Approximating the Online Set Multicover Problems Via Randomized Winnowing

Piotr Berman[1] and Bhaskar DasGupta[2]

[1] Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802.
berman@cse.psu.edu.
Supported by NSF grant CCR-O208821.
[2] Computer Science Department, University of Illinois at Chicago, Chicago, IL 60607.
dasgupta@cs.uic.edu.
Supported in part by NSF grants CCR-0206795, CCR-0208749 and a CAREER grant IIS-0346973.

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[3]. 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

[3] 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.

Cover). The unweighted versions of these problems, when the cost any set is one, will be denoted by $\mathbf{OSC}_{\mathtt{k}}$ or **OSC**.

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:

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

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

1.1 Motivations and Applications

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

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 $\mathbf{WOSC}_{\mathtt{k}}$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]).

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 one such setup, 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:

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

where $\mathbf{p} = (p_1, \ldots, p_m)$ 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 $\mathbf{p}$, which represents "wild type" (that is, normal) conditions, and a corresponding steady state $\bar{x}$ of $\mathbf{x}$, 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. 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. Consequently, an "online" experimental protocol to achieve the above goal, that gives rise to the problems $\mathbf{WOSC}_k$ and $\mathbf{OSC}_k$ is as follows:

- Change one parameter, say p_k.
- Measure the resulting steady state vector $\mathbf{x} = \xi(p)$. Experimentally, this may for instance mean that the concentration of a certain chemical represented by p_k is kept are a slightly altered level, compared to the default value $\bar{p}_k$; then, the system is allowed to relax to steady state, after which the complete state $\mathbf{x}$ 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.
- For each of the possible m experiments, in which a given p_j is perturbed, we may 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 jth canonical basis vector).

From these data, via some linear-algebraic reductions and depending on whether each experiment has the same or different cost, one can arrive at the problems $\mathbf{WOSC}_k$ and $\mathbf{OSC}_k$ with "large" k, *e.g.*, when $k \approx |V|$.

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≠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 $e^2 \approx 7.39$, 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[4] 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 corrolary of our analysis, we observe the following.
  - For **OSC**, **WOSC** and $\mathbf{WOSC}_k$ our randomized algorithm has $E[r(m, d, k)]$ 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 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

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

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 $E[r(m,d,k)]$ 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 $E[r(m,d,1)]$ for **OSC** with better constants.
- We next provide an improved analysis of $E[r(m,d,k)]$ 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 $E[r(m,d,1)]$ is at most

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

- 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.

All proofs omitted due to space limitations will appear in the full version of the paper.

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 $E[f]=\sum_{i=1}^{\infty}\Pr[f\ge i]$.*

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

// definition //
D1 **for** ($i \in V$)
D2 $\quad \mathcal{S}_i \leftarrow \{s \in \mathcal{S} : i \in S\}$

// initialization //
I1 $\mathcal{T} \leftarrow \emptyset$ // $\mathcal{T}$ is our collection of selected sets //
I2 **for** ($S \in \mathcal{S}$)
I3 $\quad \alpha p[S] \leftarrow 0$ // accumulated probability of each set //

// after receiving an element i //
A1 $\texttt{deficit} \leftarrow k - |\mathcal{S}_i \cap \mathcal{T}|$ // k is the coverage factor //
A2 **if** $\texttt{deficit} = 0$ // we need deficit more sets for i //
A3 $\quad$ finish the processing of i
A4 $\mathcal{A} \leftarrow \emptyset$
A5 **repeat** deficit **times**
A6 $\quad S \leftarrow$least cost set from $\mathcal{S}_i - \mathcal{T} - \mathcal{A}$
A7 $\quad$ insert S to $\mathcal{A}$
A8 $\mu \leftarrow c_S$ // μ is the cost of the last set added to $\mathcal{A}$ //
A9 **for** ($S \in \mathcal{S}_i - \mathcal{T}$)
A10 $\quad p[S] \leftarrow \min\left\{\frac{\mu}{c_S}\left(\alpha p[S] + |\mathcal{S}_i|^{-1}\right), 1\right\}$ // probability for this step //
A11 $\quad \alpha p[S] \leftarrow \alpha p[S] + p[S]$ // accumulated probability //
A12 $\quad$ **with** probability $p[S]$
A13 $\quad\quad$ insert S to $\mathcal{T}$ // randomized selection //
A14 $\texttt{deficit} \leftarrow k - |\mathcal{S}_i \cap \mathcal{T}|$
A15 **repeat** deficit **times** // greedy selection //
A16 $\quad$ insert a least cost set from $\mathcal{S}_i - \mathcal{T}$ to $\mathcal{T}$

Fig. 1. Algorithm **A-Universal**

The subsequent lemmas deal with N independent 0-1 random functions $\tau_1, \ldots, \tau_N$ 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 $x_i = \Pr[\tau_i = 1] = E[\tau_i]$ and $X = \sum_{i=1}^{N} x_i = E[s]$.

Lemma 3 *If* $0 \le \alpha \le X/2$ *then* $\Pr[s \le \alpha] < \mathbf{e}^{-X} X^{\alpha}/\alpha!$.

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**, **OSC**$_k$, **WOSC** and **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 1. *The expected competitive ratio* $E[r(m, d, k)]$ *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 4
(a) *For* **OSC**, **WOSC***and* $\mathbf{WOSC}_k$, *setting* $\kappa = 1$ *we obtain* $E[r(m, d, k)]$ *to be at most* $\log_2 m \ln d$ *plus lower order terms.*
(b) *For* $\mathbf{OSC}_k$, *setting* $\kappa = k$, *we obtain* $E[r(m, d, k)]$ *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:

– $\sum_{i\in\mathcal{J}} \textit{account}(i) \leq L(m) \cdot c(\mathcal{T}^*)$. 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 $\sum_{i\in\mathcal{J}} c_i(\mathcal{T}^*) \le c(\mathcal{T}^*)$; and
  - showing that, for each $i \in \mathcal{J}$, $account(i) \le L(m) \cdot c_i(\mathcal{T}^*)$.
- $\sum_{S\in\mathcal{T}^*} account(S) \le L(m) \cdot \max\{4, \ln D + 1\} \cdot c(\mathcal{T}^*)$.

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

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

- $\Delta(c_S\Lambda(S)) \ge \min\{c_S, \mu(i)\}$;
- $\Delta(account(S))$ 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 $E[\Lambda(S)]$. As a first attempt, the following lemma seems useful.

Lemma 5 $E[\Lambda(S)] \le \log_2(m+1)$.

How does $\Lambda(S)$ increase when **A-Universal** handles its element i? A preliminary glance at tha 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 $Ave\Lambda(S)$ in Lemma 5 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 $c_S \log_2(1 + \frac{\mu(i)}{c_S})$). Intead of separately arguing for this case repeatedly at various places, we handle this by extending the upper bound for $E[\Lambda(S)]$ in Lemma 5 so that we can consider this last increment of $c_S \log_2(m \cdot \alpha p[S] + 1)$ also to be at least $c_S \log_2(1 + \frac{\mu(i)}{c_S})$. We omit the details here, but to summarize, 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 6 *If* $S \in \mathcal{S}_i \cap \mathcal{T}^* - \mathcal{T}$ *then* $\Delta(c_S\Lambda(S)) \geq \min\{c_S, \mu(i)\}$.

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 $S(1), S(2), \ldots$ 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 $A(1), \ldots, A(\mathtt{deficit})$. 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 6.

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 $D^{-1} = L(m)\kappa/d$. We will define a random events $E(i, a)$ so that the probability of event $E(i, a)$ is a function of the form $p(\xi(i), a)$ and when such an event happens, we charge *account*(i) with some $f(\xi(i), a)$. We will show in the next subsection that the event $E(i, a)$ can be appropriately defined such that the expected sum of charges is *sufficiently small*, *i.e.*, that $\sum_a p(X, a) f(X, a) < D^{-1}$.

3.5 Charges to Heavy Sets

Let $\psi = \max\{1, \ln D - 1\}$. Suppose that we charge each heavy set S with an amount of ψ of ξ plus the two additional amounts, for a total of $\max\{3, \ln D + 1\}$. Then, $\Delta(c_S \Lambda(S)) \geq \min\{1, c_S\} \geq 1$ and the maximum charge is within a factor $\max\{3, \ln D + 1\}$ of $\Delta(c_S \Lambda(S))$.

If $\psi(a(i)+1) \geq \xi(i)$ we have no problem because we charge $a(i)+1$ accounts, each with at most ψ. Otherwise we need to charge *account*(i) with $\xi(i) - \psi(a(i)+1)$. We describe this case using the following events: $E(i, a)$ means that $a(i) \leq a$.

Let us identify $E(i, a)$ with a zero-one random funtion, and $\mathtt{charge}(i, \psi, \ell, x)$ is the formula for the charge to *account*(i) assuming we use ψ, $\ell\psi \leq X \leq (\ell+1)\psi$ and $\xi(i) = x$. If $E(i, \ell - 1)$ happens, we have to charge *account*(i) with $x - \ell\psi$; if $E(i, \ell - 2)$ happens than $E(i, a - 1)$ also happens, so we charged $x - \ell\psi$, but we need to charge *account*(i) with another ψ. One can see that for each $a \leq \ell - 2$, if $E(i, a)$ happens we charge *account*(i) with ψ. One can see that

$$\mathtt{charge}(i, \psi, \ell, x) = E(i, \ell - 1)(\xi(i) - \ell\psi) + \psi \sum_{j=0}^{\ell-2} E(i, \psi, j).$$

Let $C(\psi, \ell, x)$ be the estimate of $E[\mathtt{charge}(i, \psi, \ell, x)]$ that uses Lemma 3:

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

Lemma 7 *If* $\psi \geq 2$, $x \geq 1$ *and* $\ell = \lfloor x/\psi \rfloor$ *then* $C(\psi, \ell, x) \leq e^{-(\psi+1)}$.

As a result of the above lemma, setting $\psi = \max\{2, \ln D - 1\}$ we conclude that the average charge to *account*(i) is at most 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 $\mathsf{E}[r(m, d, k)]$.

4.1 Improved Performance Bounds for OSC

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

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 $\mathtt{deficit} \in \{0, 1, 2, \ldots, k\}$. For improved analysis, we change Algorithm **A-Universal***slightly*, namely, line A10 (with $\mu = c_S = 1$)

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

is changed to

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

Theorem 3. *With the above modification of Algorithm* **A-Universal**,
$\mathsf{E}[r(m, d, k)] \leq \begin{cases} \left(\frac{1}{2} + \log_2 m\right) \cdot \left(2 \ln \frac{d}{k} + 3.4\right) + 1 + 2 \log_2 m & \text{if } k \leq (2\mathbf{e}) \cdot d \\ 1 + 2 \log_2 m & \text{otherwise} \end{cases}$

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

Lemma 1. *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* $\mathbf{WOSC}_k$.

Acknowledgments. The second author wishes to thank the organizers of the Online Algorithms 2004 Workshop (OLA-2004) in Denmark for invitation which provided motivatations 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-degre 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.