

Asking the Right Questions: Model-driven Optimization using Probes

Ashish Goel
Department of Management
Sci. & Engg. and (by
courtesy) Computer Science
Stanford University
Stanford, CA
ashishg@stanford.edu

Sudipto Guha
Department of Computer and
Information Sciences
University of Pennsylvania
Philadelphia, PA
sudipto@cis.upenn.edu

Kamesh Munagala
Department of Computer
Science
Duke University
Durham, NC
kamesh@cs.duke.edu

ABSTRACT

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

We present a framework for studying such problems, and present several scenarios arising in anomaly detection in complex systems, monitoring extreme values in sensor networks, load shedding in data stream systems, and estimating rates in wireless channels and minimum latency routes in networks, which can be modeled in this framework with the appropriate objective functions.

Even for several simple objective functions, we show the problems are NP-HARD. We present greedy algorithms with good performance bounds. The proof of the performance bounds are via novel sub-modularity arguments.

Categories and Subject Descriptors

F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity

General Terms

Algorithms, Theory

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PODS'06, June 26–28, 2006, Chicago, Illinois, USA.
Copyright 2006 ACM 1-59593-318-2/06/0003 ...\$5.00.

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 [5, 3]. As another example, a system like Eddies [2] 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. Note that this is not the same as minimizing residual entropy (Krause and Guestrin [22]) which minimizes uncertainty of the joint distribution – we are concerned with minimizing the uncertainty of an optimization that depends on the joint distribution (like the maximum value – even this simple problem turns out to be NP-HARD). Minimizing residual entropy will most often involve probing a different set of variables than those required for estimating best the specific function at hand (refer Example 1.3); therefore, the problems are orthogonal.

In this paper, we study several natural optimization problems where the inputs are random variables corresponding to parameters of the data model, in the setting that the values of one or more inputs can be observed and resolved by paying some cost. We show that even for the simplest of optimization problems, this choice becomes non-trivial and intractable, motivating the development of algorithmic techniques to address them. Abstractly, the class of problems we propose to solve can be formulated as follows:

PROBLEM 1. *We are given the distributions of n non-negative independent random variables X_1, \dots, X_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 $f(X_1, X_2, \dots, X_n)$, 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 ?

Often the only access we have to the data is to run a simpler, smaller query or use sampled estimates. The maintenance of good samples or estimates of a parameter is a challenge in itself. If we can judiciously choose which parameters need a finer or more accurate estimate, we can avoid maintaining very accurate estimates of all parameters, and only refine our estimates when needed. This motivates the following question.

PROBLEM 2. *Can we achieve the above (Problem 1) with only the access to samples of the distribution?*

In this paper we answer both questions in the affirmative. The notion of refining uncertainty has been considered in an *adversarial setting* by several researchers [26, 12, 18, 7]. 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., [9, 8, 11, 4, 5, 3, 28] 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. To demonstrate the benefit of probing and using the stochasticity, we consider a few examples.

1.1 Motivating Examples

Extreme Value Estimation: We first consider a sensor network where the root server monitors the maximum value, which is a specific case of TOP-K monitoring considered in [4, 28]. The probability distributions of the values at various nodes are known to the server. However, probing all nodes to find out their actual values is undesirable since it costs battery life at all nodes. Consider the simplest setting where the network connecting the nodes to the server is a one-level tree, and probing a node consumes battery power of that node. Given a bound on the total battery life consumed, the goal of the root server is to maximize (in expectation) its estimate of the maximum value. The problem maps to our formulation as follows:

1. X_i = Random variable denoting value at node i . This distribution is known to the server.
2. c_i = Battery life consumed at node i when probed by the server.
3. C = Bound on total battery consumed by the probing.
4. $f = \max_{i \in S} X_i$, where S is the set of probed nodes.

Of course, if the maximum value among the probed set is less 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. The reason is presented in Section 4. We now show the benefit of probing with an example.

EXAMPLE 1.1. *If only one probe is allowed, the root server's estimate is $\max_i \mathbf{E}[X_i]$. If the resource constraint is sufficient to probe all nodes, this estimate improves to $\mathbf{E}[\max_i X_i]$, since the server can find the exact values at all nodes and return the maximum value. Even if all X_i are Bernoulli $B(1, p)$ with $p < 1/n$, the former value is p and the latter is $1 - (1 - p)^n \approx np$. Therefore, probing nodes can improve the expected value returned significantly. The gap is at least n/K for sum of TOP-K.*

Route Selection in Networking: 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 [1, 14] 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:

1. X_i are the distributions of route latencies.
2. The probing cost c_i is a function of the delay and load incurred in detecting latency of route i .
3. The budget C is the total load and processing time that can be incurred by the route choosing algorithm.
4. $f = \min_{i \in S} X_i$, where S is the set of routes probed.

The goal is to choose that set of routes to probe which minimizes the expected value of the smallest latency detected. This is defined as the **MINIMUM-ELEMENT**. As with **MAXIMUM ELEMENT**, the solution can use both probed and unprobed variables; we show in Section 4 that this additional aspect can be ignored. We illustrate the benefit of probing using the following example.

EXAMPLE 1.2. *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 yield an estimate which is **exponentially smaller**, which means that if there is a low utilization, we will very likely find it.*

Query Optimization: In the context of long-running queries in a data stream processing engine, consider an overloaded system where the scheduler has to shed load, i.e., decide which queries to terminate, in order to maximize processing rate [29]. Each query contributes a given amount to the system load. A priori, the output rate of each query is only known in distributional form (since the query is long running). When the system suddenly becomes overloaded, the query optimizer needs to terminate some queries based on current output rates and loads of each query so that the output rates of the retained queries is maximized. However since load increases abruptly, the optimizer has to make this decision quickly, meaning that it has limited time to observe

the current processing rates of each query. The load shedding problem can be now modeled as a KNAPSACK problem in our framework as follows:

1. Each item i corresponds to a long running query.
2. Profit X_i corresponds to the rate of the query, and is a random variable.
3. Size s_i corresponds to the contribution of the query to the system load.
4. Probing cost c_i corresponds to the time required to exactly estimate the current rate of the query.
5. Budget C is the time available to the query processor to determine what load to shed.
6. Knapsack capacity B corresponds to the total system load permitted.

In this special case $f = \max_{T \subseteq S; \sum_{i \in T} s_i \leq B} \sum_{i \in T} X_i$. The above is a KNAPSACK problem¹ where the profit of item i with size s_i and observation cost c_i follows distribution X_i , and the knapsack capacity is B . The goal is to choose a subset S of items (or variables) whose total observation cost is at most C , such that *after* observing them we choose (possibly) a subset T which maximizes the expected profit. Note that the (maximum) KNAPSACK problem generalizes the problem of estimating the maximum value and sum of TOP-K estimation. Naturally, we can (and do) define a variant where the profit of an item (job) is fixed and the size (modeling duration of time) is a random variable.

Anomaly Detection, Data Mining: In networks and complex systems, event and performance logs are used to track anomalies like unbalanced load [24]. In a real-time environment, we are interested in finding the anomaly as fast as possible since there may be secondary effects or cost (e.g., virus spread) associated with delay. Low utilization of resources is usually an indicator of such an anomaly. In long running systems, which are typical in these environments, distributions of various performance parameters are often recorded. An anomaly detection algorithm, which wants to detect problems as they are arising, would not have time to process all the performance logs, and must judiciously choose which to process. This can be naturally modeled as a MINIMUM-ELEMENT problem.

1.2 Technical Hurdles

Consider the problem of estimating the maximum value in a sensor network in the simple case when the probing costs of all nodes are equal. Let m denote the constraint on the number of nodes that can be probed. It would appear that the optimal strategy would be to choose the m nodes with the highest expected values. The example to compute the MAXIMUM below shows that this need not be the case.

EXAMPLE 1.3. *There are 3 distributions X_1, X_2 and X_3 corresponding to the values at the three nodes. We let $m = 2$. Distribution X_1 is 2 with probability $\frac{1}{2}$ and 1 with probability $\frac{1}{2}$. Distribution X_2 is 1 with probability $\frac{1}{2}$ and 0 with probability $\frac{1}{2}$. Distribution X_3 is 2 with probability $\frac{1}{5}$ and 0*

¹In general, we may use both unprobed and probed variables in the solution. The objective function f becomes quite involved in this case and we relegate further discussion to Section 4. However to solve the general problem, it turns out that we have to solve the subproblem where only a subset T of the probed set S of jobs are retained.

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 maximum value of 1.5, while probing X_1, X_3 yields an expected maximum value of 1.6. Minimizing residual entropy [22] 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, this problem (and the MINIMUM-ELEMENT problem) becomes NP-HARD for arbitrary distributions even with uniform costs. However from the perspective of approximation guarantees, the two problems are very different. For the MINIMUM-ELEMENT problem, 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?

1.3 Variants of the Model

There are several variants to the basic model which are both of theoretical and practical interest. We focus on two aspects of the proposed framework which we plan to address in future work.

Adaptive Observations: Our problem formulation assumes the probing is *non-adaptive*. This means we first decide the *entire* set of variables to observe, and are then told the values of the observed variables. The variables are therefore observed in parallel. This is a reasonable assumption in many situations where there is not enough time to make adaptive decisions; for instance, load shedding in an overloaded data stream system [29]. Also in scenarios where the probe returns an answer after a delay (due to network or latency) adaptive probing significantly increases the latency of the answer. This is not desired in any query optimizer or monitoring scenario. However, we note that in some situations the probing can be adaptive [11, 28].

Correlations: Our model assumes the random variables are independent. In scenarios like sensor networks and processing performance logs, the values are correlated from one time step to another. Similarly, in query processing, the running times of the queries are correlated if they share predicates. Modeling correlations tractably is itself an interesting area of research [11], representing joint distributions of an arbitrarily large subset is non-trivial. Under restrictions, some of our results carry over to the correlated setting.

1.4 Results

1. We introduce the problem of model driven optimization in the presence of observations. 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.
2. For the MINIMUM-ELEMENT problem, we 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.

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 an approximation ratio $O(\log \log \frac{\min_{i \in S^*} \mathbf{E}[X_i]}{\mathbf{E}[\min_{i \in S^*} X_i]})$ where S^* is the optimal solution. We show significantly improved results for special cases of distributions which are quite common in practice. This problem is discussed in Section 2.

3. We consider the KNAPSACK problem in Section 3. The KNAPSACK problem subsumes the sum of TOP-K and MAXIMUM-ELEMENT problems, and is NP-HARD by reduction from the MINIMUM-ELEMENT problem. We present constant factor approximations to the expected profit to the KNAPSACK problem in two variants: one with profits as random variables and the other with random sizes.
4. In terms of techniques, we combine an involved sub-modularity argument along with the analysis of the best *fractional* solutions. Although the analyses are complicated, the algorithms are natural.

Related Work: Other than the literature discussed already, the work which appears closest to the results in this paper are by Dean *et. al.*, [10]. They consider 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 is made *after* the observations. There also has been ongoing research in Multi-stage stochastic optimization [20, 13, 17, 15, 16, 27], however most of this literature also involves making irrevocable commitments.

2. MINIMUM-ELEMENT

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 the that $c_i \leq C$ for all i . We further assume the distributions are specified as discrete distributions² 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 $f(\Phi) = M$.

²Continuous models introduce the issue of how the input is specified, for most smooth continuous distributions we can use a histogram with polynomially many pieces or we can use the blackbox sampling method discussed in the next section. Note that any polytime algorithm will implicitly construct a representation with polynomially many pieces.

2.1 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 section.

DEFINITION 2.1. 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 $A \in \mathbb{Z}^{m \times n}$, 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 $b_j \in \mathbb{Z}^+$ and $b_{j+1} \geq b_j$.

Suppose we are given a column-monotone CIP with a cost budget C and our goal is to determine whether the optimum value of the CIP 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 2.2. The r -GAP-CIP problem with polynomially bounded (in n and m) coefficients A_{ji} reduces, in polynomial 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 $\mu_j - \mu_{j-1} = q^{bj}$. Distribution X_i has observation cost c_i , and $\Pr[X_i \geq \mu_{j-1}] = q^{-A_{ji}}$. Observe that column-monotonicity is crucial for this definition to correspond to a valid probability distribution; the requirement that A_{ji} 's be polynomially

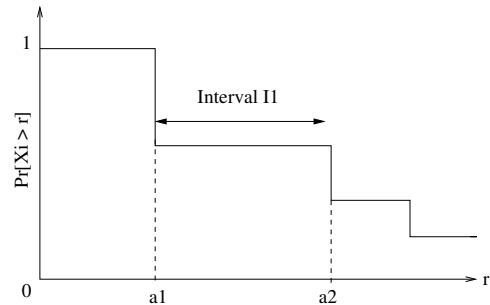


Figure 1: Notation used in MINIMUM-ELEMENT.

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 ,

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

Therefore,

$$\mathbf{E}[\min_{i \in S} X_i] = \sum_j (\mu_j - \mu_{j-1}) \Pr[\min_{i \in S} X_i \geq \mu_{j-1}] \leq m.$$

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

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

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 2.3. *It is NP-HARD to obtain any 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 2.2, 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 2.2. A hardness of approximating column-monotone CIPs will immediately lead to a stronger hardness result for the MINIMUM-ELEMENT problem via Lemma 2.2.

2.2 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 [25]. *Sub-modularity* is a discrete analogue of convexity which is the basis of many greedy approximation algorithms. We formally define sub-modularity next.

DEFINITION 2.4. *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 2.5. *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. $\alpha_j = \Pr[(\min_{Y \in A} Y) \geq a_j] = \Pr_{Y \in A} [Y \geq a_j]$.
2. $\beta_j = \Pr[(\min_{Y \in C} Y) \geq a_j] = \Pr_{Y \in C} [Y \geq a_j]$.
3. $\gamma_j = \Pr[X \geq a_j]$.

The following sequence of statements are immediate:

1. The α_j, β_j and γ_j values are non-negative and monotonically non-increasing with increasing j .
2. By the independence of the variables,

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

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

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

Figure 2: Greedy Algorithm for MINIMUM-ELEMENT .

3. Therefore,

$$\frac{f(A \cup \{X\})}{f(A)} = \frac{\sum_j l_j \alpha_j \gamma_j}{\sum_j l_j \alpha_j}$$

and

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

From above, we have

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

This implies $\log \frac{1}{j}$ is a sub-modular function. \square

The connection between sub-modular functions and the greedy algorithm is captured by the following theorem [19, 23] which generalizes to arbitrary costs the result in [25] for unit costs.

THEOREM 2.6 ([25, 19, 23]). *Given a non-decreasing sub-modular function $g()$ 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.*

1. The minimal greedy set T_1 which violates the cost constraint by at most one element has $g(T_1) - g(\Phi) \geq (1 - 1/e)(g(S^*) - g(\Phi))$.
2. Let T_2 be the maximal greedy set that obeys the cost constraint. Let $T_3 = \operatorname{argmax}\{g(T_2), \max_{i \in U} g(\{i\})\}$. Then $g(T_3) - g(\Phi) \geq \frac{1}{2}(1 - 1/e)(g(S^*) - g(\Phi))$.
3. For any ϵ , the greedy algorithm finds a maximum value set T_ϵ such that $g(T_\epsilon) \geq g(S^*) - \epsilon$ with cost $C \log \frac{g(S^*) - g(\Phi)}{\epsilon}$.

Intuitively, sub-modularity ensures that current greedy choice has cost per unit increase in value of $g()$ at most the corresponding value for the optimal solution. For MINIMUM-ELEMENT, let S^* denote the optimal solution using cost C . Also, let $V = \mathbf{E}[\min_{i=1}^n X_i]$.

THEOREM 2.7. *The greedy algorithm for MINIMUM ELEMENT achieves a $(1 + \epsilon)$ approximation to $f(S^*)$ with cost parameter $\tilde{C} = C(\log \log \frac{M}{V} + \log \frac{1}{\epsilon})$.*

PROOF. In the above theorem, set $g = \log \frac{1}{j}$. Also since $f(\Phi) = M$, we have $g(\Phi) = \log \frac{1}{M}$. Let S denote the greedy set. Suppose $f(S) \leq (1 + \epsilon)f(S^*)$. Therefore, $g(S) \geq g(S^*) - \log(1 + \epsilon) \geq g(S^*) - \epsilon$. Therefore, $T_\epsilon = S$ in the above theorem, implying its cost $\tilde{C} \leq C(\log \log \frac{M}{f(S^*)} + \log \frac{1}{\epsilon})$. Since $f(S^*) \geq V$, we have $\tilde{C} \leq C(\log \log \frac{M}{V} + \log \frac{1}{\epsilon})$. \square

Note: If we had used $f()$ (which is submodular) as suggested by a naive greedy algorithm then we would have needed a 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 2.8. *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 $i = 1, 2, \dots, K$. Distribution X_i ($i = 1, 2, \dots, K$) takes value 2^{2^i} with probability $(1 - 2^{-2^{i+1} + \epsilon})$ and takes value 2^{2^r} otherwise for $r = K + 1$. All distributions have unit cost. Optimum solution uses two distributions Y_1 and Y_2 such that the survival probability at the start of interval I_i for both of them is $2^{-2^i + \epsilon}$. The value of the optimal solution is K and its cost is 2.

We claim that greedy first chooses X_K . If greedy chose any other distribution then on the last interval I_K the contribution would be at least $(2^{2^r} - 2^{2^{r-1}})2^{-2^{r-1} + \epsilon}$. Since $2^{2^2} > 2^{2^1}$ for $x > 1$, the contribution is at least $2^{2^r - 1}2^{-2^{r-1} + \epsilon} \geq 2^{2^r - 1 + \epsilon}$. $\mathbf{E}[X_K] \leq 2^\epsilon + 2^{2^r - 1}$, which is smaller.

At this point, the contribution from the interval I_K to greedy is 2^ϵ . The contribution of the previous interval I_{K-1} is $2^{2^{r-1}}$. Clearly, greedy will reduce the contribution to this interval. Arguing inductively, greedy picks X_{K-1}, X_{K-2} , and so on. This shows that the greedy algorithm chooses all of $X_K, \dots, X_{K - \log \log K}$ in order to be competitive on the objective, and therefore spends cost $\Omega(K)$. \square

2.3 Improved Approximation Algorithm

We now show an improved algorithms when the random variables have small range. This is useful in many real life situations.

THEOREM 2.9. *A modified greedy algorithm that starts from a carefully chosen initial solution achieves a $(1 + \epsilon)$ approximation with cost $O(C \log m)$ for discrete distributions on m intervals. Further if the values of the discrete distributions come from a polynomially bounded range of values $\{0, 1, \dots, M - 1\}$ then the approximation ratio (on the cost) of a modified greedy algorithm is $O(\log \log M)$.*

PROOF. We present a $O(\log m)$ approximation (on the cost) to the MINIMUM-ELEMENT problem 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 .

DEFINITION 2.10. *The survival density function (SDF) of a random variable Y is $F(r) = \Pr[Y \geq r]$.*

Let SDF of variable X_i be denoted by $F_i(r) = \Pr[X_i \geq r]$; the value of $F_i()$ for the interval I_j is therefore p_{ij} . Let $a_{ji} = \log \frac{1}{p_{ij}}$. Thus the matrix $[a]$ is column-monotone.

We first guess the objective function value X^* . There are polynomially many guesses if the guesses are in powers of $(1 + \epsilon)$. We then write the following CIP where y_i is an indicator variable which is 1 if variable X_i is probed.

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

This program essentially insists that if S is the chosen set of variables then the area under the SDF of the solution,

$Pr[\min_{i \in S} X_i \geq r]$, is at most X^* in each interval I_j . This is satisfied by the optimum solution because the *entire area* under the SDF of the optimum solution is X^* . The optimal solution is therefore feasible for this program with $\sum_i c_i y_i \leq C$. In addition, any feasible integer solution to this program has objective value at most mX^* . We now use the following:

PROPOSITION 2.11. [6, 21]. *If a CIP with m constraints has a feasible solution, we can find a solution with approximation ratio $O(\log m)$, i.e., cost $O(C \log m)$ in this case.*

Note that the value of the solution found is mX^* since any feasible solution has at most this value. We now run the greedy algorithm starting with this solution and adding the input distributions greedily until the solution value is at most X^* . Using the above analysis of the greedy algorithm, this incurs a cost of at most $O(C \log \log \frac{mX^*}{X^*}) = O(C \log \log m)$. Therefore, the approximation is $O(\log m)$ on the cost.

To prove the second part, suppose the discrete distributions are integer valued in the range $\{0, 1, \dots, M-1\}$. We first group the intervals so that the lengths are increasing in powers of 2. There are $\log M$ groups. It is easy to see that the optimal solution discretized so that the SDF is uniform within each group has value at most $2X^*$.

We write the covering program over these $\log M$ groups and round it. The cost of the solution is $O(C \log \log M)$, and this achieves an objective value of $2X^* \log M$. We then run the greedy algorithm, which reduces the objective value to X^* by spending an additional cost of $O(C \log \log M)$. Therefore, the overall approximation on the cost is $O(\log \log M)$. \square

3. KNAPSACK

We consider two variants of the knapsack problem: the first where the profits are random variables and the second where the sizes are random variables. In each case, the goal is to choose a subset of items to observe such that the expected profit of packing the best subset of these items into the knapsack is maximized. We present greedy algorithms which achieve a constant factor approximation to the optimal expected profit.

3.1 Random Profits

We first consider the problem when the profits are random variables. The profit of item i follows distribution X_i . Item i has size $s_i \leq B$, and the knapsack capacity is B . The goal is to choose a subset S of distributions whose total cost is at most C , in order to maximize $g(S) = \mathbf{E}[\max_{Q \subseteq S, \sum_{i \in Q} s_i \leq B} \sum_{i \in Q} X_i]$.

The road-map of the proof: The function $g(S)$ is *not* sub-modular. However, we can define a different function $f(S)$ which is sub-modular, and relate $g(S)$ and $f(S)$; thereby showing $g(S)$ is approximately sub-modular. However, the hurdles are not over, we may not be able to compute $f(S)$ for arbitrary subset, and have to use an estimate $\hat{f}(S)$. For polynomially bounded values, these estimates can be based on *Black-Box* sampling of the data. Since this is a very natural and common variant of the problem, we present a $0.5(1 - \frac{1}{e}) - \frac{1}{n}$ approximation for this case based on easy to implement algorithms. The more general case, where the profits can be exponentially large with exponentially small probabilities is considered in Section 3.1.1.

LEMMA 3.1. *For any subset S , let $f(S) = \mathbf{E}[\max_{y \geq 0, y_i \leq 1, \sum_{i \in S} s_i y_i \leq B} \sum_{i \in S} X_i y_i]$ then $g(S) \geq 0.5f(S)$.*

PROOF. The function $f(S)$ denotes the expected ‘‘fractional’’ profit when S is used. $f(S)$ is computed by averaging over all samples of the profit obtained by packing the items in order of decreasing profit to size ratio, with the possibility of the last item packed fractionally, i.e., having a fractional y_i . In any scenario of values of profits of items, the integer profit is at least half the fractional profit. Therefore, $g(S) \geq 0.5f(S)$. \square

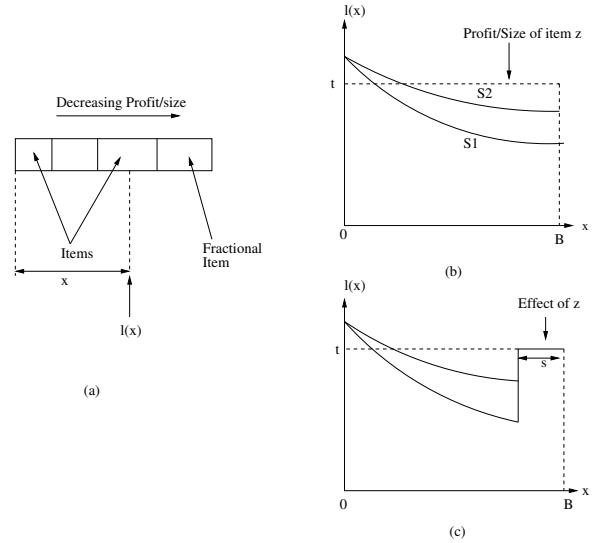


Figure 3: (a). The definition of $l(x)$ in Lemma 3.2. (b), (c). Effect of adding z in Lemma 3.2.

LEMMA 3.2. *The function $f(S)$ is sub-modular.*

PROOF. Consider any two sets $S_1 \subseteq S_2$ and a variable $z \notin S_2$. Consider any sample of profits from the joint distribution for items in $S_2 \cup \{z\}$. This naturally defines samples from the joint distributions over $S_1 \cup \{z\}$, S_2 , and S_1 by restricting the sample to these items.

Consider the items in decreasing order of ratio of profit to size in this sample. In the ordering for $S_1 \cup \{z\}$, the item z appears at no later a position than in the ordering for $S_2 \cup \{z\}$. We now show that if the addition of z increases the (fractional) profit of S_2 , it increases the profit of S_1 by at least that amount.

For a given set S , sort the items in decreasing order of profit to size ratio. For every $x \in [0, B]$, consider the fractional solution when the knapsack size is restricted to x (which is the prefix of items the sorted order with total size x) and let $l(x)$ denote the least profit to size ratio of any item in this solution. This is illustrated in Figure 3(a). Plot $l(x)$ as a function of x for S . The area under the curve is precisely the profit of the knapsack solution for items in S . The curve is also monotonically non-increasing.

Let $l_1(x)$ denote the curve for S_1 , and $l_2(x)$ the same for S_2 . Note that $l_2(x) \geq l_1(x)$ for all $x \in [0, B]$. Let the profit per unit size of z be t , and its size be s .

Now consider adding z to the two sets. If z is such that size s' fits in the knapsack in the solution for $S_2 \cup \{z\}$, the

size that fits in $S_1 \cup \{z\}$ is $s'' \geq s'$. The increase in profit (by adding z) for S_1 is $\sum_{x=B-s''}^B (t - l_1(x))$ and the quantity for S_2 is $\sum_{x=B-s'}^B (t - l_2(x))$. The latter quantity is always smaller. Refer Figure 3(b) and 3(c) for intuition. This implies the marginal increase in profit by the addition of z to S_1 will be at least as large as that to S_2 in every sample. Since f is the average of the profit over these samples, $f(S_1 \cup \{z\}) - f(S_1) \geq f(S_2 \cup \{z\}) - f(S_2)$. This implies f is sub-modular. \square

The greedy algorithm (Fig. 4) therefore yields a $\frac{1}{2}(1 - \frac{1}{e})$ approximation to the expected fractional profit using Theorem 2.6. Since the fractional profit is at most twice the best integer profit, our approximation ratio is $\frac{1}{4}(1 - 1/e)$.

```

KNAPSACK
 $X_m \leftarrow \operatorname{argmax}_{X_i} \mathbf{E}[X_i]$ .
 $S \leftarrow \Phi$ 
While ( $\sum_{i \in S} c_i \leq C$ )
     $X_q \leftarrow \operatorname{argmax}_i \frac{\tilde{f}(S \cup \{X_i\}) - \tilde{f}(S)}{c_i}$ .
     $S \leftarrow S \cup \{X_q\}$ 
endwhile
 $X_f \leftarrow$  last variable chosen in above loop.
Output  $\operatorname{argmax} (\tilde{f}(S \setminus \{X_f\}), \tilde{f}(\{X_m\}))$ 

```

Figure 4: Greedy Algorithm for KNAPSACK.

Estimating the value of f : We now show how to approximate $f(S)$ efficiently. Given a subset S , the function $f(S)$ is estimated by sampling from the joint distribution of profits of items in S . For every sample, compute the best fractional profit and average this value over all samples. The greedy algorithm uses the estimates \tilde{f} obtained from the samples instead of the function f .

Let $i^* = \operatorname{argmax}_i \mathbf{E}[X_i]$. Let $t_{\max} = \mathbf{E}[X_{i^*}]$. For any distribution X_i , let $X_i \leq mt_{\max}$. We assume m is polynomially bounded. We remove this restriction in Section 3.1.1. The greedy algorithm is run on items i for which $\mathbf{E}[X_i] \geq t_{\max}/n^2$. The contribution of all i with $\mathbf{E}[X_i] \leq t_{\max}/n^2$ to any $f(S)$ is at most t_{\max}/n . Since the final solution found in Figure 4 has value at least t_{\max} , ignoring these “small” items i changes the solution by at most a factor of $(1 - 1/n)$. Therefore we only need to estimate $f(S)$ for which $\mathbf{E}[\tilde{f}(S)] = f(S) \geq t_{\max}/n^2$.

LEMMA 3.3. *Let S be any set such that $f(S) \geq t_{\max}/n^2$. A sample size of $3mn^6$ is sufficient to estimate f approximately by \tilde{f} such that $|\tilde{f}(S) - f(S)| \leq 1/nf(S)$ with probability at least $1 - e^{-n}$.*

PROOF. By the assumption $X_i \leq mt_{\max}$, the maximum profit in any sample is at most mnt_{\max} . Since $\mathbf{E}[\tilde{f}(S)] \geq t_{\max}/n^2$ By a suitable application of Chernoff bounds: $\Pr[|f(S) - \tilde{f}(S)| \geq \frac{f(S)}{n}] \leq e^{-\frac{3mn^6 t_{\max}}{2mn^5 t_{\max}}} \leq e^{-n}$. \square

The following theorem follows immediately. Note that the theorem holds even if the distributions are specified as Black boxes from which we can sample.

THEOREM 3.4. *The greedy algorithm for KNAPSACK (presented in Figure 4), using the estimates \tilde{f} , computes a $0.25(1 - 1/e) - O(\frac{1}{n})$ approximation with very high probability.*

PROOF. Since there are 2^n sets in all, by the previous lemma and the union bound, with high probability, $(1 - 1/n)f(S) \leq \tilde{f}(S) \leq (1 + 1/n)f(S)$ for all such sets S . Let S^* denote the optimal set. We have $|\tilde{f}(S^*) - f(S^*)| \leq \frac{f(S^*)}{n}$. The greedy algorithm using the samples, computes S_0 such that $\tilde{f}(S_0) \geq \tilde{f}(S^*)0.5(1 - 1/e)$. Again, $|\tilde{f}(S_0) - f(S_0)| \leq \frac{f(S_0)}{n} \leq \frac{f(S^*)}{n}$. Therefore, $f(S_0) \geq f(S^*)(0.5(1 - 1/e) - 2/n)$. \square

3.1.1 Exponentially Large Profits

We now present the solution to the KNAPSACK problem when the profits are random variables taking on possibly exponentially large values. One of the hurdles to efficient sampling in this case is the presence of very high profit items which occur with very (exponentially) low probability. The total profit from these “problem” situations is considered separately. In these situations, at least one of the variables is in the low probability high profit scenario.

Let $Y^* = \max_i \mathbf{E}[X_i]$. Let OPT be the value of the optimal solution.

PROPOSITION 3.5. $Y^* \leq OPT \leq (n + 1)Y^*$.

The above proposition follows from the fact that the best solution can pick everything (regardless of sizes) and Y^* will at least pick the largest profit item (in expectation).

Let $\mathbf{E}_i[X] = \int_{r=l}^{\infty} r f_X(r) dr$, where f_X denotes the probability density function of random variable X .

LEMMA 3.6. *For any set S of random variables queried, let J denote the set of events s.t. at least one of the X_i ’s queried is above $n^3 Y^*$. The contribution of all events in J to the expected profit of the knapsack is at most $\sum_{i \in S} \mathbf{E}_{n^3 Y^*}[X_i] + \frac{Y^*}{n}$ and at least $(1 - \frac{1}{n^2}) \sum_{i \in S} \mathbf{E}_{n^3 Y^*}[X_i]$ with probability at least $1 - 1/n^2$.*

PROOF. By Markov’s inequality, $\Pr[X_i \geq n^3 Y^*] \leq \frac{1}{n^3}$. Now conditioned on the fact that $X_i \geq n^3 Y^*$, the maximum contribution from all the other variables is at most nY^* . The net contribution of X_i is therefore at most $\mathbf{E}_{n^3 Y^*}[X_i] + nY^* \frac{1}{n^3}$. Summing over all i proves the first part of the claim.

For the second part, simply observe that conditioned on the event $X_i \geq n^3 Y^*$, the probability that there is some other j such that $X_j \geq n^3 Y^*$ is at most $\frac{1}{n^2}$. \square

The final algorithm chooses the solution of larger value among these three:

1. Choose just the highest expected profit item.
2. Compute the set S with cost at most C which maximizes $\sum_{i \in S} \mathbf{E}_{n^3 Y^*}[X_i]$. This is simply an instance of knapsack where the profits are the $\mathbf{E}_{n^3 Y^*}[X_i]$ and the sizes are the c_i . The value of the solution can be approximated to factor of $(1 - \epsilon)$ using the standard dynamic programming algorithm. In any scenario, we choose at most one item from this set to place in the knapsack.
3. Ignore profit values larger than $n^3 Y^*$ in the distributions. Compute the set S that maximizes $f(S)$ (the expected fractional knapsack profit of choosing set S) subject to the cost constraint using the greedy algorithm. By Theorem 3.4, this can be approximated to factor of $\frac{1}{4}(1 - \frac{1}{e})$ with high probability using estimation of f by sampling combined with the greedy algorithm.

THEOREM 3.7. *The approximation ratio of the above algorithm is at least $\frac{1}{8}(1 - \frac{1}{e})$ on the profit, while respecting the cost constraint.*

PROOF. In any optimal solution which chooses set S to observe, let J be the set of events where one of the observed variables has profit larger than $n^3 Y^*$.

If the optimal solution obtains at least half its expected profit from events in J , by Lemma 3.6, the profit of these events is at most $\sum_{i \in S} \mathbf{E}_{n^3 Y^*} [X_i] + \frac{Y^*}{n}$. The set T chosen in the second step approximately maximizes this quantity. The profit from T is at least $(1 - \frac{1}{n^2}) \sum_{i \in T} \mathbf{E}_{n^3 Y^*} [X_i]$. The profit of the solution is at least Y^* by the first step. Therefore, this solution is within a factor of $1/2$ of the optimal profit.

If the optimal solution obtains more than half its profit from events not in J , choose a set in the third step which is a $\frac{1}{4}(1 - 1/e)$ approximation to the best possible profit if all distributions were truncated at $n^3 Y^*$. \square

3.2 Random Job Sizes

Consider the situation where the sizes of the items are random variables and the profits are deterministic values. Item i has size which is a random variable X_i , profit t_i , and observation cost c_i . The knapsack capacity is B . We assume $0 \leq X_i \leq B$ for all items i . The goal is to choose a set S^* of items to probe and estimate the exact sizes of such that $\sum_{i \in S^*} c_i \leq C$, and the expected profit $g(S^*)$ of packing in the knapsack is maximized, where for a set S , $g(S) = \mathbf{E}[\max_{Q \subseteq S, \sum_{i \in Q} X_i \leq B} \sum_{i \in Q} t_i]$.

Denote by $f(S)$ the fractional profit that is obtained by probing the set S and packing a subset in the knapsack. Therefore, $f(S) = \mathbf{E}[\max_{\bar{y} \geq 0, y_i \leq 1, \sum_{i \in S} X_i y_i \leq B} \sum_{i \in S} t_i y_i]$. As shown above, $g(S) \geq 0.5f(S)$. Using the same proof ideas as in the previous subsections it is easy to show the following.

LEMMA 3.8. *The function $f(S)$ is submodular.*

LEMMA 3.9. *A sample size of $3n^4$ is sufficient to estimate f approximately by \tilde{f} such that $(1 - 1/n)f(S) \leq \tilde{f}(S) \leq (1 + 1/n)f(S)$ with probability at least $1 - e^{-n}$.*

By Theorem 2.6, we approximate $f(S)$ to a factor of $1/2(1 - 1/e)$, which implies a $\frac{1}{4}(1 - 1/e)$ approximation to the optimal value of $g(S)$. Using the estimates \tilde{f} instead of f we get (including for Black-box sampling),

THEOREM 3.10. *The greedy algorithm computes a $\frac{1}{4}(1 - 1/e) - \frac{1}{n}$ approximation with very high probability.*

4. 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. For instance, in the MINIMUM-ELEMENT problem, 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 first show that the restriction of using only the probed set does not matter in the case of finding the MINIMUM-ELEMENT (and similarly for MAXIMUM ELEMENT).

THEOREM 4.1. *In order to achieve the same (or better) objective value 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 the expression $\mathbf{E}[\min(\min_{Y \in S^*} Y, \mathbf{E}[X^*])]$. 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

However for KNAPSACK with profits as random variables, the issue is more complicated since we can use both unprobed and probed values in the solution. The objective function $f(S)$ for a set of probed items S is the best expected profit of packing items into the knapsack when the profit of items in S are their observed values, and the profits for items not in S are their expected values. This function is no longer sub-modular.

The algorithm separates the problem into the probed and unprobed parts: If a variable X_i is not probed, its profit is simply $\mathbf{E}[X_i]$. Therefore, the profit of the unprobed part is at most the profit of the knapsack instance where all profits are their expectations. For the profit of the probed part, use the algorithm for knapsack from Section 3 to compute a $O(1)$ approximation. Of the two solutions, choose the one with the larger value. Since the optimal solution can be similarly decomposed, we have the following theorem.

THEOREM 4.2. *For KNAPSACK, the above algorithm yields an $O(1)$ approximation to the optimal profit.*

5. 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. As future work, we plan to enhance the model with adaptive observations, correlated random variables, other metrics measuring the trade-off (like the difference between value and cost), observing time-evolving processes, and optimizing more complex objective functions.

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

Ashish Goel's research is supported by an NSF Career award and an Alfred P. Sloan faculty fellowship. Sudipto Guha's research is supported in part by an Alfred P. Sloan Research Fellowship and by an NSF Award CCF-0430376. Kamesh Munagala's research is supported in part by NSF CNS-0540347.

6. 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 and P. Bizarro. Proactive reoptimization. In *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data*, 2005.
- [6] R. Carr, L. Fleischer, V. Leung, and C. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proc. of the Annual ACM-SIAM Symp. on Discrete Algorithms*, 2000.
- [7] M. Charikar, R. Fagin, J. Kleinberg, P. Raghavan, and A. Sahai. Querying priced information. *Proc. of the Annual ACM Symp. on Theory of Computing*, 2000.
- [8] F. Chu, J. Halpern, and J. Gehrke. Least expected cost query optimization: What can we expect? *Proc. of the ACM Symp. on Principles of Database Systems*, 2002.
- [9] F. Chu, J. Halpern, and P. Seshadri. Least expected cost query optimization: An exercise in utility. In *Proc. of the ACM Symp. on Principles of Database Systems*, 1999.
- [10] B. Dean, M. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Proc. of the Annual Symp. on Foundations of Computer Science*, 2004.
- [11] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proc. of the 2004 Intl. Conf. on Very Large Data Bases*, 2004.
- [12] T. Feder, R. Motwani, R. Panigrahy, C. Olston, and J. Widom. Computing the median with uncertainty. *SIAM J. Comput.*, 32(2), 2003.
- [13] A. Goel and P. Indyk. Stochastic load balancing and related problems. *Proc. of the Annual Symp. on Foundations of Computer Science*, 1999.
- [14] 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.
- [15] A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. *Proc. of the Annual ACM Symp. on Theory of Computing*, 2004.
- [16] A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: LP rounding approximation algorithms for stochastic network design. In *Proc. of the Annual Symp. on Foundations of Computer Science*, pages 218–227, 2004.
- [17] 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. of the Annual ACM-SIAM Symp. on Discrete Algorithms*, 2004.
- [18] S. Khanna and W-C. Tan. On computing functions with uncertainty. In *Proc. of the ACM Symp. on Principles of Database Systems*, 2001.
- [19] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999.
- [20] J. Kleinberg, Y. Rabani, and É. Tardos. Allocating bandwidth for bursty connections. *SIAM J. Comput.*, 30(1), 2000.
- [21] S. Kolliopoulos and N. Young. Tight approximation results for general covering integer programs. *Proc. of the Annual Symp. on Foundations of Computer Science*, 2001.
- [22] A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. *Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI 2005)*, 2005.
- [23] A. Krause and C. Guestrin. A note on the budgeted maximization on submodular functions. *Technical Report CMU-CALD-05-103*, 2005.
- [24] M. L. Massie, B. N. Chun, and D. E. Culler. The Ganglia distributed monitoring system: Design, implementation, and experience. *Parallel Computing*, 30(7), 2004.
- [25] 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.
- [26] C. Olston. *Approximate Replication*. PhD thesis, Stanford University, 2003.
- [27] D. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as discrete optimization. *Proc. of the Annual Symp. on Foundations of Computer Science*, 2004.
- [28] 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. of the Intl. Conf. on Data Engineering*, 2006.
- [29] N. Tatbul, U. Etintemel, S. Zdonik, M. Chemiack, and M. Stonebraker. Load shedding in a data stream manager. In *Proc. of the Intl. Conf. on Very Large Data Bases*, 2003.