the variable not in S^* with the smallest expectation. The strategy is to probe S^* and if the minimum value observed is larger than $\mathbf{E}[X^*]$, output X^* . The value of the solution is given by $\mathbf{E}[\min(\min_{Y \in S^*} Y, \mathbf{E}[X^*])]$. Consider now the solution that probes $S^* \cup \{X^*\}$. The value of this solution is $\mathbf{E}[\min(\min_{Y \in S^*} Y, X^*)]$. It is easy to see that this value is smaller than the value of the optimal strategy for the mixed model. \square

7 Conclusions

We have presented a framework (along with simple greedy algorithms) for studying the cost-value trade-off in resolving uncertainty based on the objective function being optimized. This paradigm will increasingly play a role in model-driven optimization in sensor networks and other complex distributed systems.

In the context of MINIMUM-ELEMENT our work presents interesting open questions. First, there is a huge gap between the lower bounds and approximation ratios we show. Can this gap be closed? In particular, can logarithmic hardness be shown for the general case, and NP-HARDNESS for the case where the domain is restricted to be poly-bounded. Furthermore, can the algorithms be extended to the case where the observations are adaptive, *i.e.*, based on the results of previous observations? In subsequent work [Guha *et al.* 2008], a poly-time optimal algorithm is presented for the problem of adaptively probing for the maximum value, with the objective being to maximize (over adaptive choices of S), the quantity $\mathbf{E}[\max_{i \in S} X_i] - \alpha \sum_{i \in S} c_i$. The same algorithm also optimizes $\mathbf{E}[\min_{i \in S} X_i] + \alpha \sum_{i \in S} c_i$ over adaptive

choices of S . However, these techniques do not extend to the non-adaptive MINIMUM-ELEMENT problem with cost budgets considered in this paper. We leave this aspect as a challenging open question.

Acknowledgments: We thank Shivnath Babu, Utkarsh Srivastava, Sampath Kannan and Brian Babcock for helpful discussions.

References

- [1] A. Akella, B. M. Maggs, S. Seshan, A. Shaikh, and R. K. Sitaraman. A measurement-based analysis of multihoming. In *ACM SIGCOMM Conference*, pages 353–364, 2003.
- [2] R. Avnur and J. M. Hellerstein. Eddies: Continuously adaptive query processing. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 261–272, 2000.
- [3] B. Babcock and S. Chaudhuri. Towards a robust query optimizer: A principled and practical approach. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 119–130, 2005.
- [4] B. Babcock and C. Olston. Distributed top-k monitoring. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 28–39, 2003.
- [5] S. Babu, P. Bizarro, and D. J. DeWitt. Proactive reoptimization. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pages 107–118, 2005.
- [6] M. Bagnoli and T. Bergstrom. Log-concave probability and its applications. *Working paper, University of Michigan*, 1989.
- [7] R. Carr, L. Fleischer, V. Leung, and C. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 106–115, 2000.
- [8] M. Charikar, R. Fagin, J. Kleinberg, P. Raghavan, and A. Sahai. Query strategies for priced information. *J. Comput. Syst. Sci.*, 64(4), 2002.
- [9] F. Chu, J. Halpern, and J. Gehrke. Least expected cost query optimization: What can we expect? *Proc. ACM Symp. on Principles of Database Systems (PODS)*, pages 293–302, 2002.
- [10] F. Chu, J. Halpern, and P. Seshadri. Least expected cost query optimization: An exercise in utility. In *Proc. ACM Symp. on Principles of Database Systems (PODS)*, pages 138–147, 1999.
- [11] B. Dean, M. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *Proc. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 208–217, 2004.
- [12] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proc. Intl. Conf. on Very Large Data Bases (VLDB)*, pages 588–599, 2004.
- [13] T. Feder, R. Motwani, R. Panigrahy, C. Olston, and J. Widom. Computing the median with uncertainty. *SIAM J. Comput.*, 32(2), 2003.
- [14] A. Goel, S. Guha, and K. Munagala. Asking the right questions: Model-driven optimization using probes. In *Proc. ACM Symp. on Principles of Database Systems (PODS)*, pages 203–212, 2006.

- [15] A. Goel and P. Indyk. Stochastic load balancing and related problems. In *Proc. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 579–586, 1999.
- [16] S. Guha and K. Munagala. Model-driven optimization using adaptive probes. In *Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 308–317, 2007.
- [17] S. Guha and K. Munagala. Sequential design of experiments via linear programming. *Computing Research Repository, arXiv:0805.2630*, 2008. Preliminary version appeared in the ACM Symposium on Theory of Computing (STOC), 2007.
- [18] S. Guha, K. Munagala, and S. Sarkar. Information acquisition and exploitation in multi-channel wireless systems. *Computing Research Repository, arXiv:0804.1724*, 2008. Preliminary version appeared in the ACM SIGMETRICS Conference, 2006.
- [19] P. K. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall. Improving the reliability of internet paths with one-hop source routing. In *6th Symposium on Operating System Design and Implementation (OSDI)*, pages 183–198, 2004.
- [20] A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In *Proc. ACM Symp. on Theory of Computing (STOC)*, pages 417–426, 2004.
- [21] A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: LP rounding approximation algorithms for stochastic network design. In *Proc. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 218–227, 2004.
- [22] N. Immorlica, D. Karger, M. Minkoff, and V. Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 691–700, 2004.
- [23] S. Khanna and W-C. Tan. On computing functions with uncertainty. In *Proc. ACM Symp. on Principles of Database Systems (PODS)*, 2001.
- [24] J. Kleinberg, Y. Rabani, and É. Tardos. Allocating bandwidth for bursty connections. *SIAM J. Comput.*, 30(1), 2000.
- [25] S. Kolliopoulos and N. Young. Tight approximation results for general covering integer programs. In *Proc. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 522–528, 2001.
- [26] A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proc. 21st Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 324–331, 2005.
- [27] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-I. *Math Programming*, 14(1):265–294, 1978.
- [28] C. Olston. *Approximate Replication*. PhD thesis, Stanford University, 2003.
- [29] D. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as discrete optimization. In *Proc. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 228–237, 2004.
- [30] A. Silberstein, R. Braynard, C. Ellis, K. Munagala, and J. Yang. A sampling based approach to optimizing top-k queries in sensor networks. In *Proc. Intl. Conf. on Data Engineering (ICDE)*, pages 68–77, 2006.