Approximating the Online Set Multicover Problems Via Randomized Winnowing*

Piotr Berman†
Department of Computer Science and Engineering
Pennsylvania State University
University Park, PA 16802
Email: berman@cse.psu.edu

Bhaskar DasGupta‡
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
Email: dasgupta@cs.uic.edu

June 11, 2005

Abstract

In this paper, we consider the weighted online set k-multicover problem. In this problem, we have an universe V of elements, a family $\mathcal{S}$ of subsets of V with a positive real cost for every $S \in \mathcal{S}$, and a "coverage factor" (positive integer) k. A subset $\{i_0, i_1, \ldots\} \subseteq V$ of elements are presented online in an arbitrary order. When each element i_p is presented, we are also told the collection of all (at least k) sets $\mathcal{S}_{i_p} \subseteq \mathcal{S}$ and their costs in which i_p belongs and we need to select additional sets from $\mathcal{S}_{i_p}$ if necessary such that our collection of selected sets contains *at least* k sets that contain the element i_p. The goal is to *minimize* the *total cost* of the selected sets[1]. In this paper, we describe a new randomized algorithm for the online multicover problem based on the randomized winnowing approach of [11]. This algorithm generalizes and improves some earlier results in [1]. We also discuss lower bounds on competitive ratios for *deterministic algorithms* for general k based on the approaches in [1].

1 Introduction

In this paper, we consider the Weighted Online Set k-multicover problem (abbreviated as $\mathbf{WOSC}_k$) defined as follows. We have an universe $V = \{1, 2, \ldots, n\}$ of elements, a family $\mathcal{S}$ of subsets of U with a cost (positive real number) c_S for every $S \in \mathcal{S}$, and a "coverage factor" (positive integer) k. A subset $\{i_0, i_1, \ldots\} \subseteq V$ of elements are presented in an arbitrary order. When each element i_p is presented, we are also told the collection of all (at least k) sets $\mathcal{S}_{i_p} \subseteq \mathcal{S}$ in which i_p belongs and we need to select additional sets from $\mathcal{S}_{i_p}$, if necessary, such that our collection of sets contains *at least* k sets that contain the element i_p. The goal is to minimize the total cost of the selected sets. The special case of $k = 1$ will be simply denoted by **WOSC** (Weighted Online Set Cover). The unweighted versions of these problems, when the cost any set is one, will be denoted by $\mathbf{OSC}_k$ or **OSC**.

*A preliminary version of these results will appear in 9th Workshop on Algorithms and Data Structures, Waterloo, Canada, August 15-August 17, 2005.

†Supported by NSF grant CCR-0208821.

‡Supported in part by NSF grants CCR-0206795, CCR-0208749 and IIS-0346973.

[1]Our algorithm and competitive ratio bounds can be extended to the case when a set can be selected at most a prespecified number of times instead of just once; we do not report these extensions for simplicity.

The performance of any online algorithm can be measured by the *competitive ratio*, *i.e.*, the ratio of the total cost of the online algorithm to that of an optimal offline algorithm that knows the entire input in advance; for randomized algorithms, we measure the performance by the *expected* competitive ratio, *i.e.*, the ratio of the expected cost of the solution found by our algorithm to the optimum cost computed by an adversary that knows the entire input sequence and has no limits on computational power, but who is *not familiar* with our random choices.

The following notations will be used uniformly throughout the rest of the paper unless otherwise stated explicitly:

- V is the universe of elements;
- $m = \max_{i \in V} |\{S \in \mathcal{S} \mid i \in S\}|$ is the maximum *frequency*, *i.e.*, the maximum number of sets in which any element of V belongs;
- $d = \max_{S \in \mathcal{S}} |S|$ is the maximum set size;
- k is the coverage factor.

None of m*,* d *or* $|V|$ *is known to the online algorithm in advance.*

1.1 Motivations and Applications

There are several applications for investigating the online settings in $\mathbf{WOSC}_k$. Below we mention two such applications.

1.1.1 Client/Server Protocols [1]

Such a situation is modeled by the problem **WOSC** in which there is a network of servers, clients arrive one-by-one in arbitrary order, and the each client can be served by a subset of the servers based on their geographical distance from the client. An extension to $\mathbf{WOSC}_k$handles the scenario in which a client must be attended to by at least a minimum number of servers for, say, reliability, robustness and improved response time. In addition, in our motivation, we want a distributed algorithm for the various servers, namely an algorithm in which each server locally decide about the requests without communicating with the other servers or knowing their actions (and, thus for example, not allowed to maintain a potential function based on a subset of the servers such as in [1]).

1.1.2 Reverse Engineering of Gene/Protein Networks [2, 4, 6, 9, 10, 14, 15]

We briefly explain this motivation here due to lack of space; the reader may consult the references for more details. This motivation concerns unraveling (or "reverse engineering") the web of interactions among the components of complex protein and genetic regulatory networks by observing global changes to derive interactions between individual nodes. In this application our attention is focused solely on one such approach, originally described in [9, 10], further elaborated upon in [2, 14], and reviewed in [6, 15]. Here one assumes that the time evolution of a vector of state variables $x(t) = (x_1(t), \ldots, x_n(t))$ is described by a system of differential equations:

$$\frac{\partial \vec{x}}{\partial t} = f(\vec{x}, \vec{p}) \equiv \begin{cases} \frac{\partial x_1}{\partial t} & = & f_1(x_1, \ldots, x_n, p_1, \ldots, p_m) \\ \frac{\partial x_2}{\partial t} & = & f_2(x_1, \ldots, x_n, p_1, \ldots, p_m) \\ & \vdots & \\ \frac{\partial x_n}{\partial t} & = & f_n(x_1, \ldots, x_n, p_1, \ldots, p_m) \end{cases}$$

where $\vec{p} = (p_1, \ldots, p_m)$ is a vector of parameters, such as levels of hormones or of enzymes, whose half-lives are long compared to the rate at which the variables evolve and which can be manipulated but remain constant during any given experiment. The components $x_i(t)$ of the state vector represent quantities that can be in principle measured, such as levels of activity of selected proteins or transcription rates of certain genes. There is a reference value $\bar{p}$ of $\vec{p}$, which represents "wild type" (that is, normal) conditions, and a corresponding steady state $\bar{x}$ of $\vec{x}$, such that $f(\bar{x}, \bar{p}) = 0$. We are interested in obtaining information about the Jacobian of the vector field f evaluated at $(\bar{x}, \bar{p})$, or at least about the signs of the derivatives $\partial f_i / \partial x_j(\bar{x}, \bar{p})$. For example, if $\partial f_i / \partial x_j > 0$, this means that x_j has a positive (catalytic) effect upon the rate of formation of x_i. To be precise, we want to find as much information as possible about the unknown matrix A which is the Jacobian matrix $\partial f / \partial x$. The critical assumption is that, while we may not know the form of f, we often do know that *certain parameters* p_j *do not directly affect certain variables* x_i. This amounts to *a priori* biological knowledge of specificity of enzymes and similar data. Such a knowledge can be summarized by a binary matrix $C = (c_{ij}) \in \{0, 1\}^{n \times m}$, where "$c_{ij} = 0$" means that p_j does not appear in the equation for $\dot{x}_i$, that is, $\partial f_i / \partial p_j \equiv 0$. In our current context, each row of C correspond to a set, each column of C correspond to an element, and 0-1 entries indicate the memberships of elements in sets. A crucial contribution of the above-mentioned references in this context is as follows. Suppose that we solve this set-multicover instance in which each element is covered at least some β times. Then with $\beta = n - 1$ we can recover the elements of A *uniquely up to a scalar multiple* and with $\beta = n - k$ for some small k we can recover the elements of A up to a modest ambiguity that can be tolerated in practice. This is done via computing another matrix $B = (b_{ij}) \in \mathbb{R}^{n \times m}$ and then using some linear algebraic manipulations. When the j^{th} column of A, that is the corresponding element of our set-multicover with all the sets containing it, is revealed, one also performs an experimental protocol such as the one described below:

- Change (perturb) parameter p_j.
- Measure the resulting steady state vector $\vec{x} = \xi(p)$. Experimentally, this may for instance mean that the concentration of a certain chemical represented by p_j is kept are a slightly altered level, compared to the default value $\bar{p}_j$; then, the system is allowed to relax to steady state, after which the complete state $\vec{x}$ is measured, for example by means of a suitable biological reporting mechanism, such as a microarray used to measure the expression profile of the variables x_i.
- Estimate the n "sensitivities"

  $$b_{ij} = \frac{\partial \xi_i}{\partial p_j}(\bar{p}) \approx \frac{1}{\bar{p}_j - p_j} \left(\xi_i(\bar{p} + p_j e_j) - \xi_i(\bar{p}) \right)$$

  for $i = 1, \ldots, n$ (where $e_j \in \mathbb{R}^m$ is the j^{th} canonical basis vector).

The cost of doing these experiments is associated with the weights of the sets, the unweighted case being the simplest case when we just wish to minimize the number of experiments.

The offline version of the problem, corresponding to the offline version of the set-multicover problem considered in [4], is the case when the matrix C is known precisely without any ambiguity. This is done in [4] by solving a linear program and then via appropriate randomized rounding. The online version of the problem, considered in this paper, is more suited to the case when C is not known without significant ambiguity which can either only be precisely resolved when the actual DNA microarray experiment is performed or will need to be tolerated by the algorithm. Without any special knowledge about the ambiguities of the elements of C, we can just perform the experiments in an online fashion in an arbitrary order. Instead of running the costly linear programming computations after every experiment, we simply choose the covering sets in a computationally fast online fashion. Notice that our randomized winnowing approach is adaptable to minor errors in the data. At the end of all experiments, we can then have two choices. If we are satisfied with the quality of the solution presented by the online set-multicover solutions, we need to do nothing more. Otherwise, we can run any suitable offline algorithm for the problem just once; if the algorithm is of branch-and-bound type then we also put the estimate of the cost of the online algorithm as an upper bound for the true optimal cost.

1.2 Summary of Prior Work

Offline versions $\mathbf{SC}_1$ and $\mathbf{SC}_k$ of the problems $\mathbf{WOSC}_k$ and $\mathbf{OSC}_k$, in which all the $|V|$ elements are presented at the same time, have been well studied in the literature. Assuming $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{\log\log n})$, the $\mathbf{SC}_1$ problem cannot be approximated to within a factor of $(1-\varepsilon)\ln|V|$ for any constant $0 < \varepsilon < 1$ in polynomial time [7]; a slightly weaker lower bound under the more standard complexity-theoretic assumption of P$\neq$NP was obtained by Raz and Safra [13] who showed that there is a constant c such that it is NP-hard to approximate the $\mathbf{SC}_1$ problem to within a factor of $c\ln|V|$. An instance of the $\mathbf{SC}_k$ problem can be $(1+\ln d)$-approximated in $O(|V|\cdot|\mathcal{S}|\cdot k)$ time by a simple greedy heuristic that, at every step, selects a new set that covers the maximum number of those elements that has not been covered at least k times yet [8, 16]; these results was recently improved upon in [4] who provided a randomized approximation algorithm with an expected performance ratio that about $\ln(d/k)$ when d/k is at least about $\mathbf{e}^2 \approx 7.39$, and for smaller values of d/k it decreases towards 1 as a linear function of $\sqrt{d/k}$.

Regarding previous results for the online versions, the authors in [1] considered the **WOSC** problem and provided both a deterministic algorithm with a competitive ratio of $O(\log m \log|V|)$ and an almost matching lower bound of $\Omega\left(\frac{\log|\mathcal{S}|\log|V|}{\log\log|\mathcal{S}|+\log\log|V|}\right)$ on the competitive ratio for any deterministic algorithm for almost all values[2] of $|V|$ and $|\mathcal{S}|$. The authors in [3] provided an efficient randomized online approximation algorithm and a corresponding matching lower bound (for any randomized algorithm) for a different version of the online set-cover problem in which one is allowed to pick at most k sets for a given k and the goal is to maximize the number of presented elements for which at least one set containing them was selected on or before the element was presented. To the best of our knowledge, there are no prior non-trivial results for either $\mathbf{WOSC}_k$ or $\mathbf{OSC}_k$ for general $k > 1$.

1.3 Summary of Our Results and Techniques

Let $r(m, d, k)$ denote the competitive ratio of any online algorithm for $\mathbf{WOSC}_k$ as a function of m, d and k. In this paper, we describe a new randomized algorithm for the online multicover problem based on the randomized winnowing approach of [11]. Our main contributions are then as follows:

- We first provide an uniform analysis of our algorithm for all cases of the online set multicover problems. As a corollary of our analysis, we observe the following.
  - For **OSC**, **WOSC** and $\mathbf{WOSC}_k$ our randomized algorithm has $\mathsf{E}[r(m, d, k)]$ equal to $\log_2 m \ln d$ plus small lower order terms. While the authors in [1] did obtain a deterministic algorithm for **OSC** with $O(\log m \log|V|)$ competitive ratio, the advantages of our approach are more uniform algorithm with simpler analysis, as well as better constant

[2]To be precise, when $\log_2|V| \le |\mathcal{S}| \le \mathbf{e}^{|V|^{\frac{1}{2}-\delta}}$ for any fixed $\delta > 0$; we will refer to similar bounds as "almost all values" of these parameters in the sequel.

factors and usage of the maximum set size d rather than the larger universe size $|V|$ in the competitive ratio bound. Unlike the approach in [1], our algorithm does not need to maintain a global potential function over a subcollection of sets.

  – For (the unweighted version) $\mathbf{OSC}_k$ for general k the expected competitive ratio $\mathsf{E}[r(m,d,k)]$ decreases logarithmically with increasing k with a value of roughly $5\log_2 m$ in the limit for all sufficiently large k.

- We next provide an improved analysis of $\mathsf{E}[r(m,d,1)]$ for **OSC** with better constants.

- We next provide an improved analysis of $\mathsf{E}[r(m,d,k)]$ for $\mathbf{OSC}_k$ with better constants and asymptotic limit for large k. The case of large k is important for its application in reverse engineering of biological networks as outlined in Section 1.1. More precisely, we show that $\mathsf{E}[r(m,d,1)]$ is at most

$$\begin{cases} \left(\frac{1}{2}+\log_2 m\right)\cdot\left(2\ln\frac{d}{k}+3.4\right)+1+2\log_2 m, & \text{if } k\le(2\mathbf{e})\cdot d \\ 1+2\log_2 m, & \text{otherwise} \end{cases}$$

- Finally, we discuss lower bounds on competitive ratios for *deterministic algorithms* for $\mathbf{OSC}_k$ and $\mathbf{WOSC}_k$ general k using the approaches in [1]. The lower bounds obtained are $\Omega\left(\max\left\{1,\frac{\log\frac{|\mathcal{S}|}{k}\log\frac{|V|}{k}}{\log\log\frac{|\mathcal{S}|}{k}+\log\log\frac{|V|}{k}}\right\}\right)$ for $\mathbf{OSC}_k$ and $\Omega\left(\frac{\log|\mathcal{S}|\log|V|}{\log\log|\mathcal{S}|+\log\log|V|}\right)$ for $\mathbf{WOSC}_k$ for almost all values of the parameters.

2 A Generic Randomized Winnowing Algorithm

We first describe a generic randomized winnowing algorithm **A-Universal** below in Fig. 1. The winnowing algorithm has two scaling factors: a multiplicative scaling factor $\frac{\mu}{c_S}$ that depends on the particular set S containing i and another additive scaling factor $|\mathcal{S}_i|^{-1}$ that depends on the number of sets that contain i. These scaling factors quantify the appropriate level of "promotion" in the winnowing approach. In the next few sections, we will analyze the above algorithm for the various online set-multicover problems. The following notations will be used uniformly throughout the analysis:

- $\mathcal{J}\subseteq V$ be the set of elements received in a run of the algorithm.

- $\mathcal{T}^*$ be an optimum solution.

2.1 Probabilistic Preliminaries

For the analysis of Algorithm **A-Universal**, we will use the following combinatorial and probabilistic facts and results.

Fact 1 *If* f *is a non-negative integer random function, then* $\mathsf{E}[f]=\sum_{i=1}^{\infty}\Pr[f\ge i]$.

Fact 2 *The function* $f(x)=x\mathbf{e}^{-x}$ *is maximized for* $x=1$.

The subsequent lemmas deal with N independent 0-1 random variables $\tau_1,\dots,\tau_N$ called trials with event$\{\tau_i=1\}$ is the *success* of trial number i and $s=\sum_{i=1}^{N}\tau_i$ is the number of successful trials. Let $x_i=\Pr[\tau_i=1]=\mathsf{E}[\tau_i]$ and $X=\sum_{i=1}^{N}x_i=\mathsf{E}[s]$.

```
// definition //
for (i ∈ V)
    S_i ← {s ∈ S : i ∈ S}

// initialization //
T ← ∅                // T is our collection of selected sets //
for (S ∈ S)
    αp[S] ← 0        // accumulated probability of each set //

// after receiving an element i //
deficit ← k − |S_i ∩ T|        // k is the coverage factor //
if deficit = 0                 // we need deficit more sets for i //
    finish the processing of i
A ← ∅
repeat deficit times
    S ←least cost set from S_i − T − A
    insert S to A
μ ← c_S              // μ is the cost of the last set added to A //
for (S ∈ S_i − T)
    p[S] ← min{ (μ/c_S) (αp[S] + |S_i|^{-1}) , 1 }   // probability for this step //
    αp[S] ← αp[S] + p[S]                 // accumulated probability //
    with probability p[S]
        insert S to T                    // randomized selection //
deficit ← k − |S_i ∩ T|
repeat deficit times                     // greedy selection //
    insert a least cost set from S_i − T to T
```

Figure 1: Algorithm **A-Universal**

Lemma 3 *If* $0 < 2\alpha \leq X + 1$ *than* $\Pr[s = \alpha] > \Pr[s = \alpha - 1]$.

Proof. Our elementary events are 0/1 vectors $\tau = (\tau_1, \ldots, \tau_N)$. Let E_α be the event $\{s = \alpha\}$, *i.e.* the set of elementary events with α 1's. Given $\tau \in E_{\alpha-1}$ we can form an elementary event from E_α by converting some 0 into 1. If we do it with τ_i, the call the result τ^i; observe that $\Pr\left[\tau^i\right] > x_i \Pr[\tau]$. Therefore the sum of probabilities of elementary events formed from τ is at least $\Pr[\tau] \sum_{i:\ \tau_i=0} x_i \geq (X - \alpha + 1)\Pr[\tau] \geq \alpha \Pr[\tau]$.

This shows that the sum of probabilities of the multi-set of elementary events formed from elements of $E_{\alpha-1}$ is larger than $\alpha \Pr[E_{\alpha-1}]$; in turn, every elements in this multi-set belongs to E_α and it is present in this multi-set exactly α times. Thus $\Pr[E_\alpha] \geq \alpha^{-1}\alpha \Pr[E_{\alpha-1}]$. ❑

Lemma 4 *If* $0 \leq \alpha \leq X/2$ *then* $\Pr[s \leq \alpha] < \mathbf{e}^{-X} X^\alpha / \alpha!$.

Proof. The case of $\alpha = 0$ is easy since $\Pr[s \leq 0] = \Pi_{i=1}^n (1 - x_i) < \Pi_{i=1}^n \mathbf{e}^{-x_i} = \mathbf{e}^{-X}$. So, we assume in the remaining that $\alpha > 0$.

We will show how to alter the probabilities so that X remains constant and $\Pr[s \leq \alpha]$ does not decrease. Let $x_0 = x_1 + x_2$, $s' = s - \tau_1 - \tau_2$ and let $q_\alpha = \Pr[s' \leq \alpha]$. We assume that $x_0 \leq 1$. Then

$$\begin{aligned}
\Pr[s \leq \alpha] &= \Pr[\tau_1 = \tau_2 = 0 \ \& \ s' \leq \alpha] + \Pr[\tau_1 + \tau_2 = 1 \ \& \ s' \leq \alpha - 1] \\
&\quad\quad + \Pr[\tau_1 = \tau_2 = 1 \ \& \ s' \leq \alpha - 2] \\
&= (1 - x_1)(1 - x_1)q_\alpha + [(1 - x_1)x_2 + x_1(1 - x_2)]q_{\alpha-1} + x_1 x_2 q_{\alpha-2} \\
&= (1 - x_0 + x_1 x_2)q_\alpha + (x_0 - 2x_1 x_2)q_{\alpha-1} + x_1 x_2 q_{\alpha-2} \\
&= [P = (1 - x_0)q_\alpha + x_0 q_{\alpha-1}] + x_1 x_2(q_\alpha - 2q_{\alpha-1} + q_{\alpha-2}) \\
&= P + x_1 x_2(\Pr[s' = \alpha] - \Pr[s' = \alpha - 1])
\end{aligned}$$

If we keep $x_1 + x_2$ fixed, P is constant and we maximize the latter expression when $x_1 = x_2$ (because $2\alpha \leq (X - x_1 - x_2) + 1$, by Lemma 3, the difference of probabilities in the parenthesis is positive).

This shows that $\Pr[s = \alpha]$ is maximized when all x_i's are equal. We can "pad" the vector of x_i's with zeros, *i.e.* add trials with zero probability of success. This shows that we can overestimate our probability when we go to the limit with $N \to \infty$ and all x_i's equal to X/N. We can now finish the proof by observing the following from standard estimates in probability theory:

$$\lim_{N\to\infty} \frac{N!}{(N-\alpha)!\alpha!}\left(1 - \frac{X}{N}\right)^{N-\alpha}\left(\frac{X}{N}\right)^{\alpha} = \frac{X^\alpha}{\mathbf{e}^X \alpha!}$$

❑

3 An Uniform Analysis of Algorithm A-Universal

In this section, we present an uniform analysis of Algorithm **A-Universal** that applies to all versions of the online set multicover problems, *i.e.*, **OSC**, $\mathbf{OSC}_k$, **WOSC** and $\mathbf{WOSC}_k$. Abusing notations slightly, define $c(\mathcal{S}') = \sum_{S\in\mathcal{S}'} c_S$ for any subcollection of sets $\mathcal{S}' \subseteq \mathcal{S}$. Our bound on the competitive ratio will be influenced by the parameter κ defined as: $\kappa = \min_{i\in\mathcal{J} \ \& \ S\in\mathcal{S}_i\cap\mathcal{T}^*} \left\{ \frac{c(\mathcal{S}_i \cap \mathcal{T}^*)}{c_S} \right\}$.

It is easy to check that $\kappa = \begin{cases} 1 & \text{for } \mathbf{OSC} \text{ and } \mathbf{WOSC} \\ k & \text{for } \mathbf{OSC}_k \\ \geq 1 & \text{for } \mathbf{WOSC}_k \end{cases}$. The main result proved in this section is the following theorem.

Theorem 5 *The expected competitive ratio* $\mathsf{E}[r(m,d,k)]$ *of Algorithm* **A-Universal** *is at most*

$$\max\left\{1 + 5(\log_2(m+1)+1),\ 1 + (1 + \log_2(m+1))\left(2 + \ln\left(\frac{d}{\kappa(\log_2(m+1)+1)}\right)\right)\right\}$$

Corollary 6
(a) *For* **OSC**, **WOSC***and* $\mathbf{WOSC}_k$*, setting* $\kappa = 1$ *we obtain* $\mathsf{E}[r(m,d,k)]$ *to be at most* $\log_2 m \ln d$ *plus lower order terms.*
(b) *For* $\mathbf{OSC}_k$*, setting* $\kappa = k$*, we obtain* $\mathsf{E}[r(m,d,k)]$ *to be at most*

$$\max\left\{6 + 5\log_2(m+1),\ 1 + (1 + \log_2(m+1))\left(2 + \ln\left(\frac{d}{k\log_2(m+1)}\right)\right)\right\}$$

.

In the next few subsections we prove the above theorem.

3.1 The Overall Scheme

We first roughly describe the overall scheme of our analysis. The average cost of a run of **A-Universal** is the sum of average costs that are incurred when elements $i \in \mathcal{J}$ are received. We will account for these costs by dividing these costs into three parts $\text{cost}_1 + \sum_{i \in \mathcal{J}} \text{cost}_2^i + \sum_{i \in \mathcal{J}} \text{cost}_3^i$ where:

cost$_1 \leq c(\mathcal{T}^*)$ upper bounds the *total* cost incurred by the algorithm for selecting sets in $\mathcal{T} \cap \mathcal{T}^*$.

cost$_2^i$ is the cost of selecting sets from $\mathcal{S}_i - \mathcal{T}^*$ in line A13 for each $i \in \mathcal{J}$.

cost$_3^i$ is the cost of selecting sets from $\mathcal{S}_i - \mathcal{T}^*$ in line A16 for each $i \in \mathcal{J}$.

We will use the accounting scheme to count these costs by creating the following three types of accounts:

account$(\mathcal{T}^*)$;	
account(S)	for each set $S \in \mathcal{T}^* - \mathcal{T}$;
account(i)	for each received element $i \in \mathcal{J}$.

cost_1 obviously adds at most 1 to the average competitive ratio; we will charge this cost to *account*$(\mathcal{T}^*)$. The other two kinds of costs, namely $\text{cost}_2^i + \text{cost}_3^i$ for each i, will be distributed to the remaining two accounts. Let $L(m)$ be a function of m satisfying $L(m) \leq 1 + \log_2(m+1)$ and let $D = \frac{d}{\kappa(\log_2(m+1)+1)}$. The distribution of charges to these two accounts will satisfy the following:

- $\sum_{i \in \mathcal{J}}$ *account*$(i) \leq L(m) \cdot c(\mathcal{T}^*)$. This claim in turn will be satisfied by:
  - dividing the optimal cost $c(\mathcal{T}^*)$ into pieces $c_i(\mathcal{T}^*)$ for each $i \in \mathcal{J}$ such that $\sum_{i \in \mathcal{J}} c_i(\mathcal{T}^*) \leq c(\mathcal{T}^*)$; and
  - showing that, for each $i \in \mathcal{J}$, *account*$(i) \leq L(m) \cdot c_i(\mathcal{T}^*)$.
- $\sum_{S \in \mathcal{T}^*}$ *account*$(S) \leq L(m) \cdot \max\{4, \ln D + 1\} \cdot c(\mathcal{T}^*)$.

This will obviously prove an expected competitive ratio of at most the maximum of $1 + 5(\log_2(m+1)+1)$ and $1 + (\log_2(m+1)+1)(2 + \ln D)$, as promised.

We will perform our analysis from the point of view of each received element $i \in \mathcal{J}$. To define and analyze the charges we will define several quantities:

$\mu(i)$	the value of μ calculated in line A8 after receiving i
$\xi(i)$	the sum of $\alpha p[S]$'s over $S \in \mathcal{S}_i - \mathcal{T}^*$ at the time when i is received
$a(i)$	$\lvert \mathcal{T} \cap \mathcal{S}_i - \mathcal{T}^* \rvert$ at the time when i is received
$\Lambda(S)$	$\log_2(m \cdot \alpha p[S] + 1)$ for each $S \in \mathcal{S}$; it changes during the execution of **A-Universal**

Finally, let $\Delta(X)$ denote the amount of change (increase or decrease) of a quantity X when an element i is processed.

3.2 The role of $\Lambda(S)$

Our goal is to ensure that $\sum_{S \in \mathcal{T}^* - \mathcal{T}}$ *account*(S) is bounded by at most $\max\{4, \ln D + 1\}$ times $\sum_{S \in \mathcal{T}^*} c_S \Lambda(S)$. For a $S \in \mathcal{S}_i \cap \mathcal{T}^* - \mathcal{T}$ corresponding to the case when element $i \in \mathcal{J}$ is processed, we will do this by ensuring that $\Delta($*account*$(S))$, the change in *account*(S), is at most a *suitable* multiple of $\Delta(c_S \Lambda(S))$. Roughly, we will partition the sets in $\mathcal{T}^* - \mathcal{T}$ into the so-called "heavy" and "light" sets that we will define later and show that

- for a light set, $\Delta(\mathit{account}(S))$ will be at most $\Delta(c_S\Lambda(S))$, and
- for a heavy set $\Delta(\mathit{account}(S))$ will be at most $\max\{4, \ln D + 1\}\Delta(c_S\Lambda(S))$.

The general approach to prove that $\Delta(\mathit{account}(S))$ is at least some multiple of $\Delta(c_S\Lambda(S))$ will generally involve two steps:

- $\Delta(c_S\Lambda(S)) \geq \min\{c_S, \mu(i)\}$;
- $\Delta(\mathit{account}(S))$ is at most a multiple of $\min\{c_S, \mu(i)\}$.

Of course, such an approach makes sense only if we can prove an upper bound on $\mathsf{E}[\Lambda(S)]$. As a first attempt, the following lemma seems useful.

Lemma 7 $\mathsf{E}[\Lambda(S)] \leq \log_2(m+1)$.

Proof. Because $\Lambda(S)$ is a concave function of $\alpha p[S]$, by Jensen's inequality, $\mathsf{E}[\Lambda(S)] = \mathsf{E}[\log_2(m \cdot \alpha p[S] + 1)] \leq \log_2(m \cdot \mathsf{E}[\alpha p[S]] + 1)$ Thus, it suffices to show that $\mathsf{E}[\alpha p[S]] \leq 1$.

Consider the sequence of consecutive values of $p[S]$, say $\vec{p} = (p_0, p_1, p_2, \ldots)$ with $0 \leq p_0 < p_1 < p_2 < \cdots \leq 1$. The final value of $\alpha p[S]$ can be thought of as being the result of the following experiment: make independent Bernoulli trials with success probabilities $p_0, p_1, \ldots$ until we obtain the first success; if we succeed at trial number j, the outcome is $p_0 + \ldots + p_j$. If we never succeed, the outcome is the sum of all the probabilities. This experiment describes the fact that if we succeed, S becomes an element of $\mathcal{T}$ and we stop growing the accumulated probability of S. Let $f(\vec{p})$ or just f denote the result of the above experiment.

Let $f' = \lceil f \rceil$. By Fact 1, $\mathsf{E}[f] \leq \mathsf{E}[f'] = \sum_{j=1}^{\infty} \Pr[f' \geq j] = \sum_{j=1}^{\infty} \Pr[f' > j-1]$. Now, $\Pr[f' > j-1]$ is the probability of the event that we had no success while performing trials with the sum of probabilities that was at least $j-2$; be Lemma 4 with $\alpha = 0$ this is at most e^{2-j}. As a result, $\mathsf{E}[f]$ is at most the sum of a fixed convergent series. This implies that there exists a finite supremum of values $\mathsf{E}[f(\vec{p})]$, say, F. This supremum can be obtained with a particular value of p_0, say X.

Let $\vec{p'}$ be the sequence $\vec{p}$ without the first term; $\mathsf{E}[f(\vec{p})] = X + (1-X)\mathsf{E}\left[f(\vec{p'})\right]$. Therefore we have $F = X + (1-X)F$, and this implies $F = 1$. ❑

How does $\Lambda(S)$ increase when **A-Universal** handles its element i? A preliminary glance at the algorithm suggests the following. First we calculate μ in line A8, then we calculate $p[S]$ in line A10 to be at least $\frac{\mu(i)}{c_S}\frac{1}{m}(m \cdot \alpha p[S] + 1)$, then we increase $\alpha p[S]$ by $p[S]$, thus we increase $m \cdot \alpha p[S] + 1$ by a factor of at least $1 + \frac{\mu(i)}{c_S}$. Therefore $\log_2(m \cdot \alpha p[S] + 1)$ seems to increase by at least $\log_2(1 + \frac{\mu(i)}{c_S})$.

However, some corrections may need to be made to the upper bound of $\mathit{Ave}\Lambda(S)$ in Lemma 7 to ensure that $\log_2(m \cdot \alpha p[S] + 1)$ increases by at least $\log_2(1 + \frac{\mu(i)}{c_S})$ for the very last time $p[S]$ and consequently $\alpha p[S]$ is updated. The reason for this is that in line A10 of algorithm AUn we calculate

$$p[S] \leftarrow \min\left\{\frac{\mu}{c_S}\left(\alpha p[S] + |\mathcal{S}_i|^{-1}\right), 1\right\}$$

instead of calculating just

$$p[S] \leftarrow \frac{\mu}{c_S}\left(\alpha p[S] + |\mathcal{S}_i|^{-1}\right)$$

and it may be the case that $\frac{\mu}{c_S}\left(\alpha p[S] + |\mathcal{S}_i|^{-1}\right) > 1$. Note that for each S such a problem may occur only once and for the last increment since if we calculate $p[S] = 1$ then S is surely inserted to $\mathcal{T}$.

Thus, the very last increment of $\Lambda(S) = \log_2(m \cdot \alpha p[S] + 1)$ may be smaller than $\log_2(1 + \frac{\mu(i)}{c_S})$ (and, consequently, the very last increment of $c_S\Lambda(S)$ may be smaller than $c_S \log_2(1 + \frac{\mu(i)}{c_S})$). Instead of separately arguing for this case repeatedly at various places, we handle this by extending the upper bound for $\mathsf{E}[\Lambda(S)]$ in Lemma 7 so that we can consider this last increment of $c_S \log_2(m \cdot \alpha p[S] + 1)$ also to be at least $c_S \log_2(1 + \frac{\mu(i)}{c_S})$. Now, let us observe the following:

- First, consider $c_S > \mu(i)$. In this case, $\log_2(1+\frac{\mu(i)}{c_S}) < 1$. Consequently, the additive correction that we need in the upper bound for $\Lambda(S)$ is at most 1.

- Otherwise, $c_S \leq \mu(i)$. The "intended" increase in the value of $c_S\Lambda(S)$, namely $c_S \log_2(1+\frac{\mu(i)}{c_S})$, is an increasing function of c_S. Thus, the worst situation happens for the largest possible value of c_S, in our case $c_S = \mu(i)$. Therefore we have the following situation: $\alpha p[S] + \frac{1}{m} = 1 + x$, $c_S = \mu(i)$, we calculate $p[S] = 1 + x$, and we have to "round it down" to 1; this yields an "accounting deficit" of x in the new calculated value $2 + x - \frac{1}{m}$ of $\alpha p[S]$ when the processing of element i is finished. However, we already tried to select S and the sum of probabilities of this trials is $1 + x - \frac{1}{m}$. Since we did not already select S, by Lemma 4 with $\alpha = 0$ this problem happens with probability below $\mathbf{e}^{-x+\frac{1}{m}}$. Thus the expected size of the accounting deficit is $x \cdot \mathbf{e}^{-x+\frac{1}{m}}$, which is maximum for $x = 1$ by Fact 2. Therefore, the expected additive correction that we need in the upper bound for $\Lambda(S)$ is at most

$$
\begin{aligned}
& \mathsf{E}\left[\frac{\log_2(m\cdot\alpha p[S]+1)}{\log_2(m\cdot(\alpha p[S]-x)+1)}\right] \\
\leq\ & \mathsf{E}\left[\log_2\left(\frac{\alpha p[S]}{\alpha p[S]-x}\right)\right] \\
=\ & \mathsf{E}\left[\log_2\left(1+\frac{x}{\alpha p[S]-x}\right)\right] \\
\leq\ & \mathsf{E}\left[\log_2(1+x)\right] && \text{(since } \alpha p[S]-x \geq 1\text{)} \\
\leq\ & \log_2(1+\mathsf{E}[x]) && \text{(by Jensen's inequality))} \\
\leq\ & \log_2(1+\mathbf{e}^{\frac{1}{m}-1}) < 1
\end{aligned}
$$

In summary, we can alter the definition of $\Lambda(S)$ so that for $S \in \mathcal{S}_i \cap \mathcal{T}^* - \mathcal{T}$

- if $c_S \geq \mu(i)$, $\Delta(\Lambda(S)) \geq \log_2(1 + \frac{\mu}{c_S})$;
- if $c_S \leq \mu(i)$, $\Delta(\Lambda(S)) \geq 1$;
- the expected final value of $\Lambda(S)$ is $L(m) < 1 + \log_2(m + 1)$.

Now we are able to prove the following lemma.

Lemma 8 *If* $S \in \mathcal{S}_i \cap \mathcal{T}^* - \mathcal{T}$ *then* $\Delta(c_S\Lambda(S)) \geq \min\{c_S, \mu(i)\}$.

Proof. If $c_S \leq \mu(i)$, $\Delta(\Lambda(S)) \geq 1$ and thus $\Delta(c_S\Lambda(S)) \geq c_S$. If $c_S \geq \mu(i)$, $\Delta(\Lambda(S)) \geq \log_2(1 + \frac{\mu}{c_S})$; thus it suffices to show that

$$
c_S \log_2\left(1 + \frac{\mu(i)}{c_S}\right) \geq \mu(i) \quad \equiv \quad \log_2\left(1 + \frac{\mu(i)}{c_S}\right) \geq \frac{\mu(i)}{c_S}
$$

which is true because $\log_2(1+x) - x$ is a decreasing function and thus $\log_2(1+x) - x \geq 1$ for $x \leq 1$. ❑

3.3 Definition of Light/Heavy Sets and Charges To Light Sets

When an element i is received, we will make charges to *account*(S) for $S \in \mathcal{S}_i \cap \mathcal{T}^* - \mathcal{T}$. Note that these are accounts of *at least* $\mathtt{deficit} + a(i)$ many sets. We number these sets as $S(1), S(2), \dots$ in nondecreasing order of their costs with. We will define the *last* $a(i) + 1$ sets in this ordering as *heavy* and the rest as *light*.

Consider the sets inserted to $\mathcal{A}$ in lines A5-7, say $A(1), \dots, A(\mathtt{deficit})$. We pessimistically assume that except for its last — and most costly — element, $\mathcal{A}$ is inserted to T in line A16. We charge the cost of that to the accounts of light sets — these sets will not receive any other charges. More specifically, we charge $c_{A(j)}$ to *account*$(S(j))$. Because $c_{A(j)} \leq \min\{c_{S(j)}, \mu(i)\}$, this charge is not larger than $\Delta(c_{S(j)}\Lambda(S(j)))$ by Lemma 8.

3.4 Charges to *account*(i)

The sum of charges to accounts of heavy set and *account*(i) can be estimated as $\mu(i)\xi(i) + 2\mu(i)$, where the part $\mu(i)\xi(i) + \mu(i)$ refers to line A13 and the remaining part $\mu(i)$ refers to the cost of line A16 that is not attributed to the accounts of light sets. *To simplify our calculations, we rescale the costs of sets so* $\mu(i) = 1$ *and thus* $c_S \geq 1$ *for each heavy set* S *and the sum of charges to accounts of heavy set and* account(i) *is simply* $\xi(i) + 2$.

We associate with i a piece $c_i(\mathcal{T}^*)$ of the optimum cost $c(\mathcal{T}^*)$:

$$c_i(\mathcal{T}^*) = \sum_{S \in \mathcal{S}_i \cap \mathcal{T}^*} c_S/|S| \quad \leq \quad \frac{1}{d} c(\mathcal{S}_i \cap \mathcal{T}^*) \quad \leq \quad \frac{\kappa}{d}\mu(i) \quad = \quad \kappa/d.$$

It is then easy to verify that

$$\sum_{i \in \mathcal{J}} c_i(\mathcal{T}^*) \leq \sum_{i \in \mathcal{J}} \frac{1}{d} c(\mathcal{S}_i \cap \mathcal{T}^*) \leq c(\mathcal{T} \cap \mathcal{T}^*) \leq c(\mathcal{T}^*)$$

As explained in the overview of this approach, we will charge *account*(i) in such a way that on average it receives $D^{-1} = L(m)\kappa/d$. In the next subsection, we will show how to define a set of random events $\mathcal{E}(i, b)$ so that the probability of the event $\mathcal{E}(i, b)$ is a function of the form $\mathcal{P}(i, b)$ and, when such an event happens, we charge *account*(i) with some amount $\mathcal{F}(\xi(i), b)$. We show in the next subsection that the event $\mathcal{E}(i, b)$ can be appropriately defined such that the expected sum of charges is *sufficiently small*, *i.e.*, $\sum_b \mathcal{P}(i, b) \cdot \mathcal{F}(\xi(i), b) \leq D^{-1}$.

3.5 Charges to Heavy Sets

Let $\psi = \max\{2, \ln D - 1\}$. Suppose that we charge each heavy set S with an amount of ψ of ξ plus the two additional amounts, for a total of $\max\{4, \ln D + 1\}$. Then, $\Delta(c_S\Lambda(S)) \geq \min\{1, c_S\} \geq 1$ and the maximum charge is within a factor $\max\{4, \ln D + 1\}$ of $\Delta(c_S\Lambda(S))$.

If $\psi(a(i) + 1) \geq \xi(i)$ we have no problem because we can charge $a(i) + 1$ accounts, each with at most ψ.

Otherwise, we have $\psi(a(i) + 1) < \xi(i)$. We charge each heavy set with ψ, thus we need to charge *account*(i) with $\xi(i) - \psi(a(i) + 1)$. To describe this case we introduce the event $\mathcal{E}(i, b)$ mentioned in the previous subsection: $\mathcal{E}(i, b)$ is the event $a(i) \geq b$.

Abusing notations slightly, let us identify $\mathcal{E}(i, b)$ with a zero-one random function denoting this event, *i.e.*, $\mathcal{E}(i, b) = \begin{cases} 1 & \text{if } a(i) \geq b \\ 0 & \text{otherwise} \end{cases}$. Let $\mathtt{charge}(i, \psi, \ell, x)$ be the formula for the charge to *account*(i) assuming we use ψ with $\ell\psi \leq x = \xi(i) \leq (\ell + 1)\psi$. We can estimate $\mathtt{charge}(i, \psi, \ell, x)$ in the following manner:

- If $\mathcal{E}(i, \ell-1)$ happens, then $a(i)+1=\ell$, the total charge to all the heavy sets is $\ell\psi$ and thus we have to charge *account*(i) with $x-\ell\psi$.
- if $\mathcal{E}(i, \ell-2)$ happens then $\mathcal{E}(i, \ell-1)$ also happens, so we charged *account*(i) with $x-\ell\psi$ already, but we need to charge *account*(i) with an additional amount of ψ.
- Continuing in a similar manner, it follows that for each $b \leq \ell-2$, if $\mathcal{E}(i, b)$ happens we charge *account*(i) with an additional amount of ψ.

Thus we get the following estimate:

$$\mathsf{E}\left[\mathsf{charge}(i, \psi, \ell, x)\right] = \Pr\left[\mathcal{E}(i, \ell-1)=1\right] \cdot (x-\ell\psi) + \psi \sum_{j=0}^{\ell-2} \Pr\left[\mathcal{E}(i, j)=1\right].$$

Since $\psi(a(i)+1) < \xi(i)$ and $\psi \geq 2$, $a(i)+1$ is less than $\frac{1}{2}\xi(i)$. Thus, we can use Lemma 4 with $X = x = \xi(i)$ and $a = j$ to obtain $\Pr\left[\mathcal{E}(i,j)=1\right] < \mathbf{e}^{-x}\frac{x^j}{j!}$ for $j = \ell-1, \ell-2, \ldots, 0$. Let $C(\psi, \ell, x)$ be the estimate of of $\mathsf{E}\left[\mathsf{charge}(i, \psi, \ell, x)\right]$ thus obtained:

$$C(\psi, \ell, x) = \mathbf{e}^{-x}\left(\frac{x^{\ell-1}}{(\ell-1)!}(x-\ell\psi) + \psi\sum_{j=0}^{\ell-2}\frac{x^j}{j!}\right).$$

Lemma 9 *If* $\psi \geq 2$, $x \geq 1$ *and* $\ell = \lfloor x/\psi \rfloor \geq 1$ *then* $C(\psi, \ell, x) \leq \mathbf{e}^{-(\psi+1)}$.

Proof. We first consider the case of $\ell = 1$. Because $\mathcal{E}(i, -1)$ is not possible, $\mathsf{charge}(i, \psi, 1, x) = \mathcal{E}(i, 0)(x-\psi)$ and $C(\psi, 1, x) = \mathbf{e}^{-x}(x-\psi)$. Now since $\frac{\partial}{\partial x}C(\psi, 1, x) = \mathbf{e}^{-x}(-x+\psi+1)$, $C(\psi, 1, x)$ is maximized for $x = \psi+1$ with a maximum value of $\mathbf{e}^{-(\psi+1)}$.

For $\ell \geq 2$ the summation part of the formula for $C(\psi, \ell, x)$ is non-trivial; in that case one can calculate that

$$\frac{\partial}{\partial x}C(\psi, \ell, x) = \mathbf{e}^{-x}\frac{x^{\ell-2}}{(\ell-1)!}(-x^2 + \ell(\psi+1)x - (\ell^2-1)\psi).$$

As we see, this derivative is a product of a positive function with a trinomial. This trinomial has the maximum for $x = \ell(\psi+1)/2$, so in our range, $\ell\psi \leq x \leq (\ell+1)\psi$, it is decreasing. For $x = \ell\psi$ the value of the trinomial is $\psi > 0$, and for $x = \ell\psi + 2/\ell$ the value of the trinomial is $2 - \psi - 4\ell^{-2} < 0$. Therefore the maximum must occur in the interval between $\ell\psi$ and $\ell\psi + 2/\ell$ and it will suffice to prove our claim in this range.

For $x = \ell\psi + z$ with $0 < z < 2/\ell$ the inequality we want to prove is equivalent to

$$\mathsf{LHS} = \frac{(\ell\psi+z)^{\ell-1}}{(\ell-1)!}z + \psi\sum_{j=0}^{\ell-2}\frac{(\ell\psi+z)^j}{j!} \leq \mathbf{e}^{(\ell-1)\psi-1+z} = \mathsf{RHS} \tag{1}$$

Suppose that (1) is true for some ψ; then for $\psi' = \psi + \varepsilon$ RHS increases by a factor of $\mathbf{e}^{(\ell-1)\varepsilon}$, while each monomial $\frac{(\ell\psi+z)^j}{j!}$, for $j = 0, 1, \ldots, \ell-1$, increases by a factor of $\left(1 + \frac{\varepsilon}{\psi+\frac{z}{\ell}}\right)^j \leq \left(1+\frac{\varepsilon}{\psi}\right)^{\ell-1} < \mathbf{e}^{(\ell-1)\frac{\varepsilon}{\psi}}$ and thus the entire LHS increases by a factor of at most $\psi\mathbf{e}^{(\ell-1)\frac{\varepsilon}{\psi}} < \mathbf{e}^{(\ell-1)\varepsilon}$. Because LHS increases less that RHS, the inequality for ψ implies that for $\psi + \varepsilon$ and thus for every higher value. For this reason it suffices to prove the inequality for $\psi = 2$ and for $\ell\psi < x < \ell\psi + 2/\ell$ (thus, for $0 < z < 2/\ell$). For $\psi = 2$, our claim is reduces to

$$\mathsf{LHS} = \frac{(2\ell+z)^{\ell-1}}{(\ell-1)!}z + 2\sum_{j=0}^{\ell-2}\frac{(2\ell+z)^j}{j!} \leq \mathbf{e}^{2\ell-3+z} = \mathsf{RHS}$$

For convenience, let $y = 2\ell + z$. Thus, we need to prove

$$\mathsf{LHS} = \frac{y^{\ell-1}}{(\ell-1)!}(y-2\ell) + 2\sum_{j=0}^{\ell-2}\frac{y^j}{j!} \le \mathbf{e}^{y-3} = \mathsf{RHS}$$

subject to $2\ell < y < 2\ell + \frac{2}{\ell}$. Since $\ell \ge 2$, $y < 2\ell + \frac{2}{\ell} < 2(\ell+1)$ and thus $y - 2\ell < 2$. Thus $\mathsf{LHS} < 2\sum_{j=0}^{\ell-1}\frac{y^j}{j!}$, and since, by the well-known series expansion, $\mathbf{e}^y = \sum_{j=0}^{\infty}\frac{y^j}{j!}$ it suffices to show that

$$2\mathbf{e}^3\sum_{j=0}^{\ell-1}\mathsf{T}_j \le \sum_{j=0}^{\infty}\mathsf{T}_j$$

for $\ell \ge 2$, $2\ell < y < 2\ell + \frac{2}{\ell}$ and $\mathsf{T}_j = \frac{y^j}{j!}$. First, we verify by induction that $\mathsf{T}_j \ge \sum_{i=0}^{j-1}\mathsf{T}_i$ for $1 \le j \le \ell$. Note that for $1 \le j \le \ell$, $\mathsf{T}_j/\mathsf{T}_{j-1} = y/j > 2$. For the basis case of $j = 1$, it is therefore obvious. Otherwise, $\mathsf{T}_j > 2\mathsf{T}_{j-1} > \mathsf{T}_{j-1} + \sum_{i=0}^{j-2}\mathsf{T}_i = \sum_{i=0}^{j-1}\mathsf{T}_i$ by inductive hypothesis. Thus, it suffices to show that

$$2\mathbf{e}^3\mathsf{T}_\ell \le \sum_{j=0}^{\infty}\mathsf{T}_j$$

For $\ell + 1 \le j \le 2\ell$, $\mathsf{T}_j/\mathsf{T}_{j-1} = y/j > 1$. Thus, $\sum_{j=0}^{\infty}\mathsf{T}_j \ge \ell\mathsf{T}_\ell$, and thus it suffices to show that $2\mathbf{e}^3\mathsf{T}_\ell \le \ell \cdot \mathsf{T}_\ell$ which holds provided $\ell \ge 2\mathbf{e}^3 \approx 40.17$. Thus, the claim holds for $\ell > 40$.

For $2 \le \ell \le 40$ and $\psi = 2$, we can verify our claim by easy numerical calculation. Notice that we just need to verify $\mathsf{C}(2, \ell, x_0) \le \mathbf{e}^{-3}$ where x_0 is the real root of the quadratic function $f(x) = -x^2 + 3\ell x - 2(\ell^2 - 1)$ that lies in the range $2\ell < x < 2\ell + 2/\ell$. By numerical calculation, one can tabulate the results as shown in Table 1 and verify that $\mathsf{C}(2, \ell, x_0) < 0.049 < \mathbf{e}^{-3}$.

❑

Now, since $\psi = \max\{2, \ln \mathsf{D} - 1\} \ge 2$ we conclude using Lemma 9 that the average charge to *account*(i) is at most $\mathbf{e}^{-\ln \mathsf{D}} = \mathsf{D}^{-1}$.

4 Improved Analysis of Algorithm A-Universal for Unweighted Cases

In this section, we provide improved analysis of the expected competitive ratios of Algorithm **A-Universal** or its minor variation for the unweighted cases of the online set multicover problems. These improvements pertain to providing improved constants in the bound for $\mathsf{E}[r(m, d, k)]$. The following notations will be used in this section:

$\sigma p[i] = \sum_{S \in \mathcal{S}_i} p[S]$;
$\sigma\alpha p[i] = \sum_{S \in \mathcal{S}_i} \alpha p[S]$;
$\widetilde{\mathsf{T}}$ is the set of elements of T for which line A9 was executed.

4.1 Improved Performance Bounds for OSC

Theorem 10 $\mathsf{E}[r(m, d, 1)] \le \begin{cases} \log_2 m \ln d, & \textit{if } m > 15 \\ \left(\frac{1}{2} + \log_2 m\right)(1 + \ln d), & \textit{otherwise} \end{cases}$

ℓ	x_0	$C(2, \ell, x_0)$
40	80.049938	0.000000267802482750
39	78.051215	0.000000367770130466
38	76.052559	0.000000505162841918
37	74.053975	0.000000694037963620
36	72.055470	0.000000953753092710
35	70.057050	0.000001310973313578
34	68.058722	0.000001802442476141
33	66.060495	0.000002478811076980
32	64.062378	0.000003409926108503
31	62.064382	0.000004692144890365
30	60.066519	0.000006458452590756
29	58.068802	0.000008892465898008
28	56.071247	0.000012247826675415
27	54.073872	0.000016875076361489
26	52.076697	0.000023258920058581
25	50.079746	0.000032069930688629
24	48.083046	0.000044236337186173
23	46.086630	0.000061043767052413
22	44.090537	0.000084273925651732
21	42.094810	0.000116397546202183
20	40.099505	0.000160843029165595
19	38.104686	0.000222370693445282
18	36.110434	0.000307594429791974
17	34.116844	0.000425709065373619
16	32.124038	0.000589504628397967
15	30.132169	0.000816780125566277
14	28.141428	0.001132311971151022
13	26.152067	0.001570588251431389
12	24.164414	0.002179590204991318
11	22.178908	0.003025980931596380
10	20.196152	0.004202124182703906
9	18.216991	0.005835328094363729
8	16.242641	0.008099376451161879
7	14.274917	0.011227174827357965
6	12.316625	0.015519482245119539
5	10.372281	0.021333034990024608
4	8.449490	0.028995023101223379
3	6.561553	0.038468799615120751
2	4.732051	0.048129928161242959

Table 1: Verification of $C(2, \ell, x_0) < \mathbf{e}^{-3}$ for $2 \leq \ell \leq 40$.

In the rest of the section, we prove the above theorem via a series of claims. Note that for **OSC** we substitute $\mu = c_S = k = 1$ in the psuedocode of Algorithm **A-Universal** and that $\mathsf{deficit} \in \{0,1\}$.

Lemma 11 *For any* $T \in \mathcal{T}^*$, $\mathsf{E}\left[|\widetilde{T}|\right] \leq \begin{array}{ll} \frac{1}{2} + \log_2 m, & \textit{if } m \leq 5 \\ \log_2 m, & \textit{otherwise} \end{array}$

Proof. Let p_ℓ be the value of $p[T]$ that is computed in line A10 during ℓ^{th} time when a new element of T is received and the algorithm executes line A9 making it an element of $\widetilde{T}$; define $p_0 = 0$ for notational convenience. For $\ell > 0$, since we ensure $p_\ell \geq m^{-1} + \sum_{j=0}^{\ell-1} p_j$, we have $p_\ell \geq 2^{\ell-1}m^{-1}$.

For integer $i > 0$, the event $|\widetilde{T}| \geq i$ happens when our first $i-1$ attempt to select T with positive probability failed. Hence, $\mathsf{Pr}\left[|\widetilde{T}| \geq 1\right] = 1$ and, for any integer $i > 1$,

$$\mathsf{Pr}\left[|\widetilde{T}| \geq i\right] = \prod_{k=0}^{i-1}(1 - p_k) \leq \prod_{k=1}^{i-1}(1 - 2^{k-1}m^{-1}) = \prod_{k=0}^{i-2}(1 - 2^k m^{-1})$$

Thus, by Fact 1,

$$\mathsf{E}\left[|\widetilde{T}|\right] \leq 1 + \sum_{\substack{i>1 \\ p_{i-1}<1}} \mathsf{Pr}\left[|\widetilde{T}| \geq i\right] \leq 1 + \sum_{\substack{i>1 \\ 2^{i-2}<m}} \prod_{k=0}^{i-2}(1-2^k m^{-1}) = 1 + \sum_{\substack{i \\ 2^i<m}} \prod_{k=0}^{i}(1-2^k m^{-1}) = \phi(m)$$

We can immediately observe that (1) $\phi(m)$ increases with m and (2) $\phi(2m) = 1 + (1-(2m)^{-1})\phi(m)$. Suppose that $\phi(m_0) \leq \log_2 x m_0$ for some $x < 1$; then $k \geq 0$ and $x2^k m_0 \leq m \leq 2^k m_0$ implies

$$\phi(m) < \phi(2^k m_0) \leq k + \phi(m_0) < k + \log_2 x m_0 \leq \log_2 m.$$

We can calculate the values of $\phi(m)$ for $m = 2, 7, \ldots, 18$ and observe that (a) $\phi(m) < 0.5 + \log_2 m$ for all of them, (b) $\phi(m) < \log_2 m$ for $m \geq 6$, (c) $\phi(m) < \log_2 m - 1$ for $m \geq 10$. ❑

Obviously $\mathsf{E}[|\mathcal{T}|]$ is equal to the sum of probabilities used in line A12 plus the number of times we execute line A16. Let $\xi(i)$ be the value of $\sigma\alpha p[i]$ *at the time* the algorithm receives element i as the input. If the test of line A2 is false, the sum of probabilities used in line A10 is $\xi(i) + 1$, while by Lemma 4 with $\alpha = 0$ line A16 is executed with probability at most $\frac{1}{\mathbf{e}} < 0.37$, so the contribution of i to the expected cost is smaller than $\xi(i) + 1.37$.

Lemma 12 *For* $T \in \mathcal{T}^*$, *if* $|\widetilde{T}| > 0$ *then* $\mathsf{E}\left[\sum_{i\in\widetilde{T}} \xi(i)\right] < \mathsf{E}\left[|\widetilde{T}|\right]\left(\ln|T| - \ln \mathsf{E}\left[|\widetilde{T}|\right]\right)$.

Proof. Before the condition in line A2 is evaluated for element i the algorithm performs independent random selections of sets from $\mathcal{S}_i$ with the sum of probabilities of success equal to $\xi(i)$. By Lemma 4 with $\alpha = 0$ the probability that all these selections fail, and thus the test in line A2 is false, is $\mathsf{Pr}\left[i \in \widetilde{T}\right] < \mathbf{e}^{-\xi(i)}$. Let Γ be a parameter to be established later, and let $\zeta(i) = \max\{0, \xi(i) - \ln|T| + \Gamma\}$. Clearly,

$$\mathsf{E}\left[\sum_{i\in\widetilde{T}} \xi(i)\right] \leq \mathsf{E}\left[|\widetilde{T}|\right](\ln|T| - \Gamma) + \sum_{i\in T} \mathsf{Pr}\left[i \in \widetilde{T}\right]\zeta(i)$$

Let $T' = \{i \in T : \ \zeta(i) > 0\}$. Then

$$\sum_{i\in T} \mathsf{Pr}\left[i \in \widetilde{T}\right]\zeta(i) \leq \sum_{i\in T'} \mathbf{e}^{-\zeta(i)-\ln|T|+\Gamma}\zeta(i) = |T|^{-1}\mathbf{e}^{\Gamma}\sum_{i\in T'} \mathbf{e}^{-\zeta(i)}\zeta(i) < \mathbf{e}^{\Gamma-1}.$$

where the last inequality follows from Fact 2 and $T' \subseteq T$. Thus,

$$E\left[\sum_{i\in\widetilde{T}} \xi(i)\right] \le E\left[|\widetilde{T}|\right]\left(\ln|T| - \Gamma + \frac{\mathbf{e}^{\Gamma-1}}{E\left[|\widetilde{T}|\right]}\right)$$

We can use $\Gamma = 1 + \ln E\left[|\widetilde{T}|\right]$ to get the desired estimate. ❑

Now, we are ready to finish the proof of the claim on $E[r(m,d,1)]$ in the theorem.

$$\begin{aligned} E[r(m,d,1)] = \frac{E[|\mathcal{T}|]}{|\mathcal{T}^*|} &< \frac{\sum_{T\in\mathcal{T}^*} E[\sum_{i\in\widetilde{T}} \xi(i)+1.37]}{|\mathcal{T}^*|} \\ &< \frac{\sum_{T\in\mathcal{T}^*} E[|\widetilde{T}|]\left(\ln|T| - \ln E[|\widetilde{T}|]+1.37\right)}{|\mathcal{T}^*|} \qquad \text{(by Lemma 12)} \\ &= E\left[|\widetilde{T}|\right]\left(\ln|T| - \ln E\left[|\widetilde{T}|\right] + 1.37\right) \end{aligned}$$

The last quantity is an increasing function of $E\left[|\widetilde{T}|\right]$, so we can replace it with its overestimate. For every $m \ge 2$ we can use estimate $E\left[|\widetilde{T}|\right] \le 0.5 + \log_2 m$ and the fact that $\ln(0.5 + \log_2 2) > 0.37$. For $m \ge 16$ we can use estimate $E\left[|\widetilde{T}|\right] \le \log_2 m$ and the fact that $\ln\log_2 16 > 1.37$.

4.2 Improved Performance Bounds for $\mathbf{OSC}_k$

Note that for $\mathbf{OSC}_k$ we substitute $\mu = c_S = 1$ in the psuedocode of Algorithm **A-Universal** and that $\mathsf{deficit} \in \{0,1,2,\ldots,k\}$. For improved analysis, we change Algorithm **A-Universal***slightly*, namely, line A10 (with $\mu = c_S = 1$)

A10 $\quad p[S] \leftarrow \min\left\{\left(\alpha p[S] + |\mathcal{S}_i|^{-1}\right), 1\right\}$ // probability for this step //

is changed to

A10' $\quad p[S] \leftarrow \min\left\{\left(\alpha p[S] + \mathsf{deficit}\cdot|\mathcal{S}_i|^{-1}\right), 1\right\}$ // probability for this step //

Theorem 13 *With the above modification of Algorithm* **A-Universal**,

$$E[r(m,d,k)] \le \begin{cases} \left(\frac{1}{2} + \log_2 m\right)\cdot\left(2\ln\frac{d}{k} + 3.4\right) + 1 + 2\log_2 m & \textit{if } k \le (2\mathbf{e})\cdot d \\ 1 + 2\log_2 m & \textit{otherwise} \end{cases}$$

We now proceed with the proof of the above theorem. As before, $\mathcal{T}^*$ is an optimal solution and for $T \in \mathcal{T}^*$ we define $\widetilde{T}$ as the set of elements of T for which line S3 was executed. Since Lemma 11 is still true with the same proof, we have $E\left[\widetilde{T}\right] \le \log_2 m + \frac{1}{2}$ for all m.

We will distribute the average cost of the obtained solution as follows. Each element of $\widetilde{T}$ gives a charge to T and a charge to its elements. If the algorithm have received the set of element $X \subseteq U$, then clearly $|\mathcal{T}^*| \ge \frac{|X|\cdot k}{d}$; our goal is to give charges to the elements so that their expected sum equals $xk/d \le |\mathcal{T}^*|$.

We will again perform an analysis of the average cost of receiving an element i for which the test in line A2 is false. We define or redefine the following notations:

$\sigma\alpha p[i] = \sum_{S\in\mathcal{S}_i-\mathcal{T}^*} \alpha p[S]$;
$\xi(i)$ is the value of $\sigma\alpha p[i]$ when line A1 is executed for i;
$\beta(i) = |(\mathcal{S}_i \cap \mathcal{T}^*) - (\mathcal{S}_i \cap \mathcal{T})|$;
$\psi(i) = |(\mathcal{S}_i \cap \mathcal{T}) - (\mathcal{S}_i \cap \mathcal{T}^*)|$;

The value of $\mathsf{deficit}$ in line A1 is at most $\beta(i) - \psi(i)$. Element i will belong to some $\widetilde{T}$ only if $\psi(i) < \beta(i)$. We will view $\xi(i)$ and $\beta(i)$ as fixed parameters of the event when i is received. The quantity $\psi(i)$ is the number of successes in independent trials with success probabilities that add to $\xi(i)$. Let $p(i) = \Pr[\psi(i) < \beta(i)]$.

We charge element i with a value of $\pi_e(i) = \frac{k}{dp(i)}$. The intuition is that, because we make this charge with probability $p(i)$, on an average it equals $p(i)\pi_e(i) = k/d$ and the sum of these charges therefore cannot be larger that $|\mathcal{T}^*|$. We then distribute the remaining cost equally among $\psi(i) < \beta(i)$ many elements of $(\mathcal{S}_i \cap \mathcal{T}^*) - (\mathcal{S}_i \cap \mathcal{T})$.

Clearly, each of the value of $\mathsf{deficit}$ computed in line A1 and computed in line A14 cannot exceed $\beta(i)$. The term $\mathsf{deficit} \cdot |\mathcal{S}_i|^{-1}$ in line A10' adds at most $\mathsf{deficit}$ to the sum of probabilities computed in line A10', thus the cost attributable to this term, as well as the cost due to line A16 add to at most 2 per $T \in \mathcal{T}^*$. It remains to estimate the cost due to the terms $\alpha p[S]$. We decrease this cost by the charge made to i, so each set $T \in \mathcal{T}^*$ such that $i \in \widetilde{T}$ receives a charge of at most $\pi_s(i) = \max\left\{0, \frac{\xi(i)-\pi_e(i)}{\beta(i)}\right\} = \max\left\{0, \frac{\xi(i)-\frac{k}{dp(i)}}{\beta(i)}\right\}$.

The expected number of sets selected by us is therefore at most

$$\begin{array}{rl} & \sum_{T\in\mathcal{T}^*}\sum_{i\in\widetilde{T}}(\pi_s(i)+2) + \sum_{i\in X} p(i)\pi_e(i) \\ \leq & |\mathcal{T}^*| \cdot \sum_{i\in\widetilde{T}} \pi_s(i) + 2\cdot|\widetilde{T}|\cdot|\mathcal{T}^*| + \frac{|X|\cdot k}{d} \\ \leq & \left(\left(\frac{1}{2}+\log_2 m\right)\pi_s(i) + 2\log_2 m + 1\right)\cdot|\mathcal{T}^*| \end{array}$$

which means we need to estimate the quantity $\pi_s(i)$. For this, we first need to calculate a bound for $p(i)$. Remember that $\psi(i)$ is the number of successes of a set of independent trials with success probabilities that add up to $\xi(i)$. The standard Chernoff bound theorem [5, 12] states that if we have a set of independent trials with the sum of success probabilities μ, the probability that the number of successes is below $(1-\delta)\mu$ is below $\mathbf{e}^{-\delta^2\mu/2}$. In our case, $\mu = \xi(i)$ and $(1-\delta)\mu$ is $\beta(i)$. We introduce the following notations for simplicity: $\beta = \beta(i)$, $\phi = \xi(i)/\beta$ and $\kappa = d/k$. Now $\mu = \phi\beta$ and $\delta = (\phi-1)/\phi$; thus via Chernoff bound we have $p(i) < \mathbf{e}^{-\frac{(\phi-1)^2}{2\phi^2}\phi\beta} = \mathbf{e}^{-\frac{(\phi-1)^2}{2\phi}\beta}$. Hence $\pi_s(i) < \max\left\{0, \phi - \frac{1}{\kappa\beta}\mathbf{e}^{\frac{(\phi-1)^2}{2\phi}\beta}\right\} < \max\left\{0, \phi - \frac{1}{\kappa\beta}\mathbf{e}^{(\frac{\phi}{2}-1)\beta}\right\}$. By using simple calculus and the fact that $\beta \geq 1$, it can be shown that the maximum value of the function $f(\phi) = \phi - \frac{1}{\kappa\beta}\mathbf{e}^{(\frac{\phi}{2}-1)\beta}$ is at most $2\ln\kappa + 2\ln(2\mathbf{e}) < 2\ln\kappa + 3.4$. This shows that $\pi_s(i) < \begin{cases} 2\ln\kappa + 3.4 & \text{if } k < (2\mathbf{e})\cdot d \\ 0 & \text{otherwise} \end{cases}$.

4.3 Lower Bounds on Competitive Ratios for $\mathbf{OSC}_k$ and $\mathbf{WOSC}_k$

Lemma 14 *For any* k*, there exists an instance of* $\mathbf{OSC}_k$ *and* $\mathbf{WOSC}_k$ *for almost all values of* $|V|$ *and* $|\mathcal{S}|$ *such that any deterministic algorithm must have a competitive ratio of* $\Omega\left(\max\left\{1, \frac{\log\frac{|\mathcal{S}|}{k}\log\frac{|V|}{k}}{\log\log\frac{|\mathcal{S}|}{k}+\log\log\frac{|V|}{k}}\right\}\right)$ *for* $\mathbf{OSC}_k$ *and* $\Omega\left(\frac{\log|\mathcal{S}|\log|V|}{\log\log|\mathcal{S}|+\log\log|V|}\right)$ *for* $\mathbf{WOSC}_k$.

Proof. First we prove the result for $\mathbf{OSC}_k$. It suffices to show the result when k is a sufficiently large constant and at most a large constant fraction of $\max\{|V|, |\mathcal{S}|\}$, *e.g.*, $k \leq \frac{1}{16}\max\{|V|, |\mathcal{S}|\}$. Alon *et al.* [1] provided an instance of **OSC** with m' sets and n' elements with such that the optimal (offline) cover contains just one set but any online cover must use $\Omega\left(\frac{\log m' \log n'}{\log\log m' + \log\log n'}\right)$ sets. Consider a given k. We will use one additional element x and k additional sets such that x appears in all these sets. To make these k sets mutually different, we will use an additional $O(\log k)$ elements (which we will never present) and add a distinct subset of these additional elements to each set. We will also have k copies of the instances of Alon *et al.* [1] with elements renamed to make each copy distinct from the rest; each element of each copy is also added to exactly $k-1$ of the k additional sets we mentioned at first. The total number of elements is $|V| = kn' + 1 + \log k \leq (k+1)n'$, and the total number of sets is $|\mathcal{S}| = (k+1)m'$. We first present the element x to force the adversary to select the k additional sets; these sets also cover any element in the k copies of Alon *et al.* [1] exactly $k-1$ times. After this, we present the elements in the k copies of Alon *et al.* [1] following their scheme, presenting elements in one copy completely before presenting elements in the next copy. Now the optimal uses at most $2k$ sets, whereas by a reasoning similar to that in Alon *et al.* [1] the online algorithm must use $\Omega\left(k \cdot \frac{\log m' \log n'}{\log\log m' + \log\log n'}\right)$ sets.

To prove the result for $\mathbf{WOSC}_k$, we again use one additional element x plus $O(\log k)$ additional elements (that we will never present) to create k additional sets such that x appears in all these sets. We set the cost of this set to be arbitrarily close to zero, say ε. This time we just use one copy of the instance of Alon *et al.* [1] with each set of cost 1 and, as before, each element of this copy is also added to exactly $k-1$ of the k additional sets we mentioned at first. We again first present the element x to force the adversary to select the k additional sets; these sets also cover any element in the copy of Alon *et al.* [1] exactly $k-1$ times. After this, we present the elements in the copy of Alon *et al.* [1] following their scheme. Since ε can be set arbitrarily close to zero, our result follows. ❑

Acknowledgments. The second author wishes to thank the organizers of the Online Algorithms 2004 Workshop (OLA-2004) in Denmark for invitation which provided motivations to look at these cover problems.

References

[1] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. *The online set cover problem*, 35th annual ACM Symposium on the Theory of Computing, pp. 100-105, 2003.

[2] M. Andrec, B.N. Kholodenko, R.M. Levy, and E.D. Sontag. *Inference of signaling and gene regulatory networks by steady-state perturbation experiments: Structure and accuracy*, J. Theoretical Biology, in press.

[3] B. Awerbuch, Y. Azar, A. Fiat and T. Leighton. *Making Commitments in the Face of Uncertainty: How to Pick a Winner Almost Every Time*, 28th annual ACM Symposium on the Theory of Computing, pp. 519-530, 1996.

[4] P. Berman, B. DasGupta and E. Sontag. *Randomized Approximation Algorithms for Set Multicover Problems with Applications to Reverse Engineering of Protein and Gene Networks*, to appear in the special issue of Discrete Applied Mathematics on computational biology (the conference version in 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, LNCS 3122, K. Jansen, S. Khanna, J. D. P. Rolim and D. Ron (editors), Springer Verlag, pp. 39-50, August 2004).

[5] H. Chernoff. *A measure of asymptotic efficiency of tests of a hypothesis based on the sum of observations*, Annals of Mathematical Statistics, 23: 493–509, 1952.

[6] E.J. Crampin, S. Schnell, and P.E. McSharry. *Mathematical and computational techniques to deduce complex biochemical reaction mechanisms*, Progress in Biophysics & Molecular Biology, 86, pp. 77-112, 2004.

[7] U. Feige. *A threshold for approximating set cover*, Journal of the ACM, Vol. 45, 1998, pp. 634-652.

[8] D. S. Johnson. *Approximation Algorithms for Combinatorial Problems*, Journal of Computer and Systems Sciences, Vol. 9, 1974, pp. 256-278.

[9] B. N. Kholodenko, A. Kiyatkin, F. Bruggeman, E.D. Sontag, H. Westerhoff, and J. Hoek. *Untangling the wires: a novel strategy to trace functional interactions in signaling and gene networks*, Proceedings of the National Academy of Sciences USA 99, pp. 12841-12846, 2002.

[10] B. N. Kholodenko and E.D. Sontag. *Determination of functional network structure from local parameter dependence data*, arXiv physics/0205003, May 2002.

[11] N. Littlestone. *Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm*, Machine Learning, 2, pp. 285-318, 1988.

[12] R. Motwani and P. Raghavan. *Randomized Algorithms*, Cambridge University Press, New York, NY, 1995.

[13] R. Raz and S. Safra. *A sub-constant error-probability low-degree test and sub-constant error-probability PCP characterization of NP* , proceedings of the 29th Annual ACM Symposium on Theory of Computing, pp. 475-484, 1997.

[14] E.D. Sontag, A. Kiyatkin, and B.N. Kholodenko. *Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data*, Bioinformatics 20, pp. 1877-1886, 2004.

[15] J. Stark, R. Callard and M. Hubank. *From the top down: towards a predictive biology of signaling networks*, Trends Biotechnol. 21, pp. 290-293, 2003.

[16] V. Vazirani. *Approximation Algorithms*, Springer-Verlag, July 2001.