

Partial Information Spreading with Application to Distributed Maximum Coverage

Keren Censor Hillel* and Hadas Shachnai†
Department of Computer Science, Technion
Haifa 32000, Israel

February 18, 2010

Abstract

This paper addresses *partial information spreading* among n nodes of a network. As opposed to traditional information spreading, where each node has a message that must be received by all nodes, we propose a relaxed requirement, where only n/c nodes need to receive each message, and every node should receive n/c messages, for some $c \geq 1$.

As a key tool in our study we introduce the novel concept of *weak conductance*, a generalization of classic graph conductance which allows to analyze the time required for partial information spreading. We show the power of weak conductance as a measure of how well-knit the components of a graph are, by giving an example of a graph family for which the conductance is $O(n^{-2})$, while the weak conductance is as large as $1/2$. For such graphs, weak conductance can be used to show that partial information spreading requires time complexity of $O(\log n)$.

Finally, we demonstrate the usefulness of partial information spreading in solving the *maximum coverage* problem, which naturally arises in circuit layout, job scheduling and facility location, as well as in distributed resource allocation with a global budget constraint. Our algorithm yields a constant approximation factor and a constant deviation from the given budget. For graphs with a constant weak conductance, this implies a scalable time complexity for solving a problem with a global constraint.

Keywords: Distributed computing, randomized algorithms, weak conductance, partial information spreading, maximum coverage, approximation algorithms

*Keren Censor Hillel is a full-time student at the Technion. Supported in part by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities and by the *Israel Science Foundation* (grant number 953/06). Email: ck-eren@cs.technion.ac.il.

†Email: hadas@cs.technion.ac.il.

1 Introduction

Many distributed applications require the nodes in a network to spread information throughout the network in order to perform a global task. The problem of *information spreading* is to distribute the messages sent by each of the nodes in a network to all other nodes. Information spreading algorithms have been extensively studied (see, e.g. [7, 10, 16, 17]). We consider the synchronous push/pull model of communication, where each node chooses in each round a random *neighbor* to exchange information with.

The time required for achieving information spreading depends on the structure of the communication graph, or more precisely, on how well-connected it is. The notion of graph *conductance*, defined by Sinclair [24], gives a measure of the connectivity of a graph. Roughly speaking, the conductance of a graph G , denoted by $\Phi(G)$, is a value in $[0, 1]$: This value is large for graphs that are well-connected (e.g., cliques), and small for graphs that are not (i.e., graphs which have many communication bottlenecks). Graph conductance plays a pivotal role in analyzing algorithms for such NP-hard optimization problems as clustering and graph partitioning, as well as in recent studies of social networks (e.g. [10]). In distributed computing, it has been shown that the time required for information spreading crucially depends on the conductance of the underlying communication graph [4, 9, 10, 20]. In particular, Mosk-Aoyama and Shah [20] show that, for any $\delta \in (0, 1)$, information spreading can be achieved in $O(\frac{\log n + \log \delta^{-1}}{\Phi(G)})$ rounds with probability at least $1 - \delta$. This implies that information spreading is faster on graphs with large conductance.

Some graphs have a small conductance, implying that they are not well-connected and therefore require many rounds of communication for information spreading. Nevertheless, for some of these graphs we can do better if we do not require the information of every node to reach every other node in the network. This is the focus of our paper.

We define *partial information spreading*, where the condition that each node receives the information of all other nodes (to which we refer as *full information spreading*) is relaxed to smaller amounts of information. Formally, for some values $c \geq 1$ and $\delta \in (0, 1)$, we require that with probability at least $1 - \delta$ every message reaches at least n/c nodes, and every node receives at least n/c messages.¹ We call an algorithm that fulfills this requirement (δ, c) -*spreading*. Indeed, the special case where $c = 1$ corresponds to full information spreading.

As a key tool in our study we introduce the novel concept of *weak conductance*, a generalization of graph conductance which allows to analyze the time required for partial information spreading. We show the power of weak conductance as a measure of how well-knit the components of a graph are, by giving an example of a graph family for which the conductance is $O(n^{-2})$, while the weak conductance is as large as $1/2$. For such graphs, weak conductance can be used to show that partial information spreading requires time complexity of $O(\log n)$.

We demonstrate the usefulness of partial information spreading in solving the classic *maximum coverage* problem defined as follows. Given is a universe of m elements, each having some nonnegative weight, and n subsets of the elements; also, given is an integer $K \geq 1$. We need to select a collection of K subsets so as to maximize the total weight of the covered elements. Coverage problems are at the heart of resource allocation problems in communication networks and information systems (see, e.g., [25, 26]). In particular, the maximum coverage problem and its variants naturally arise in circuit layout, job scheduling and facility location (see, e.g., [2, 18] and a comprehensive survey in [13]). We give an algorithm for the special case of the problem where each element belongs to exactly two subsets. In a distributed network, this is the problem of selecting K nodes with the goal of maximizing the total number of covered edges. Consider, for example, a monitoring system for the traffic flow on the links of the network. The system can handle in each

¹For simplicity of the presentation, we assume throughout the paper that c and n/c are integers; however, all the results hold for any $c, n/c \geq 1$.

phase the data collected from at most K nodes on the status of their neighboring links, for some $K \geq 1$. Then the objective is to select in each phase a subset of K nodes which cover the maximum number of (unmonitored) links.² Our algorithm yields a constant approximation factor and a constant deviation from the given budget. For graphs with a constant weak conductance, this implies a scalable time complexity for solving a problem with a global constraint. The same algorithm can be applied to more general instances of maximum coverage, where each element has a weight and can appear in arbitrary number of subsets, as well as for approximating the *budgeted maximum coverage* problem (see in Section 3).

Main Contributions: Our first main contribution is in generalizing the definition of conductance to *weak conductance*. Roughly speaking, rather than measuring connectivity of the whole graph, weak conductance is the minimal conductance among subsets of n/c nodes that are induced by the *best* partition of the graph into c subgraphs (we give the precise definition in Section 2).

We prove that partial information spreading is fast on graphs which have a large weak conductance, although they may have small conductance and therefore do not enable fast full information spreading. Specifically, we prove that for any $\delta \in (0, 1)$, partial information spreading can be achieved in $O(\frac{\log n + \log \delta^{-1}}{\Phi_c(G)})$ rounds, where $\Phi_c(G)$ is the weak conductance of the graph. Moreover, we give examples of families of graphs for which partial information spreading is significantly faster than full information spreading.

Our second main contribution is in showing that for solving maximum coverage, we can do well enough with only partial information spreading. In Section 3 we show how to solve the maximum coverage problem in a distributed manner with a *constant* approximation factor, given a partial information spreading algorithm. Our result implies that for graphs with a large weak conductance (of $\Omega(1/\log n)$) our algorithm has a scalable time complexity, in spite of the need to address a global constraint.

Finally, in Section 4 we extend our results to networks without node identities. This model captures well ad-hoc and mobile networks, which lack infrastructure such as IP addresses, limiting the knowledge a node can gain on the structure of the network. We borrow the technique of [20], which allows the nodes to estimate sums of values of other nodes despite the possibility of duplicated messages, and show how it can be embedded in our algorithm for maximum coverage, in order to obtain a constant approximation, while using only partial information spreading.

Related Work: Communication models vary in different studies. For example, Karp et al. [16] consider the *random phone-call* model, where in each round very node chooses a random node to communicate with, assuming that the communication graph is complete. Our results hold for arbitrary communication graphs.

Previously, Avin and Brito [3] and Avin and Ercal [4] analyzed the partial cover time of a random walk. This notion measures the time required for a random walk to visit a large fraction of the nodes, but not all nodes (as measured by the cover time). A random walk is related to our model, since the process of relaying a message in the graph corresponds to a random walk, however, as opposed to the single random walk considered in [3, 4], our model consists of many parallel random walks for every message, as a new random walk begins in each round.

Dolev et al. [23] considered gossip in multi-channel radio networks, where in each round a node chooses a channel on which to participate. The paper introduces the ϵ -gossip problem in which $(1 - \epsilon)n$ of the messages need to be fully spread in the network. This differs from our definition of partial information spreading, since we require *all* messages to be *partially* spread in the network.

Georgiou et al. [12] investigated *majority gossip* for solving consensus. The requirement of majority gossip is that each node receives the message of a majority of the nodes, guaranteeing some overlap of

²See also in [25].

received messages. This is strictly stronger than our definition of partial spreading, in which a node may receive only $n/2$ messages (or less, for larger values of c).

Some modifications to the definition of graph conductance have been proposed in the past (see, e.g., [15, 19, 21]), but all are different in essence from the concept of weak conductance presented in this work.

The maximum coverage problem, which is known to be NP-hard [11], has been widely studied in the sequential setting. For the case of unit costs, a $(1/\alpha_k)$ -approximation algorithm follows from the works of [6, 13, 14, 22, 27], where $\alpha_k = 1 - (1 - 1/k)^k$, which decreases as k increases, and tends to $1 - 1/e$ as $k \rightarrow \infty$. Budgeted maximum coverage can be approximated within factor $e/(e - 1)$ (see, e.g., [18]), and this is the best possible, unless $P = NP$ [8]. In the distributed setting, Subhadrabandhu et al. [25] developed a constant factor approximation algorithm which uses network-wide broadcasts. In contrast, our algorithm avoids spreading information network-wide, because of the large number of rounds that may be required.

2 Partial Information Spreading

The time required for an information spreading algorithm to complete, i.e., for every node to receive every piece of information, has been previously analyzed using the notion of graph conductance [20]:

$$\Phi(G) = \min_{S \subseteq V, |S| \leq n/2} \varphi(S, V),$$

where

$$\varphi(S, V) = \frac{\sum_{i \in S, j \in V \setminus S} P_{i,j}}{|S|} \quad (1)$$

and P is the stochastic matrix associated with the communication of the nodes. Notice that the conductance satisfies $0 \leq \Phi(G) \leq 1$, since for every $i \in S$ we have $\sum_{j \in V \setminus S} P_{i,j} \leq \sum_{j \in V} P_{i,j} = 1$.

As mentioned in [20], this definition differs from the traditional definition of conductance [24]:

$$\Phi(G) = \min_{S \subseteq V, \pi(S) \leq 1/2} \frac{Q(S, V \setminus S)}{\pi(S)},$$

where $\pi(S) = \sum_{i \in S} \pi(i)$, π is the stationary probability vector of the matrix P , and

$$Q(S, V \setminus S) = \sum_{i \in S, j \in V \setminus S} Q(i, j) = \sum_{i \in S, j \in V \setminus S} \pi(i) P_{i,j}.$$

However, for a symmetric stochastic matrix P the definitions are equivalent.³ We can obtain a symmetric matrix for any graph, by taking

$$P_{i,j} = \begin{cases} \frac{1}{d_{max}} & \text{if } (i, j) \in E \\ 1 - \frac{d_i}{d_{max}} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

where d_i is the degree of node i , and $d_{max} = \max_{i \in V} d_i$ is the maximum degree in the graph. This matrix is slightly different than our model of communication in which a node i chooses a neighbor j with probability $1/d_i$. Furthermore, we avoid the assumption that the nodes have knowledge of the value of

³Recall that if P is symmetric then the stationary distribution is uniform.

d_{max} . Nevertheless, for every node i we have $\frac{1}{d_i} \geq \frac{1}{d_{max}}$, which implies that the spreading of information in our model can only be faster than by using the above matrix P . Indeed, let $\tilde{\Phi}(G)$ be the conductance as calculated for the transition matrix \tilde{P} , where

$$\tilde{P}_{i,j} = \begin{cases} \frac{1}{d_i} & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

then the conductance $\tilde{\Phi}(G)$ using the matrix \tilde{P} is at least the conductance $\Phi(G)$ using P . The following lemma is proved in Appendix A.

Lemma 1 *For every graph $G = (V, E)$ we have $\tilde{\Phi}(G) \geq \Phi(G)$.*

The conductance of a graph measures how well it is connected. Consider a clique on all n nodes, which is a well-connected graph. We associate with the graph a stochastic symmetric matrix P where $P_{i,j} = 1/(n-1)$ for every $1 \leq i \neq j \leq n$, and $P_{i,i} = 0$ for every $1 \leq i \leq n$. This implies that the conductance is $\left(\frac{\binom{n}{2}(n-\frac{n}{2})\frac{1}{n-1}}{\frac{n}{2}}\right) / \frac{n}{2} = \frac{n}{2(n-1)}$ which is $\Theta(1)$. On the other hand, a path of n nodes is associated with a matrix P in which $P_{i,j} = 1/2$ for every two neighbors i and j , $P_{i,i} = 1/2$ for the two nodes at the ends of the path, and $P_{i,j} = 0$ otherwise. A path has conductance $(\frac{1}{2}) / \frac{n}{2} = \frac{1}{n}$ and indeed, a path admits many communication bottlenecks. Graphs with small conductance require more rounds of communication to achieve full information spreading.

Since we only require a relaxed spreading guarantee, we introduce the concept of *weak conductance* in order to analyze partial information spreading. While conductance provides a measure of the connectivity of the whole graph, weak conductance measures the *best* connectivity among any partition of the graph into c equal-sized components. Formally, for an integer c such that n/c is an integer we define $\vec{V} = (V_1, \dots, V_c)$ to be a partition of V into c sets of size n/c , i.e., $V_i \subseteq V$ and $|V_i| = n/c$, for every $1 \leq i \leq c$. The weak conductance of a graph $G = (V, E)$ is defined as:

$$\Phi_c(G) = \max_{\vec{V}} \left\{ \min_{1 \leq i \leq c} \left\{ \min_{S \subseteq V_i, |S| \leq |V_i|/2} \varphi(S, V_i) \right\} \right\},$$

where $\varphi(S, V)$ is defined in equation (1). Indeed, in the special case where $c = 1$, the weak conductance of G is equal to its conductance, namely, $\Phi_1(G) = \Phi(G)$.

Before we proceed to use weak conductance to analyze partial information spreading, we show how it can serve as a refined measure of connectivity, by examining several graph classes. First, consider a clique on all n nodes. The weak conductance of a clique is $\left(\frac{\binom{n}{2c}(\frac{n}{c} - \frac{n}{2c})\frac{1}{n-1}}{\frac{n}{2c}}\right) / \frac{n}{2c} = \frac{n}{2c(n-1)}$, which, as the conductance, is also $\Theta(1)$ if c is a constant.⁴ The weak conductance of a path is $(\frac{1}{2}) / \frac{n}{2c} = \frac{c}{n}$, which, as the conductance, is also $\Theta(1/n)$ if c is a constant. For the two examples above, the weak conductance is in the same order as the conductance for some constant $c \geq 1$.

We now give an example of a graph with very small conductance (which is bad for fast information spreading) but a large weak conductance. Since a clique has a large conductance, and a path has a small conductance, we introduce the *c-barbell* graph, which is a generalization of the *barbell* graph, consisting of a path of c cliques, where each contains n/c nodes (see Figure 1). The *c-barbell* graph is associated with the transition matrix P for which $P_{i,j} = 1/\binom{n}{c}$ for every two neighbors, $P_{i,i} = 1/\binom{n}{c}$ for every node i that does not connect two cliques, and $P_{i,i} = 0$ for every node i connecting two cliques. While the conductance

⁴This implies that using weak conductance we obtain stronger guarantees than (δ, c) -spreading.

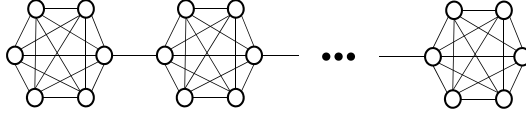


Figure 1: The c -barbell graph is a path of c equal-sized cliques. It is an example of a graph with small conductance and large weak conductance.

of this graph is $(1/(\frac{n}{c})) / \frac{n}{2} = \frac{2c}{n^2}$, the weak conductance is $\left(\left(\frac{n}{2c}\right)\left(\frac{n}{c} - \frac{n}{2c}\right)\frac{1}{n/c}\right) / \frac{n}{2c} = \frac{1}{2}$. For any constant $c \geq 1$, this implies a conductance of $\Theta(1/n^2)$ while the weak conductance improves to $\Theta(1)$.

Indeed, the barbell graph has been studied before [1, 5] as a graph for which information spreading requires a large number of rounds (in [1] the context is random walks, which is closely related, since the path of a message can be viewed as a random walk on the graph). Our definition of weak conductance and relaxed requirement of partial information spreading greatly improve the guarantees that can be obtained for this graph. There are additional families of graphs that have a similar property of small conductance and large weak conductance. Examples include rings of cliques and other structures with c equal-sized well-connected components that are connected by only a few edges. Notice that for a graph to have a large weak conductance, it need not even be connected. For example, a graph consisting of c disconnected cliques has a large weak conductance, but its conductance is equal to zero.

Next, we proceed to the analysis of the partial information spreading algorithm. Recall that in every round, each node i randomly chooses a neighbor j with probability $1/d_i$ and exchanges information with it. Let $G = (V, E)$ be the underlying communication graph, and let \vec{V} be the partition that realizes the weak conductance of G .

Consider a node i and assume w.l.o.g. that $i \in V_\ell$, for some $1 \leq \ell \leq c$. Let $S_i(\tau) \subseteq V_\ell$ denote the set of nodes of V_ℓ that received the message $m(i)$ of node i by round τ , and let X_j be an indicator random variable for the receipt of the message $m(i)$ from a node in $S_i(\tau)$ at a node $j \in V_\ell$, in round $\tau + 1$. Then, for $|S_i(\tau)| \leq n/2c$ we have:

$$\begin{aligned} E[|S_i(\tau + 1)| \mid S_i(\tau)] &= |S_i(\tau)| + \sum_{j \in V_\ell \setminus S_i(\tau)} E[X_j \mid S_i(\tau)] = |S_i(\tau)| + \sum_{k \in S_i(\tau), j \in V_\ell \setminus S_i(\tau)} P_{k,j} \\ &= |S_i(\tau)| \left(1 + \frac{\sum_{k \in S_i(\tau), j \in V_\ell \setminus S_i(\tau)} P_{k,j}}{|S_i(\tau)|} \right) \geq |S_i(\tau)| (1 + \Phi_c(G)). \end{aligned} \quad (2)$$

From here, the analysis proceeds exactly as the proof of Lemma 4 in [20]. The proof considers two phases of the algorithm, the first is while less than $n/2c$ of the nodes received $m(i)$, and the second is until all n/c nodes receive the message. The evolving of $S_i(t)$ in each phase is examined using sub-martingales, for which inequality (2) suffices to carry out the rest of the analysis. Although the analysis is for the number of nodes that receive a message $m(i)$, a similar argument addresses the number of messages that node i receives. This is because we are using a push/pull model of communication, along with a symmetric transition matrix P , which implies that the probability of a node i receiving a message $m(j)$ equals the probability of node j receiving the message $m(i)$. This gives our main result:

Theorem 2 For any $\delta \in (0, 1)$, the number of rounds required for (δ, c) -spreading is $O\left(\frac{\log n + \log \delta^{-1}}{\Phi_c(G)}\right)$.

Notice that the result of Theorem 2 matches the result of [20] for $c = 1$.

For a graph with a constant weak conductance, by taking $\delta = O(1/n)$ we obtain $(1/n, c)$ -spreading in $O(\log n)$ rounds.

We note that the above analysis provides a stronger guarantee than (δ, c) -spreading, since it implies a partition into c sets, where in each set every node receives all the messages sent within that set. For our specific application of an algorithm for maximum coverage, we could obtain an approximation algorithm that exceeds the budget by a factor of at most c , by considering the network as partitioned and solving the problem separately within each set. However, a graph with a large weak conductance may be very different from the c -barbell graph, and hence far from providing such a ‘good’ partition. An example is the clique on all n nodes, which is also a graph with a large weak conductance, but in this case, if each node receives n/c of the messages there is a very small probability that the messages received admit a partition similar to the above example. Therefore, in general, for graphs with large weak conductance we can only aim for each message to be received by n/c nodes (and for each node to receive n/c messages), but not necessarily as a well-structured partition.

3 Distributed Maximum Coverage

In this section we present a distributed algorithm which uses partial information spreading for approximating maximum coverage. The problem that we consider is defined as follows.

Definition 1 *In the distributed maximum coverage problem, each node is given the number of nodes n and the budget K and should return a value in $\{true, false\}$, such that the number of nodes that return true is K and the number of edges that are covered by the nodes that return true is maximized.*

We are interested in bi-criteria (α, β) -approximation algorithms, which exceed the given budget by a factor of α , while guaranteeing a cover that is at least a factor β of an optimal cover with the given budget.

We show a randomized algorithm for maximum coverage, which in expectation obtains an (α, β) -approximation with constant α and β . Later, we show that for values of K that are not too small, e.g., $K = \Omega(\log n)$, these approximation factors are obtained with high probability, and not only in expectation.

We first give some intuition to the difficulty in obtaining an efficient distributed algorithm for maximum coverage. As discussed above, using an information spreading algorithm allows approximating maximum coverage within a constant factor. However, as shown in Section 2, some networks require a large number of rounds to achieve full information spreading, and therefore we wish to avoid it. By allowing only partial information spreading, we can no longer guarantee that a node knows the degrees in the graph, and certainly not the structure of the graph. Knowing only half of the degrees is insufficient, since the unknown degrees may be very large, in which case the node should not choose itself for the cover, or very small, in which case perhaps it should. Even knowledge of the maximal and/or average degrees does not seem to be sufficient.

Nevertheless, we present a constant-approximation algorithm for maximum coverage that uses only partial information spreading. Let Spr be a $(\delta, 2)$ -spreading algorithm with a round complexity of R_{Spr} (e.g., the partial information spreading algorithm in Section 2). The idea is that the nodes use the algorithm Spr to construct a distributed algorithm for solving the maximum coverage problem with a given budget K , by partially spreading their degrees, and at the same time estimating the number of nodes in certain predetermined ranges of degrees. The latter information is then also spread using the algorithm Spr .

For simplicity, we assume that $n = 2^t$ for some integer $t \geq 1$, although our results hold for any value of n . We denote by $m(v)$ the message that node v spreads in algorithm Spr . We assume that every node v always receives its own message $m(v)$. First, we define below the local variables maintained by each node.

We define $t + 1$ sets $D_1, D_2, \dots, D_{\log n + 1}$, that partition the set of nodes according to their degree:

$$D_i = \{v \in V \mid d(v) \in (n/2^i, n/2^{i-1}]\}, \quad i = 1, \dots, \log n + 1.$$

For every i , $1 \leq i \leq t + 1$, we denote by n_i the number of nodes in the set D_i , i.e., $n_i = |D_i|$. The goal of each node v is to obtain good estimates $n_i(v)$ of these sizes, while the initial information a node has is only the number of nodes n , and the budget K allowed for covering. Therefore, initially, $n_i(v) = 1$ if $v \in D_i$, and $n_i(v) = 0$ otherwise.

For our analysis to go through, the actual information that the nodes spread is about the values $\tilde{n}_i(v) = \sum_{j=1}^i n_j(v)$, which are the estimates of $\tilde{n}_i = \sum_{j=1}^i n_j$, rather than the values $n_i(v)$ themselves. To this end, each node v also maintains $t + 1$ static boolean variables $b_i(v)$, for every i , $1 \leq i \leq t + 1$, such that $b_i(v) = 1$ if and only if $v \in \bigcup_{j=1}^i D_j$.

The estimate of a node is updated according to two types of information it gathers. First, the node receives messages from a set of nodes U with the information $\tilde{n}_i(u)$, for $u \in U$. In addition, the node estimates the sum $\sum_{u \in U} b_i(u)$. The estimate $\tilde{n}_i(v)$ will then be updated to the maximum of these values. The pseudocode appears in Algorithm 1.

Algorithm 1 Algorithm for maximum coverage, code for node v

```

1:  repeat 3 times:
2:    run Spr with message  $m(v)$  with the sequence  $\langle b_1(v), \tilde{n}_1(v) \rangle, \dots, \langle b_{t+1}(v), \tilde{n}_{t+1}(v) \rangle$ 
3:    for  $i = 1$  to  $t + 1$ :
4:      estimate  $\hat{n}_i(v) = \sum_{u \in U} b_i(u)$  according to the set of received messages  $U$ 
5:      update  $\tilde{n}_i(v) = \max_{u \in U} \tilde{n}_i(u)$  according to the set of received messages  $U$ 
6:      update  $\tilde{n}_i(v) = \max \{ \tilde{n}_i(v), \hat{n}_i(v) \}$ 
7:    let  $m = i$  such that  $v \in D_i$ 
8:    if  $\tilde{n}_m(v) \leq K$  then return true // return true with probability  $p(v) = 1$ 
9:    else if  $\tilde{n}_{m-1}(v) \leq K$  then return true with probability  $p(v) = \frac{K}{\tilde{n}_m(v)}$ 
10:   else return false // return true with probability  $p(v) = 0$ 

```

The algorithm consists of three iterations, in each of which a node v invokes the information spreading algorithm *Spr* and updates the estimates of the values $\tilde{n}_i(v)$. As our proof will show, since the spreading algorithm promises only that half of the messages are received by each node, we need three iterations of it in order for our guarantees of the maximum coverage algorithm to hold. We also note that more iterations cannot improve these guarantees, since it may be the case that the $(\delta, 2)$ -spreading algorithm induces two disjoint subsets of $n/2$ nodes and each node receives all the messages within its subset.

It is easy to see that the number of rounds of Algorithm 1 is in the same order as the number of rounds of the spreading algorithm *Spr*, since we have three iterations of it. In addition, each message contains $O(\log n)$ variables, each of size $O(\log n)$ bits. Therefore, we get the following round and bit complexity:

Lemma 3 *The round complexity of Algorithm 1 is $O(R_{Spr})$. The bit complexity per message $m(v)$ of a node v is $O(\log^2 n)$.*

We now prove the approximation factors of the algorithm. Throughout the rest of the analysis, we assume that *Spr* obtained the required spreading in all three iterations. This event happens with probability at least $1 - 3\delta$, since *Spr* is a $(\delta, 2)$ -spreading algorithm.

We first bound the expected number of nodes that return *true*. First, the next lemma bounds the expected number of nodes that return *true* in a given set D_i . This bound itself is not enough for guaranteeing a constant deviation from the budget, since the number of sets is $t + 1 = \log n + 1$. However, we will use it later for some of the sets, while the others will be bounded more carefully.

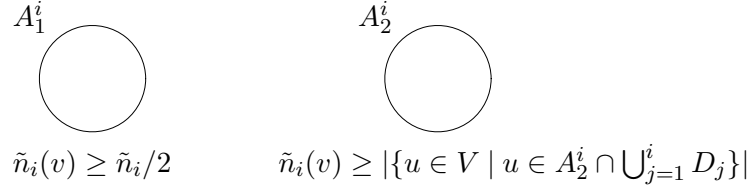


Figure 2: The partition of nodes into the sets A_1^i and A_2^i in Lemma 4.

We use the following notation, which considers the algorithm after the *third* iteration. For every i , $1 \leq i \leq t+1$, we partition the set of all nodes V into two sets A_1^i and A_2^i , such that nodes in A_1^i received a message from some node u with $\tilde{n}_i(u) \geq \tilde{n}_i/2$, and nodes in A_2^i did not.

Lemma 4 *For every i , $1 \leq i \leq t+1$, the expected number of nodes in D_i that return true is at most $3K$.*

Proof: Since Spr is $(\delta, 2)$ -spreading, the first iteration contains at least $\tilde{n}_i \cdot \frac{n}{2}$ messages by nodes with $b_i(u) = 1$. A simple pigeon-hole argument implies that there is a node v^* that receives at least $\tilde{n}_i/2$ out of these messages.

The node v^* estimates $\hat{n}_i(v^*) = \sum_{u \in U} b_i(u)$ according to the set U of received messages. Since v^* receives at least $\tilde{n}_i/2$ messages with $b_i(u) = 1$, we have that $\tilde{n}_i(v^*) \geq \tilde{n}_i/2$ after line 6.

In the second iteration, at least $n/2$ nodes receive the message $m(v^*)$, and therefore at least $n/2$ nodes have $\tilde{n}_i(u) \geq \tilde{n}_i/2$ after line 6.

Now, consider the partition of all nodes after the third iteration into A_1^i and A_2^i . If $v \in D_i$ is in A_1^i then $\tilde{n}_i(v) \geq \tilde{n}_i/2$. Otherwise, let x be the number of nodes in D_i that are in A_2^i . These nodes do not receive in the third iteration any of the messages with $\tilde{n}_i(u) \geq \tilde{n}_i/2$, but each of them still receives at least $n/2$ messages, since our spreading algorithm is $(\delta, 2)$ -spreading. This implies that each node v of D_i which is in A_2^i receives all messages from nodes in A_2^i , and hence has $\tilde{n}_i(v) \geq x$ (see Figure 2).

A node $v \in D_i$ in A_1^i returns *true* with probability at most $\frac{K}{\tilde{n}_i/2}$, unless $\tilde{n}_i(v) < K$, in which case v returns *true* with probability 1. Either all nodes in $v \in D_i \cap A_1^i$ have $\tilde{n}_i(v) \geq K$ and the expected number of nodes in $D_i \cap A_1^i$ that return *true* is at most

$$\tilde{n}_i \cdot \frac{K}{\tilde{n}_i/2} \leq 2K,$$

or there is a node $v \in D_i \cap A_1^i$ for which $\tilde{n}_i(v) < K$, but then $n_i \leq 2K$ and again the expected number of nodes in $D_i \cap A_1^i$ that return *true* is at most $2K$.

A node $v \in D_i$ in A_2^i returns *true* with probability at most $\frac{K}{x}$, unless $x < K$, in which case v returns *true* with probability 1. In the latter case we have at most K nodes in $D_i \cap A_2^i$, and therefore the expected number of nodes in $D_i \cap A_2^i$ that return *true* is at most K . Otherwise, if $x \geq K$ then the expected number of nodes in $D_i \cap A_2^i$ that return *true* is at most $x \cdot \frac{K}{x} = K$.

Therefore, the expected number of nodes in D_i that return *true* is at most $2K + K = 3K$, which completes the proof. \blacksquare

We are now ready to prove the upper bound on the expected number of nodes that return *true*. We denote by ℓ the minimal index such that $\tilde{n}_\ell > 2K$. Ideally, we would like to choose the nodes in D_i for $i < \ell$ and perhaps some of the nodes in D_ℓ such that their total number is K , in order to exceed the budget by no more than a constant fraction of it. We define

$$Bad_i = \{v \in D_i \mid \tilde{n}_{i-1}(v) \leq 2K \text{ and } \tilde{n}_{i-1} > 2K\},$$

D_1	...	D_ℓ	...	D_h	...	D_{t+1}
$\tilde{n}_{\ell-1} \leq 2K$		At most $3K$ return <i>true</i> by Lemma 4	At most K return <i>true</i> by definition of h	At most $3K$ return <i>true</i> by Lemma 4		No node returns <i>true</i>

Figure 3: The bounds on the number of nodes that return *true* in the sets D_i , as proved in Theorem 5.

which is the set of nodes in D_i that estimate that they are in $\bigcup_{j=1}^{\ell-1} D_j$, but are actually not there. These are nodes that may be chosen by the algorithm and exceed the budget K . We wish to bound the number of such nodes to derive a bound on the deviation from the budget K .

Theorem 5 *The expected number of nodes that return true in Algorithm 1 is at most $9K$.*

Proof: From the definition of ℓ , it is clear that the number of nodes that return *true* in $\bigcup_{j=1}^{\ell-1} D_j$ is at most $2K$ (because $2K$ is a bound on the total number of nodes in these sets).

Applying Lemma 4 for $i = \ell$ implies that in D_ℓ there are at most $3K$ nodes that return *true*.

We now define h to be the minimal index such that $|\bigcup_{j=\ell+1}^h \text{Bad}_j| > K$. These are nodes that return *true* from the sets $D_{\ell+1}, \dots, D_h$. By the definition of h , there are at most K nodes in $\bigcup_{j=\ell+1}^{h-1} D_j$ that return *true*.

Again, applying Lemma 4 for $i = h$ implies that in D_h there are at most $3K$ nodes that return *true*.

Finally, we claim that in D_i , for $i > h$, no node returns *true*. This is because either a node $v \in D_i$ is in A_1^ℓ , in which case the node has $\tilde{n}_{i-1}(v) \geq \tilde{n}_\ell(v) \geq \tilde{n}_\ell/2 > 2K/2 = K$ and it returns *false*, or the node $v \in D_i$ is in A_2^ℓ . In the latter case, node v receives a message from every node $u \in A_2^\ell$. Every node u in $\bigcup_{j=\ell+1}^h \text{Bad}_j$ is in A_2^ℓ , otherwise, by the proof of Lemma 4, u has $\tilde{n}_\ell(u) > 2K/2 = K$ and hence u returns *false*, which contradicts the assumption that u is in $\bigcup_{j=\ell+1}^h \text{Bad}_j$. Since v receives a message from every node u in $\bigcup_{j=\ell+1}^h \text{Bad}_j$, we have that $\tilde{n}_i(v) \geq \tilde{n}_h(v) \geq |\bigcup_{j=\ell+1}^h \text{Bad}_j| > K$, where the last inequality follows from the definition of h . Hence, in this case v also returns *false*.

Therefore, in total we have at most $2K + 3K + K + 3K = 9K$ nodes that return *true* (see Figure 3). ■

We now bound the number of edges covered by Algorithm 1. Let s be the minimal index such that $|\bigcup_{j=1}^s D_j| \geq K$. We prove the following lemma in Appendix B.

Lemma 6 *The expected number of nodes in $\bigcup_{j=1}^s D_j$ that return true is at least K and every node in $\bigcup_{j=1}^{s-1} D_j$ returns true.*

The following theorem gives the expected number of edges covered by our solution, and its proof appears in Appendix B.

Theorem 7 *Let ALG be the expected value of the cover obtained by Algorithm 1. Then, $ALG \geq OPT/4$, where OPT is the value of an optimal solution.*

By choosing $\delta = O(1/n)$ the expected approximation factors remain the same, and the above analysis gives the following main theorem that summarizes the properties of Algorithm 1.

Theorem 8 *Algorithm 1 yields in expectation a $(9, 4)$ -approximation to the maximum coverage problem, with a round complexity of $O(R_{Spr})$, and $O(\log^2 n)$ bits per message $m(v)$ of every node v .*

Improvements to the Analysis: The expected approximation factors are proved by analyzing and summing the probabilities $p(v)$ of every node v to return *true*. Since we consider the sum of n independent Bernoulli random variables, we can use a standard Chernoff bound to obtain constant approximation factors for the number of covered edges, as well as for the relative deviation from the budget, K , with a probability of at least $1 - \rho$, where ρ is exponentially small in the expectation, $O(K)$. Indeed, for values of K that are not too small, such as $K = \Omega(\log n)$ (which is still scalable), this implies constant approximation factors *with high probability* rather than in expectation, where high probability refers to probabilities that are $O(1 - \frac{1}{\text{poly}(n)})$.

In addition, we remark that better approximation factors can be obtained by modifying the partition of V to the sets D_i , $i \geq 1$, as follows. For some $\gamma \in (0, 1)$,

$$D_i = \{v \in V \mid d(v) \in (n/(1+\gamma)^i, n/(1+\gamma)^{i-1}]\}, \quad i = 1, \dots, \log_{1+\gamma} n + 1.$$

A respective modification of Theorem 7 now gives an approximation factor of $\beta = 2(1 + \gamma) = 2 + 2\gamma$ for maximum coverage. This implies that our algorithm exhibits a tradeoff between the approximation factor and the size of the messages $m(v)$, as summarized in the next result.

Theorem 9 *For every $\gamma > 0$, there is an algorithm that obtains in expectation a $(9, 2 + 2\gamma)$ -approximation to the maximum coverage problem, with a round complexity of $O(R_{Spr})$ and $O(\log n \cdot \log_{1+\gamma} n)$ bits per message $m(v)$ of every node v .*

When using the partial information spreading algorithm from Section 2 and plugging the round complexity of Theorem 2 along with $\delta = O(1/n)$ into Theorem 9 we obtain:

Corollary 10 *For every $\gamma > 0$, there is an algorithm that obtains in expectation a $(9, 2 + 2\gamma)$ -approximation to the maximum coverage problem, with a round complexity of $O\left(\frac{\log n}{\Phi_2(G)}\right)$ and $O(\log n \cdot \log_{1+\gamma} n)$ bits per message $m(v)$ of every node v .*

For graphs with a constant weak conductance, such as the barbell graph, this implies a scalable number of rounds.

Extensions: If each edge $e \in E$ has a weight $w(e)$, we modify our algorithm to use $w_i = \sum_{\exists j: e=(i,j)} w(e)$ instead of the degree d_i (which corresponds to the case of unit weights).

Moreover, for the budgeted maximum coverage problem, where each node v is associated with some non-negative cost $c(v)$, we can obtain similar approximation factors by modifying the algorithm to scale the probabilities $p(v)$ according to the costs $c(v)$.⁵

For more general instances of maximum coverage, where each element may belong to at most f sets, instead of just two (e.g., an f -hypergraph, in our setting), Algorithm 1 yields an approximation factor of $\beta = 2f$ instead of $\beta = 4$, and the analysis remains the same.

4 Networks Without Node Identities

In some distributed systems, nodes do not possess unique identities. In this section we consider such a model, where each node has some local numbering of its neighbors, but the nodes are not equipped with global identities, and therefore are limited in gaining knowledge about the structure of the network. Thus, even with a full information spreading algorithm it is not clear how to solve the maximum coverage problem.

⁵We give the details in the full version of the paper.

The main issue that arises in this setting is that nodes cannot distinguish between a duplicated message and more than one different message with the same content. For our maximum coverage algorithm to go through, this requires the nodes to use a different method for counting the number of nodes in each category D_i . While updating an estimate $\tilde{n}_i(v)$ to the maximal estimate that node v receives is not affected by the lack of identities, summing the values $\sum_{u \in U} b_i(u)$, where U is the set of nodes from which v receives a message, highly depends on having no duplicates.

We use the framework of Mosk-Aoyama and Shah [20] for computing separable functions, and modify it to fit our partial information spreading rather than the full information spreading assumed there. Instead of sending the values $b_i(v)$, each node v generates exponential random variables $W_i(v)$ with rate $b_i(v)$ (and hence mean $1/b_i(v)$). However, such a value may be equal to 0. To overcome this, if $b_i(v) = 0$ we replace it by a small but positive value $b_i(v) = 1/2n$. The idea is that the minimum of exponential random variables is also an exponential random variable, whose rate is the sum of their rates. Each node now takes the minimal value $W = \min_{u \in U} W_i(u)$ and takes $1/W$ as its estimate of $\sum_{u \in U} b_i(u)$. With some probability, this estimate is close to the correct sum.

To obtain a close estimate with *high probability*, each node generates and sends r variables $W_i^j(v)$, where $1 \leq j \leq r$, with rate $b_i(v)$. A node v then calculates for every j , $1 \leq j \leq r$, the minimum $W^j = \min_{u \in U} W_i^j(u)$ according to the set of received messages U , and takes $est(v) = r/(\sum_{1 \leq j \leq r} W^j)$ as its estimate of $sum_U(v) = \sum_{u \in U} b_i(u)$. The motivation for generating more random variables is to guarantee better bounds on the probability of an estimate that is close to the correct sum; the chosen value of r is determined below.

Formally, we say that an estimate $est(v)$ is *close* to the correct sum $sum_U(v)$ if

$$est(v) \in [(1 - \epsilon)sum_U(v), (1 + \epsilon)sum_U(v)],$$

for some parameter $0 < \epsilon < 1/2$. Provided that the algorithm *Spr* achieves the required spreading, we have that for a given node v the probability that the estimate of v is far from the correct sum is:

$$\Pr(est(v) \notin [(1 - \epsilon)sum_U(v), (1 + \epsilon)sum_U(v)]) \leq O(e^{-O(\epsilon^2 r)}), \quad (3)$$

which for $r = \Theta(\epsilon^{-2} \log \delta^{-1})$ is at most δ (see [20, Lemma 2]).

In the analysis of [20], this also implies that the estimates of *all* nodes are within this range, since it assumes a full information spreading algorithm and therefore all the nodes have the same minimum $\min_{u \in V} W_i^j(u)$. We cannot use the same observation in our case, since different nodes calculate their estimate according to different sets U of received messages. Using the union bound to simply sum these probabilities over all nodes results in a very weak bound on the total probability of good estimates. However, careful inspection of our analysis of Algorithm 1 shows that we need the estimate of $sum_U(v) = \sum_{u \in U} b_i(u)$ to be close to the correct value only in a few cases, as described next, since other nodes update their estimate according to the maximal estimate they receive. We modify Lemma 4 as follows:

Lemma 11 *For every i , $1 \leq i \leq t + 1$, with probability at least $1 - 2\delta$, the expected number of nodes in D_i that return true is at most $(3 + 6\epsilon)K$.*

Proof: We only state the differences from the proof of Lemma 4. With probability at least $1 - \delta$, the node v^* , which obtains at least $\tilde{n}_i/2$ messages from nodes with $b_i(u) = 1$, satisfies inequality (3) and therefore may now have an estimate as small as $(1 - \epsilon)\tilde{n}_i/2$.

In addition, all the nodes in A_2^i receive each other's messages, and therefore we apply the bound in inequality (3) only once to get that with probability $1 - \delta$ every $v \in A_2^i$ has $\tilde{n}_i(v) \geq (1 - \epsilon)x$.

This implies that the expected number of nodes that return *true* in a set D_i is at most $(2/(1 - \epsilon) + 1/(1 - \epsilon))K \leq (3 + 6\epsilon)K$, since $1/(1 - \epsilon) \leq 1 + 2\epsilon$ for $0 < \epsilon \leq 1/2$. ■

Notice that applying Lemma 11 for $i = \ell$ and $i = h$ implies that this adds a term of at most 4δ to our probability of failing to achieve the desired approximation, in addition to the 3δ by the guarantees of *Spr*.

We adjust the definition of ℓ to be the minimal index such that $\tilde{n}_\ell > 2K/(1 - \epsilon)$, and the definition of h to be the minimal index such that $|\bigcup_{j=\ell+1}^h \text{Bad}_j| > K/(1 - \epsilon)$. This induces a bound of $(2(3 + 6\epsilon) + 2(1 + 2\epsilon) + (1 + 2\epsilon))K = (9 + 18\epsilon)K$ in Theorem 5.

For the lower bound on the number of covered edges, we adjust the definition of s to be the minimal value for which $|\bigcup_{j=1}^s D_j| \geq K/(1 + \epsilon)$ (instead of K). Now, in the proof of Lemma 6 we use the sets A_1^i and A_2^i for both $i = s - 1$ and $i = s$. For $i = s - 1$ this implies that every node v in $\bigcup_{j=1}^{s-1} D_j$ has $\tilde{n}_{s-1}(v) \leq (1 + \epsilon) \cdot \tilde{n}_{s-1} < K$, and therefore returns *true*. There are \tilde{n}_{s-1} such nodes. For $i = s$ this implies that every node v in D_s has $\tilde{n}_s(v) \leq (1 + \epsilon) \cdot \tilde{n}_s$. Plugging this into the calculation gives that the number of nodes that return *true* in $\bigcup_{j=1}^s D_j$ is at least $K/(1 + \epsilon)$. This implies another factor of $(1 + \epsilon)$ in the approximation of Theorem 7.

Notice that this adds another term of 4δ to the probability of failing to achieve the desired approximation, hence we have a probability of at least $1 - 11\delta$ for our algorithm to obtain a $(9 + 18\epsilon, 2(1 + \gamma)(1 + \epsilon))$ -approximation for maximum coverage. As before, we choose $\delta = O(1/n)$, which gives:

Theorem 12 *If the nodes in the network do not have identities, there is an algorithm that yields in expectation a $(9 + 18\epsilon, 2(1 + \gamma)(1 + \epsilon))$ -approximation for the maximum coverage problem, with a round complexity of $O(R_{Spr})$, and $O(\epsilon^{-2} \cdot \log^2 n \cdot \log_{1+\gamma} n)$ bits per message $m(v)$ of every node v .*

The smaller we take ϵ , the better the approximation guarantee. However, the cost is in having a large r , which blows up the size of messages sent. If we take a small constant ϵ we get that the approximation factors α, β are still constants, and the size of a message $m(v)$ remains polylogarithmic in n .

We remark that for simplicity of presentation the above analysis only aims to show a constant approximation factor, and that the approximation factors may be improved.

5 Discussion

This paper discusses the notion of partial information spreading and defines the weak conductance of a communication graph as a tool to measure the time needed for partial information spreading. We believe that weak conductance will turn out to be useful in analyzing other properties of graphs as well. An interesting line of research is to relate the weak conductance of a graph to its algebraic properties, as an analogue to the bounds on the conductance, $1 - 2\Phi(G) \leq \lambda_1 \leq 1 - \frac{\Phi(G)^2}{2}$, where λ_1 is the second eigenvalue of the transition matrix [24].

We showed how partial information spreading can be embedded in an approximation algorithm for solving the problem of maximum coverage. It is an open question whether better algorithms exist for this problem.

In addition, as a further research direction we propose the question of achieving other types of partial information spreading, which can be useful in designing distributed algorithms for solving other problems.

Acknowledgments: The authors thank Hagit Attiya and Ariel Kulik for helpful suggestions and comments on an earlier version of this paper.

References

- [1] N. Alon, C. Avin, M. Koucky, G. Kozma, Z. Lotker, and M. R. Tuttle. Many random walks are faster than one. In *SPAA '08: Proceedings of the twentieth annual symposium on Parallelism in algorithms and architectures*, pages 119–128, New York, NY, USA, 2008. ACM.
- [2] D. Amzallag, J. Naor, and D. Raz. Coping with interference: From maximum coverage to planning cellular networks. In *WAOA*, pages 29–42, 2006.
- [3] C. Avin and C. Brito. Efficient and robust query processing in dynamic environments using random walk techniques. In *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 277–286, New York, NY, USA, 2004. ACM.
- [4] C. Avin and G. Ercal. Bounds on the mixing time and partial cover of ad-hoc and sensor networks. In *Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on*, pages 1–12, 2005.
- [5] M. Borokhovich, C. Avin, and Z. Lotker. Tight bounds for algebraic gossip on graphs. *arXiv:1001.3265v1 [cs.IT]*, 2010.
- [6] M. Conforti and G. Cornuejols. Submodular set functions, matroids and the greedy algorithm: Tight worstcase bounds and some generalizations of the rado-edmonds theorem. *Discrete Applied Math*, 7:257–274, 1984.
- [7] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *PODC '87: Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 1–12, New York, NY, USA, 1987. ACM.
- [8] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.
- [9] S. L. Flavio Chierichetti and A. Panconesi. Almost tight bounds for rumour spreading with conductance. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010)*, to appear, 2010.
- [10] S. L. Flavio Chierichetti and A. Panconesi. Rumour spreading and graph conductance. In *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1657–1663, 2010.
- [11] M. R. Garey and D. S. Johnson. “strong” NP-completeness results: Motivation, examples, and implications. *J. ACM*, 25(3):499–508, 1978.
- [12] C. Georgiou, S. Gilbert, R. Guerraoui, and D. R. Kowalski. On the complexity of asynchronous gossip. In *PODC '08: Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pages 135–144, New York, NY, USA, 2008. ACM.
- [13] D. S. Hochbaum. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. In *Approximation algorithms for NP-hard problems*, pages 94–143, Boston, MA, USA, 1997. PWS Publishing Co.
- [14] D. S. Hochbaum and A. Pathria. Analysis of the greedy approach in problems of maximum k-coverage. *Naval Research Logistics*, 45(6):615–627, 1998.

- [15] R. Kannan, L. Lovász, and R. Montenegro. Blocking conductance and mixing in random walks. *Comb. Probab. Comput.*, 15(4):541–570, 2006.
- [16] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 565, Washington, DC, USA, 2000. IEEE Computer Society.
- [17] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, page 482, Washington, DC, USA, 2003. IEEE Computer Society.
- [18] S. Khuller, A. Moss, and J. S. Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999.
- [19] L. Lovász and R. Kannan. Faster mixing via average conductance. In *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 282–287, New York, NY, USA, 1999. ACM.
- [20] D. Mosk-Aoyama and D. Shah. Computing separable functions via gossip. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing (PODC)*, pages 113–122, New York, NY, USA, 2006. ACM.
- [21] D. Mosk-Aoyama and D. Shah. Information dissemination via network coding. In *2006 IEEE International Symposium on Information Theory*, pages 1748–1752, July 2006.
- [22] G. Nemhauser and L. Wolsey. Maximizing submodular set functions: Formulations and studies of algorithms. *Studies on Graphs and Discrete Programming*, pages 279–301, 1981.
- [23] R. G. Shlomi Dolev, Seth Gilbert and C. Newport. Gossiping in a multi-channel radio network (an oblivious approach to coping with malicious interference). In *Proceedings of the 21st International Symposium on Distributed Computing (DISC)*, pages 208–222, 2007.
- [24] A. Sinclair. *Algorithms for random generation and counting: a Markov chain approach*. Birkhauser Verlag, Basel, Switzerland, Switzerland, 1993.
- [25] D. Subhadrabandhu, F. Anjum, S. Kannan, and S. Sarkar. Domination and coverage guarantees through distributed computation. In *Proceedings of 43rd Annual Allerton Conference on Communication, Control and Computing*, Allerton, Monticello, Illinois, September 28-30, 2005.
- [26] K. Suh, Y. Guo, J. Kurose, and D. Towsley. Locating network monitors: Complexity, heuristics, and coverage. *Comput. Commun.*, 29(10):1564–1577, 2006.
- [27] R. V. Vohra and N. G. Hall. A probabilistic analysis of the maximal covering location problem. *Discrete Appl. Math.*, 43(2):175–183, 1993.

A Proof of Lemma 1

Recall that for every graph G , $\Phi(G)$ is the conductance according to the symmetric transition matrix P where

$$P_{i,j} = \begin{cases} \frac{1}{d_{max}} & \text{if } (i,j) \in E \\ 1 - \frac{d_i}{d_{max}} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

and $\tilde{\Phi}(G)$ is the conductance according to the transition matrix \tilde{P} , where

$$\tilde{P}_{i,j} = \begin{cases} \frac{1}{d_i} & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Lemma 1 [restated] *For every graph $G = (V, E)$ we have $\tilde{\Phi}(G) \geq \Phi(G)$.*

Proof: Recall that the stationary distribution π_P is uniform, hence $\pi_P(i) = 1/n$ for every node i . This implies that

$$\begin{aligned} \Phi(G) &= \min_{S \subseteq V, \pi(S) \leq 1/2} \frac{\sum_{i \in S, j \in V \setminus S} \pi(i) P_{i,j}}{\pi(S)} \\ &= \min_{S \subseteq V, |S| \leq n/2} \frac{\sum_{i \in S, j \in V \setminus S} (\frac{1}{n} \cdot \frac{1}{d_{max}})}{|S| \frac{1}{n}} = \min_{S \subseteq V, |S| \leq n/2} \frac{|E(S, V \setminus S)|}{|S| d_{max}}, \end{aligned}$$

where $E(S, V \setminus S) = \{e \in E \mid e = (i, j), i \in S, j \in V \setminus S\}$ is the set of edges of the cut $(S, V \setminus S)$.

On the other hand, the stationary distribution $\pi_{\tilde{P}}$ satisfies $\pi_{\tilde{P}}(i) = d_i/2m$ for every node i , where $m = |E|$. This implies that

$$\begin{aligned} \tilde{\Phi}(G) &= \min_{S \subseteq V, \pi(S) \leq 1/2} \frac{\sum_{i \in S, j \in V \setminus S} \pi(i) P_{i,j}}{\pi(S)} \\ &= \min_{S \subseteq V, \sum_{i \in S} d_i \leq m} \frac{\sum_{i \in S, j \in V \setminus S} (\frac{d_i}{2m} \cdot \frac{1}{d_i})}{\sum_{i \in S} \frac{d_i}{2m}} = \min_{S \subseteq V, \sum_{i \in S} d_i \leq m} \frac{|E(S, V \setminus S)|}{\sum_{i \in S} d_i}. \end{aligned}$$

Therefore, to prove that $\tilde{\Phi}(G) \geq \Phi(G)$, we need to prove that

$$\min_{S \subseteq V, \sum_{i \in S} d_i \leq m} \frac{|E(S, V \setminus S)|}{\sum_{i \in S} d_i} \geq \min_{S \subseteq V, |S| \leq n/2} \frac{|E(S, V \setminus S)|}{|S| d_{max}}. \quad (4)$$

It is easy to see that for any given set $S \subseteq V$ we have

$$\frac{|E(S, V \setminus S)|}{\sum_{i \in S} d_i} \geq \frac{|E(S, V \setminus S)|}{|S| d_{max}},$$

but it is not necessarily the case that the minimum in both expressions in inequality (4) is taken over the same sets $S \subseteq V$. However, if there is a set $S \subseteq V$ for which $\sum_{i \in S} d_i \leq m$ but $|S| > n/2$, then for $\bar{S} = V \setminus S$ we have $|\bar{S}| \leq n/2$, and in addition:

$$\frac{|E(S, V \setminus S)|}{\sum_{i \in S} d_i} \geq \frac{|E(S, V \setminus S)|}{m} \geq \frac{|E(S, V \setminus S)|}{\sum_{i \in \bar{S}} d_i} \geq \frac{|E(S, V \setminus S)|}{|\bar{S}| d_{max}},$$

where the first two inequalities follow from the fact that $\sum_{i \in S} d_i \leq m$. This completes our proof since for every set $S \subseteq V$ taken in the left-hand side of inequality (4) there is a set taken in the right-hand side whose value is at least as small. \blacksquare

B Proofs omitted from Section 3

In this Section we provide the proofs omitted from Section 3. Recall that s is the minimal index such that $|\bigcup_{j=1}^s D_j| \geq K$.

Lemma 6 [restated] *The expected number of nodes in $\bigcup_{j=1}^s D_j$ that return true is at least K and every node in $\bigcup_{j=1}^{s-1} D_j$ returns true.*

Proof: There are less than K nodes in $\bigcup_{j=1}^{s-1} D_j$, and therefore less than K nodes u with $b_{s-1}(u) = 1$. This implies that every node v has $\tilde{n}_{s-1}(v) \leq \tilde{n}_{s-1} < K$. Therefore, every node in $\bigcup_{j=1}^{s-1} D_j$ returns true. The total number of these nodes is \tilde{n}_{s-1} .

Now consider nodes in D_s . A node v in D_s returns true with probability at least $\frac{K}{\tilde{n}_s(v)}$ (if $\tilde{n}_s(v) \leq K$ then v returns true with probability 1). Similar to the previous argument, every node v has $\tilde{n}_s(v) \leq \tilde{n}_s$.

The expected number of nodes in D_s that return true is therefore

$$\sum_{v \in D_s} \frac{K}{\tilde{n}_s(v)} \geq (\tilde{n}_s - \tilde{n}_{s-1}) \frac{K}{\tilde{n}_s} = K - \frac{K\tilde{n}_{s-1}}{\tilde{n}_s},$$

which implies that the expected number of nodes that return true in $\bigcup_{j=1}^s D_j$ is at least

$$\sum_{v \in \bigcup_{j=1}^s D_j} p(v) = \sum_{v \in \bigcup_{j=1}^{s-1} D_j} p(v) + \sum_{v \in D_s} p(v) \geq \tilde{n}_{s-1} + (K - \frac{K\tilde{n}_{s-1}}{\tilde{n}_s}) \geq K,$$

where the last inequality follows from the fact that $\tilde{n}_s \geq K$, by the definition of s . ■

We prove the approximation factor of Algorithm 1, as stated in Section 3.

Theorem 7 [restated] *Let ALG be the expected value of the cover obtained by Algorithm 1. Then, $ALG \geq OPT/4$, where OPT is the value of an optimal solution.*

Proof: We denote by v_1, \dots, v_K the nodes of an optimal solution ordered according to decreasing degrees, and by $u_1, \dots, u_{K'}$ the nodes of the solution of Algorithm 1 ordered according to decreasing degrees. Recall that by Lemma 6 we have that $K' \geq K$. We use the simple observation that the number of edges covered by a set of nodes is at least half of the sum of their degrees (because we count each edge twice in the worst case). We therefore have:

$$\begin{aligned} ALG &\geq \frac{1}{2} \sum_{i=1}^{K'} d(u_i) \geq \frac{1}{2} \sum_{i=1}^K d(u_i) \geq \frac{1}{2} \left(\sum_{i=1}^{\tilde{n}_{s-1}} d(u_i) + \sum_{i=\tilde{n}_{s-1}+1}^K d(u_i) \right) \\ &\geq \frac{1}{2} \left(\sum_{i=1}^{\tilde{n}_{s-1}} d(v_i) + \frac{1}{2} \sum_{i=\tilde{n}_{s-1}+1}^K d(v_i) \right) \geq \frac{1}{4} \left(\sum_{i=1}^K d(v_i) \right) = \frac{1}{4} OPT, \end{aligned}$$

where the inequality on the third line is because all the nodes from the first $s-1$ sets are selected by our algorithm and these are the \tilde{n}_{s-1} nodes with the largest degrees, and the other nodes are all from the set D_s , and therefore have degree at least $n/2^s$ while the largest degree in the set can be at most twice this value.

This gives the desired approximation ratio. ■