Optimal Information Gathering on the Internet with Time and Cost Constraints*

Oren Etzioni Steve Hanks Tao Jiang† Omid Madani

Dept. of Comp. Sci. and Eng.
University of Washington
Seattle, WA 98195-2350
Email: {etzioni,hanks,jiang,madani}@cs.washington.edu

Abstract

The World Wide Web provides access to vast amounts of information, but content providers are considering charging for the information and services they supply. Thus the consumer may face the problem of balancing the benefit of asking for information against the cost (both in terms of money and time) of acquiring it. We study information-gathering strategies that maximize the expected value to the consumer. In our model there is a single information request, which has a known *benefit* to the consumer. To satisfy the request, *queries* can be sent simultaneously or in sequence to any of a finite set of independent *information sources*. For each source we know the monetary cost of making the query, the amount of time it will take, and the probability that the source will be able to provide the requested information. A *policy* specifies which sources to contact at which times, and the *expected value* of the policy can be defined as some function of the likelihood that the policy will yield an answer, the expected benefit, and the monetary cost and time delay associated with executing the policy. The problem is to find an expected-value-maximizing policy.

We explore four variants of the objective function V : (i) V consists only of the benefit term subject to threshold constraints on both total cost and total elapsed time. (ii) V is linear in the expected total cost of the policy subject to the constraint that the total elapsed time never exceeds some deadline, (iii) V is linear in the expected total elapsed time subject to the constraint that the total cost never exceeds some threshold, and (iv) V is linear in the expected total monetary cost and the expected time delay of the policy. The problems of devising an optimal querying policy for all four variants, and approximating an optimal querying policy for variants (iii) and (iv) are shown to be NP-hard. For (i), and with a mild simplifying assumption for (iii), we give a fully polynomial time approximation scheme. For (ii), we consider *batched* querying policies, and design an $O(n^2)$ time approximation algorithm with ratio $\frac{1}{2}$ and a polynomial time approximation scheme for optimal single-batch policies, and an $O(kn^2)$ time approximation algorithm with ratio $\frac{1}{5}$ for optimal k-batch policies.

Key Words. Approximation algorithm, batched policy, computational complexity, information gathering, information retrieval, Internet, scheduling, World Wide Web.

*Research supported in part by Office of Naval Research grant 92-J-1946, ARPA / Rome Labs grant F30602-95-1-0024, a gift from Rockwell International Palo Alto Research, National Science Foundation grant IRI-9357772, Natural Science and Engineering Research Council of Canada Research grant OGP0046613, and Canadian Genome Analysis and Technology Grant GO-12278.

†Present address: Dept. of Comp. Sci., McMaster University, Hamilton, Ont L8S 4K1, Canada. Email: jiang@maccs.mcmaster.ca

1 Introduction

The Internet is rapidly becoming the foundation of an information economy. Valuable information sources include on-line travel agents, nationwide Yellow Pages, job listing services, on-line malls, and many more. Currently, most of this information is available free of charge, and as a result parallel search tools such as MetaCrawler[18] and BargainFinder [10] respond to requests by querying numerous information sources simultaneously to maximize the information provided and minimize delay. However, information providers may start charging for their services [8, 12, 14]. Billing protocols to support an "information marketplace" have been announced by large players such as Visa and Microsoft [17] and by researchers [20].

Once billing mechanisms are in place, consumers of information may face the problem of balancing the benefit of obtaining information against the cost (both monetary and temporal) of obtaining it. Information providers will differ in the quality of the information they provide as well as the amount they charge and the speed at which they deliver information. The consumer thus faces the problem of developing a schedule of queries to the providers that maximizes expected value, which can be expressed in terms of (1) the benefit associated with a successful query, (2) the likelihood that a particular query will yield successful results, (3) the cost of making a query, and (4) the amount of time it takes.

This paper analyzes the "query scheduling" problem for a number of variants of the objective function. We begin by stating the problem precisely, then summarize our main results.

1.1 The Model

The basic problem is to find a *policy* for obtaining the answer to a (single) query. The policy will dictate which information source will be queried and when. To define a policy we begin with a (finite) set of information sources, $s_1, \ldots, s_n$. For each source s_i we introduce a cost parameter c_i and a duration parameter d_i. The former is the monetary cost assessed when the source is activated and the latter is the amount of time it takes the source to process the query. The cost and duration are known with certainty and are charged whether or not the source returns an answer to the query. [1] Finally we have p_i, the probability that s_i will return an answer to the query. Success probabilities are independent for distinct sources, and whether or not s_i will answer a query is uncertain but consistent: if s_i successfully answers a query it will always do so subsequently, and if it fails to answer a query it will always fail to do so subsequently.[2] We assume that accurate estimates of these parameters are obtainable from the history of the interactions with the information sources.

A policy can be represented as a sequence of pairs $\mathcal{P} = (s_{i_1}, t_1), (s_{i_2}, t_2), \ldots, (s_{i_m}, t_m)$, where $t_1 \leq t_2 \leq \cdots \leq t_m$. This specifies that source s_{i_1} will be initiated at t_1, s_{i_2} will be initiated at t_2, and so on. An execution of the policy is terminated either when some source returns a correct answer or when the policy has been exhausted. Since each source in a policy succeeds probabilistically, a policy generates a probability distribution over *outcomes*, where each outcome is one possible way that the policy might be played out. We use $S(\mathcal{O})$, $C(\mathcal{O})$ and $T(\mathcal{O})$ to denote the outcome's success (1 or 0),

[1] All our results can be extended to the case when the cost is charged only if the query is successful. Cost *expectations* in place of costs should be used in a few of the models.

[2] As a result it is never profitable to query a source more than once.

Objective fn	time threshold	linear in time
cost threshold	**TT**: max $\mathbf{E}[R \cdot S(\mathcal{O})]$ s.t. $\forall \mathcal{O}$ $C(\mathcal{O}) \leq \varsigma$ and $T(\mathcal{O}) \leq \tau$	**TL**: max $\mathbf{E}[R \cdot S(\mathcal{O}) - T(\mathcal{O})]$ s.t. $\forall \mathcal{O}$ $C(\mathcal{O}) \leq \varsigma$
linear in cost	**LT**: max $\mathbf{E}[R \cdot S(\mathcal{O}) - C(\mathcal{O})]$ s.t. $\forall \mathcal{O}$ $T(\mathcal{O}) \leq \tau$	**LL**: max $\mathbf{E}[R \cdot S(\mathcal{O}) - C(\mathcal{O}) - T(\mathcal{O})]$

Table 1: The four objective functions. Here, $\mathcal{O}$ denotes a possible outcome of the policy $\mathcal{P}$ to be found.

total cost and duration, respectively. The *value* of an outcome $\mathcal{O}$ is a function of $S(\mathcal{O})$, $C(\mathcal{O})$ and $T(\mathcal{O})$. The first component of the value function of an outcome $\mathcal{O}$ is always a constant reward R if the query was answered and 0 otherwise. The function additionally contains two additive components, one a function of $C(\mathcal{O})$ and one a function of $T(\mathcal{O})$. The *expected value* of a policy $\mathcal{P}$, denoted $V(\mathcal{P})$, is simply the expectation of the values of all its outcomes.

Our objective is to find a policy to maximize the expected value. We will consider four versions of the objective function: linear and threshold versions of the cost and time components. With suitable scaling of the monetary and time costs, the four objective functions assume the forms given in Table 1. We will hereafter refer to the four problems by their acronyms: **TT** for threshold in cost and time, **LT** for linear in cost and threshold in time, **TL** for threshold in cost and linear in time, and **LL** for linear in cost and time. Note that in the threshold cases we try to find a policy with the maximum expected value subject to the constraint that the policy never violates the threshold. In the remainder of the paper, assume without loss of generality that $R = 1$, unless otherwise stated.

1.2 "Batched" Policies

The results in this paper concerning the model **LT** will reflect one more simplifying assumption: that the duration parameters d_i are the same for each source. This assumption is powerful because it allows us to consider scheduling sources in simultaneous "batches:" all sources will be scheduled at $t = 0, d, 2d, \ldots$, where d is the common duration.

Although not fully general, this is a reasonable model of the current and probable future state of information access on the Internet. The current common mode of providing information is to supply small amounts of information quickly and cheaply (rather than process large-scale lengthy requests) [18]. As a result the duration for processing a single query relative to the user's time threshold is typically small. Furthermore, the number of providers continues to grow dramatically. In the case in which there are many information providers but each takes a short amount of time, the assumption of equal process duration may be an excellent approximation: the error introduced by assuming equal times will tend to be small relative to the amount of time the user is willing to wait for his information (and thus will not affect the quality of the schedule significantly), yet the sheer number of potential providers will still require an algorithm to choose carefully among its sources since a simple policy of completely serial or parallel queries is liable to be a very bad one.

1.3 Summary of Our Results

We show that finding an optimal policy is NP-hard for models **TT** and **LT**. Reductions from the problems in **LT** and **TT** show that even approximating an optimal policy for models **LL** and **TL** is NP-hard. A fully polynomial time approximation scheme (FPTAS) is obtained for the model **TT**,

using an extension of the well-known rounding technique for Knapsack [7]. The FPTAS also works for the model **TL** under a weak assumption: every source is "profitable" individually according to the **TL** objective function, *i.e.* for every source s_i, $Rp_i - d_i \geq 0$. The approximation algorithms for the case **LT**, where the objective function is linear in total cost subject to a time threshold, are perhaps the most interesting technically. We assume that all sources have the same time duration, and consider batched policies with a bounded number of batches. We will first present an $O(n^2)$ time approximation algorithm for optimal single-batch policies with ratio $\frac{1}{2}$, and then extend it to a polynomial time approximation scheme (PTAS). For any constant $r > 1$, the PTAS runs in time $O(n^{r+1})$ to achieve an approximation ratio $\frac{r-1}{r+1}$. The algorithms are simple and are similar to the ones in [16] for Knapsack, but the analyses are more sophisticated. We then design an approximation algorithm with ratio $\frac{1}{5}$ for optimal k-batch policies, running in time $O(kn^2)$. The algorithm is based on the ratio $\frac{1}{2}$ algorithm for single-batch policies, but it also involves some new ideas.

1.4 Related Work

Scheduling problems have been studied in many contexts including job-shop scheduling, processor allocation, etc. However, our Internet-inspired query scheduling problem has a unique flavor due to the need to balance the competing time and cost constraints on policies with unbounded parallelism. We here consider a number of alternative models that have appeared in the literature, underscoring the difference from our own. If we constrain the policies to be serialized, then an optimal solution can be found in polynomial time (see Section 4 for the **LT** case). Similar problems have been addressed in [5, 9, 13, 19] and elsewhere. The difference in this paper is the ability to query any number of sources in parallel. [4, 6] study scheduling tasks with unlimited parallelism, but their models are different because all tasks have to be executed successfully, whereas in our model a successful answer from any single source suffices. Furthermore, the positive results in [4, 6] are restricted to an exponential time dynamic programming algorithm and some heuristics. Another model of optimal information gathering has recently been studied in [3]. There, the objective is to find a query policy that minimizes the expected value of a linear combination of the total dollar cost and total time cost. A constant ratio approximation algorithm is obtained. Note that their model omits the positive reward associated with the successful completion of a query, which changes the nature of the problem as far as the design of approximation algorithms is concerned.

This paper provides complete proofs and adds new results to the work appearing in [2]. The paper is organized as follows. The hardness results for all four models are given in the next section. Sections 3 and 4 present the approximation algorithms with their analyses for optimal single-batch policies and optimal k-batch policies in the **LT** model. The FPTAS's for the two models involving a cost threshold are given in Section 5. The proofs of some technical claims are provided in the appendix.

2 The Complexity of Computing Optimal Querying Policies

We first prove that computing an optimal policy in models **TT** and **LT** is NP-hard. The proofs are reductions from the Partition Problem: Given a finite multiset S of positive integers $w_i \in S$, is there a subset $I \subset S$ such that $\sum_{w_i \in I} w_i = \frac{1}{2} \sum_{w_i \in S} w_i$. The only subtlety is that we have to use exponential numbers in the constructions.

Theorem 2.1 *Finding an optimal policy in model* **TT** *is NP-hard.*

Proof. It is clear that any optimal policy in this model is in fact a single-batch policy. Hence we only need to consider single-batch policies. We show that Partition reduces to this problem. Assume that the duration parameters of all the sources are less than the deadline. Then, the expected value of any policy $\mathcal{P}$ in this model is

$$V(\mathcal{P}) = 1 - \prod_{s_i \in \mathcal{P}} (1 - p_i).$$

Thus, maximizing $V(\mathcal{P})$ subject to a cost threshold is equivalent to minimizing $\prod_{s_i \in \mathcal{P}}(1 - p_i)$ under the same constraint, which in turn means maximizing $\sum_{s_i \in \mathcal{P}} - \ln(1 - p_i)$ under the same constraint.

Consider an instance of Partition consisting of a set $S = \{w_1, \ldots, w_n\}$ of integers, and let $C = \sum_{i=1}^{n} w_i$. For each source s_i, let its cost $c_i = w_i$, success probability $p_i = 1 - (1 + 1/C)^{-w_i}$, and time duration $d_i = 0$. Take the cost threshold to be $C/2$, and the time threshold to be some positive number. The expression $(1+1/C)^{-w_i}$ can be evaluated using the standard repeated squaring technique or a binomial series expansion approximation. It will become clear that we only have to keep at most $3 \log C + \log n$ precision bits during the process.

Clearly, $V(\mathcal{P}) \leq 1 - (1 + 1/C)^{-\frac{C}{2}}$ for any feasible policy $\mathcal{P}$. Now, let $x = \sum_{w_i \in \mathcal{P}} w_i$ and treat x as a continuous variable. Consider function $h(x) = 1 - (1 + 1/C)^{-x}$, which is clearly increasing in x. Since

$$(1 + 1/C)^{-\frac{C}{2}+1} - (1 + 1/C)^{-\frac{C}{2}} = (1 + 1/C)^{-\frac{C}{2}}(1/C) > 1/(\sqrt{e}C),$$

where e is the natural constant, there is a separation of at least $1/(\sqrt{e}C)$ in the value of the function $h(x)$ between point $\frac{C}{2}$ and any point less than or equal to $\frac{C}{2} - 1$. If we keep $3 \log C + \log n$ bits during calculation of each $1 - (1 + 1/C)^{-w_i}$, then each p_i would have $2 \log C + \log n$ precision bits. Thus, the precision of our evaluation of $V(\mathcal{P}) = 1 - \prod_{s_i \in \mathcal{P}}(1 - p_i)$ is at least $2 \log C$ bits, which is sufficient to allow us to distinguish between the case $\sum_{s_i \in \mathcal{P}} c_i = \frac{C}{2}$ and the case $\sum_{s_i \in \mathcal{P}} c_i < \frac{C}{2}$, due to the above $1/(\sqrt{e}C)$ separation. ■

For the problem instance **LT** we prove a stronger result by showing that a special case of the problem is NP-hard.

Theorem 2.2 *Finding an optimal single-batch policy for the* **LT** *objective function is NP-hard.*

Proof. Note that the objective in the single-batch case is to find a set $\mathcal{P}$ of sources to query in parallel such that the function

$$V(\mathcal{P}) = R(1 - \prod_{s_i \in \mathcal{P}} (1 - p_i)) - \sum_{s_i \in \mathcal{P}} c_i \tag{1}$$

is maximized. This is equivalent to minimizing the quantity $R \prod_{s_i \in \mathcal{P}}(1-p_i) + \sum_{s_i \in \mathcal{P}} c_i$ over all possible sets of sources.

Consider an instance of Partition consisting of a set $S = \{w_1, \ldots, w_n\}$ of integers, and let $C = \sum_{i=1}^{n} w_i$. Define the parameters for the optimal single-batch policy problem as follows:

$$\begin{aligned} c_i &= w_i \\ p_i &= 1 - (1 + 1/C)^{-w_i} \end{aligned}$$

$$R = [\ln(1+1/C)]^{-1}(1+1/C)^{\frac{C}{2}}$$

Again, it will become clear that only $4\log C + \log n$ bits must be kept in the calculation of p_i's and R. For any subset $S_1 \subseteq S$ we have:

$$\begin{aligned}\sum_{w_i \in S_1} c_i + R \prod_{w_i \in S_1}(1-p_i) &= \sum_{w_i \in S_1} w_i + [\ln(1+1/C)]^{-1}(1+1/C)^{\frac{C}{2}} \prod_{w_i \in S_1}(1+1/C)^{-w_i} \\ &= \sum_{w_i \in S_1} w_i + [\ln(1+1/C)]^{-1}(1+1/C)^{\frac{C}{2}-\sum_{w_i \in S_1} w_i}\end{aligned}$$

Again, let $x = \sum_{w_i \in S_1} w_i$ and treat x as a continuous variable. We want to locate the minimum of the following function

$$h(x) = x + [\ln(1+1/C)]^{-1}(1+1/C)^{\frac{C}{2}-x}.$$

Setting the derivative to zero,

$$\begin{aligned} h'(x) &= 1 + [\ln(1+1/C)]^{-1}[-\ln(1+1/C)](1+1/C)^{\frac{C}{2}-x} = 0 \\ \Rightarrow \quad 1 &= [\ln(1+1/C)]^{-1}\ln(1+1/C)(1+1/C)^{\frac{C}{2}-x} \\ \Rightarrow \quad 1 &= (1+1/C)^{\frac{C}{2}-x} \\ \Rightarrow \quad x &= \frac{C}{2}. \end{aligned}$$

We also note that the second derivative of $h(x)$ is always positive, which shows the convexity of the function:

$$h''(x) = [\ln(1+1/C)]^{-1}[\ln(1+1/C)]^2(1+1/C)^{\frac{C}{2}-x} > 0.$$

The following shows that there is a separation of $\Omega(1/C^2)$ in the value of the function $h(x)$ between the points $\frac{C}{2}$ and $\frac{C}{2} \pm 1$. Assume that $C > 2$.

$$\begin{aligned} h(\frac{C}{2}-1) - h(\frac{C}{2}) &= C^{-1}[\ln(1+1/C)]^{-1} - 1 \\ &> C^{-1}(\frac{6C^3}{6C^2-3C+2}) - 1 \\ &= \frac{3C-2}{6C^2-3C+2} \\ &> 1/(6C^2). \\ h(\frac{C}{2}+1) - h(\frac{C}{2}) &= 1 - (\frac{1}{C+1})(\ln(1+1/C))^{-1} \\ &> 1 - (\frac{1}{C+1})(\frac{2C^2}{2C-1}) \\ &> 1/(6C^2). \end{aligned}$$

Therefore, we only have to keep $4\log C + \log n$ precision bits in the calculation of R and p_i's. Hence the reduction can be done in polynomial time. ■

Hence, the problem of deciding whether the expected value of some policy for the single-batch case exceeds a certain threshold is NP-hard. This problem readily reduces to a problem in the **LL** model where the duration parameters of the sources are all set to the threshold in question. In this case, it is not hard to see that there is a policy with a positive value for the **LL** model problem if and only if there is a policy for the single-batch model problem with expected value greater than the threshold. Below, by a *positive approximation* we mean constructing a policy which has positive expected value if and only if the value of an optimal policy is positive. Positive approximation is a very relaxed approximation criterion and we just argued that even positively approximating the optimal in model **LL** is hard. A similar reduction from model **TT** shows that positively approximating problems in model **TL** is NP-hard as well:

Theorem 2.3 *Positively approximating an optimal policy for the objective functions in the models* **TL** *or* **LL** *is NP-hard.*

3 Approximating Optimal Single-Batch Policies

In this and the following sections our focus in on the **LT** model. In this section we consider policies that send out all their queries in a single batch, *i.e.* all queries are sent in parallel at time $t = 0$. We present an algorithm that approximates the optimal single-batch policy with ratio $1/2$, then develop a PTAS. Although the PTAS is a straightforward extension of the ratio $1/2$ algorithm, its analysis is very different.

Recall again that a single-batch policy is just a set of sources, and our goal is to maximize the objective function in equality 1.

The following simple facts and definitions will be useful in this and the next sections. The first lemma shows the subadditivity of the objective function for batched policies.

Lemma 3.1 *Let* OPT_0 *be an optimal* k*-batch policy. For any partition of* OPT_0 *into two subpolicies* OPT_1 *and* OPT_2*, where the sources in* OPT_1 *and* OPT_2 *are scheduled in the same batches as they are in* OPT_0*,* $V(\mathrm{OPT}_0) \le V(\mathrm{OPT}_1) + V(\mathrm{OPT}_2)$*.*

Proof. We prove it for $k = 2$; the extension to the general k is straightforward. For each $i = 0, 1, 2$ and $j = 1, 2$, let $P_{i,j}$ and $C_{i,j}$ be the collective success probability and cost of the sources in batch j of OPT_i, respectively. Then

$$V(\mathrm{OPT}_0) = P_{0,1} - C_{0,1} + (1 - P_{0,1})(P_{0,2} - C_{0,2}) = P_{1,1} - C_{1,1} + P_{0,1} - P_{1,1} - C_{2,1} + (1 - P_{0,1})(P_{0,2} - C_{0,2}).$$

Since $P_{0,1} = P_{1,1} + P_{2,1} - P_{1,1}P_{2,1}$, $P_{0,1} - P_{1,1} = P_{2,1}(1 - P_{1,1}) \le P_{2,1}$. Hence,

$$P_{1,1} - C_{1,1} + P_{0,1} - P_{1,1} - C_{2,1} \le P_{1,1} - C_{1,1} + P_{2,1} - C_{2,1}.$$

Similarly we have

$$P_{0,2} - C_{0,2} \le P_{1,2} - C_{1,2} + P_{2,2} - C_{2,2}.$$

Because OPT_0 is optimal, $P_{1,2} - C_{1,2} \geq 0$ and $P_{2,2} - C_{2,2} \geq 0$. Therefore,

$$\begin{aligned} V(\mathrm{OPT}_0) &\leq (P_{1,1} - C_{1,1} + P_{2,1} - C_{2,1}) + (1 - P_{0,1})(P_{1,2} - C_{1,2} + P_{2,2} - C_{2,2}) \\ &= (P_{1,1} - C_{1,1} + (1 - P_{0,1})(P_{1,2} - C_{1,2})) + P_{2,1} - C_{2,1} + (1 - P_{0,1})(P_{2,2} - C_{2,2}) \\ &\leq (P_{1,1} - C_{1,1} + (1 - P_{1,1})(P_{1,2} - C_{1,2})) + P_{2,1} - C_{2,1} + (1 - P_{2,1})(P_{2,2} - C_{2,2}) \\ &= V(\mathrm{OPT}_1) + V(\mathrm{OPT}_2). \quad \blacksquare \end{aligned}$$

Lemma 3.2 *Suppose that $\mathcal{P}$ is any k-batch policy, i is an index between 1 and k, and s_j is a source not appearing in $\mathcal{P}$. Let $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ denote the subpolicies consisting of the the first $i-1$ batches, the i-th batch, and the last $k-i$ batches of $\mathcal{P}$, respectively. Also denote the expected cost and collective success probability of the sources in policy $\mathcal{P}_l$ as C_l and P_l, $l = 1, 2, 3$. Then adding s_j to the i-th batch of policy $\mathcal{P}$ increases its expected value by*

$$\begin{aligned} V(\mathcal{P} \cup \{s_j\}) - V(\mathcal{P}) &= (1 - P_1)(p_j(1 - P_2)(1 - P_3 + C_3) - c_j) \\ &= (1 - P_1)p_j((1 - P_2)(1 - P_3 + C_3) - c_j/p_j) \end{aligned} \tag{2}$$

In particular, if $k = i = 1$, the net increase is

$$V(\mathcal{P} \cup \{s_j\}) - V(\mathcal{P}) = p_j(1 - P_2) - c_j = p_j(1 - P_2 - c_j/p_j) \tag{3}$$

Proof. The expected values of the policies $\mathcal{P}$ and $\mathcal{P} \cup \{s_j\}$ can be written as

$$V(\mathcal{P}) = (P_1 - C_1) + (1 - P_1)((P_2 - C_2) + (1 - P_2)(P_3 - C_3))$$

$$V(\mathcal{P} \cup \{s_j\}) = (P_1 - C_1) + (1 - P_1)((P_2 + p_j - P_2 p_j - C_2 - c_j) + (1 - P_2)(1 - p_j)(P_3 - C_3))$$

Taking the difference gives us the lemma. $\blacksquare$

Thus, in the case of $k = 1$, adding s_j to the policy $\mathcal{P}$ results in an increased expected value iff the collective failure probability $1 - P_2$ of the sources in $\mathcal{P}$ is strictly greater than the cost-to-success-probability ratio c_j/p_j of source s_j. Observe that the increased value $p_j(1 - P_2 - c_j/p_j)$ is proportional to the success probability p_j as long as the ratio c_j/p_j is kept constant. Also observe that, for general k, $1 - P_3 + C_3 = 1 - (P_3 - C_3) = 1 - V(\mathcal{P}_3)$. It follows from Lemma 3.2 that a source s_i with $c_i > p_i$ is not useful if $V(\mathcal{P}_3) \geq 0$. Hence we can assume from now on that $p_i \geq c_i$ for all i.

We say that a source s is *profitable in a policy* $\mathcal{P}$ if s is queried in $\mathcal{P}$ and dropping it from $\mathcal{P}$ would not increase the expected value of $\mathcal{P}$. The above lemma states that source s_j in batch i of the k-batch policy $\mathcal{P}$ is profitable in $\mathcal{P}$ if and only if $c_j/p_j \leq (1 - P_{2,j})(1 - V(\mathcal{P}_3))$, where $P_{2,j}$ is the collective success probability of sources in batch i *excluding* source s_j. A policy $\mathcal{P}$ is *irreducible* if every source in $\mathcal{P}$ is profitable in $\mathcal{P}$. Clearly, every optimal k-batch policy is irreducible.

3.1 A Ratio $\frac{1}{2}$ Approximation Algorithm

Our algorithm, as shown in Figure 1, is somewhat similar to the greedy approximation algorithm for Knapsack given in [16], though the analysis of its performance is more complex.

The algorithm Pick-a-Star sorts the sources in ascending order of the ratio c_i/p_i. It then goes over each source s_i, picks it and then picks the rest from the sorted list (with s_i removed) until either the

```
1.  Sort the sources so that c_1/p_1 ≤ ··· ≤ c_n/p_n.
2.  APPR = ∅.   (* APPR is the best policy found so far. *)
3.  For i := 1 to n
4.    S := {s_i}.   (* S is the current policy constructed. *)
5.    Q := 1 − p_i.   (* Q is the collective failure probability of S. *)
6.    For j := 1 to n, where j ≠ i
7.      If Q ≥ c_j/p_j then (* profitability check *)
8.        S := S ∪ {s_j}.
9.        Q := Q(1 − p_j).
10.     else exit to step 11.
11.   If V(APPR) < V(S) then APPR := S.
12. Output policy APPR.
```

Figure 1: The algorithm Pick-a-Star.

list is exhausted or it reaches a source s_j which cannot be profitable, meaning that the profitability criterion:

$$\prod_{k=i \text{ or } k<j} (1 - p_k) \geq c_j/p_j$$

is not satisfied. Equality 3 in Lemma 3.2 and the comments following the lemma explain the choice of the criterion. Pick-a-Star keeps track of the policy with the highest expected value over the iterations. Clearly the running time is $O(n^2)$.

Now we analyze the performance of Pick-a-Star and show that it results in an expected value that is at least half of the optimum. Let APPR be the policy obtained by Pick-a-Star and OPT an optimal single-batch policy. Since Pick-a-Star picks the first source optimally (*i.e.* through exhaustive search), $V(\text{APPR}) \geq V(\{s_i\}) = p_i - c_i$ for all $i \leq n$. Thus, without loss of generality, we may assume $|\text{APPR}| > 1$. Moreover, we will assume henceforth that the first source picked by Pick-a-Star is the "most profitable" source in OPT, *i.e.* some source s_i with the maximum $V(\{s_i\})$ over all sources in OPT. Let s_{last} be the last source picked by Pick-a-Star. We can assume that the collective failure probability of APPR is at least the ratio c_{last}/p_{last}, because otherwise we could modify APPR by decreasing p_{last} while keeping c_{last}/p_{last} constant until the collective failure probability of APPR becomes equal to c_{last}/p_{last}. This is possible since the collective failure probability of APPR $- \{s_{last}\}$ is greater than c_{last}/p_{last}. By Lemma 3.2 such modification could only worsen the expected value of APPR. Note that if s_{last} appears also in OPT, then it is treated as a different copy and kept intact. Hence we do not change the expected value of OPT in this case (or in any other case). Note also that this potential modification does not affect the first source picked by Pick-a-Star since $|\text{APPR}| > 1$.

Define $S_0 = \text{APPR} \cap \text{OPT}$, $S_1 = \text{APPR} - S_0$, and $S_2 = \text{OPT} - S_0$. For each $i = 0, 1, 2$, let C_i and P_i be the collective cost and success probability of the sources in S_i. Note that if the success probability of source s_{last} is modified in APPR as mentioned above and s_{last} also appears in OPT, then the two copies of s_{last} in APPR and OPT are viewed as distinct sources and thus s_{last} will not be included in S_0. On the other hand, if s_{last} is not modified in APPR and s_{last} appears in OPT, then the two copies of s_{last} in APPR and OPT are viewed as the same source and thus s_{last} will be

included in S_0. Observe that

$$\forall s_i \in S_1 \forall s_j \in S_2, c_i/p_i \leq c_j/p_j \tag{4}$$

Let us first consider the (easier) case in which $S_2 = \emptyset$. Observe that $S_1 \subseteq \{s_1, \ldots, s_{last}\}$. Since the collective failure probability of $\mathrm{APPR} - \{s_{last}\}$ is greater than $c_{last}/p_{last} \geq \cdots \geq c_1/p_1$, every element of S_1 is profitable in the set $\mathrm{APPR} - \{s_{last}\}$. By Lemma 3.2, $V(\mathrm{APPR} - \{s_{last}\}) \geq V(\mathrm{OPT} - \{s_{last}\})$. We also know that $V(\mathrm{APPR}) \geq V(\mathrm{APPR} - \{s_{last}\})$ by Lemma 3.2. Since $V(\mathrm{APPR}) \geq V(\{s_{last}\})$ and $V(\mathrm{OPT}) \leq V(\mathrm{OPT} - \{s_{last}\}) + V(\{s_{last}\})$ by Lemma 3.1,

$$2V(\mathrm{APPR}) \geq V(\mathrm{OPT} - \{s_{last}\}) + V(\{s_{last}\}) \geq V(\mathrm{OPT}).$$

Now suppose that $S_2 \neq \emptyset$. Since OPT is irreducible and Pick-a-Star picks sources until no remaining source can be profitable, $S_1 \neq \emptyset$. Let $m = |S_2|$ and $l = |S_1|$. Let

$$\begin{aligned} \alpha_1 &= \max_{s_i \in S_1} \frac{c_i}{p_i(1-P_0)} \leq \frac{c_{last}}{p_{last}(1-P_0)} \\ \alpha_2 &= \min_{s_i \in S_2} \frac{c_i}{p_i(1-P_0)} \end{aligned}$$

By relation 4, clearly $\alpha_1 \leq \alpha_2$. The next lemma relating α_1, α_2 to P_1, P_2 is a key to our analysis.

Lemma 3.3 *(i)* $\alpha_1 \leq 1 - P_1 \leq \alpha_2$ *and (ii)* $1 - P_2 \geq \alpha_2^{\frac{m}{m-1}}$.

Proof. Recall that we have assumed that $(1-P_0)(1-P_1) \geq c_{last}/p_{last}$. Thus, $1 - P_1 \geq \alpha_1$. Since Pick-a-Star stopped before picking anything from S_2, $(1-P_0)(1-P_1) \leq c_i/p_i$ for any $s_i \in S_2$. These prove (i). To prove (ii), let $p_{min} = \min_{s_i \in S_2} p_i$. Since OPT is irreducible, $(1-P_0)(1-P_2)/(1-p_{min}) \geq c_{min}/p_{min} \geq (1-P_0)\alpha_2$. So,

$$(1-P_2)^{\frac{m-1}{m}} \geq (1-P_2)/(1-p_{min}) \geq \alpha_2.$$

I.e. $1 - P_2 \geq \alpha_2^{\frac{m}{m-1}}$. ■

Now we want to find a lower bound for the ratio

$$\frac{V(\mathrm{APPR})}{V(\mathrm{OPT})} = \frac{P_0 - C_0 + (1-P_0)P_1 - C_1}{P_0 - C_0 + (1-P_0)P_2 - C_2} \tag{5}$$

Since $V(S_0) \geq V(S_2)/m$ by the choice of the first source picked by Pick-a-Star and the fact that S_0 is irreducible,

$$V(\mathrm{OPT}) \leq V(S_0) + V(S_2) \leq (m+1)V(S_0).$$

This implies

$$\frac{(1-P_0)P_2 - C_2}{P_0 - C_0 + (1-P_0)P_2 - C_2} \leq \frac{m}{m+1}.$$

Define

$$r = \frac{(1-P_0)P_1 - C_1}{(1-P_0)P_2 - C_2} \tag{6}$$

To obtain a lower bound of $1/2$ for the ratio in equality 5, we need

$$\frac{1}{m+1} + r\frac{m}{m+1} \geq \frac{1}{2}, \quad \textit{i.e.} \quad r \geq \frac{m-1}{2m}, \tag{7}$$

which we show below. The following lemma gives a clean lower bound for ratio r.

Lemma 3.4

$$r \geq \min_{\alpha_1 \leq 1-P_1 \leq \alpha_2} \frac{P_1 - l(1-(1-P_1)^{1/l})\alpha_1}{(1-\alpha_2^{m/(m-1)})(1-\alpha_2)}.$$

Proof. Observe that

$$\begin{aligned} r &= \frac{(1-P_0)P_1 - (\sum_{s_i \in S_1} p_i \frac{c_i}{p_i})}{(1-P_0)P_2 - C_2} \\ &\geq \frac{(1-P_0)P_1 - (\sum_{s_i \in S_1} p_i)(1-P_0)\alpha_1}{(1-P_0)P_2 - C_2} \\ &\geq \frac{P_1 - (\sum_{s_i \in S_1} p_i)\alpha_1}{P_2 - C_2/(1-P_0)} \\ &\geq \frac{P_1 - (\sum_{s_i \in S_1} p_i)\alpha_1}{P_2 - P_2\alpha_2} \\ &\geq \frac{P_1 - (\sum_{s_i \in S_1} p_i)\alpha_1}{(1-\alpha_2^{m/(m-1)})(1-\alpha_2)}. \end{aligned}$$

The last step follows from (ii) of Lemma 3.3. Since

$$\sum_{s_i \in S_1} (1-p_i) \geq l[\prod_{s_i \in S_1} (1-p_i)]^{1/l} = l(1-P_1)^{1/l},$$

$\sum_{s_i \in S_1} p_i \leq l - l(1-P_1)^{1/l}$. Hence the lemma follows from (i) of Lemma 3.3. ■

Now we try to simplify the lower bound function[3].

Claim 3.5 *The ratio*

$$\frac{P_1 - l(1-(1-P_1)^{1/l})\alpha_1}{(1-\alpha_2^{m/(m-1)})(1-\alpha_2)}$$

is increasing in P_1 when $1 - P_1 \geq \alpha_1$.

By Lemma 3.3, the smallest P_1 can be is $1 - \alpha_2$. The above ratio is clearly decreasing in $\alpha_1 \leq \alpha_2$. For convenience, let $x = \alpha_2$. We set $P_1 = 1 - x$ and $\alpha_1 = x$, and we get,

$$r \geq \frac{1}{1-x^{m/(m-1)}}[\frac{1-x-l(1-x^{1/l})x}{1-x}] \tag{8}$$

Claim 3.6 *The right hand side of inequality 8 is nonincreasing in l.*

Taking the limit

$$\lim_{l\to\infty} l(1-x^{1/l}) = \lim_{l\to\infty} \frac{1-x^{1/l}}{1/l} = \lim_{l\to\infty} \frac{(-x^{1/l})(\ln x)(-1/l^2)}{-1/l^2} = -\ln x,$$

we get

$$r \geq \frac{1}{1-x^{m/(m-1)}}[\frac{1-x+x\ln x}{1-x}] \tag{9}$$

[3]The proofs of the claims appear in the appendix.

Claim 3.7 *The right hand side of inequality 9 is decreasing in $x \in (0,1)$.*

Since the right-hand-side expression is undefined at $x = 1$, we take the limit

$$\begin{aligned} r &\geq \lim_{x\to 1}[\frac{1}{1-x^{m/(m-1)}}(\frac{1-x+x\ln x}{1-x})] \\ &= \lim_{x\to 1}\frac{1-x+x\ln x}{1-x-x^{\frac{m}{m-1}}+x^{\frac{2m-1}{m-1}}} \\ &= \lim_{x\to 1}\frac{-1+\ln x+x/x}{-1-\frac{m}{m-1}x^{\frac{1}{m-1}}+\frac{2m-1}{m-1}x^{\frac{m}{m-1}}} \\ &= \lim_{x\to 1}\frac{1/x}{-\frac{m}{m-1}\frac{1}{m-1}x^{\frac{2-m}{m-1}}+\frac{2m-1}{m-1}\frac{m}{m-1}x^{\frac{1}{m-1}}} \\ &= \frac{1}{-\frac{m}{(m-1)^2}+\frac{m(2m-1)}{(m-1)^2}} = \frac{1}{\frac{m}{m-1}(\frac{-1+2m-1}{m-1})} \\ &= \frac{m-1}{2m}. \end{aligned}$$

This verifies inequality 7 and completes the proof that $V(\text{APPR})/V(\text{OPT}) \geq 1/2$.

Theorem 3.8 *Pick-a-Star produces a single-batch policy with an expected value that is at least half of the optimum.*

3.2 Extending Pick-a-Star to a PTAS

The extension of the algorithm is straightforward. Let $r \geq 1$ be any fixed constant. The new algorithm iterates over all possible choices of at most r sources and schedules the rest of the sources based on the cost-to-success probability ratio, using the same stopping criterion. It then outputs the best policy found in all iterations. Call the new algorithm Pick-r-Stars. Clearly, it runs in $O(n^{r+1})$ time. We show that Pick-r-Stars achieves an approximation ratio of $\frac{r-1}{r+1}$. The analysis is different from the previous subsection in that we will make use of the r sources in the optimal policy with the highest success probability instead of the the most profitable ones. We would like to remark here that this new strategy does not work for Pick-a-Star, nor does our analysis of Pick-a-Star work for general r because the best lower bound that the analysis yields for the ratio defined in equality 6 is $\frac{m-1}{2m}$.

Let APPR be the policy found by Pick-r-Stars and OPT an optimal policy. As in the previous subsection, we assume without loss of generality that (i) $|\text{APPR}| > r$ and $|\text{OPT}| > r$, (ii) APPR contains the r sources in OPT with the highest success probability, and (iii) the collective failure probability of APPR is at least the ratio c_{last}/p_{last}, where s_{last} is the last source picked by Pick-r-Stars. Since OPT is irreducible, we can also assume that APPR $\not\subseteq$ OPT, because otherwise APPR = OPT.

Again, let $S_0 = \text{APPR} \cap \text{OPT}$, $S_1 = \text{APPR} - S_0$, and $S_2 = \text{OPT} - S_0$ and the corresponding collective costs and success probabilities C_i and P_i, for each $i = 0, 1, 2$. We also have $c_i/p_i \leq c_j/p_j$ for all $s_i \in S_1, s_j \in S_2$. Define $l = |S_1|$, $m = |S_2|$, and

$$\alpha_0 = \max_{s_i \in S_0} \frac{c_i}{p_i}$$

$$\begin{aligned}\alpha_1 &= \max_{s_i \in S_1} \frac{c_i}{p_i} \leq \frac{c_{last}}{p_{last}(1-P_0)} \\ \alpha_2 &= \min\{1, \min_{s_i \in S_2} \frac{c_i}{p_i}\}\end{aligned}$$

Then, we again have

$$\alpha_1 \leq (1-P_0)(1-P_1) \leq \alpha_2 \tag{10}$$

To obtain a clean lower bound for the approximation ratio $V(\text{APPR})/V(\text{OPT})$, we go through a sequence of simplifying steps. In the process we will guarantee that the ratio $V(\text{APPR})/V(\text{OPT})$ never improves and inequality 10 always holds.

First, we will assume that $|S_0| = r$, *i.e.* APPR and OPT share exactly r common sources, by the following argument. Let $s_i \in S_0$ be any source. Then,

$$\begin{aligned}\frac{V(\text{APPR})}{V(\text{OPT})} &= \frac{1-(1-P_0)-C_0+(1-P_0)P_1-C_1}{1-(1-P_0)-C_0+(1-P_0)P_2-C_2} \\ &= \frac{\frac{1}{1-p_i}-\frac{1-P_0}{1-p_i}-\frac{C_0}{1-p_i}+\frac{(1-P_0)P_1}{1-p_i}-\frac{C_1}{1-p_i}}{\frac{1}{1-p_i}-\frac{1-P_0}{1-p_i}-\frac{C_0}{1-p_i}+\frac{(1-P_0)P_2}{1-p_i}-\frac{C_2}{1-p_i}} \\ &= \frac{\frac{1-c_i}{1-p_i}-\frac{1-P_0}{1-p_i}-\frac{C_0-c_i}{1-p_i}+\frac{(1-P_0)P_1}{1-p_i}-\frac{C_1}{1-p_i}}{\frac{1-c_i}{1-p_i}-\frac{1-P_0}{1-p_i}-\frac{C_0-c_i}{1-p_i}+\frac{(1-P_0)P_2}{1-p_i}-\frac{C_2}{1-p_i}} \\ &> \frac{1-\frac{1-P_0}{1-p_i}-\frac{C_0-c_i}{1-p_i}+\frac{(1-P_0)P_1}{1-p_i}-\frac{C_1}{1-p_i}}{1-\frac{1-P_0}{1-p_i}-\frac{C_0-c_i}{1-p_i}+\frac{(1-P_0)P_2}{1-p_i}-\frac{C_2}{1-p_i}}\end{aligned}$$

The last step holds because $p_i - c_i = V(\{s_i\}) > 0$ and thus $1-c_i > 1-p_i$. Hence, we can define a new pair of APPR and OPT by removing the source s_i and dividing the cost of every remaining source in S_0 by $1-p_i$. Clearly inequality 10 still holds for the new pair.

Second, we can assume that $\alpha_1 = \alpha_2 = (1-P_0)(1-P_1)$. This can be achieved by increasing the cost of each source in S_1 and decreasing the cost of each source in S_2. So now, $c_i = \alpha_1 p_i$ for each $s_i \in S_1 \cup S_2$.

Third, we assume without loss of generality that the r sources in S_0 have the same success probability $p = 1-(1-P_0)^{1/r}$. Since these sources are assumed to have the largest success probability in OPT, $p_i \leq p$ for each $s_i \in S_2$. Moreover, we can assume that $p_i = p$ for each $s_i \in S_2$ by the following argument. If $p_i < p$, we increase p_i to p and c_i to $\alpha_1 p$. By Lemma 3.2, this improves $V(\text{OPT})$ because the set OPT is irreducible. If this results in a set that is not irreducible, we can make it irreducible by dropping some sources in S_2, again improving the expected value of OPT.

Forth, we can assume that $c_i/p_i = \alpha_0$ for all $s_i \in S_0$. The condition can be achieved by increasing the costs of the sources in S_0. This would decrease the expected value of both APPR and OPT by the same amount, and thus decrease the ratio $V(\text{APPR})/V(\text{OPT})$.

Finally, we can worsen APPR by assuming that

$$p_i = 1-(1-P_1)^{1/l} = 1-(\alpha_1/(1-P_0))^{1/l} = 1-(\alpha_1/(1-p)^r)^{1/l}$$

for each $p_i \in S_1$, as mentioned in the previous subsection.

Now we have clean formulas for the expected values:

$$\begin{aligned} V(\text{APPR}) &= 1-(1-p)^r\frac{\alpha_1}{(1-p)^r} - \alpha_0 rp - l\alpha_1[1-(\frac{\alpha_1}{(1-p)^r})^{1/l}] \\ V(\text{OPT}) &= 1-(1-p)^{r+m} - \alpha_0 rp - \alpha_1 mp \end{aligned}$$

We further simplify the formulas by getting rid of l and m.

Lemma 3.9

$$V(\text{APPR}) \geq 1-\alpha_1-\alpha_0 rp + \alpha_1 \ln(\frac{\alpha_1}{(1-p)^r}).$$

Proof. Since $\alpha_1/(1-p)^r < 1$ by inequality 10, the function $-l\alpha_1[1-(\frac{\alpha_1}{(1-p)^r})^{1/l}]$ is nonincreasing in l by Claim 3.6 in the last subsection. Taking the limit, we get

$$\begin{aligned} V(\text{APPR}) &\geq \lim_{l\to\infty} 1-\alpha_1-\alpha_0 rp - l\alpha_1(1-(\frac{\alpha_1}{(1-p)^r})^{1/l}) \\ &\geq 1-\alpha_1-\alpha_0 rp+\alpha_1\ln(\frac{\alpha_1}{(1-p)^r}) \end{aligned}$$ ■

Lemma 3.10

$$V(\text{OPT}) \leq 1+\frac{\alpha_1 p}{\ln(1-p)} - \alpha_0 rp - \frac{\alpha_1 p}{\ln(1-p)}\ln[\frac{-\alpha_1 p}{(1-p)^r\ln(1-p)}].$$

Proof. We can treat m as a continuous variable and maximize $1-(1-p)^{r+m}-\alpha_0 rp-\alpha_1 mp$ over all real values of m. Taking the derivative with respect to m and setting it to 0, we get

$$m = \frac{1}{\ln(1-p)}\ln(\frac{-\alpha_1 p}{(1-p)^r\ln(1-p)}).$$

Hence,

$$\max_m 1-(1-p)^{r+m}-\alpha_0 rp-\alpha_1 mp = 1+\frac{\alpha_1 p}{\ln(1-p)} - \alpha_0 rp - \frac{\alpha_1 p}{\ln(1-p)}\ln(\frac{-\alpha_1 p}{(1-p)^r\ln(1-p)})$$ ■

Now we are ready to show that $V(\text{APPR})/V(\text{OPT}) \geq (r-1)/(r+1)$. It suffices to prove that

$$\frac{V(\text{OPT})-V(\text{APPR})}{V(\text{APPR})} \leq \frac{2}{r-1} \tag{11}$$

We first give an overestimate of the difference $V(\text{OPT})-V(\text{APPR})$. The following simple mathematical facts for $0<p<1$ will be useful.

$$\begin{aligned} p < -\ln(1-p) &= p+\frac{p^2}{2}+\frac{p^3}{3}+\cdots < \frac{p}{1-p} \\ 0 < 1+\frac{p}{\ln(1-p)} &= 1-\frac{1}{1+p/2+p^2/3+\cdots} \\ &< 1-\frac{1}{1+p+p^2+\cdots} \\ &= p \\ -\ln(\frac{-p}{\ln(1-p)}) &= -\ln(1-(1+\frac{p}{\ln(1-p)})) \\ &> 1+\frac{p}{\ln(1-p)} \end{aligned}$$

Lemma 3.11

$$V(\mathrm{OPT}) - V(\mathrm{APPR}) < f_1(\alpha_1) = \alpha_1 p^2 - \alpha_1 p \ln(\frac{\alpha_1}{(1-p)^r}).$$

Proof. From Lemmas 3.9 and 3.10, we know

$$V(\mathrm{OPT}) - V(\mathrm{APPR}) \leq \alpha_1(1 + \frac{p}{\ln(1-p)}) - \alpha_1(1 + \frac{p}{\ln(1-p)}) \ln(\frac{\alpha_1}{(1-p)^r}) - \frac{\alpha_1 p}{\ln(1-p)} \ln(\frac{-p}{\ln(1-p)}).$$

Therefore,

$$\begin{aligned} V(\mathrm{OPT}) - V(\mathrm{APPR}) &< \alpha_1(1 + \frac{p}{\ln(1-p)}) - \alpha_1 p \ln(\frac{\alpha_1}{(1-p)^r}) + \frac{\alpha_1 p}{\ln(1-p)}(1 + \frac{p}{\ln(1-p)}) \\ &= \alpha_1(1 + \frac{p}{\ln(1-p)})^2 - \alpha_1 p \ln(\frac{\alpha_1}{(1-p)^r}) \\ &\leq \alpha_1 p^2 - \alpha_1 p \ln(\frac{\alpha_1}{(1-p)^r}). \quad \blacksquare \end{aligned}$$

Next we find an underestimate of $V(\mathrm{APPR})$. Observe that the following conditions follow from inequality 10, the simplifying assumptions on the sources in S_i (*e.g.* $\forall s_i \in S_0, c_i/p_i = \alpha_0$), and the arguments for the second claim of Lemma 3.3:

$$(1-p)^r > \alpha_1$$

$$(1-p)^{r-1} = ((1-p)^r)^{\frac{r-1}{r}} > \alpha_0$$

Lemma 3.12

$$V(\mathrm{APPR}) > f_2(\alpha_1) = \frac{r(r-1)}{2}(1-p)^{r-2}p^2 + \alpha_1 \ln(\frac{\alpha_1}{(1-p)^r}) + (1-p)^r - \alpha_1.$$

Proof.

$$\begin{aligned} V(\mathrm{APPR}) &\geq 1 - \alpha_1 - \alpha_0 rp + \alpha_1 \ln(\frac{\alpha_1}{(1-p)^r}) \\ &= 1 - (1-p)^r - \alpha_0 rp + \alpha_1 \ln(\frac{\alpha_1}{(1-p)^r}) + (1-p)^r - \alpha_1 \\ &> 1 - (1-p)^r - (1-p)^{r-1} rp + \alpha_1 \ln(\frac{\alpha_1}{(1-p)^r}) + (1-p)^r - \alpha_1 \\ &= p^2(1 + 2(1-p) + \cdots + (r-1)(1-p)^{r-2}) + \alpha_1 \ln(\frac{\alpha_1}{(1-p)^r}) + (1-p)^r - \alpha_1 \\ &> \frac{r(r-1)}{2}(1-p)^{r-2}p^2 + \alpha_1 \ln(\frac{\alpha_1}{(1-p)^r}) + (1-p)^r - \alpha_1. \quad \blacksquare \end{aligned}$$

Hence we have the following clean lower bound for the ratio $(V(\mathrm{OPT}) - V(\mathrm{APPR}))/V(\mathrm{APPR})$.

Lemma 3.13

$$\frac{V(\mathrm{OPT}) - V(\mathrm{APPR})}{V(\mathrm{APPR})} > \min_{\alpha_1 < (1-p)^r} \frac{f_1(\alpha_1)}{f_2(\alpha_1)}.$$

Lemma 3.14 *For all $\alpha_1 < (1-p)^r$, $f_1(\alpha_1)/f_2(\alpha_1) > 1/(r-1)$.*

Proof. First consider the case $\alpha_1 \geq (1-p)^{2r-1}$. Let $\alpha_1 = (1-p)^x$. Hence, $r < x \leq 2r-1$. Observe that because the function $y \ln y + 1 - y$ is positive for all $y > 0$,

$$\alpha_1 \ln(\frac{\alpha_1}{(1-p)^r}) + (1-p)^r - \alpha_1 = (1-p)^r(\frac{\alpha_1}{(1-p)^r} \ln(\frac{\alpha_1}{(1-p)^r}) + 1 - \frac{\alpha_1}{(1-p)^r}) > 0.$$

This means $f_2(\alpha_1) > \frac{r(r-1)}{2}(1-p)^{r-2}p^2$. So,

$$\begin{aligned}
\frac{f_1(\alpha_1)}{f_2(\alpha_2)} &< \frac{(1-p)^x p^2 - (x-r)(1-p)^x p \ln(1-p)}{\frac{r(r-1)}{2}(1-p)^{r-2}p^2} \\
&< \frac{(1-p)^x p^2 + (x-r)(1-p)^{x-1}p^2}{\frac{r(r-1)}{2}(1-p)^{r-2}p^2} \\
&< \frac{1+x-r}{r(r-1)/2} \\
&\leq \frac{2r}{r(r-1)} \\
&= \frac{2}{r-1}.
\end{aligned}$$

When $\alpha_1 \leq (1-p)^{2r-1}$, we claim that the function $f_2(\alpha_1) - \frac{r-1}{2}f_1(\alpha_1)$ is decreasing in α_1.

Claim 3.15 *The function*

$$\begin{aligned}
& f_2(\alpha_1) - \tfrac{r-1}{2}f_1(\alpha_1) \\
= \; & \tfrac{r(r-1)}{2}(1-p)^{r-2}p^2 + \alpha_1 \ln(\tfrac{\alpha_1}{(1-p)^r}) + (1-p)^r - \alpha_1 - \tfrac{r-1}{2}(\alpha_1 p^2 - \alpha_1 p \ln(\tfrac{\alpha_1}{(1-p)^r}))
\end{aligned}$$

is decreasing in α_1, for $\alpha_1 \leq (1-p)^{2r-1}$.

Hence, for any $\alpha_1 \leq (1-p)^{2r-1}$,

$$f_2(\alpha_1) - \frac{r-1}{2}f_1(\alpha_1) \geq f_2((1-p)^{2r-1}) - \frac{r-1}{2}f_1((1-p)^{2r-1}) > 0. \qquad \blacksquare$$

This concludes the analysis for Pick-r-Stars.

Theorem 3.16 *Pick-r-Stars produces a single-batch policy with an expected value that is at least $(r-1)/(r+1)$ of the optimum.*

4 Approximating Optimal k-Batch Policies

Here, we present an algorithm for the linear cost and time threshold (**LT**) model that approximates optimal k-batch policies with a constant ratio 1/5. Recall that our simplifying assumption here is that time durations are equal, consequently optimal batched policies exist. The k-batch approximation algorithm, called Reverse-Greedy, is illustrated in Figure 2. The algorithm works by constructing an irreducible policy. It greedily constructs the policy batch by batch, starting from the last batch

```
1.   Sort the sources so that c_1/p_1 ≤ ··· ≤ c_n/p_n.
2.   APPR = ∅.   (* APPR denotes the k-batch policy. *)
3.   For i := k downto 1
4.     S = ∅.   (* S is the best i-th batch found so far. *)
5.     Q = 1.   (* Q is the collective failure probability of S. *)
6.     For j := 1 to n, where s_j ∉ APPR
7.       S_1 := {s_j}.
8.       Q_1 := 1 − p_j.   (* Q_1 is the collective failure probability of S_1. *)
9.       For l := 1 to n, where l ≠ j and s_l ∉ APPR
10.        If Q_1(1 − V(APPR)) ≥ c_l/p_l then   (* profitability check *)
11.          S_1 := S_1 ∪ {s_l}.
12.          Q_1 := Q_1(1 − p_l).
13.        else exit to step 14.
14.      If V(S) < V(S_1) then S := S_1; Q = Q_1.
15.    For each s_j in S   (* Make S irreducible. *)
16.      If c_j/p_j > Q/(1−p_j) (1 − V(APPR))
17.        S = S − {s_j}; Q = Q/(1 − p_j);
18.    Add S to APPR as the i-th batch.
19.  Output policy APPR.
```

Figure 2: The algorithm Reverse-Greedy.

(rightmost) and going in reverse time. For each batch, it invokes the single-batch algorithm Pick-a-Star, but with a modified profitability criterion that follows from equality 2 of Lemma 3.2 (see the comments following the Lemma). Even though each source is profitable at the time it is added to a partially constructed batch, the completed batch may not be irreducible, *i.e.* some of the sources picked before the last source may become nonprofitable after the addition of the last source. Thus after each call to Pick-a-Star, the algorithm scans back over the newly created batch and drops any source that is nonprofitable. In this way, the final policy is surely irreducible. Clearly Reverse-Greedy can be implemented to run in time $O(kn^2)$.

The analysis of Reverse-Greedy makes use of Theorem 3.8. The difficulty here is that because the sources can be scheduled in different batches, some batches of an optimal k-batch policy could be arbitrarily better individually than their counterparts in APPR. To get around this, we relate an irreducible k-batch policy to its optimally serialized version. For any policy $\mathcal{P}$, let $\overline{\mathcal{P}}$ denote the optimal serial policy for the sources in $\mathcal{P}$. It is not hard to see that $V(\overline{\mathcal{P}}) \geq V(\mathcal{P})$. Note that a serial policy $\overline{\mathcal{P}}$ may violate the time threshold, however we use $V(\overline{\mathcal{P}})$ in the analysis as a means of bounding the value of an optimal scheduling of the sources in $\mathcal{P}$. First, let us characterize an optimal serial policy.

Lemma 4.1 *For any set of sources, an optimal serial policy (including all sources in the set) sorts the sources in the nondecreasing order of their cost to success probability ratios.*

Proof. Consider any serial policy $\mathcal{P} = s_{i_1}, \ldots, s_{i_m}$. Then

$$V(\mathcal{P}) = p_{i_1} - c_{i_1} + (1 - p_{i_1})(p_{i_2} - c_{i_2}) + \cdots + \prod_{j=1}^{m-1} (1 - p_{i_j})(p_{i_m} - c_{i_m}).$$

Hence swapping source s_{i_j} and source $s_{i_{j+1}}$ would result in a net value of

$$\begin{aligned} & \prod_{l=1}^{j-1}(1-p_{i_l})(p_{i_{j+1}} - c_{i_{j+1}} + (1-p_{i_{j+1}})(p_{i_j} - c_{i_j})) - \prod_{l=1}^{j-1}(1-p_{i_l})(p_{i_j} - c_{i_j} + (1-p_{i_j})(p_{i_{j+1}} - c_{i_{j+1}})) \\ = & \prod_{l=1}^{j-1}(1-p_{i_l})(p_{i_{j+1}}c_{i_j} - p_{i_j}c_{i_{j+1}}), \end{aligned}$$

which is positive if $c_{i_{j+1}}/p_{i_{j+1}} < c_{i_j}/p_{i_j}$. ■

The following property of optimal serial policies is a side product of the above lemma and will be useful in the analysis.

Corollary 4.2 *Let S_1 and S_2 be two sets of sources and $S_1 \subseteq S_2$. An optimal serial policy for S_2 gives an expected value at least that of an optimal serial policy for S_1.*

Proof. Recall our basic assumption from section 3 that $p_i \geq c_i$ for all sources s_i. As a consequence an optimal serial policy for the sources in S_2 exists that includes all the sources in S_2. Starting from such an optimal serial policy for S_2, we can gradually swap the sources that are not in S_1 towards the end of the sequence, and eventually remove them. By Lemma 4.1, no such swap or removal can increase the expected value. ■

The following lemma, which is somewhat surprising, is a key to our analysis. It states that serializing an irreducible batched policy can at most triple the expected value.

Lemma 4.3 *Let P denote the success probability of an irreducible batched policy $\mathcal{P}$. Then $V(\overline{\mathcal{P}}) \leq (2+P)V(\mathcal{P})$.*

Proof. The proof is by induction on the number of sources of the policy. Assume that the statement holds for any irreducible policy consisting of $n-1 \geq 1$ sources, and consider an irreducible policy $\mathcal{P}$ consisting of n sources. Let S denote the set of sources in the first (leftmost) batch and s_n denote a source in S. In order to maximize the difference between $V(\overline{\mathcal{P}})$ and $V(\mathcal{P})$ we assume that the cost-to-success-probability ratios of all the sources in S are at the maximum possible (without violating the irreducibility condition). We perturb the initial policy $\mathcal{P}$ and transform it into another irreducible policy $\mathcal{P}'$, by moving s_n to the left in the time-line so that it finishes before all the other sources. We increase the cost of source s_n and other sources in S by amounts to be described below. Note that policy $\mathcal{P}'$ will not be different from policy $\mathcal{P}$ if $S - \{s_n\} = \emptyset$. Due to the potential increase in costs in the transformation, we will see that $V(\mathcal{P}') \leq V(\mathcal{P})$ and $V(\overline{\mathcal{P}'}) \leq V(\overline{\mathcal{P}})$. We first show that

$$2\left(V(\mathcal{P}) - V(\mathcal{P}')\right) \geq V(\overline{\mathcal{P}}) - V(\overline{\mathcal{P}'}). \tag{12}$$

We complete the proof by showing, using the inductive hypothesis, that the statement of the lemma holds for policy $\mathcal{P}'$.

Let Q denote the collective failure probability of the sources in S and V_a denote the expected value of policy $\mathcal{P}$ without batch S. Let $S' = S - \{s_n\}$. Since the cost-to-success-probability ratios of sources in S are as high as possible, we have $c_i/p_i = Q/(1-p_i)(1-V_a)$ for $s_i \in S$ according to the definition of irreducibility. Let c'_n and c'_i be the costs of sources s_n and $s_i \in S'$ in policy $\mathcal{P}'$. Again, we set these costs to be as high as they can be: $c'_i = p_i(Q/(1-p_i)(1-p_n))(1-V_a) = c_i/(1-p_n)$

and $c'_n = p_n(Q/(1-p_n) + \sum_{s_i \in S'} c'_i - Q/(1-p_n)V_a)$. Therefore we have that $\forall s_i \in S', c'_i - c_i = c'_i - (1-p_n)c'_i = c'_i p_n$ and $c'_n - c_n = \sum_{s_i \in S'} p_n c'_i$. Thus the change in the cost of s_n is equal to the total change in the costs of the sources in S'.

We have $V(\mathcal{P}') = 1 - Q - c'_n - (1-p_n)\sum_{s_i \in S'} c'_i + QV_a$, and $V(\mathcal{P}) = 1 - Q - \sum_{s_i \in S} c_i + QV_a$. Thus $V(\mathcal{P}) - V(\mathcal{P}') = c'_n - c_n = p_n \sum_{s_i \in S'} c'_i$. In other words, in going from policy $\mathcal{P}'$ to policy $\mathcal{P}$, only the reduction in the cost of source s_n is felt in the increase of the expected value. Observe that this reduction is exactly half of the total change in all costs. Now it is easy to see that $2\,(V(\mathcal{P}) - V(\mathcal{P}')) \geq V(\overline{\mathcal{P}}) - V(\overline{\mathcal{P}'})$: In the transition from the optimal serial policy $\overline{\mathcal{P}'}$ to the optimal serial policy $\overline{\mathcal{P}}$, the reduction in the costs of both source s_n and the sources in S', which is twice the reduction in the cost of source s_n, is reflected in the increase of the expected value. However, $V(\overline{\mathcal{P}}) - V(\overline{\mathcal{P}'}) \leq 2p_n \sum_{s_i \in S'} c'_i$, since in a serial policy the effect of a cost reduction on each source is reduced due to the success probability of the sources queried earlier.

Next we show

$$V(\overline{\mathcal{P}'}) \leq (2+P)V(\mathcal{P}'). \tag{13}$$

Note that source s_n is the only scheduled source in the leftmost batch of policy $\mathcal{P}'$, and since its cost-to-success-probability ratio c_n/p_n is maximized, $c_n/p_n = 1 - V$, where V denotes the expected value of the policy $\mathcal{P}'$ excluding s_n. Again, since c_n/p_n is maximized, $V = V(\mathcal{P}')$, *i.e.* the presence of s_n in $\mathcal{P}'$ does not increase $V(\mathcal{P}')$ (see Lemma 3.2). Hence $c_n/p_n = 1 - V(\mathcal{P}')$. Consider the subpolicy A consisting of the sources in $\mathcal{P}'$ with cost-to-success-probability ratio less than $1 - V(\mathcal{P}')$ and let P_A denote its success probability. Let B denote the subpolicy consisting of the rest of the sources in $\mathcal{P}'$ with success probability P_B. The sources in B have higher cost-to-success-probability ratios than the sources in A, hence they are queried later than the sources in A in policy $\overline{\mathcal{P}'}$. Therefore we have $V(\overline{\mathcal{P}'}) = V(\overline{A}) + (1-P_A)V(\overline{B})$. Since $\mathcal{P}'$ is irreducible, $V(A) \leq V(\mathcal{P}')$, and because subpolicy A is irreducible and does not include s_n, we have $V(\overline{A}) \leq (2+P_A)V(A) \leq (2+P_A)V(\mathcal{P}')$ by induction. It is not hard to see that $V(\overline{B}) \leq P_B V(\mathcal{P}')$: Let the sources in B be $s_1, \ldots, s_m$, and let α_i denote the cost-to-success-probability ratio of s_i, where $\alpha_1 \leq \cdots \leq \alpha_m$. Then

$$\begin{aligned}
V(\overline{B}) &= p_1(1-\alpha_1) + (1-p_1)p_2(1-\alpha_2) + \cdots + (\prod_{i=1}^{m-1}(1-p_i))p_m(1-\alpha_m) \\
&\leq \left(p_1 + (1-p_1)p_2 + \cdots + (\prod_{i=1}^{m-1}(1-p_i))p_m\right)(1-\alpha_1) \\
&\leq P_B V(\mathcal{P}')
\end{aligned}$$

Therefore

$$\begin{aligned}
V(\overline{\mathcal{P}'}) = V(\overline{A}) + (1-P_A)V(\overline{B}) & \\
&\leq 2V(\mathcal{P}') + P_A V(\mathcal{P}') + (1-P_A)P_B V(\mathcal{P}') \\
&= 2V(\mathcal{P}') + (P_A + (1-P_A)P_B)V(\mathcal{P}') \\
&= (2+P)V(\mathcal{P}')
\end{aligned}$$

We complete the proof using inequalities 12 and 13:

$$V(\overline{\mathcal{P}}) \leq 2(V(\mathcal{P}) - V(\mathcal{P}')) + V(\overline{\mathcal{P}'}))$$

$$\begin{aligned} &\leq 2V(\mathcal{P}) - 2V(\mathcal{P}') + (2+P)V(\mathcal{P}') \\ &\leq 2V(\mathcal{P}) + PV(\mathcal{P}) = (2+P)V(\mathcal{P}) \end{aligned}$$ ■

A similar proof shows that lemma 4.3 also holds for the general case of an irreducible schedule with sources that can have unequal durations. However the irreducibility criterion is slightly more complicated than what is derived from 3.2 for batched schedules, and unfortunately, unlike for the case for batched schedules, we don't know how to use that fact to derive an approximation algorithm for the general case. The interested reader is referred to [11] for a proof for the general case. Interestingly, serializing a single irreducible batch of sources can at most double the value (this result was used in [2]).

Now we analyze the performance of algorithm Reverse-Greedy. Just as in the case for single-batch policies, we will also use set operations on k-batch policies when there is no ambiguity. Denote the optimal policy as OPT, and partition OPT as

$$\begin{aligned} \mathrm{OPT}_1 &= \mathrm{APPR} \cap \mathrm{OPT}, \\ \mathrm{OPT}_2 &= \mathrm{OPT} - \mathrm{OPT}_1, \end{aligned}$$

where the sources in OPT_1 and OPT_2 are scheduled in the same batches as they are in OPT. By Lemma 3.1,

$$V(\mathrm{OPT}) \leq V(\mathrm{OPT}_1) + V(\mathrm{OPT}_2).$$

We compare the performances of OPT_1 and OPT_2 with that of APPR separately.

Lemma 4.4 $V(\mathrm{OPT}_1) \leq 3V(\mathrm{APPR})$.

Proof. This follows immediately from Corollary 4.2 and Lemma 4.3. ■

Lemma 4.5 $V(\mathrm{OPT}_2) \leq 2V(\mathrm{APPR})$.

Proof. For each $i = 1, \ldots, k$, let $\mathrm{OPT}_2(i)$ and $\mathrm{APPR}(i)$ denote the subpolicies of OPT_2 and APPR consisting of the last $k - i + 1$ batches. We prove inductively, starting from the last batch, that $V(\mathrm{OPT}_2(i)) \leq 2V(\mathrm{APPR}(i))$. Without loss of generality, we may assume that $V(\mathrm{APPR}(i)) \leq V(\mathrm{OPT}_2(i))$ for all i because otherwise we could always replace the last $k - i + 1$ batches of OPT_2 with those of APPR and continue with the induction. This could only improve $V(\mathrm{OPT}_2)$.

The base of the induction is clearly true by Theorem 3.8. Note that making each batch irreducible can only increase the expected value of the batch. Suppose that the claim holds for $i + 1 \leq k$, and consider batch i. Let P_o, C_o be the collective success probability and cost of the sources in the i-th batch of OPT_2 and P_a, C_a the corresponding quantities for APPR. By Lemma 3.2,

$$V(\mathrm{APPR}(i)) - V(\mathrm{APPR}(i+1)) = P_a(1 - V(\mathrm{APPR}(i+1))) - C_a$$

$$V(\mathrm{OPT}_2(i)) - V(\mathrm{OPT}_2(i+1)) = P_o(1 - V(\mathrm{OPT}_2(i+1))) - C_o$$

Taking the ratio and noting that $V(\mathrm{APPR}(i+1)) \leq V(\mathrm{OPT}_2(i+1))$,

$$\frac{P_a(1 - V(\mathrm{APPR}(i+1))) - C_a}{P_o(1 - V(\mathrm{OPT}_2(i+1))) - C_o} = \left(\frac{1 - V(\mathrm{APPR}(i+1))}{1 - V(\mathrm{OPT}_2(i+1))}\right) \frac{P_a - C_a/(1 - V(\mathrm{APPR}(i+1)))}{P_o - C_o/(1 - V(\mathrm{OPT}_2(i+1)))}$$

$$\geq \frac{P_a - C_a/(1 - V(\mathrm{APPR}(i+1)))}{P_o - C_o/(1 - V(\mathrm{OPT}_2(i+1)))}$$
$$\geq \frac{P_a - C_a/(1 - V(\mathrm{APPR}(i+1)))}{P_o - C_o/(1 - V(\mathrm{APPR}(i+1)))}.$$

Let set S consist of all sources that do not appear in APPR$(i+1)$. Clearly, S includes all sources in OPT_2 and thus all sources in the i-th batch of OPT_2. Divide the cost of each source by $1 - V(\mathrm{APPR}(i+1))$. Then on input S, Pick-a-Star would return exactly the same set as the i-th batch of APPR with expected value being $P_a - C_a/(1 - V(\mathrm{APPR}(i+1)))$. By Theorem 3.8, the value is at least half of the optimal expected value for set S which is in turn at least $P_o - C_o/(1 - V(\mathrm{APPR}(i+1)))$. This means

$$2(V(\mathrm{APPR}(i)) - V(\mathrm{APPR}(i+1))) \geq V(\mathrm{OPT}_2(i)) - V(\mathrm{OPT}_2(i+1)),$$

and hence $2V(\mathrm{APPR}(i)) \geq V(\mathrm{OPT}_2(i))$. ■

Lemmas 4.5 and 4.4 together give the following theorem.

Theorem 4.6 *Algorithm Reverse-Greedy returns a k-batch policy with an expected value at least $1/5$ of the optimum.*

5 Approximation Algorithms for the Cost Threshold Models

We first present an FPTAS for model **TL** under a weak assumption: $p_i - d_i \geq 0$ for every source s_i (d_i is the time duration of s_i), *i.e.* every source considered is profitable by itself. The extension to model **TT** (with no restriction) is straightforward. Note that in model **TT** our goal is simply to maximize the overall probability of getting the information under the time and cost constraints.

The main idea is the rounding technique introduced in [7] for Knapsack. As mentioned before, in the cost threshold model **TL**, an optimal policy should be in fact a single-batch policy. Let $\mathcal{P} = \{s_{i_1}, \ldots, s_{i_m}\}$ be a single-batch policy, where $d_{i_1} \leq \cdots \leq d_{i_m}$. Then,

$$V(\mathcal{P}) = 1 - \prod_{j=1}^{m}(1 - p_{i_j}) - \sum_{j=1}^{m}\prod_{l=1}^{j-1}(1 - p_{i_l})p_{i_j}\, d_{i_j} - \prod_{j=1}^{m}(1 - p_{i_j})d_{i_m}.$$

We cannot apply the rounding technique to the above objective function directly because it involves subtractions. Let's rewrite the expression as

$$\begin{aligned} V(\mathcal{P}) &= \sum_{j=1}^{m}\prod_{l=1}^{j-1}(1 - p_{i_l})p_{i_j}(1 - d_{i_j}) - \prod_{j=1}^{m}(1 - p_{i_j})d_{i_m} \\ &= \sum_{j=1}^{m-1}\prod_{l=1}^{j-1}(1 - p_{i_l})p_{i_j}(1 - d_{i_j}) + \prod_{j=1}^{m-1}(1 - p_{i_j})(p_{i_m} - d_{i_m}) \end{aligned} \tag{14}$$

Since $p_i - d_i \geq 0$ by our assumption, every term is nonnegative in equation 14, and we can apply rounding as follows.

Let $\epsilon > 0$ be any desired relative error. Sort the sources in the ascending order of their time durations. We will exhaustively consider every possible choice of s_{i_m}. For each $i \leq n$, consider only policies that includes the source s_i and possibly some others from $\{s_1, \ldots, s_{i-1}\}$, subject to the same cost threshold. Let $\mathrm{OPT}(i)$ denote an optimal such policy. For simplicity, assume that $|\mathrm{OPT}(i)| > 1$. We find a trivial lower bound for $V(\mathrm{OPT}(i))$:

$$V(\mathrm{OPT}(i)) \geq L_i = \max\{p_i - d_i, \max_{\substack{j<i \\ c_j + c_i \leq \varsigma}} p_j(1 - d_j)\}.$$

Similar to [7], we formulate a new instance by rounding $p_i - d_i$ and each $p_j(1-d_j)$ *down* to the nearest multiple of $\epsilon L_i/(2i)$. But here we also need round each $1-p_j$ to the nearest power of $(1-\epsilon/(2i))^{1/(i-1)}$. In other words, we round each $\log(1-p_j)$ to the nearest multiple of $(\log(1-\epsilon/(2i)))/(i-1)$. We solve the new instance optimally. Let OPT_i denote an optimal policy for the new instance. It is sufficient to bound the difference between $V(\mathrm{OPT}(i))$ and $V(\mathrm{OPT}_i)$ and we do this by obtaining an upper bound between each term of $OPT(i)$ and the corresponding rounded value. Each term difference is upper bounded when the $p_i - d_i$ or $p_i(1-d_i)$ part of a term is at its maximum L_i and the probability factors in the unrounded term are all 1 (otherwise they are factored out and reduce the difference). Thus we obtain a maximum difference of $L_i - L_i(1-\frac{\epsilon}{2i})[(1-\frac{\epsilon}{2i})^{\frac{1}{i-1}}]^{i-1}$ for every term and there can be at most i many:

$$\begin{aligned}
V(\mathrm{OPT}(i)) - V(\mathrm{OPT}_i) &\leq i\left(L_i - L_i(1-\frac{\epsilon}{2i})[(1-\frac{\epsilon}{2i})^{\frac{1}{i-1}}]^{i-1}\right) \\
&= iL_i(1-(1-\frac{\epsilon}{2i})^2) \\
&\leq iL_i(1-(1-\frac{\epsilon}{i})) \\
&= \epsilon L_i \\
&\leq \epsilon V(\mathrm{OPT}(i))
\end{aligned}$$

Hence, the new instance approximates the original problem with the desired ratio.

We can compute OPT_i for the new instance by dynamic programming in the space $\mathcal{S}_i$ of all possible values of $V(\mathrm{OPT}_i)$. Denote $Q_i = \prod_{j<i}(1-p_j)$. By the above rounding, the cardinality of $\mathcal{S}_i$ is upper bounded by

$$i(\frac{2i}{\epsilon})(\frac{(i-1)\log Q_i}{\log(1-\epsilon/(2i))}) < i(\frac{2i}{\epsilon})(\frac{2i^2 \log Q_i}{-\epsilon}) = \frac{-4i^4 \log Q_i}{\epsilon^2},$$

which is polynomial in the input size and $1/\epsilon$. In the above inequality, the factor $\frac{2i}{\epsilon}$ represents the number of different values that $p_i - d_i$ and $p_j(1-d_j)$ can have after being rounded to the nearest multiple of $\epsilon L_i/(2i)$, and the factor $\frac{(i-1)\log Q_i}{\log(1-\epsilon/(2i))}$ represents the number of different exponents that a product of the form $\prod_{l=1}^{j-1}(1-p_{i_l})$ can have after being rounded to the nearest power of $(1-\epsilon/(2i))^{1/(i-1)}$.

Rewrite the expression in equality 14 as a nested form:

$$p_{i_1}(1-d_{i_1}) + (1-p_{i_1})[p_{i_2}(1-d_{i_2}) + (1-p_{i_2})[\cdots + (1-p_{i_{m-1}})(p_{i_m} - d_{i_m})]].$$

The form easily suggests a backward inductive algorithm. The algorithm will cycle through the list $s_{i-1}, \ldots, s_1$. For each $j = i, \ldots, 1$ and each possible value $x \in \mathcal{S}_i$, it computes and records a policy

of expected value x for the subset of sources $\{s_j, \ldots, s_i\}$ that contains the source s_i and costs the least. The above nested form allows the algorithm to find the cheapest policy of a specific expected value for subset $\{s_j, \ldots, s_i\}$ by expanding the cheapest policies of the same or lower expected values recorded before for subset $\{s_{j+1}, \ldots, s_i\}$ to potentially include the source s_j. The running time is at most $O(i^2|\mathcal{S}_i|)$, which is polynomial in the input size and $1/\epsilon$.

Theorem 5.1 *Assume that $p_i - d_i \geq 0$ for every source s_i. There is an FPTAS for the problem of computing optimal policies in model* **TL**.

Corollary 5.2 *There is an FPTAS for the problem of computing optimal policies in model* **TT**.

Proof. Recall that the objective function in this case is

$$V(\mathcal{P}) = 1 - \prod_{j=1}^{m}(1 - p_{i_j}),$$

which involves only the success probabilities of the sources queried. Hence, the above FPTAS works if we simply throw out all sources whose time duration exceeds the deadline. ■

6 Concluding Remarks

As charging for information on the Internet becomes more common, information-acquisition algorithms will have to trade off the benefits of acquiring information with the cost of doing so. Characteristics of these problems are (1) the fact that the information provided by a source cannot be fully predicted in all cases, so the benefit of asking for information can be uncertain, (2) the fact that there can be monetary and time costs associated with information requests, and (3) the fact that information providers can be accessed both serially and in parallel.

We have developed a model that takes into account these aspects of information scheduling, and have established worst-case complexity and approximation results for a variety of objective functions: those in which the value is linear in the cost or time attributes and the consumer supplies a cost and/or time threshold for acquiring information. All of these models have plausible applications for information access on the World Wide Web.

Acknowledgments

Many thanks to Richard Anderson and Richard Karp, for the discussions and their very valuable suggestions, and to the two anonymous referees for their careful reading and constructive criticism of the paper.

References

[1] D. Dreilinger. Integrating Heterogeneous WWW Search Engines. *Masters Thesis*, Colorado State University. May, 1995.

[2] O. Etzioni, S. Hanks, T. Jiang, R. M. Karp, O. Madani, and O. Waarts. Efficient Information Gathering on the Internet. *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, 1996.

[3] O. Etzioni, R. M. Karp, and O. Waarts. Efficient Access to Information Sources on the Internet. Manuscript 1996.

[4] P. Feigin and G. Harel. Minimizing costs of personnel testing programs. *Naval Research Logistics Quarterly* 29, 87-95, 1982.

[5] M. Garey. Optimal task scheduling with precedence constraints. *Discrete Mathematics*, 4, 37-56 (1973).

[6] M. Henig and D. Simchi-Levy. Scheduling tasks with failure probabilities to minimize expected cost. *Naval Research Logistics* 37,99-109, 1990.

[7] O. Ibarra and C. Kim. Fast approximation algorithms for the knapsack and sum of subsets problems. *Journal of the ACM* 22, 463-368, 1975.

[8] The LEXIS-NEXIS corporation. 1997. At `http://www.lexis-nexis.com/lncc/`

[9] J. Kadane. Quiz show problems. *Mathematical Analysis and Applications* 27, 609-623, 1969.

[10] B. Krulwich. The BargainFinder agent: Comparison price shopping on the Internet. *Bots and Other Internet Beasties*. 1996.

[11] O. Madani. Serializing an Irreducible Schedule at most Triples the Value. Manuscript 1996.

[12] The Knowledge Finder corporation, providing access to MEDLINE. 1997. At `http://www.kfinder.com/`

[13] L. Mitten. An analytic solution to the least cost testing sequence problem. *J. Indust. Eng.* 11, 1960.

[14] New York Times, June 7, 1992

[15] M. Porat. *The Information Economy*, US. Office of Telecommunications, 1977.

[16] S. Sahni. Approximation algorithms for the 0/1-knapsack problem. *Journal of the ACM* 22, 115-124, 1975.

[17] Secure Transaction Technology. In `http://www.visa.com/cgi-bin/vee/sf/set/intro.html`.

[18] E. Selberg and O. Etzioni. Multi-Service Search and Comparison Using the MetaCrawler. *Proc. 4th World Wide Web Conf.*, 195-208, Boston, MA, 1995.

[19] H. Simon and J. Kadane. Optimal problem-solving search: all-or-none solutions. *Artificial Intelligence* 6, 235-247, 1975.

[20] M. Sirbu, and J.D. Tygar. NetBill: An Internet Commerce System Optimized for Network Delivered Services. Manuscript 1995. To appear in *IEEE CompCon Conference*.

APPENDIX. Verifying the Claims.

We prove the claims made on the behavior of the various functions that came up in the main proofs.

Proof of Claim 3.5. We show that the function $P_1 - l(1-(1-P_1)^{1/l})\alpha_1$ is decreasing in P_1, where $1 - P_1 \geq \alpha_1$.

We take the derivative with respect to P_1 and get

$$1 - \alpha_1(1-P_1)^{\frac{1}{l}-1} = 1 - \frac{\alpha_1}{(1-P_1)^{\frac{l-1}{l}}}$$

The ratio is increasing in P_1 as long as $(1-P_1)^{\frac{l-1}{l}} > \alpha_1$. ■

Proof of Claim 3.6. We show that the function $h(l, p_1) = l(1-(1-p_1)^{1/l})$ is nondecreasing in l.

Taking the derivative we have

$$\frac{\partial h(l,p_1)}{\partial l} = 1 - (1-p_1)^{1/l} + \frac{\ln(1-p_1)(1-p_1)^{1/l}}{l}$$

We observe that $\frac{\partial h(l,0)}{\partial l} = 1 - 1 + 0 = 0$ and

$$\begin{aligned}\frac{\partial h(l,p_1)}{\partial p_1 \partial l} &= 1/l(1-p_1)^{1/l-1} - \frac{(1-p_1)^{1/l}}{l(1-p_1)} - \frac{(1-p_1)^{1/l-1}\ln(1-p_1)}{l^2} \\ &= -\frac{(1-p_1)^{1/l-1}\ln(1-p_1)}{l^2} \geq 0\end{aligned}$$

Therefore $\frac{\partial h(l,p_1)}{\partial l}$ is nonnegative for all $p_1 \in [0, 1.0]$. Whence $h(l, p_1)$ is nondecreasing in l. ■

Proof of Claim 3.7. We need show that the ratio $\left(\frac{1}{1-x^{m/(m-1)}}\right)\left(\frac{1-x+x\ln x}{1-x}\right)$ is decreasing in $x \in (0,1)$.

We take the derivative

$$\begin{aligned}&\left(\frac{\frac{m}{m-1}x^{\frac{1}{m-1}}}{(1-x^{\frac{m}{m-1}})^2}\right)\left(\frac{1-x+x\ln x}{1-x}\right) + \left(\frac{1}{1-x^{m/(m-1)}}\right)\left(\frac{(1-x)\ln x + 1 - x + x\ln x}{(1-x)^2}\right) \\ &= \frac{1}{(1-x^{m/(m-1)})(1-x)}\left[\left(\frac{\frac{m}{m-1}x^{\frac{1}{m-1}}}{1-x^{\frac{m}{m-1}}}\right)(1-x+x\ln x) + \left(\frac{1-x+\ln x}{1-x}\right)\right]\end{aligned}$$

The factor $\frac{1}{(1-x^{m/(m-1)})(1-x)}$ is poitive in the interval, hence it suffices to show that the term in the brackets is negative. Noting that $1 - x + x\ln x$ in the left summand is nonnegative (the derivative $\ln x$ is negative, and the function is zero at 1), and that in the right summand $1 - x + \ln x \leq 0$, we conclude that the left summand is nonnegative, while the right one is nonpositive. We will verify that

$$\frac{1}{1-x} \geq \frac{\frac{m}{m-1}x^{\frac{1}{m-1}}}{1-x^{\frac{m}{m-1}}}. \tag{15}$$

This eliminates the extra factors and makes it sufficient to show that $h_1(x) = (1-x+x\ln x) + 1 - x + \ln x < 0$. We note that $h_1(x)$ is negative at x arbitrarily close to zero, and zero at $x = 1$, and its derivative $-1 + \ln x + 1/x = \frac{1-x+x\ln x}{x}$ is nonnegative in the interval.

Rearranging 15 we need to verify that

$$\frac{\frac{m}{m-1}x^{\frac{1}{m-1}}(1-x)}{1-x^{\frac{m}{m-1}}} \leq 1.$$

The fraction is zero at $x = 0$ and taking the limit as x approaches 1 we have

$$\lim_{x\to 1} \frac{\frac{m}{m-1}x^{\frac{1}{m-1}}(1-x)}{1-x^{\frac{m}{m-1}}} = \frac{\frac{m}{m-1}(\frac{1}{m-1}-\frac{m}{m-1})}{-\frac{m}{m-1}} = 1$$

Finally, we can verify that the derivative is nonnegative in the region. We may ignore the factor $\frac{m}{m-1}$ and to simplify a little, we make the substitution $u = x^{\frac{1}{m-1}}$. Hence we want to show that the derivative of $h_2(u) = \frac{u(1-u^{m-1})}{1-u^m}$ is positive. Taking the derivative,

$$h_2'(u) = \frac{(1-mu^{m-1})(1-u^m) + mu^m(1-u^{m-1})}{(1-u^m)^2}$$

We will show that the numerator

$$1 - mu^{m-1} - u^m + mu^{2m-1} + mu^m - mu^{2m-1} = u^{m-1}((m-1)u - m) + 1$$

is positive. The function $h_3(u) = u^{m-1}((m-1)u-m)+1$ is nonnegative on the interval since $h_3(0) = 1$ and $h_3(1) = 0$ and $h_3'(u) < 0$ on the interval:

$$\begin{aligned} h_3'(u) &= (m-1)u^{m-2}((m-1)u-m) + (m-1)u^{m-1} \\ &= (m-1)u^{m-2}((m-1)u-m+u) = (m-1)u^{m-2}(m(u-1)) < 0. \end{aligned}$$

Hence $h_2'(u) \geq 0$. ■

Proof of Claim 3.15. We need to show that the function

$$\begin{aligned} & f_2(\alpha_1) - \tfrac{r-1}{2}f_1(\alpha_1) \\ = \; & \tfrac{r(r-1)}{2}(1-p)^{r-2}p^2 + \alpha_1\ln(\tfrac{\alpha_1}{(1-p)^r}) + (1-p)^r - \alpha_1 - \tfrac{r-1}{2}(\alpha_1 p^2 - \alpha_1 p\ln(\tfrac{\alpha_1}{(1-p)^r})) \end{aligned}$$

is decreasing in α_1, for $\alpha_1 \leq (1-p)^{2r-1}$. Taking the partial derivative on $f_2(\alpha_1) - \frac{r-1}{2}f_1(\alpha_1)$ with respect to α_1,

$$\begin{aligned} & \frac{\partial\left[\frac{r(r-1)}{2}(1-p)^{r-2}p^2 + \alpha_1\ln(\frac{\alpha_1}{(1-p)^r}) + (1-p)^r - \alpha_1 - \frac{r-1}{2}(\alpha_1 p^2 - \alpha_1 p\ln(\frac{\alpha_1}{(1-p)^r}))\right]}{\partial\alpha_1} \\ = \; & \ln(\frac{\alpha_1}{(1-p)^r}) + 1 - 1 - \frac{r-1}{2}p^2 + \frac{(r-1)p}{2}\ln(\frac{\alpha_1}{(1-p)^r}) + \frac{(r-1)p}{2} \\ < \; & \ln(\frac{\alpha_1}{(1-p)^r}) + \frac{(r-1)p}{2} \\ \leq \; & (r-1)\ln(1-p) + \frac{(r-1)p}{2} \\ < \; & -(r-1)p + \frac{(r-1)p}{2} \\ \leq \; & 0. \quad ■ \end{aligned}$$