Pliable Index Coding: The Multiple Requests Case

Siddhartha Brahma, Christina Fragouli
EPFL, Switzerland

***Abstract*—The Pliable Index Coding problem is a recently proposed new formulation of the Index Coding problem where each client wants *any one message* that it does not have and the server tries to "satisfy" all the clients using the side information sets of each of the clients by broadcasting coded messages. We present two generalizations of the problem. Firstly, we consider the problem of each client requiring any t messages ($t \geq 1$) that it does not have. If the cardinality of their side information sets is the same and there are n messages, then we show that $O(\min(t \log n, t + \log^2 n))$ coded broadcast messages are sufficient. For $t \geq \log n$, this shows a linear dependence on t independent of n. We also develop simple approximation algorithms for the problem and evaluate their performance through simulations. Secondly, we consider the problem of the server having incomplete side information. If the server only knows the size s of the side information sets (assumed to be all equal), we show that there exists a linear code using $\min(s+1, n-s)$ coded messages. We also show that this is tight for linear codes by proving a matching lower bound.** [1]

I. Introduction

Pliable index coding (**PICOD**), introduced in [1], is a variation of the well-known Index Coding with side-information problem (ICOD): In ICOD, a server holds n messages, m clients have as side information (SI) some subset of the n messages, and each client requests from the server *a specific message* she does not have. In PICOD, clients are "pliable", and are happy to receive *any one message* they do not already have. There are several applications that motivate this relaxation: for example, the clients might be doing an internet search, have collected some information e.g. some pages as search results, and are interested in receiving with low delay one more such page; they do not care what specifically they do receive, as long as it is new information they do not already have. We show in [1] that when $m = n$ and the SI sets of each client are of equal size, PICOD requires $O(\log n)$ message transmissions by the server, which is exponentially better than what the application of classical ICOD algorithms would require.

In this paper, we first look at a more general case of the PICOD problem, where each client would like to receive *any t messages* she does not have (in our internet search example, she wants t new pages). We call this problem *Multiple Pliable Index Coding Problem* (**MULT-PICOD**). A simple strategy would be to run t rounds of Pliable Index Coding, where in each round each client learns at least one new message. If the cardinalities of the SI sets are all equal to $s \leq n-t$, this would result in $O(t \log n)$ coded broadcast transmissions from the server. But in the extreme case where $t = n - s$, i.e. the clients want to know all the messages they do not have, a trivial solution using an erasure correcting code [2] only requires $t = n-s$ transmissions, much less than $O(t \log n)$. Thus the simple strategy is clearly not optimal. Indeed, a main contribution of this paper is that only $O(\min(t \log n, t+\log^2 n))$ transmissions are required. This shows that if t is larger than $\log n$, then the number of broadcasts needed grows *linearly* with t and is *independent* of n. We show through simulation results that this behavior can be achieved by simple approximation algorithms that use a greedy strategy.

We then look at a different generalization of PICOD, where now the server knows the cardinality but not the content of the SI sets of the clients. For instance, a server may know that all its clients have received at least s information pages, but does not want the overhead of learning which exactly are these pages for each client. If the number of clients is large compared to the number of messages or sending feedback on the entire side information sets is costly, sending just the size of the side information sets will have a much smaller overhead. This formulation is reminiscent of the erasure coding formulation, where the source would know that a receiver has received a set of coded bits, but not exactly which ones (another way to think of this formulation is that we have $\binom{n}{s}$ clients, with each client observing a different subset of size s). We call this problem *Oblivious Pliable Index Coding Problem* (**OB-PICOD**). We prove in this paper that to ensure each client knows at least $s+1$ messages (i.e. at least one extra message) the server needs to make $\min(s+1, n-s)$ broadcast transmissions using linear codes. We provide very simple constructions that achieve this. We also give a lower bound agreement that shows that with linear encoding and decoding this is the best we can achieve.

The remainder of the paper is organized as follows. After a brief survey of work on different formulations of index coding in Section II, Section III introduces the precise definition of our problems and notation used in the paper. In Section IV, we prove tight upper and lower bounds for the OB-PICOD problem. In Section V, we prove upper bounds on the number of broadcasts needed to solve an instance of MULT-PICOD. In Section VI and VII, we develop simple approximation algorithms for MULT-PICOD and illustrate their practical performance (in particular the linear dependence on t) through simulation results on random instances of the problem.

II. Related Work

The problem of Index Coding with Side Information was introduced by Birk *et. al.* [3] in the context of an application

[1]This work was supported by the European Research Council grant NOWIRE ERC-2009-StG-240317.

in satellite communication networks. Bar-Yossef *et.al.* [4] presented the first theoretical analysis of the problem and introduced the *minrank* parameter. Since then, there has been a large amount of work proving new bounds for both linear and non-linear index codes [5]–[7]. Recently techniques from interference alignment have been used to analyse index codes [8]. Various special cases and extensions have also been studied like the complimentary index coding problem [9] and index codes with near extreme rates [10]. The computation of index codes in presence of error [11] and secure index codes [12] have also been looked at.

The formulation of Pliable Index Coding was introduced in [1]. In it the authors show that the problem is NP-Hard and prove universal upper bounds logarithmic in the size of the message set. The so called bipartite index coding problem is analysed in [13], [14] where multiple clients may "want" a specific message.

III. Problem Definition

We will assume that the messages (which are bits) are embedded in a field $(\mathcal{F}, +, \cdot)$ that is large enough (this will be clarified later) and all the encodings are linear. The server has n information messages $b_1, \cdots, b_n$ and there are n clients $c_1, \cdots, c_n$. Each client c_i knows a subset of messages $b_{N_c[i]}$, where $N_c[i]$ is a strict subset of $[n]$. Here $b_{N_c[i]}$ denotes the set $\{b_j, j \in N_c[i]\}$ and $[n] = \{1, 2, \cdots, n\}$. Thus $N_c[i]$ represents the indices of the messages that client c_i has as SI. The *Oblivious Pliable Index Coding Problem* (**OB-PICOD**) models a situation where the server has limited information about the SI sets $N_c[i]$. More concretely, we assume the cardinalities of the SI sets are same i.e. $|N_c[i]| = s$ for some $0 \leq s < n$ and the server knows only s. The problem then is to construct a set of coded messages which can be broadcast to the clients such that each client can decode at least one message that it does not have. Thus given n and s the OB-PICOD problem is to devise a minimum length linear code $\mathcal{C}_O$ which consists of

1) A linear encoding function E mapping $x \in \mathcal{F}^n$ to $E(x) \in \mathcal{F}^l$, where l is the length of the code.
2) Decoding functions $D_1, \cdots, D_n$ for the n clients such that $D_i(E(x), b_{N_c[i]}) = b_{k_i}$ for *some* $k_i \in \overline{N_c[i]} = [n] \setminus N_c[i]$ for *any* SI sets $N_c[i]$ such that $|N_c[i]| = s$ for all $i \in [n]$.

Note that since the server does not have the exact SI sets, an encoding scheme should be able to deal with all possible SI sets of size s the clients may have.

In the *Multiple Pliable Index Coding Problem* (**MULT-PICOD**), each client wants to learn *any t extra messages* that it does not have. In this case, the server does know the SI sets of each client. This generalizes the PICOD problem which corresponds to the case $t = 1$. Assuming that $|N_c[i]| \leq n - t$, (**MULT-PICOD**) is to devise a minimum length linear code $\mathcal{C}_M$ which consists of

1) A linear encoding function E mapping $x \in \mathcal{F}^n$ to $E(x) \in \mathcal{F}^l$, where l is the length of the code.
2) Decoding functions $D_1, \cdots, D_n$ for the n clients such that $D_i(E(x), b_{N_c[i]}) = \{b_{k_{i,1}}, \ldots, b_{k_{i,t}}\}$ for *some distinct* $k_{i,1}, \ldots, k_{i,t} \in \overline{N_c[i]} = [n] \setminus N_c[i]$.

In proving the results claimed in this paper, we will be using the following result from linear algebra.

Lemma 3.1: Given n_0 symbols in a large enough field $\mathcal{F}$ there exist $k_0 \leq n_0$ linear combinations of the symbols in $\mathcal{F}$ such that any subset of $k_1 \leq k_0$ symbols can be recovered if the remaining $n_0 - k_1$ symbols are known.

In fact if the field size is greater than n_0, k_0 random linear combinations will have the above property with high probability [15]. This will be a key ingredient in our proofs, so we will assume that the field $\mathcal{F}$ in which the messages reside is of size greater than n (the number of messages).

IV. Tight Bounds for the OB-PICOD Problem

As mentioned above, in the OB-PICOD problem the server only knows the size of the SI sets which are assumed to be the same for all the clients. The optimal code length can be upper bounded as follows.

Lemma 4.1: If all the SI sets are of cardinality s i.e. $|N_c[i]| = s$ for $i \in [n]$, then $\min(s+1, n-s)$ coded messages are sufficient to satisfy all the clients in OB-PICOD.

Proof: There are two ways to make sure that all the clients are able to decode at lease one message that it does not have. Select any $s + 1$ messages and send them uncoded, one at a time. Since all the SI sets are of size s, there will always exist at least one message that a client will not have. As another strategy, consider $n - s$ linear combinations of all the symbols obtained by the application of Lemma 3.1. Since each client knows s symbols, it can recover at least one (in fact all) of the remaining symbols. Taking the minimum of the two strategies, we get an upper bound of $\min(s + 1, n - s)$ messages. ■

In particular when s or $n-s$ is a constant, the server can satisfy all the clients using only a constant number of broadcasts, without even knowing the exact SI sets.

We now show that this bound is tight for linear encoding and decoding. For this we will need some notation. Let $e_1, \ldots, e_n$ be the unit vectors in $\mathcal{F}^n$ i.e. e_i has a 1 (the identity element in $\mathcal{F}$) in the i-the position and 0 (the zero element in $\mathcal{F}$) elsewhere. Thus they form an orthogonal basis for $\mathcal{F}^n$. Since the server uses linear encodings, the j-th encoded message can be represented as a dot product of an encoding vector $\mathbf{A}_j$ and the message vector $\mathbf{b} = \{b_1, \ldots, b_n\}$, with operations done over the field $\mathcal{F}$. For l encoded messages , we will have the corresponding l encoding vectors $\mathbf{A} = \{\mathbf{A}_1, \ldots, \mathbf{A}_l\}$. We will denote the vector space spanned by the vectors in $\mathbf{A}$ by $Span(\mathbf{A})$. For a subset of indices $B \subseteq [n]$, e_B will denote the corresponding set of unit vectors $e_i, i \in B$.

Lemma 4.2: For linear encoding and decoding, any scheme for OB-PICOD will require at least $\min(s+1, n-s)$ messages.

Proof: The knowledge of the SI set can be expressed equivalently by the fact that the client can compute any vector in the span of $\mathbf{S} = \{e_{\alpha_1}, \ldots, e_{\alpha_s}\}$ where $\alpha_1, \ldots, \alpha_s$ are the indices of the messages that is in the SI set. For a client to be able to decode the i-th message using its SI sets and the encoded messages it is easy to see that e_i should belong to the span of $\mathbf{A} \cup \mathbf{S}$ (For a proof see [4]).

$$e_i \in Span(\mathbf{A} \cup \mathbf{S}) \tag{1}$$

Assume that client i decodes the β_i-th message. Now consider the set of all unique message indices that are decoded by the clients and call it B. We claim that $|B| \geq s+1$. Indeed, if $|B| \leq s$ then a client may have as SI set the messages corresponding to the indices in B (and maybe a few extra). But then, it cannot decode a message that it does not have as B was assumed to contain all the decoded message indices.

For simplicity, assume that $|B| = s+1$, $s+1 \leq n-s$ and w.l.o.g $B = \{1, 2, \ldots, s+1\}$. Let $\mathbf{A}_\perp$ be the projection of $\mathbf{A}$ onto the first $s+1$ coordinates i.e. the ones corresponding to B. Consider the following possible SI sets.

$$\mathbf{S}_1 = \{e_{\alpha_{1,1}}, \ldots, e_{\alpha_{1,s}}\}$$

where each of $\alpha_{1,1}, \ldots, \alpha_{1,s} \notin B$ (such a set exists as $s+1 \leq n-s$). Clearly, by the condition of decodability of a new message (Eq. 1), at least one vector in e_B should be in $Span(\mathbf{A} \cup \mathbf{S}_1)$. Again w.l.o.g let it be e_1. Since the $\alpha_{1,i}$ indices are disjoint from B, this implies that $e_{1_\perp} \in Span(\mathbf{A}_\perp)$. Now consider the SI set

$$\mathbf{S}_2 = \{e_1, e_{\alpha_{2,1}}, \ldots, e_{\alpha_{2,s-1}}\}$$

where $\alpha_{2,1}, \ldots, \alpha_{2,s-1} \notin B$. In this case an application of Eq. 1 implies that $e_2 \in Span(\mathbf{A} \cup \mathbf{S}_2)$ (w.l.o.g). Since e_2 is orthogonal to all the vectors in $\mathbf{S}_2$, $e_{2_\perp} \in Span(\mathbf{A}_\perp)$. We can continue this argument for a total of $s+1$ steps where the last SI set is

$$\mathbf{S}_{s+1} = \{e_1, \ldots, e_s\}$$

and we can derive $e_{s+1_\perp} \in Span(\mathbf{A}_\perp)$. Since the vectors in $e_{1_\perp}, \ldots, e_{s+1_\perp}$ are orthogonal and all of them lie in $Span(\mathbf{A}_\perp)$, $\mathbf{A}_\perp$ and hence $\mathbf{A}$ must contain at least $s+1$ linearly independent vectors.

The other case where $s+1 > n-s$ can be handled in a similar manner, where instead of $s+1$ sets, we will have $n-s$ SI sets for which the new messages will correspond to orthogonal vectors. Combining the two, we conclude that $\mathbf{A}$ must contain at least $\min(s+1, n-s)$ linearly independent vectors, which implies at least $\min(s+1, n-s)$ broadcasts need to be made. Finally, the assumption of $|B| = s+1$ can be removed by simple choosing the first $s+1$ of them. ■

Combining Lemma 4.1 and Lemma 4.2, we have

Theorem 4.3: For the OB-PICOD problem, if each client knows exactly s messages, then the server needs to broadcast at most $\min(s+1, n-s)$ messages to satisfy all the clients. For linear encoding and decoding, this is tight.

V. Upper Bounds for the MULT-PICOD Problem

It was shown in [1] that PICOD is NP-Hard. This immediately implies that finding the optimal code in MULT-PICOD is also NP-Hard. However, we will show in this section that tight upper bounds can be proven for the size of the codes. We assume that the server knows the SI sets and each client now wants to know t extra messages that it does not have. We can visualize an instance of MULT-PICOD using a bipartite

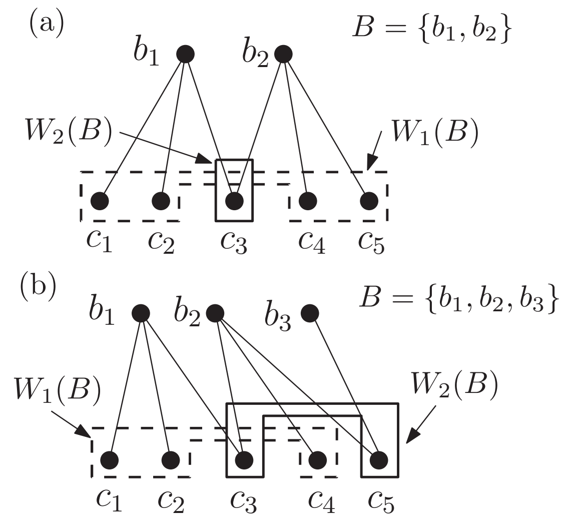

Fig. 1. A subgraph showing the coding scheme

graph G with n vertices on one side representing the messages (termed as "message vertices") and n vertices on the other side representing the clients (termed as "client vertices"). We will identify the vertices by the messages or clients they represent. There is an edge from b_j to c_i if $j \in \overline{N_c[i]}$. In Figure 1(a) shown above, b_1 is not in the SI sets of clients c_1, c_2, c_3 and hence is connected to them in G. In what follows, we will denote the neighborhood of c_i in G by $N[c_i]$ and its degree by $d(c_i) = |N[c_i]|$. We similarly define $N[b_j]$ and $d(b_j) = |N[b_j]|$.

Given a subset of message vertices (called m-vertices from here on) B, we can categorize the client vertices (called c-vertices from here on) in the neighbourhood of B according to the number of message vertices they are adjacent to. The set of c-vertices that are adjacent to exactly i m-vertices in B, is denoted by $W_i(B)$. In Figure 1(a), for $B = \{b_1, b_2\}$, $W_1(B) = \{c_1, c_2, c_4, c_5\}$ and $W_2(B) = \{c_3\}$. Note that if $b_1 + b_2$ is sent to the 4 vertices in $W_1(B)$, each of them can decode a message it does not have: c_1 and c_2 can decode b_1 as they know b_2; similarly, c_4 and c_5 can decode b_2 as they know b_1. Thus a single linear combination of the messages in B can satisfy all the c-vertices in $W_1(B)$. Similarly, two independent linear combinations of b_1, b_2 can be used by c-vertices in $W_2(B)$ to decode two messages.

In another example, consider the graph in Figure 1(b). In this case $B = \{b_1, b_2, b_3\}$, $W_1(B) = \{c_1, c_2, c_4\}$ and $W_2(B) = \{c_3, c_5\}$. Similar to above, if $b_1 + b_2 + b_3$ is sent to the 3 vertices in $W_1(B)$, each of them can decode a message it does not have. For the two vertices in $W_2(B)$, notice that they are adjacent to two different subsets of m-vertices. In this case, we can apply Lemma 3.1 to get two linear combinations such that each of c_3, c_5 can decode two messages it does not have. For any B, define $t(B) = \max\{i : \text{ s.t. } W_i(B) \neq \emptyset\}$. In general we have

Lemma 5.1: For any B there exists a set of $t(B)$ linear combinations of the messages in B such that each c-vertex in $W_i(B)$ can decode i messages it does not have, for $i \in [t(B)]$.

Our approach then will be to select a suitable subset of m-vertices such that a *large* number of c-vertices have approximately t m-vertices adjacent to them. To this end, we will use a probabilistic argument. The following main lemma proves an upper bound for the case when the SI sets have the same cardinality.

Lemma 5.2: Let $d(c_i) = d$ for $i \in [n]$. Then there is a code of length $O(\min(t \log n, t + \log^2 n))$ such that each c-vertex can decode t messages it does not have for $t \leq d$.

Proof: Randomly select a subset B_1 of m-vertices by selecting each vertex with probability $p = \frac{t}{d}$. Fix a particular c-vertex c_1 (w.l.o.g) with degree d in the graph. Let X_i denote the indicator variable which is 1 if the i-the neighbour of X_i is present in the sample B_1, for $i \in [d]$. Clearly, the X_i's are i.i.d Bernoulli random variable with $P(X_i = 1) = p$. Let $X = X_1 + \ldots + X_d$. Then we have

$$E[X] = E[\sum_{i=1}^{d} X_i] = \sum_{i=1}^{d} E[X_i] = dp = t$$

We will also need concentration bounds on $E[X]$. Since X is a sum of i.i.d Bernoulli random variables, we can use the following Chernoff bounds which are valid for any $\epsilon \in (0, 1)$ [16].

$$P(X < (1-\epsilon)E[X]) \leq e^{-\frac{\epsilon^2}{2}E[X]}$$

and

$$P(X > (1+\epsilon)E[X]) \leq e^{-\frac{\epsilon^2}{3}E[X]}$$

Assume that $t \geq 24 \log n$. If we choose $\epsilon = \sqrt{\frac{6 \log n}{t}}$, then clearly $\epsilon \leq \frac{1}{2}$. Then, substituting the values of $E[X] = t$ and ϵ, we have

$$\begin{aligned} P(X > 3E[X]/2) \leq P(X > (1+\epsilon)E[X]) &\leq e^{-\frac{\epsilon^2}{3}E[X]} \\ = e^{-\frac{6t \log n}{3t}} = e^{-2 \log n} &= n^{-2} \end{aligned}$$

Thus we can conclude that

$$P(X > 3t/2) \leq n^{-2}$$

Similarly,

$$\begin{aligned} P(X < E[X]/2) \leq P(X > (1-\epsilon)E[X]) &\leq e^{-\frac{\epsilon^2}{2}E[X]} \\ = e^{-\frac{6t \log n}{2t}} = e^{-3 \log n} &= n^{-3} \end{aligned}$$

We can conclude that

$$P(X < t/2) \leq n^{-3}$$

Combining the above two results, we have

$$P(\frac{t}{2} \leq X \leq \frac{3t}{2}) \geq 1 - \frac{1}{n^2} - \frac{1}{n^3}$$

which implies that with high probability a particular c-vertex has between $t/2$ and $3t/2$ adjacent m-vertices in B_1. Therefore, the expected number of c-vertices having the same property is at least $n - 1/n - 1/n^2$. By the probabilistic method there is at least one subset B_1 for which the expected value is reached or surpassed. This implies that there is a subset B_1 such that all c-vertices have between $t/2$ and $3t/2$ m-vertices adjacent to them. By an application of Lemma 5.1, there is a set of $3t/2$ linear combinations of the corresponding messages such that each c-vertex can decode at least $t/2$ messages it does not have. Note that if $3t/2 > d$, then number can be cut off at d.

What this shows is that if $t \geq 24 \log n$, then by using $O(t)$ broadcast messages we can make sure all the c-vertices learn at least $t/2$ new messages. We can now recursively use the same argument for a situation where each client now needs (at most) $t' = t/2$ new messages, stopping when t' becomes less than $24 \log n$. In the case when $t < 24 \log n$, we can run t rounds of the normal Pliable Index Coding algorithm proposed in [1]. In the specific case when all the c-vertices have same degree, it was shown that using $O(\log n)$ coded messages it is possible for each c-vertex to know one new message. Thus if $f(n, t)$ is the number of messages required for sending t unknown messages to each client, we get the following recurrence

$$f(n,t) \leq \begin{cases} f(n, \frac{t}{2}) + O(t) \text{ if } t \geq 24 \log n \\ O(t \log n) \text{ otherwise} \end{cases}$$

This recurrence can easily be solved to get the required bound of $f(n, t) = O(\min(t \log n, t + \log^2 n))$. ■

The above lemma can be extended in the following way, whose proof goes along the same lines as above and is omitted.

Lemma 5.3: Let $d_{max} = \max\{d(c_i), i \in [n]\}$ and $d_{min} = \min\{d(c_i), i \in [n]\}$. If $d_{max}/d_{min} \leq 2$ then there is a code of length $O(\min(t \log n, t + \log^2 n))$ such that each client can decode t messages it does not have.

For the general case of c-vertices having arbitrary degrees, we can partition them into at most $\log n$ groups such that the minimum and maximum degrees of the c-vertices in each group are within a factor of 2. For each group, we can use the above result and hence derive the following bound.

Theorem 5.4: For any MULT-PICOD instance with n message and n c-vertices, there exists a code of length $O(\min(t \log n, t + \log^2 n) \log n)$ such that all the c-vertices can decode t new messages it does not have.

For random instances of MULT-PICOD we have

Theorem 5.5: For a random MULT-PICOD instance where an edge between b_j and c_i in G exists with a constant probability q, for a large enough n, there exists a code of length $O(\min(t \log n, t + \log^2 n))$ such that all the c-vertices can decode t new messages it does not have, almost surely.

Proof: By the law of large numbers, the degree of each c-vertex is concentrated near the mean nq. That is, for a large enough n, almost surely $d(c_i) \in [n(q-\epsilon), n(q+\epsilon)]$, for any $\epsilon > 0$. If we select an $\epsilon < q/3$, almost surely the ratio of the maximum and minimum degrees is ≤ 2. Then the claim follows from Lemma 5.3. ■

VI. Approximation Algorithms

In this section, we propose a simple greedy algorithm for MULT-PICOD. It is a generalization of the *GRCOV* algorithm presented in [1]. We will work in the field $\mathcal{F} = GF(2)$. Rather than trying to find a set of m-vertices B such that $\sum_{i=t/2}^{t} |W_i(B)|$ is maximized as suggested by the upper bound arguments in the previous section, we will just try to maximize $|W_1(B)|$. After finding a particular B for which $|W_1(B)|$ is large we can use the sum of the messages in B to satisfy the c-vertices in $W_1(B)$. We also maintain a counter

$CNT[i]$ for each c-vertex i to keep track of the remaining number of messages they want.

Rather than trying to obtain B such that $|W_1(B)|$ is *maximum*, we greedily find a *maximal* such set. Let $B = \{b_{v_1}, \cdots, b_{v_j}\}$ be a set of m-vertices. B is a maximal set if for any vertex $b_{v_{j+1}} \notin B$, $|W_1(B \cup \{b_{v_{j+1}}\})| < |W_1(B)|$. To find a maximal set, we start with the null set and keep on adding m-vertices that greedily maximize $|W_1(B)|$ in each step; we stop when no further additions are possible without decreasing $|W_1(B)|$.

Once such a maximal set B_M is obtained, an encoded message consisting of the sum of all the messages in B_M is broadcast. The edges connecting the vertices in $W_1(B_M)$ to the m-vertices it can decode are removed and the value of $CNT[i]$ for each vertex in $W_1(B_M)$ is also reduced by one. The algorithm is resumed for the remaining graph, until all the $CNT[i]$ values go to zero. We call this *MULT-GRCOV*. It is shown in pseudo-code format below and a simple implementation of the algorithm has a running time of $O(n^3)$.

Algorithm 1 MULT-GRCOV(G, n, t)

Init: G is a MULT-PICOD instance with n c- and m-vertices.
Init: $\mathcal{C} = \{\}$, $CNT[i] = t$ for $i \in [n]$.
  while $\exists i$ s.t. $CNT[i] \neq 0$ **do**
    $B \leftarrow \varnothing$.
    while B is not a *maximal set* **do**
      Find message vertex b_v such that $|W_1(B \cup \{b_v\})|$ is maximized.
      $B \leftarrow B \cup \{b_v\}$.
    end while
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\sum_{u=1}^{|B|} b_{v_u}, b_{v_u} \in B\}$.
    for $c_i \in W_1(B)$ **do**
      If c_i is able to decode $b_j \in B$ using the above encoding, then delete the corresponding edge in G.
      $CNT[i] \leftarrow CNT[i] - 1$.
    end for
  end while
  Output $\mathcal{C}$.

VII. Experimental Results

In this section, we present results of extensive simulations on random instances of MULT-PICOD to evaluate the performance of the algorithm presented in the previous section. We generate instances of MULT-PICOD by randomly choosing each edge of G with probability 0.5 i.e. a client knows a particular message with probability 0.5. Then, for a given value of n (number of clients/messages) and t (number of messages each client wants to know) we run *MULT-GRCOV* several times and take an average of the length of the code the algorithm produces. If in an instance, the size of the SI set for a particular c-vertex is s and $t < n - s$, then $CNT[i]$ is initialized to $n - s$. The plot is shown in Figure 2.

In the figure, notice that for each n, after a short initial phase, the average number of bits required grows linearly

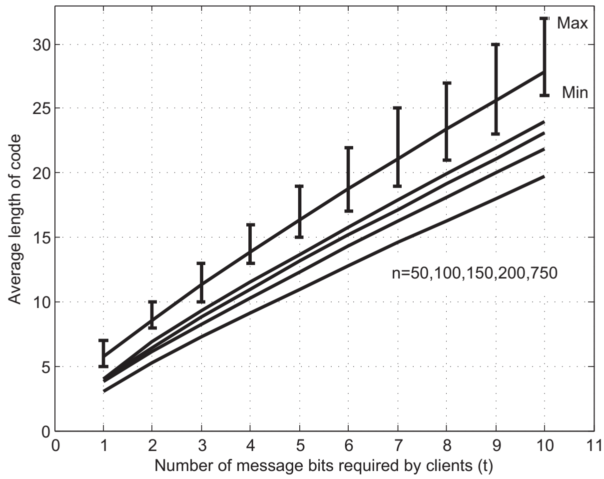

Fig. 2. Performance of *MULT-GRCOV* for varying n and t.

with t. This matches the trend expected from Theorem 5.5. Further, the contours representing different values of n are approximately parallel i.e. the slopes are independent of n. This also matches the trend suggested by a $O(t+\log^2 n)$ bound for large enough t. The error bars at the top, that represents the maximum (and minimum) length of the code encountered for random instances corresponding to $n = 750$, also shows an approximately linear trend.

References

[1] S. Brahma and C. Fragouli, "Pliable index coding," in *Proc. of IEEE ISIT*, 2012, pp. 2251–2255.
[2] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, 1997.
[3] Y. Birk and T. Kol, "Informed-source coding-on-demand (iscod) over broadcast channels," in *Proc. of IEEE INFOCOM*, vol. 3, 1998, pp. 1257–1264.
[4] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," in *Proc. of IEEE FOCS*, 2006, pp. 197–206.
[5] N. Alon, E. Lubetzky, U. Stav, A. Weinstein, and A. Hassidim, "Broadcasting with side information," in *Proc. of IEEE FOCS*, 2008, pp. 823–832.
[6] E. Lubetzky and U. Stav, "Nonlinear index coding outperforming the linear optimum," *IEEE Transactions on Information Theory*, vol. 55, no. 8, pp. 3544–3551, 2009.
[7] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Index coding via linear programming," *arXiv preprint arXiv:1004.1379*, 2010.
[8] H. Maleki, V. Cadambe, and S. Jafar, "Index coding: An interference alignment perspective," in *Proc. of IEEE ISIT*, 2012, pp. 2236–2240.
[9] M. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg, "On the complementary index coding problem," in *Proc. of IEEE ISIT*, 2011, pp. 244–248.
[10] S. Dau, V. Skachek, and Y. Chee, "Optimal index codes with near-extreme rates," in *Proc. of IEEE ISIT*, 2012, pp. 2241–2245.
[11] ——, "Index coding and error correction," in *Proc. of IEEE ISIT*, 2011, pp. 1787–1791.
[12] ——, "On secure index coding with side information," in *Proc. of IEEE ISIT*, 2011, pp. 983–987.
[13] A. Tehrani, A. Dimakis, and M. Neely, "Bipartite index coding," in *Proc. of IEEE ISIT*, 2012, pp. 2246–2250.
[14] M. Neely, A. Tehrani, and Z. Zhang, "Dynamic index coding for wireless broadcast networks," in *Proceedings of IEEE INFOCOM*, 2012, pp. 316–324.
[15] C. Fragouli and E. Soljanin, "Network coding fundamentals," *Foundations and Trends in Networking*, vol. 2, no. 1, 2007.
[16] D. Dubhashi and A. Panconesi, *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2012.