Budgeted Maximum Coverage with Overlapping Costs: Monitoring the Emerging Infections Network [*]

Donald E. Curtis[†] Sriram V. Pemmaraju[†] Philip Polgreen[‡]

Abstract

The Emerging Infections Network (EIN) (`http://ein.idsociety.org/`) is a CDC supported "sentinel" network of over 1400 members (currently), designed to connect clinical infectious disease specialists and public health officials. Members primarily communicate through an EIN managed listserv and discuss disease outbreaks, treatment protocols, effectiveness of vaccinations and other disease-control and prevention mechanisms, etc. Recently, researchers at Google and Yahoo! Research have used search engine query logs to tap into the online "wisdom of crowds" and produce disease outbreak trends for flu. Following this work, there is now interest in trying to monitor EIN discussions more carefully to disseminate timely and accurate information on clinical events of possible interest to health officials.

We model the problem of monitoring a listserv, such as the EIN, as a type of budgeted maximum coverage problem that we call *Budgeted Maximization with Overlapping Costs (BMOC).* Even though BMOC seems superficially similar to the budgeted maximum coverage problem considered by Khuller et al. (*Inf. Process. Lett.*, 1999), our problem is fundamentally different from an algorithmic point of view, due to its cost structure. We observe that the greedy algorithm that provides a constant-factor approximation to the budgeted maximum coverage problem can be arbitrarily bad for BMOC. We also present a reduction to BMOC from the *k-densest subgraph* problem that provides evidence indicating that obtaining a constant-factor approximation for our problem might be quite challenging. Nevertheless, experimental runs of the greedy algorithm on the EIN data show that greedy performs remarkably well relative to OPT. We identify a feature of our EIN data, that we call the *overlap condition*, and show that the greedy algorithm does indeed yield a constant-factor approximation guarantee if the overlap condition is satisfied. Using an implementation of the greedy algorithm for BMOC on the EIN data, we identify small sets of "bellwether" users who are good predictors of important discussions. We provide evidence to show that tracking just these users reduces the cost of monitoring the EIN significantly without causing any important discussions to be missed.

1 Introduction

The Emerging Infections Network (EIN) (`http://ein.idsociety.org/`) is a "sentinel" network of clinical infectious disease specialists, primarily from the United States, created in 1995 by the Infectious Diseases Society of America with a Cooperative Agreement Program award from the Centers for Disease Control (CDC). The goal of the EIN is to assist the CDC and other public health authorities with surveillance of emerging infectious diseases and related phenomena (new treatment protocols, possible side effects of new vaccines, etc). To achieve its goal, the EIN maintains a private listserv open to infectious disease specialists, CDC investigators, and public health officials. There are currently over 1400 subscribers who receive roughly 3 emails per day. Since its inception, the EIN listserv has served over 2800 discussions on the identification of new infectious diseases, treatments, and policy implications.

There are a few features that distinguish the EIN listserv from other online mailing lists. Each submission (*post*) to the EIN listserv is sent to the EIN coordinator, a person responsible for managing the mailing list. The EIN coordinator is responsible for screening and filtering each post by fixing grammatical errors, providing links to citations, and removing any identifying patient information. Each post received by the EIN coordinator is either the start of a new *thread*, if that post is about a new topic, or a response to a previous post in an ongoing thread. Posts are collected throughout the day and bundled into a *mailing* which is broadcast to all subscribers the following morning.

Recent work at Google [9] (`http://www.google.org/flutrends/`) and Yahoo! Research [15] has focused on using search engine query terms as a means of tracking the spread of influenza. Last spring as news of swine flu spread, numerous projects were initiated that used Twitter posts to track and observe the spread of the infection

[*] Support for this research was provided by a National Institutes of Health Young Investigator Award (Polgreen), and by the National Institutes of Health grant NIAID-R21-AI081164 (all three authors).

[†] Department of Computer Science, University of Iowa, Iowa City, IA. {dcurtis,sriram}@cs.uiowa.edu

[‡] Department of Internal Medicine, University of Iowa, Iowa City, IA. philip-polgreen@uiowa.edu

(see this project at Iowa [16] for an example). The EIN provides very different kind of information to public health officials compared to the large scale online efforts that attempt to tap into the "wisdom of the crowds." Even though the EIN is sometimes the first to detect or report an outbreak, its real utility comes later when clinical aspects of emerging infectious diseases get discussed. For example, last spring when news of the H1N1 virus was everywhere in the popular media, the EIN was relatively quiet on this topic. However, the EIN is currently buzzing with H1N1 related posts as doctors and public health officials get ready to deal with a large number of cases. EIN members are discussing not just the emergence or spread of H1N1, but its treatment, vaccine administration, patient care, etc. [1, 2, 3]. One EIN member recently posted their concern about H1N1 vaccine reacting to neural tissue and causing Guillain-Barré Syndrome (GBS), a rare disorder resulting in limb weakness and paralysis. One responder identified a possible case of this and another pointed to historical evidence supporting the original concern. Further discussion amplified these concerns and provided information to the CDC which has instituted a case-finding protocol to monitor the situation, not only for GBS but for all immunization side-effects. Another EIN member identified a situation where healthcare workers were refusing to treat patients with H1N1 due to fear of exposure. Responders noted similar experiences, identified ethical concerns, and suggested policies. Occasionally discussion on the EIN can lead to discovery of previously unknown virus strains. For example, a post on the EIN in 2005 reported a number of severe pneumonia cases caused by the adenovirus, a common cause of respiratory illness [8]. Responses on the EIN mailing list helped identify these initial instances as a rare strain of community-acquired pneumonia which was previously unrecognized and later dubbed "the killer cold."

Identifying threads that are important is currently ad hoc, done by simply reading all the posts that make their way to the EIN. There is significant interest in improving the accuracy and timeliness with which this information is identified so that it can be distributed to the CDC and other healthcare organizations. Motivated by this need and the expectation that the EIN will grow in size in the near term, our goal is to develop a simple, low-cost procedure that can be used to *sample* traffic on the EIN and predict the emergence of important threads. Such a procedure will help focus the attention of doctors and public health officials to important, emerging discussions on the EIN. Ideally, we want to be able to identify threads that have the potential to become "important," and ignore threads that are "noise." Our approach is to look at historical EIN data (we have EIN traffic data from Feb. 1997 to May 2009) and identify users who typically participate in the early stages of many important threads, but are involved in very few unimportant threads. If we are able to identify such "bellwether" users, then tracking these users can quickly point people who make policies to emerging important threads that are in their early stages of evolution, without inundating them with irrelevant information.

Suppose we have identified a set S of these "bellwether" users. Anyone wanting to identify important discussions, can follow this simple monitoring procedure:

> An unmarked thread t is marked "to be monitored" as soon as a member of S posts to t. Thread t is closely monitored until it dies.

The problem is then to find a set S of EIN participants who act as "bellwethers." That is, find a set S of users who participate in many important threads, but do not participate in many unimportant threads.

The above monitoring procedure presupposes a classification of threads into *important threads*, those that signal emerging phenomena worth closely following and *unimportant threads*, those that are irrelevant from the point of view of infectious disease concerns. This classification can be done in an automated manner or by consultation with a infectious diseases expert. This classification can also be probabilistic: to each thread t we associate a probability $p(t)$ of being important (and therefore a probability $1 - p(t)$ of being unimportant). To find a set of users via whom we can track important threads, we need to make precise the notion of *participation* in a thread. Since we are interested in early detection, we use a parameter m and say that a user u participates in a thread t if u makes a post to thread t within the first m mailings of the thread. Once these notions are defined precisely, we can associate with every subset S of users a *reward* $r(S)$ and a *cost* $c(S)$. $r(S)$ can be defined as the number of important threads that users in S participate in. In other words, $r(S)$ is the number of important threads that will be monitored if the set S of users is tracked. $c(S)$ can be defined as the number of unimportant threads that users in S participate in. In other words, $c(S)$ is the number of unimportant threads that will have to be monitored if the set S of users is tracked. More general definitions of reward and cost are possible. For example, we could associate with each thread t a weight $w(t)$ and define $r(S)$ as the sum of the weights of important threads that users in S participate in. The definition of $c(S)$ can be generalized in a similar manner. If the notion of important and unimportant threads is defined probabilistically, then the definitions of reward and cost can be extended to refer to expected values. In this setting, good choices for S are obtained by solving the following budgeted maximization problem:

$$\max_{S \subseteq U} \; r(S) \text{ s. t. } c(S) \leq B$$

Here U is the set of all users and B is a given cost budget.

It is easy to see that all of the different versions of the reward function $r : 2^U \to \mathbb{R}^+$ mentioned above are *submodular*. Recall that a function $f : 2^U \to \mathbb{R}^+$ is said to be *submodular* if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ forall $A, B \subseteq U$. The problem of maximizing submodular set functions has a long history dating back to the 70's [14]. In their seminal work, Nemhauser et al. [14] consider the problem of maximizing a given submodular set function $f : 2^U \to \mathbb{R}^+$ and show that a simple greedy algorithm yields a $(1 - \frac{1}{e})$-approximation for this problem. Subsequently, several researchers have considered the problem of maximizing a submodular set function over all sets that satisfy a given constraint [4, 10]; more recently these results have been extended to problems with multiple constraints [12, 11]. Specifically, Khuller et al. [10] suppose that each element $u \in U$ is associated a *cost* $c(u)$ and the *cost* $c(S) = \sum_{u \in S} c(u)$. Their problem is to find a subset $S \subseteq U$ with maximum $f(S)$ from among all sets $S \subseteq U$ satisfying $c(S) \leq B$, they call this the Budgeted Maximum Coverage (BMC) problem. The BMC problem is used by Leskovec et al. [13] and El-Arini et al. [6] in their work on monitoring the blogosphere. Our budgeted maximization problem turns out to be fundamentally different on account of its cost structure. For two users $u, u' \in U$, $c(\{u, u'\})$ could be much smaller than $c(u) + c(u')$ because of a substantial overlap in the unimportant threads that u and u' participate in. Later we consider the greedy algorithm of Khuller et al. [10] that yields a constant-factor approximation for the BMC problem and construct a simple instance of our problem for which this greedy algorithm performs arbitrarily poor. We also show a reduction from the *k-densest subgraph* [7] problem to our problem that provides some indication that our budgeted maximization problem with "overlapping costs" might be much harder from an approximation point of view than the problem with linear costs.

1.1 Results We model the problem of monitoring a listserv, such as the EIN, as a type of budgeted maximum coverage problem. Even though our problem seems superficially similar to the budgeted maximum coverage problem considered by Khuller et al. [10], from an algorithmic point of view they are fundamentally different. The budget constraint of Khuller et al. [10] is linear, whereas ours is not. We show that the simple greedy algorithm that works well for the problem of Khuller at al. [10] performs arbitrarily poor on some instances of our problem. Furthermore, by showing a reduction from the *k-densest subgraph* [7] problem we provide some evidence to indicate that obtaining a constant-factor approximation for our problem might be quite challenging. Nevertheless, experimental runs of the greedy algorithm on the EIN data show that greedy performs remarkably well relative to OPT. We identify a feature of our EIN data, that we call the *overlap condition*, and show that the greedy algorithm does indeed provide a constant-factor approximation guarantee if the overlap condition is satisfied. Using an implementation of our greedy algorithm on the EIN data, we select a set of "bellwether" users to track and reduce the work involved in monitoring the EIN for a year by over 75%. Additionally, we provide evidence that this set of users participates in all of the important threads, while keeping the participation in "noisy" threads very low.

2 The Reward-Cost Model

Let T denote the set of threads, U denote the set of users, and $G = (T, U, E)$ denote the *user-thread graph*, a bipartite graph with edges $\{u, t\}$, $u \in U$, $t \in T$, whenever user u *participates* in thread t. We will make the notion of participation precise later. For any $u \in U$, let $N(u)$ denote the threads that user u participates in and for any subset $S \subseteq U$ of users let $N(S) = \cup_{u \in S} N(u)$. Associated with each thread $t \in T$, there is a probability $p(t)$ of thread t being important and a positive weight $w(t)$. For any subset $S \subseteq U$ of users, we define the set functions $r : 2^U \to \mathbb{R}^+$ and $c : 2^U \to \mathbb{R}^+$ as:

$$\begin{aligned} r(S) &= \sum_{t \in N(S)} p(t) \cdot w(t) \\ c(S) &= \sum_{t \in N(S)} (1 - p(t)) \cdot w(t) \end{aligned}$$

For the most part, in this paper we focus on the deterministic setting where $p(t) \in \{0, 1\}$ for each $t \in T$ and use T^+ to denote *important* threads, i.e., those threads t with $p(t) = 1$, and T^- to denote *unimportant* threads, i.e., those threads t with $p(t) = 0$. For ease of exposition we usually assume $w(t) = 1$ for all $t \in T$. The *budgeted maximization problem with overlapping costs* (BMOC) problem takes as input a user-thread graph $G = (U, T, E)$, probabilities $p : T \to [0, 1]$, weights $w : T \to \mathbb{R}^+$, a $B \in \mathbb{R}^+$ and aims to find a subset $S \subseteq U$ that maximizes $r(S)$ while satisfying the budget constraint $c(S) \leq B$.

Threads	Users	Posts	Mailings per thread			Posts per thread			People per thread		
			Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.
2833	1451	13,502	2.85 ±1.91	1.00	18.00	4.77 ±4.62	1.00	58.00	4.417 ±3.98	1.0	34.00

Figure 1: Summary statistics of number of mailings, number of posts, and number of users per thread.

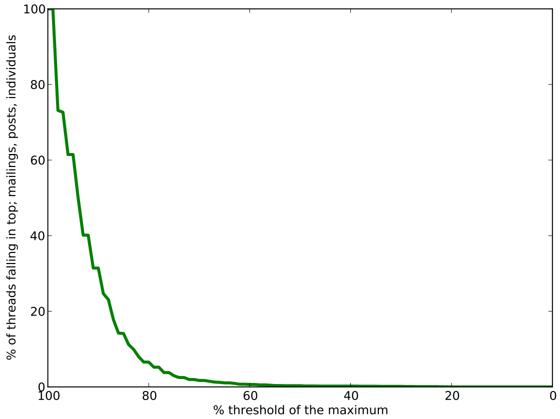
2.1 Choosing Important Threads One way to classify threads into important and unimportant threads is to consult infectious disease specialists. For example, one might survey EIN subscribers or have an online rating system in place. Since these approaches suffer from low response rate and are not currently in place, we develop an automated procedure for picking important threads by assuming that any thread worth monitoring closely will have sufficient EIN activity and therefore such threads can be identified by characteristics such as (a) number of mailings, (b) number of posts, and (c) number of distinct participants. Summary statistics of threads with respect to each of these characteristics are shown in Figure 1.

As one would expect, the distributions of the number of threads with respect to each of these characteristics are heavy-tailed. One simple way to pick “important” threads by paying attention to all three characteristics is the following. Let M^* be the maximum number of mailings in any thread, P^* be the maximum number of posts in any thread, and D^* be the maximum number of distinct participants in any thread (see Figure 1). For each value of a threshold parameter $thresh$, $0 \leq thresh \leq 100$, let $T^+(thresh)$ be the set of threads whose number of mailings are within $thresh$ % of M^*, number of posts are within $thresh$ % of P^*, and number of participants are within $thresh$ % of D^*. Figure 2a shows the cardinality of $T^+(thresh)$ for each $thresh$, $0 \leq thresh \leq 100$.

2.2 Criteria for Participation Since we are interested in early detection of potentially interesting threads, we focus on posts to a thread that are made very early on in life the thread. Specifically, we assign values to a parameter m and say that a user u *participates* in a thread t if u posts to t with the first m mailings of t. In our experiments, we use values $1, 2$, and 3.

3 A Greedy Algorithm for BMOC

Khuller et al. [10] present a simple greedy algorithm for the budgeted maximum coverage problem in which the budget constraint is linear and show that this algorithm guarantees a $\frac{1}{2}\left(1 - \frac{1}{e}\right)$-factor approximation ratio. When combined with an enumeration technique,



(a)

		Imp.	Unimp.
60%	Mean Mailings	12.00	2.79
	Mean # Distinct Users	25.11	4.28
	Mean # Posts	33.28	4.58
70%	Mean Mailings	9.88	2.73
	Mean # Distinct Users	20.23	4.14
	Mean # Posts	25.06	4.41
80%	Mean Mailings	6.98	2.56
	Mean # Distinct Users	14.79	3.69
	Mean # Posts	16.95	3.91

(b)

Figure 2: (a) Plots the percentage of threads whose number of mailings, number of posts, and number of participants are all within $thresh$ % of the corresponding maximum values of these characteristics. Our experiments use $thresh = 60$, $thresh = 70$, and $thresh = 80$ to pick out three candidate subsets of important threads. Since $T^+(thresh) \supset T^+(thresh')$ for $thresh > thresh'$, we obtain larger sets of important threads as we increase $thresh$ from 60 to 80. With $x = 60$, we pick up 18 (out of 2818) important threads, with $x = 70$, we pick up 47 (out of 2818) important threads, and with $x = 80$, we pick up 183 (out of 2818) important threads. (b) As $thresh$ increases the set of important threads grows larger, but the distinction between important and unimportant threads measured by number of mailings, number of posts, and number of distinct users becomes less pronounced.

this algorithm provides a $\left(1 - \frac{1}{e}\right)$-factor approximation ratio. To state this greedy algorithm in the context of our problem, we need notation for incremental reward and cost of adding a user to our current solution. Let $S \subseteq U$ and $u \in U \setminus S$. Then,

$$\begin{aligned} r(S,u) &= |\{t \in T^+ \mid t \notin N(S), t \in N(u)\}| \\ c(S,u) &= |\{t \in T^- \mid t \notin N(S), t \in N(u)\}| \end{aligned}$$

Algorithm 1 gives pseudocode for the greedy algorithm, which we call Greedy, combining two algorithms, which we call GreedyRatio and GreedyReward. GreedyRatio starts with an empty set S of users and repeatedly adds to S a user u who maximizes $\frac{r(S,u)}{c(S,u)}$ and whose addition to S does not violate the budget constraint. Similarly, GreedyReward starts with an empty set S of users and repeatedly adds to S a user u who maximizes $r(S,u)$ and whose addition to S does not violate the budget constraint. Let S' be the output of GreedyRatio and S'' be the output of GreedyReward. The algorithm Greedy runs GreedyRatio and GreedyReward and returns either S' or S'', whichever has the greater reward.

It is easy to construct an instance of BMOC for which Greedy performs arbitrarily poorly (see Figure 3).

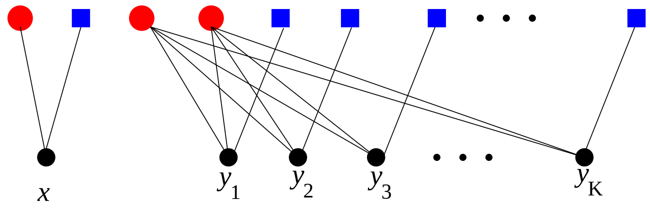


Figure 3: A user-thread graph with red vertices (circles) denoting unimportant threads and blue vertices (squares) denoting important threads. For this instance with budget $B = 2$, Greedy will pick x and obtain a reward of 1, whereas the optimal solution consists of $\{y_1, y_2, \ldots, y_K\}$ for a reward of K.

3.1 BMOC May be Difficult to Approximate Further bad news about BMOC is that even a special case of BMOC is at least as hard as the k-DensestSubgraph problem. The k-DensestSubgraph problem [7] takes as input a graph $G = (V, E)$ and seeks to find a subset of k vertices that induce a subgraph of G with maximum number of edges. The best known approximation algorithm for this problem yields an approximation factor of $O(n^{\alpha})$ where $\alpha < \frac{1}{3}$ [7]. Improving this approximation factor is an important open problem in the area of approximation algorithms. Chekuri [5] has sketched a simple

Algorithm 1 Greedy Algorithm for BMOC

```
GreedyRatio($U$)
$S' \leftarrow \emptyset$
$U' \leftarrow U$

while $U' \neq \emptyset$ do
    Pick $u \in U'$ that maximizes: $\frac{r(S',u)}{c(S',u)}$
    if $c(S' \cup \{u\}) \leq B$ then
        $S' \leftarrow S' \cup \{u\}$
    end if
    $U' \leftarrow U' \setminus \{u\}$
end while
return $S'$

GreedyReward($U$)
$S'' \leftarrow \emptyset$
$U' \leftarrow U$

while $U' \neq \emptyset$ do
    Pick $u \in U'$ that maximizes: $r(S'',u)$
    if $c(S'' \cup \{u\}) \leq B$ then
        $S'' \leftarrow S'' \cup \{u\}$
    end if
    $U' \leftarrow U' \setminus \{u\}$
end while
return $S''$

Greedy($U$)
$S' \leftarrow$ GreedyRatio($U$)
$S'' \leftarrow$ GreedyReward($U$)
if $r(S') \geq r(S'')$ then
    return $S'$
else
    return $S''$
end if
```

reduction from the k-DENSESTSUBGRAPH problem to BMOC that shows that a β-approximation algorithm for BMOC will imply a β-approximation algorithm for k-DENSESTSUBGRAPH. In fact, the reduction is from k-DENSESTSUBGRAPH to a special case of BMOC in which each user is connected to exactly 3 threads (as in our bad example in Figure 3). We present this reduction as evidence that it may be difficult to find approximation algorithms for BMOC that guarantee a small approximation factor such as a constant or even logarithmic (in the input size).

THEOREM 3.1. *If there is a β-approximation algorithm for BMOC, there is a β-approximation algorithm for k-DENSESTSUBGRAPH.*

Proof. Start with the instance $\{G = (V, E), k\}$ of k-DENSESTSUBGRAPH and construct a user-thread graph H with thread set $V \cup E$ and and user set E. Connect each user $e = \{u, v\}$ to three threads: u, v, and e. Designate E as the set of important threads and V as the set of unimportant threads. Finally, set the budget B to k. Let us call a solution $S \subseteq U$ to BMOC *maximal* if for all users $u \in U \setminus S$, $c(S \cup \{u\}) > c(S)$. It is easy to verify the following claim.

Claim: G has an induced subgraph with k vertices and m edges iff H has a maximal subset S of users with $c(S) = k$ and $r(S) = m$.

Now, suppose there exists a β-approximation algorithm A for BMOC. Start with an instance $\{G, k\}$ of the k-densest subgraph problem. Transform it as specified above to an instance H of BMOC and run A on H. The solution is a set of users S such that

$$c(S) \leq k, \qquad r(S) \geq \beta \cdot OPT$$

where OPT is the maximum reward of a subset S^* of users in H satisfying $c(S^*) \leq k$. Without loss of generality suppose that both S and S^* are maximal. Then $|S^*| = OPT$ and by the above Claim, S^* is an edge set in G of maximum size that is induced by a set of at most k vertices. The set S returned by the algorithm A is also a set of edges induced by a set of at most k vertices and since $|S| \geq \beta \cdot |S^*|$, we have a solution to the k-densest subgraph problem that is within a factor β of OPT. Note that if $|S| < k$ we can arbitrarily add vertices of G to S until $|S| = k$.

3.2 The Overlap Condition

In the bad example for the greedy algorithm, in Figure 3, the unimportant threads have high average degree (i.e., $(2K + 1)/3$) relative to the average degree of important threads (which is just 1). While this is possible in general for BMOC, our specific criteria for identifying important and unimportant threads from the EIN data makes this unlikely for our instances of the problem. We now formalize this heuristic notion, calling it the *overlap* condition and show that if we assume that the overlap condition holds, then GREEDY provides a $\frac{1}{2}(1-\frac{1}{e})$-approximation. In fact, assuming the overlap condition we can obtain a $(1-\frac{1}{e})$-approximation by using GREEDY in combination with the enumeration technique described by Khuller et al. [10].

Let S_i denote the set of the first i users selected by GREEDYRATIO. Let U_i denote the remaining users, i.e., $U \setminus S_i$ and let G_i denote the bipartite graph obtained from the user-thread graph G by deleting $S_i \cup N(S_i)$. Thus the users in G_i are those in U_i and the threads in G_i are those that are not "covered" by users in S_i. For any subset $U' \subseteq U_i$ of users, let $G_i[U']$ denote the bipartite subgraph of G_i induced by $U' \cup N(U')$. Let $\delta^+(i, U')$ (respectively, $\delta^-(i, U')$) denote the average degree of the important (respectively, unimportant) threads in $G_i[U']$. We define the *overlap* condition as:

$$\forall i, \forall U' \subseteq U_i : \delta^+(i, U') \geq \alpha \cdot \delta^-(i, U'). \tag{3.1}$$

for some constant universal α. Let $r(S_i, U')$ (respectively, $c(S_i, U')$) denote the number of important (respectively, unimportant) threads in $N(U') \setminus N(S_i)$. It is easy to verify that

$$\begin{aligned} \delta^+(i, U') &= \frac{\sum_{u \in U'} r(S_i, u)}{r(S_i, U')} \\ \delta^-(i, U') &= \frac{\sum_{u \in U'} c(S_i, u)}{c(S_i, U')}. \end{aligned}$$

and therefore the overlap condition can be equivalently stated as

$$\forall i, \forall U' \subseteq U_i : \frac{\sum_{u \in U'} r(S_i, u)}{r(S_i, U')} \geq \alpha \cdot \frac{\sum_{u \in U'} c(S_i, u)}{c(S_i, U')}. \tag{3.2}$$

for some constant universal α.

Let OPT be an optimal set of users. Suppose that after some number of iterations, GREEDYRATIO has selected a set S of users. In the next iteration, GREEDYRATIO considers an element $u \notin S$ that maximizes $\frac{r(S,u)}{c(S,u)}$. This element may or may not be added to S depending on whether adding u to S causes the budget constraint to be violated. Suppose that r is the number of iterations executed by GREEDYRATIO until the first user $u \in OPT$ is considered, but rejected (due to violation of the budget constraint). Suppose that ℓ users have been selected by GREEDYRATIO during these r iterations. Label these users $u_1, u_2, \ldots, u_\ell$ in the order in which they were selected by GREEDYRATIO and let $u_{\ell+1}$ be the first user in OPT considered but rejected. Let j_i be the iteration in which user u_i was considered. Finally, let $S_0 = \emptyset$, $S_i = S_{i-1} \cup \{u_i\}$ for each $i = 1, 2, \dots, \ell$.

The following lemma uses the overlap condition to extend a lemma in [10] to instances of BMOC in which the overlap constraint holds. The calculations in the subsequent lemmas are similar to those in [10], we include these mainly for completeness.

Lemma 3.1. *If the overlap condition is satisfied, then after each iteration $j_i, i = 1, 2, \dots, \ell + 1$,*

$$r(S_{i-1}, u_i) \geq \alpha \cdot \frac{c(S_{i-1}, u_i)}{B}\Big(r(OPT) - r(S_{i-1})\Big).$$

Proof. For each user $u \in OPT \setminus S_{i-1}$, due to the greedy choice of u_i, the ratio $\frac{r(S_{i-1},u)}{c(S_{i-1},u)}$ is at most $\frac{r(S_{i-1},u_i)}{c(S_{i-1},u_i)}$. Therefore,

$$\sum_{u \in OPT \setminus S_{i-1}} r(S_{i-1}, u) \leq \frac{r(S_{i-1}, u_i)}{c(S_{i-1}, u_i)} \sum_{u \in OPT \setminus S_{i-1}} c(S_{i-1}, u)$$

This can be rewritten as

$$\frac{\sum_{u \in OPT \setminus S_{i-1}} r(S_{i-1}, u)}{\sum_{u \in OPT \setminus S_{i-1}} c(S_{i-1}, u)} \leq \frac{r(S_{i-1}, u_i)}{c(S_{i-1}, u_i)}. \tag{3.3}$$

According to the overlap condition:

$$\frac{\sum_{u \in OPT \setminus S_{i-1}} r(S_{i-1}, u)}{\sum_{u \in OPT \setminus S_{i-1}} c(S_{i-1}, u)} \geq \alpha \cdot \frac{r(S_{i-1}, OPT \setminus S_{i-1})}{c(S_{i-1}, OPT \setminus S_{i-1})}.$$

Combining this with 3.3 yields

$$\alpha \cdot \frac{r(S_{i-1}, OPT \setminus S_{i-1})}{c(S_{i-1}, OPT \setminus S_{i-1})} \leq \frac{r(S_{i-1}, u_i)}{c(S_{i-1}, u_i)}. \tag{3.4}$$

Substituting into the above inequality the fact that $c(S_{i-1}, OPT \setminus S_{i-1}) \leq c(OPT) \leq B$, we get

$$r(S_{i-1}, OPT \setminus S_{i-1}) \leq \frac{B}{\alpha} \cdot \frac{r(S_{i-1}, u_i)}{c(S_{i-1}, u_i)}$$

It is easy to see that $r(OPT) - r(S_{i-1})$ is at most $r(S_{i-1}, OPT \setminus S_{i-1})$. This leads to

$$r(OPT) - r(S_{i-1}) \leq \frac{B}{\alpha} \cdot \frac{r(S_{i-1}, u_i)}{c(S_{i-1}, u_i)}$$

Moving terms around, yields the lemma.

Lemma 3.2. *If the overlap condition is satisfied, then for iterations $j_i, i = 1, 2, \dots, \ell + 1$,*

$$r(S_i) \geq \left[1 - \prod_{k=1}^{i} \left(1 - \alpha \frac{c(S_{k-1}, u_k)}{B}\right)\right] r(OPT)$$

Proof. The proof follows by induction on the iterations $j_i, i = 1, 2, \dots, \ell + 1$. For iteration j_1 we have $r(S_1) = r(S_0, u_1)$ and need to prove that $r(S_1) \geq \alpha \frac{c(S_0,u_1)}{B} r(OPT)$. For each user $u \in U$, due the greedy choice of u_1, $\frac{r(S_0,u)}{c(S_0,u)}$ is at most $\frac{r(S_0,u_1)}{c(S_0,u_1)}$. Thus,

$$\sum_{u \in OPT} r(S_0, u) \leq \frac{r(S_0, u_1)}{c(S_0, u_1)} \sum_{u \in OPT} c(S_0, u)$$

Which can be rewritten as:

$$\frac{\sum_{u \in OPT} r(S_0, u)}{\sum_{u \in OPT} c(S_0, u)} \leq \frac{r(S_0, u_1)}{c(S_0, u_1)}$$

And combining with the overlap condition and using the fact that $c(OPT) \leq B$ we get:

$$\frac{r(S_0, u_1)}{c(S_0, u_1)} \geq \alpha \frac{r(OPT)}{B}$$

Thus,

$$r(S_1) = r(S_0, u_1) \geq \alpha \frac{c(S_0, u_1)}{B} r(OPT)$$

Assuming the lemma holds for iterations $j_i, i = 1, .., i - 1$ we show it holds for j_i:

$$\begin{aligned}
r(S_i) &= r(S_{i-1}) + r(S_{i-1}, u_i) \\
&\geq r(S_{i-1}) + \alpha \frac{c(S_{i-1}, u_i)}{B} \left(r(OPT) - r(S_{i-1})\right) \\
&= \left[1 - \alpha \frac{c(S_{i-1}, u_i)}{B}\right] r(S_{i-1}) + \\
&\quad \alpha \frac{c(S_{i-1}, u_i)}{B} r(OPT) \\
&\geq \left[1 - \alpha \frac{c(S_{i-1}, u_i)}{B}\right] \cdot \\
&\quad \left[1 - \prod_{k=1}^{i-1} \left(1 - \alpha \frac{c(S_{k-1}, u_k)}{B}\right)\right] r(OPT) + \\
&\quad \alpha \frac{c(S_{i-1}, u_i)}{B} r(OPT) \\
&= \left[1 - \prod_{k=1}^{i} \left(1 - \alpha \frac{c(S_{k-1}, u_k)}{B}\right)\right] r(OPT)
\end{aligned}$$

Theorem 3.2. *If an instance of the user-thread graph $G = (U, T, E)$ satisfies the overlap condition with respect to an execution of Algorithm* GreedyRatio *then the set S of users returned by Algorithm* Greedy *satisfies*

$$r(S) \geq \frac{1}{2}\left(1 - \frac{1}{e^{\alpha}}\right) \cdot OPT,$$

where OPT is the maximum reward associated with any set of users whose cost is at most the budget B.

	Pairs	Triples
Total	271784	68456236
OC Holds	271490(99.98%)	68443062(99.98%)
Min. Factor (α)	0.704	2.96
Avg. Factor (α)	0.647	2.90

Table 1: Results from analyzing the overlap condition for all pairs and triples of users. *OC Holds* shows the number of sets $U' \subseteq U$ for which $\delta^+(U') \geq \delta^-(U')$. *Min. Factor* (respectively *Avg. Factor*) shows the smallest value (respectively average) value of $\frac{\delta^+(U')}{\delta^-(U')}$ over all U'.

Proof. Consider iteration $\ell + 1$. Using lemma 3.2 and the fact that $c(S_{\ell+1}) > B$ we have:

$$\begin{aligned} r(S_{\ell+1}) &\geq \left[1 - \prod_{k=1}^{\ell+1}\left(1 - \alpha\frac{c(S_{k-1}, u_k)}{B}\right)\right] r(OPT) \\ &\geq \left[1 - \prod_{k=1}^{\ell+1}\left(1 - \alpha\frac{c(S_{k-1}, u_k)}{c(S_{\ell+1})}\right)\right] r(OPT) \\ &\geq \left[1 - \left(1 - \frac{\alpha}{\ell+1}\right)^{\ell+1}\right] r(OPT) \\ &\geq \left(1 - \frac{1}{e^\alpha}\right) r(OPT) \end{aligned}$$

Thus,

$$r(S_{\ell+1}) = r(S_\ell) + r(S_\ell, u_{\ell+1}) \geq (1 - \frac{1}{e^\alpha}) r(OPT)$$

Since $r(S_0, u_{\ell+1})$ is at most the maximum reward for a single user we have $r(S_0, u_{\ell+1}) \leq r(S'')$, the reward given by GreedyReward. This gives us:

$$r(S_\ell) + r(S'') \geq r(S_\ell) + r(S_\ell, u_{\ell+1}) \geq \left(1 - \frac{1}{e^\alpha}\right) r(OPT)$$

Therefore either the reward given by GreedyRatio, $r(S') \geq r(S_\ell)$ or the reward given by GreedyReward, $r(S'')$ is greater than or equal to $\frac{1}{2}\left(1 - \frac{1}{e^\alpha}\right) r(OPT)$.

The overlap condition, as equivalently stated in (3.1) and (3.2) is required to be satisfied for every i and $U' \subseteq U_i$ for Theorem 3.2. We "tested" the overlap condition for the EIN data in a limited way by considering all pairs and triples of users (see Table 1).

Specifically, when $i = 0$, $S_i = \emptyset$, $U_i = U$ and the overlap condition reduces to

$$\forall U' \subseteq U : \delta^+(U') \geq \alpha \cdot \delta^-(U'),$$

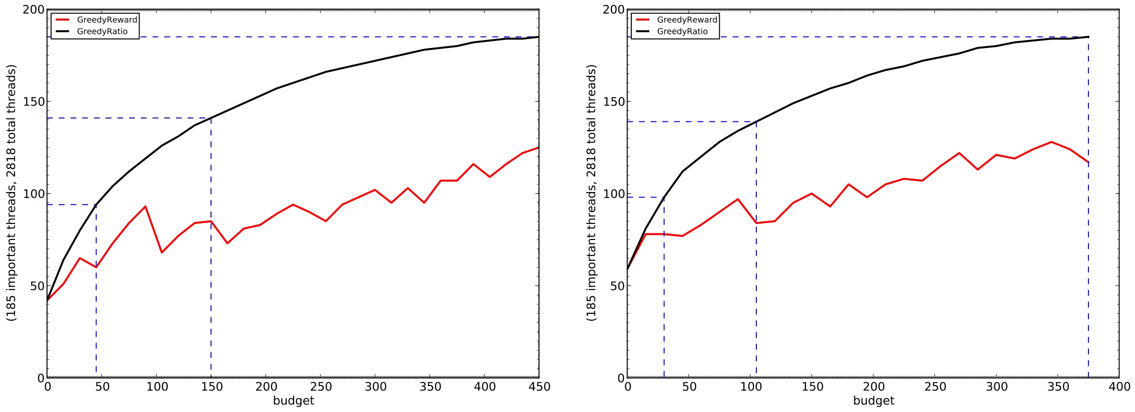
where $\delta^+(U')$ (respectively, $\delta^-(U')$) is the average degree of the important threads (respectively, unimportant threads) in the subgraph of G induced by $U' \cup N(U')$. As stated, Theorem 3.2 uses the "worst case" value of the universal constant α. It can be strengthened to use the "average" value of α, leading to a better approximation factor. We postpone further discussion of this to the full version of the paper.

4 Experiments on BMOC

Choosing a particular threshold *thresh* (60, 70, or 80), as described in Section 2.1, induces a partition of the set of threads into important and unimportant threads. By fixing a value for the participation parameter m (1, 2, 3, or ∞), as described in Section 2.2, we fix the threads each individual has participated in. Having fixed *thresh* and m, we consider all values of the budget B, starting with $B = 0$, until we achieve full coverage of all important threads. Fixing values for *thresh*, m, and B creates an instance of BMOC that we use as input to Greedy.

4.1 Greedy Performance Figure 4 shows plots for solutions found by GreedyRatio and GreedyReward for instances with $thresh = 80$ and participation parameter values $m = 2$ and $m = 3$. Recall that Greedy simply returns the better of the solutions produced by GreedyRatio and GreedyReward. Results shown here are similar for all *thresh* and m values we considered. We can view the reward of a solution returned by GreedyRatio or GreedyReward as a function of B. Note that neither of these functions are monotonic in B – simple examples are easy to construct for both algorithms. As a result, one simple improvement to these algorithms is to consider all values $B' = 1, 2, \ldots, B$ as the budget, run GreedyRatio and GreedyReward with each value of B' as the budget, and return as a solution, the subset that has maximum reward over all values of B'. Table 2 focuses on specific points on the plots in Figure 4, analyzing these more closely. In particular, this analysis focuses on points that provide 50%, 75%, and 100% coverage of the important threads.

4.2 Analysis of Selected Users The majority of active users on the EIN are doctors either in private practice, with only clinical responsibilities, or at an academic institution, where they have clinical and research responsibilities. Table 3a shows the distribution of users selected by Greedy (for $thresh = 80$ and full coverage) by whether they are at an academic institution, in private practice, or elsewhere. These results nicely match the expectations of the third author that doctors in private practice tend to initiate more important threads, possibly because they have more clinical experience and have fewer colleagues with whom they can discuss is-



(a)

(b)

Figure 4: Plots showing the reward of solutions produced by GreedyRatio and GreedyReward with *thresh* = 80 and (a) $m = 2$ and (b) $m = 3$. The x-axis shows the budget and the y-axis shows reward. The black (respectively, red) line shows the reward produced by the GreedyRatio (respectively, GreedyReward) algorithm. The dotted lines mark points of interest, corresponding to 50%, 75%, and 100% coverage of important threads, discussed further in Table 2.

thresh	m	c		
		50%	75%	100%
80% (185)	1	164.0	404.0	949.0
	2	45.0	148.0	436.0
	3	30.0	103.0	363.0
	∞	15.0	60.0	205.0

Table 2: The cost of solutions that achieve 50%, 75%, and 100% coverage of important threads (corresponding to points from the plots shown in Figure 4). The key findings reported in this table are (a) the cost of full (respectively, 75%) coverage is roughly 10 (respectively, 3) times the cost of half coverage and (b) relaxing the requirement of early detection (i.e., increasing m from 1 to 3) decreases costs significantly.

sues face-to-face. Such users tend to turn to the EIN more frequently with important concerns. On the other hand first responders and later responders in important threads tend to be evenly distributed between doctors at academic institutions and those in private practice. Table 3b shows that selected users (at *thresh* = 80, full coverage) are geographically spread out quite evenly across the U.S. even though geographic coverage was not a criteria used in our algorithms.

4.3 Analysis of Selected Threads Using the procedure mentioned in the introduction, the set S of selected users can be used to mark threads as "to be

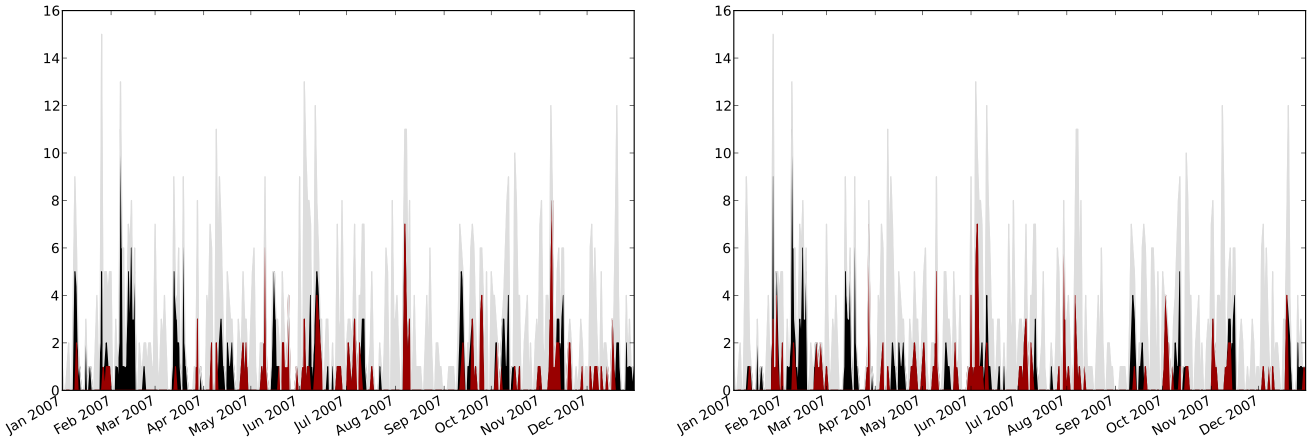
m	Total	Academic	Private	Other	Unknown
1	126	30(32.97%)	57(62.64%)	4(4.40%)	35
2	161	34(45.33%)	34(45.33%)	7(9.93%)	86
3	158	36(48.65%)	32(43.24%)	6(8.11%)	84
∞	186	33(42.86%)	35(45.45%)	9(11.69%)	109

(a)

m	Avg Distance (±)	Max Distance
2	230.57(±169.23)	885.08
3	212.89(±141.28)	714.59

(b)

Table 3: (a) Distribution of users selected by Greedy (with *thresh* = 80, full coverage) by whether they are at an academic institution, private practice, or elsewhere. The column Total shows the total number of users selected by our algorithm. (b) The geographic spread of users selected by Greedy (with *thresh* = 80, full coverage) is shown here. For example, with $m = 2$, every point in the continental U.S. is within 231 miles of a selected user, on average. These statistics were obtained by sampling 10 million points uniformly at random; more accurate results can be obtained by constructing Voronoi diagrams.



(a)

(b)

Figure 5: Plots of the number of posts read per day for 2007. The results are from running Greedy with *thresh* $= 80$, full coverage, and (a) $m = 2$ and (b) $m = 3$. The black and red marks correspond to posts that are read; black marks correspond to posts to important threads and red marks correspond to posts to unimportant threads. The gray marks show the number of posts made to any thread, regardless of whether the thread was marked.

monitored.” Ideally, we would like the number of “to be monitored” threads small relative to the total number of threads. Table 4a shows the number of threads and posts observed in 2007 and the number of threads that would have been marked and number of posts that would have been read, had this procedure been in place then. For both $m = 2$ and $m = 3$, the number of marked threads are about a fourth of the total and the number of posts are about a third of the total. The per-day totals for traffic to the EIN for $m = 2$ (figure 5a) and $m = 3$ (figure 5b) over time for the 2007 year. Of the posts that are read, more than half are important. Table 4b shows, for each value of m, the mailing at which important threads would have been marked using this procedure. Together the two tables show that as we go from $m = 2$ to $m = 3$ the cost of monitoring falls (40 threads to 38 threads, 144 posts to 140 posts) accompanied by a delay in marking a few threads (5).

4.4 Greedy Versus OPT For instances of BMOC where the number of neighbors of important threads is small, *OPT* can be calculated in a reasonable time (just by brute force). Figure 6 shows two plots comparing *OPT* with solutions returned by Greedy. Note that in both Figure 6a and Figure 6b, Greedy and *OPT* are identical for the most part and when they are not identical, *OPT* is only marginally better. Even though

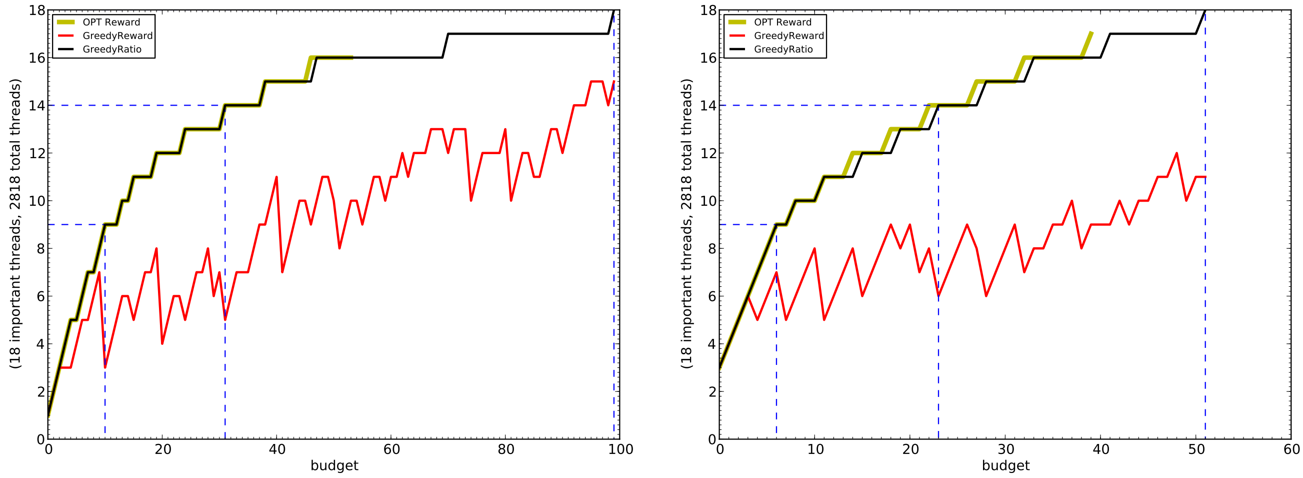
m		Total	Marked	Imp.	Unimp.
2	Threads	229	54	14	40
	Posts	1015	314	170	144
3	Threads	229	52	14	38
	Posts	1015	289	149	140

(a)

m	Mailing Marked		
	1st	2nd	3rd
2	2	12	
3	2	7	5

(b)

Table 4: EIN traffic statistics for the year 2007 for full coverage at *thresh* $= 80$.



(a)

(b)

Figure 6: Plots comparing the performance of GREEDY with *OPT* at *thresh* = 60. The x-axis is the budget and the y-axis is the reward. The black line shows the reward produced by the GREEDY and the shaded line shows *OPT*. (a) Participation defined as $m = 2$ (b) Participation defined as $m = 3$.

the BMOC problem seems difficult to approximate in general, our limited experiments show that the GREEDY produces a near-optimal solution for the EIN data. This may be partly explained by overlap condition mentioned earlier. A small improvement is obtained by enhancing GREEDY with a small "look-ahead." That is, at each step we can consider adding a subset of users, such as pairs or triples, rather only considering single users. Figure 7 shows the same plots as in section4.1 with the additional results found by modifying the GREEDYRATIO algorithm to consider pairs of users $u, u^{'} \in U$ at each iteration. While it doesn't make a significant improvement overall, the improvements are noticeable. With a very minor modification to the algorithm we find a better solution, at the cost of running time. We could improve this solution further by considering triples, or larger subsets, of users.

5 Open Problems and Further Analysis

This paper models the problem of monitoring listservs such as the EIN as a budgeted maximization problem with overlapping costs (BMOC). There are many refinements of the model that seem worth pursuing, e.g., clustering the users by geographic locations or via the social network induced by postings.

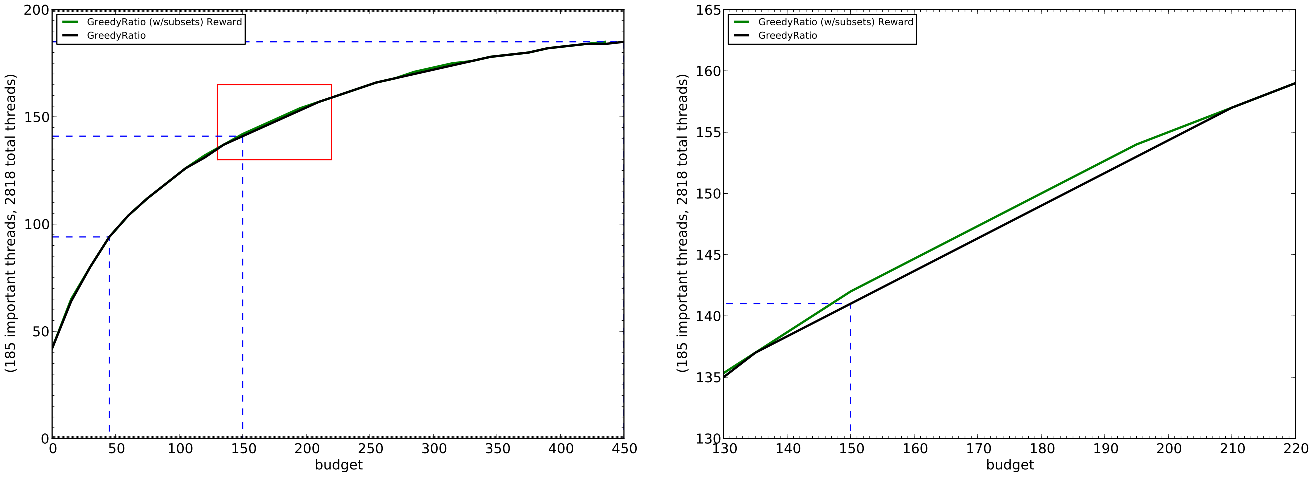
On the more algorithmic front we are interested in the computational complexity of approximating BMOC. One the positive side, we are interested in designing approximation algorithms that provide non-trivial approximation guarantees for BMOC. Due to the fact that the k-densest subgraph problem, for which the best known approximation is $O(n^{\frac{1}{3}})$ (see [7]), is reducible to an instance of BMOC, we believe coming up with an approximation algorithm for BMOC that gives a guarantee better than $O(n^{\frac{1}{3}})$ to be difficult. But at this time, even an $O(n^{\alpha})$-approximation for constant $\alpha < 1$ is unknown for BMOC. On the negative side, we are interested in proving a hardness of approximation result for BMOC that shows that it is "strictly" harder than Budgeted Maximum Coverage (BMC). It is known that BMC is inapproximable to a factor better than $(1 - \frac{1}{e})$ [10]. BMOC is at least as hard and it is our belief that in general, BMOC is a much harder problem than BMC due to the overlapping costs aspect of the problem.

Acknowledgments We'd like to thank Susan Beekmann RN, MPH (EIN Program Coordinator) for her many contributions to this work and all the members of the CompEpi group at the University of Iowa (http://compepi.cs.uiowa.edu/) for their comments and suggestions.

References

[1] EIN: H1N1 and HCW Reassignment Questions. *IDSAnews*, Sep 2009. http://news.idsociety.org/idsa/issues/2009-09-01/3.html.

[2] EIN: H1N1 Vaccine and Guillain-Barr Syndrome. *IDSAnews*, 19(8), Aug 2009. http://news.idsociety.org/idsa/issues/2009-08-01/4.html.



(a)

(b)

Figure 7: Plots showing the (small) improvement obtained by enhancing GREEDY with "look-ahead;" all subsets of size at most 2 are considered as candidates in each iteration. Here *thresh* = 80 and $m = 2$. The x-axis is the budget and the y-axis is the reward. The black line shows the reward produced by GREEDY without "look-ahead" and the green line shows the improvement due to "look-ahead." (a) shows the full plot and (b) shows the zoomed portion given by the red box in (a).

[3] EIN: Treatment Options for H1N1 Pneumonia and Antiviral Use in Infants. *IDSAnews*, 19(7), Jul 2009. http://news.idsociety.org/idsa/issues/2009-07-31/5.html.

[4] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint (extended abstract). In *IPCO '07: Proceedings of the 12th international conference on Integer Programming and Combinatorial Optimization*, pages 182–196, Berlin, Heidelberg, 2007. Springer-Verlag.

[5] Chandra Chekuri. Personal Communication, 2009.

[6] Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–298, New York, NY, USA, 2009. ACM.

[7] Uriel Feige, Guy Kortsarz, and David Peleg. The dense k-subgraph problem. *Algorithmica*, 29:2001, 1999.

[8] Centers for Disease Control. CDC - Adenoviruses, Jan 2005. http://www.cdc.gov/ncidod/dvrd/revb/respiratory/eadfeat.htm.

[9] Ginsberg J., Mohebbi MH., Patel RS., Brammer L., Smolinski MS., and Brilliant L. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–4, 2008.

[10] Samir Khuller, Anna Moss, and Joseph (Seffi) Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999.

[11] Ariel Kulik, Hadas Shachnai, and Tami Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 545–554, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[12] Jon Lee, Vahab S. Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 323–332, New York, NY, USA, 2009. ACM.

[13] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie S. Glance. Cost-effective outbreak detection in networks. In *KDD*, pages 420–429, 2007.

[14] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions-1. *Mathematical Programming*, 14:265–294, 1978.

[15] Polgreen PM, Chen Y, Pennock DM, and Nelson FD. Using internet searches for influenza surveillance. *Clin. Infect. Dis.*, 47(11):1443–8, 2008.

[16] Alessio Signori. Monitoring swine flu using twitter, Sept 2009. http://www.cs.uiowa.edu/~asignori/projects/twitter-monitor-swine-flu/.