Abstract

Motivated by applications in distributed storage and distributed computation, we introduce embedded index coding (EIC). EIC is a type of distributed index coding in which nodes in a distributed system act as both broadcast senders and receivers of information. We show how embedded index coding is related to index coding in general, and give characterizations and bounds on the communication costs of optimal embedded index codes. We also define task-based EIC, in which each sending node encodes and sends data blocks independently of the other nodes. Task-based EIC is more computationally tractable and has advantages in applications such as distributed storage, in which senders may complete their broadcasts at different times. Finally, we give heuristic algorithms for approximating optimal embedded index codes, and demonstrate empirically that these algorithms perform well.

1 Introduction

1.1 Motivation

In index coding, defined by Bar-Yossef, Birk, Jayram and Kol in [3], sender(s) encode data blocks into messages which are broadcast to receivers. The receivers already have some of the data blocks, and the goal is to take advantage of this “side information” in order to minimize the number of messages broadcast. For example, if node $r_1$ knows a data block $b_1$ and node $r_2$ knows block $b_2$, a sender $S$ can broadcast $b_1 \oplus b_2$. Then $r_1$ can cancel out $b_1$ and $r_2$ can cancel $b_2$ such that both nodes learn a distinct new block from a single broadcast message.

Index coding is typically studied in the models depicted in Figures 1a and 1b, where the senders are distinct from the receivers. In this paper, we consider a setting—depicted in Figure 1c—where the senders are the receivers. This is similar to a “peer-to-peer” network model, but in this setting nodes are always communicating by broadcasting to the full network, rather than communicating with each other directly. This model is motivated by applications in distributed storage and distributed computation. For example, in coded computation, e.g. [19], the shuffle phase consists of nodes communicating computed values with each other.

We call this model embedding index coding (EIC). EIC can be seen as a special case of the multi-sender index coding model in Figure 1b. In this paper, we will demonstrate that by considering EIC as a special case, we can prove new results and design faster algorithms than are available for the more general multi-sender index coding problem.

We also introduce a new notion of solution to an embedded index coding problem called a task-based solution. In a task-based solutions, the communication can be partitioned into independent tasks, so that each receiver is only reliant on a single sender to get a particular block. This can be seen as a generalization...
Figure 1: Communication model for (a) centralized index coding with sender $S$, receivers $r_1, ..., r_5$; (b) general multi-sender index coding with senders $s_1, ..., s_4$ and receivers $r_1, ..., r_5$; and (c) embedded index coding, a special case of (b) with joint sender and receiver nodes $r_1 = s_1, ..., r_5 = s_5$.

of *Instantly Decodable Network Codes* [14] which have been studied with similar motivation (see Remark 1). Task based solutions are also related to *Locally Decodable Index Codes* [24] (see Remark 2).

As we will see, there are efficient heuristics to find good task-based solutions to EIC problems. Moreover, task-based solutions can be more robust to failures or delays: if a sender’s messages are corrupted or lost, the messages from other senders can still be used fully to decode data blocks.

1.2 Outline and Contributions

In Section 2 we review related work in more detail. In Section 3 we formally define the EIC problem and several notions of solution. In Section 4 we show how EIC problems relate to more general index coding and we analyze how different notions of solutions are related. In Section 5 we provide algorithms for approximating optimal EIC solutions and demonstrate empirically that they perform well.

Our contributions can be summarized as follows.

1. We define *embedded index codes*, a type of distributed index code in which nodes function as both broadcast senders and receivers.

2. We define *task-based index coding*, which seems more computationally tractable than a general solution to an EIC problem, and can be thought of as relaxing the concept of *instantly decodability* in network codes.

3. We prove several results establishing relationships between centralized (single-sender) index coding, EIC, and task-based EIC. In particular, we show that the optimal communication for a general EIC problem is only a factor of two worse than the optimal communication in the centralized model; we give characterizations and bounds for the optimal communication cost of the best task-based solutions to an EIC problem; and we show separations between the three models.

4. Based on the (proofs of) the bounds described above, we design heuristics for designing general EICs and and task-based EICs. We give empirical evidence that these approximation algorithms perform well.

2 Related Work

In this section we briefly review related work. Index coding was first introduced by [3], based on the Informed-Source Coding on Demand (ISCOD) model proposed by [1], and many extensions and variations have been
studied, including non-linear index coding [20] and multi-sender index coding [26]. We focus on linear index coding, where the messages broadcast are linear combinations of the original data.

The work of [3] characterized the number of broadcasts required to solve an index coding problem in terms of the minrank (c.f. Definition [3]) of a relevant graph. The minrank is difficult to compute exactly, and a number of approximations and heuristics have been studied for computing optimal linear index codes [6, 7, 25, 29, 31, 32]. We will also use the minrank, and heuristics for computing it, in our approach.

Embedded index codes are a special case of the linear multi-sender index codes in [13] and [17], which both consist of multiple senders and multiple receivers, but as two distinct and non-overlapping sets of nodes; this is the setting depicted in Figure 1b. In [17], rank minimization is used in an approach similar to our method. The approaches of [13, 17] can also be applied to EIC, and we compare these approaches in more detail in Section 5.

The embedded model in Figure 1c has been studied before in [10]. In that work, the authors study a special case of EIC, where each node wants all of the data blocks it does not already have. In this setting, they develop a greedy algorithm which uses a near-optimal number of broadcasts. However, their approach crucially uses the fact that every node wants every block, and does not seem to generalize to the general EIC setting that we study here.

While our coding scheme is deterministic, our multi-sender network model is similar to those studied with composite coding, an approach based on randomized coding [1]. Multi-sender models and achievable rate regions using composite coding are defined in [15, 16, 27]; to the best of our knowledge these results are not directly applicable to our scheme.

Index coding is a special instance of the network coding problem (e.g., [18]), in which source nodes send information over a network containing intermediate nodes, which may modify messages, in addition to receiver nodes. It has also been shown that network coding instances can be reduced to index coding instances [3, 9]. Real-Time Instantly Decodable Network Codes (IDNC’s) [14] aim to minimize completion delay of the communication task, rather than the index coding goal of minimizing total number of messages. Our task-based solutions are a generalization of instant decodability in index codes (see Remark 1).

Task-based solutions are also related to the notion of locally decodable index codes. An index coding solution has locality r if each node uses at most r received symbols to decode any message symbol. There is a tradeoff between optimal broadcast rate and locality of solutions for a given index coding problem [24]. When r = 1, locally decodable index codes are a special case of task-based schemes, although the notions diverge for more general r (see Remark 2).

Our construction is motivated by the problem of data shuffling for coded computation, such as in [19], or for distributed storage systems which need to redistribute data among the nodes. In data shuffling, after an initial round of computation, nodes each contain some amount of intermediate results, which then need to be shared with other nodes to continue the computation. Other connections between index coding and distributed storage have been established, but are not directly related to our work. These include the relationship between an optimal recoverable distributed storage code and a general optimal index code [24] and the duality of linear index codes and Generalized Locally Repairable codes was shown by [2, 28].

Finally, index coding techniques can also be applied to coded caching (e.g., [21], [11] and references therein), in which nodes may request and store data dynamically. Coded multicasting similar to index coding has been applied to decentralized coded caching [22], and our work could also be applied in coded caching.

**Subsequent work.** In our work, we introduce the notion of task-based schemes for EIC, and develop heuristics for these schemes. However, we left it as an open problem to understand the limitations of task-based schemes relative to other schemes. Since our work first appeared, Haviv has solved this problem by giving tight bounds on the gap between task-based schemes and centralized schemes for EIC [12]. Briefly, this work shows that there for any graph G, the length of the best task-based scheme is at most quadratically worse than the best scheme without the task-based restriction, and also shows that there exist graphs where this gap is asymptotically tight.
3 Framework

In this section we formally describe our model for Embedded Index Coding.

We assume that there is a set of $m$ data blocks, $\mathcal{D} \in (\mathbb{F}_2^2)^m$, where each data block is an element of $\mathbb{F}_2^2$; when convenient, we will view $\mathcal{D} \in \mathbb{F}_2^{m \times \ell}$ as an $m \times \ell$ boolean matrix with the $m$ data blocks as rows. These $m$ data blocks are stored on $n$ storage nodes; each node $i$ stores a subset of the data blocks, and some data blocks may be stored on multiple nodes. We assume that each node can perform local computations and can broadcast information over an error-free channel to all the other nodes. In this work, we focus on a linear model, where each node is restricted to computing $\mathbb{F}_2$-linear combinations of data blocks.

An Embedded Index Coding (EIC) problem is defined in terms of which data blocks each node has and needs. It will be convenient to represent these “has” and “needs” relationships in terms of binary matrices $B$ and $R$ respectively.

**Definition 1.** An Embedded Index Coding (EIC) problem is specified by a pair of matrices $B, R \in \mathbb{F}_2^{n \times m}$ s.t. $\text{supp}(B) \cap \text{supp}(R) = \emptyset$.

Informally, the interpretation should be that in an EIC problem $(R, B)$, a node $u$ needs block $a$ if $R_{ua} = 1$ and has block $b$ if $B_{ub} = 1$.

Each node $u$ will broadcast a set of $b_u \in \mathbb{N}$ linear combinations of the blocks it has, and the goal is for each node to be able to recover all of the blocks that it needs. We formalize this in the following definition.

**Definition 2.** For an EIC problem $(R, B)$ a linear broadcast solution that solves $(R, B)$ is a collection of matrices $\beta^{(1)}, \ldots, \beta^{(n)}$ and integers $h_1, \ldots, h_n$ with $\beta^{(u)} \in \mathbb{F}_2^{h_u \times m}$ for each $u \in [n]$ so that:

- For each $u \in [n]$ and each $a \in [m]$ so that $B_{ua} = 0$, the $a$th column of $\beta^{(u)}$ is zero.
- For each $u \in [n]$ and each $a \in [m]$ so that $R_{ua} = 1$, there is some vector $\alpha^{(u, a)} \in \mathbb{F}_2^{\sum \ell h_r + m}$ so that
  \[ e_a = \alpha^{(u, a)} \cdot \begin{bmatrix} \beta^{(1)} \\ \beta^{(2)} \\ \vdots \\ \beta^{(n)} \\ \text{diag}(B_u) \end{bmatrix} \cdot \mathcal{D}, \]

  where $B_u$ is the row of $B$ indexed by $u$ and $\text{diag}(B_u)$ is the matrix with $B_u$ on the diagonal. Above, $e_j$ denotes the $j$th standard basis vector.
- The length of an EIC solution is $\Sigma_u h_u$, the number of symbols broadcast. We also refer to this as the communication cost of the solution.

To use a linear broadcast solution, each node $u$ computes and broadcasts $\beta^{(u)} \cdot \mathcal{D}$, where we view $\mathcal{D} \in \mathbb{F}_2^{m \times \ell}$ as a matrix whose rows are the data blocks. This can be computed locally because the only non-zero columns of $\beta^{(u)}$ correspond to non-zero entries of row $B_u$, i.e. blocks node $u$ has.

In order to decode the blocks it wants, each node $u$ uses the fact that

\[
\text{block } a = e_u \cdot \mathcal{D} = \alpha^{(u, a)} \cdot \begin{bmatrix} \beta^{(1)} \\ \beta^{(2)} \\ \vdots \\ \beta^{(n)} \\ \text{diag}(B_u) \end{bmatrix} \cdot \mathcal{D},
\]

and thus block $a$ is a linear combination (given by $\alpha^{(u, a)}$) of the broadcasts $\beta^{(1)} \mathcal{D}, \ldots, \beta^{(n)} \mathcal{D}$ that node $u$ receives and the data blocks that $u$ already has.
3.1 Problem Graph and Problem Matrix

We next define some representations of embedded index coding problems (extending the work of [3]) which will be useful in studying the length of solutions and the construction of algorithms.

We begin by defining a graph $G$ which captures an EIC problem. The vertices of $G$ will correspond to requirement pairs of the EIC problem, defined as follows.

**Definition 3.** Given an EIC problem $(R, B)$, the set of requirement pairs for $(R, B)$ is $P = \{(u, a) \in [n] \times [m] : R_{ua} = 1\}$.

Now we can formally define the problem graph $G$ for an EIC problem $(R, B)$.

**Definition 4.** Given an EIC problem $(R, B)$, the problem graph $G = (V, E)$ corresponding to $(R, B)$ is the graph $G$ with vertices $V = \{v_{(u,a)} : (u, a) \in P\}$ and (directed) edges $E = \{(v_{(u,a)}, v_{(w,b)}) : B_{ub} = 1 \text{ or } a = b\}$.

That is, for $(u, a)$ and $(w, b)$ in $P$, there are two reasons that there could be an edge from the vertex $v_{(u,a)}$ to the vertex $v_{(w,b)}$: either the node $u$ has the block $b$ that the node $w$ wants, or else the two blocks $a$ and $b$ are the same block. As we will see, these two types of edges play two different roles.

Figure 2 shows two examples of problem graphs. In Figure 2a, all edges indicate where a node has a block that another is requesting, i.e. cases where $(v_{(u,a)}, v_{(w,b)}) \in E(G)$ because $B_{ub} = 1$. In Figure 2b, dashed edges indicate pairs of vertices which represent two requests for the same block, i.e. cases where $(v_{(u,a)}, v_{(w,b)}), (v_{(w,b)}, v_{(u,a)}) \in E(G)$ because $a = b$.

**Definition 5.** Given a graph $G = (V, E)$, we say matrix $A \in \mathbb{F}_2^{|V| \times |V|}$ fits $G$ if:

1. $A_{kk} = 1$ for all $k \in |V|$ and
2. for any \( k, \ell \in |V|, (k, \ell) \notin E \) implies that \( A_{k,\ell} = 0 \).

Thus if \( M \) is the adjacency matrix of \( G \) and matrix \( A \) fits \( G \), the non-zero entries of \( A \) (other than the diagonal) are a subset of the non-zero entries of \( M \).

**Definition 6.** The minrank of a graph \( G \) in field \( \mathbb{F}_2 \), denoted \( \minrk_2(G) \), is the rank of the lowest-rank matrix \( A \) over \( \mathbb{F}_2 \) which fits \( G \):

\[
\minrk_2(G) := \min \{ \rk_2(A) : A \text{ fits } G \}
\]

In Section 4.1, we will show how our definition of a problem graph generalizes the side information graph defined for index coding (that is, the centralized case of Figure 1(a), where each node requests a single unique block). In this setting, it was shown in [3] that \( \minrk_2(G) \) is the length of the optimal index code. We will show later how the minrank can also be used in computing solutions for EIC problems.

### 3.2 Task-Based Solutions

We now define a task-based solution, which is a particular type of solution to an embedded index coding problem. As we will see, we can design efficient heuristics to find task-based solutions, and additionally task-based solutions may be more useful in settings with node failures.

**Definition 7.** A task \( T = (k, M) \) is defined by a sender node \( k \) and a set of pairs

\[
M \subseteq \{ (u, a) \in P : B_{ka} = 1 \}
\]

Informally, if \( T = (k, M) \) and \( (u, a) \in M \), then this means that it is part of the node \( k \)'s task to send the block \( a \) to the node \( u \). Notice that this is not completely general: it rules out the possibility that the node \( u \) could recover the block \( a \) from two separate sender nodes.

A task-based solution is one built out of tasks. We formally define this as follows.

**Definition 8.** A task-based solution to an EIC problem \((R, B)\) with requirement pairs \( P \) is a linear broadcast solution \( \beta^{(1)}, \ldots, \beta^{(n)} \) so that \( \beta^{(\ell)} \in \mathbb{F}_2^{h_{\ell} \times m} \), such that for each \((u, a) \in P\), there is an \( \ell \in [n] \) and a coefficient vector \( \alpha^{(u,a)}_\ell \in \mathbb{F}_2^{h_{\ell} \times m} \) such that

\[
e_a = \alpha^{(u,a)}_\ell \cdot \begin{bmatrix} \beta^{(\ell)} \\ \diag(B_{ua}) \end{bmatrix}.
\]

We say that such a node \( \ell \) is responsible for \((u, a)\) in the task \( T \).

Informally, a task-based solution is a linear solution in which each node \( u \) decodes each requested block \( a \) using only messages from one sender node \( \ell \) who is responsible for \((u, a)\). That is, \( \ell \) broadcasts a vector \( \beta^{(\ell)} \cdot \mathcal{D} \), and \( u \) should be able to recover \( a \) from this vector and its local side information.

A task-based solution to \((R, B)\) is related to the corresponding problem graph \( G = (V, E) \) by specifying a partition of the vertices. Let \( N^+(v_{(u,a)}) \subseteq V \) denote the out-edge neighborhood of a vertex \( v_{(u,a)} \in V \): that is, \( N^+(v_{(u,a)}) = \{ v_{(w,b)} : (v_{(u,a)}, v_{(w,b)}) \in E \} \).

**Definition 9.** For an EIC problem \((R, B)\) with problem graph \( G \), define the sender neighborhood of node \( u \in [n] \) as:

\[
N_u = \{ v_{(w,b)} \in V : B_{ub} = 1 \}.
\]

That is, the sender neighborhood \( N_u \) of a node \( u \) is the set of vertices in \( V \) corresponding to node-block pairs \((w, b)\) so that the node \( u \) has the block \( b \) (and thus could use \( u \) could send \( b \) to \( w \)). In terms of the problem graph \( G \), \( N_u \subseteq \cup \cap N^+(v_{(u,a)}) \).

Figure 3 shows examples of sender neighborhoods for the problem graph examples shown in Figure 2a.
Remark 1. Finding tasks \((k,M)\) which maximize \(|M|\) while minimizing total broadcast messages is a generalization of Instantly Decodable Network Codes (IDNCs) [14]. More precisely, solving the IDNC problem on sender neighborhood \(N_k\) for some node \(k\) finds the task \((k,M)\) with maximal \(|M|\) such that only one message needs to be broadcast by sender \(k\) to satisfy all \((u,a) \in M\).

Remark 2. Task-based solutions are also related to locally decodable index codes (LDICs) [24]. In an LDIC, a (centralized) index coding solution has locality \(r\) if each node uses at most \(r\) of the broadcast messages to decode any one block. In the case that \(r = 1\), the natural generalization of LDICs to the decentralized setting is a special case of a task-based scheme. When \(r > 1\), the two notions are different, but they have a similar flavor of restricting the information that can be used to reconstruct a single block.

Remark 3. Each node \(k\) and its sender neighborhood \(N_k\) (or any subset of \(N_k\)) together form an instance of an index coding problem with a single source: node \(k\) is a source which has all blocks requested by nodes in \(N_k\). Thus the communication model is the same as in [3], but it is not necessarily a single unicast problem (see Definition 13); that is, it is not the case that each node wants a unique block.

Definition 10. Let \(G\) be a problem graph with sender neighborhoods \(N_1, \ldots, N_n\). A neighborhood partition is a set \(\{\tilde{N}_1, \ldots, \tilde{N}_n\}\) such that

1. \(\tilde{N}_i \subseteq N_i\) for all \(i = 1, \ldots, n\),
2. \(\tilde{N}_i \cap \tilde{N}_j = \emptyset\) for any \(i, j \in [n]\),
3. \(\bigcup_{i \in [n]} \tilde{N}_i = V(G)\).

Given an EIC problem with problem graph \(G\) and task-based solution \(T\), there is a corresponding neighborhood partition \(\{\tilde{N}_1, \ldots, \tilde{N}_n\}\): each vertex \(v_{(u,a)}\) in \(G\) belongs to the \(\tilde{N}_i\) such that \(i \in [n]\) is responsible for \((u,a)\) in \(T\). Furthermore, any neighborhood partition trivially corresponds to at least one task-based solution, in which each sender \(i \in [n]\) broadcasts each block requested by a node in \(\tilde{N}_i\) as a separate message.
For example, there is a task-based solution for the EIC problem shown in Figure 2a using sender neighborhoods \( N_y = \{u, w, x\} \) and \( N_u = \{y\} \). The messages for the task executed by node \( y \) are \( D_1 \oplus D_2 \) and \( D_2 \oplus D_3 \), and the message broadcast by node \( u \) for its task is \( D_4 \). Then nodes \( u, w, \) and \( x \) each decode their requested block from the task executed by node \( y \), and node \( y \) decodes its request from the task executed by node \( u \). Task-based solutions like this can be easier to compute than a distributed solution in general, and they allow some independence between nodes: in the example, nodes \( w \) and \( x \) do not need to wait for any node other than node \( y \) to be able to decode their requested block.

Remark 4. While we only study task-based solutions on the EIC model, task-based solutions can also be defined for multi-sender index coding in general.

### 3.3 Centralized Solutions

We will later compare decentralized solutions to embedded index coding problems to an idealized centralized index coding solution. To that end, we define a solution to a an embedded index coding problem which assumes some oracle server exists with access to all of \( D \) (and has no requirements itself).

**Definition 11.** For an EIC problem defined by \((R, B)\), a centralized linear broadcast solution which solves \((R, B)\) is a matrix \( \beta \) and \( h \in \mathbb{N} \) with \( \beta \in \mathbb{F}_h^{m \times n} \) such that for each \( u \in [n] \) and each \( a \in [m] \) with \( R_{ua} = 1 \), there is some vector \( \alpha^{(u,a)} \in \mathbb{F}_h^{n+m} \) so that
\[
e_a = \alpha^{(u,a)} \cdot \left[ \begin{array}{c} \beta \\ \text{diag}(B_u) \end{array} \right].
\]

Finally, we use the following symbols to denote the optimal lengths for each type of solution:

**Definition 12.** Let \((C)_{(R,B)}\) denote the minimum length of a centralized linear solution to the EIC problem \((R, B)\) as defined in Definition 11.

Let \((D)_{(R,B)}\) denote the minimum length of a decentralized linear broadcast solution to the EIC problem \((R, B)\) as defined in Definition 2.

Let \((T)_{(R,B)}\) denote the minimum length of a decentralized and task-based solution to the EIC problem \((R, B)\) as defined in Definition 8.

### 4 Minimum Code Lengths and Relationships

In this section, we analyze the values of \((C)_{(R,B)}, (D)_{(R,B)}, \) and \((T)_{(R,B)}\) for a given \((R, B)\). We drop \((R, B)\) from the notation when comparing two of these under the same \((R, B)\) in general. While it has been shown that graph-theoretic upper and lower bounds on minrank can have significant separation [30], they are still useful in comparing the achievable minimum lengths in different solution types for EIC problems.

#### 4.1 \((C)_{(R,B)}\) and minrank of the Problem Graph

First, we discuss an idealized centralized solution to an EIC problem, and introduce some useful machinery.

The work [3] defines the side information graph for an index coding problem. We show how our problem graph is an effective generalization of the side information graph such that the same technique of using minrank to find an optimal centralized solution applies. The side information graph as defined by [3] is equivalent to a Problem Graph (Definition 4) for any single unicast EIC problem \((R, B)\) (defined below).

**Definition 13.** An EIC specified by \((R, B)\) is a single unicast index coding problem if

1. every node requests exactly one data block and
2. each data block is requested by exactly one node.
Definition 14. Given an EIC problem \((R, B)\) and a problem graph \(G = (V, E)\), we define the column repetition function \(\phi_{(R, B)} : \mathbb{F}_2^{V \times m} \rightarrow \mathbb{F}_2^{V \times |V|}\) as follows. Given a matrix \(A \in \mathbb{F}_2^{V \times m}\), construct a matrix 
\[
\phi_{(R, B)}(A) \in \mathbb{F}_2^{V \times |V|}
\]
so that for \(v_{(u,a)} \in V\), the column of \(\phi(A)\) indexed by \(v_{(u,a)}\) is equal to the \(a\)th column of \(A\). Additionally, we will denote the image of \(\phi_{(R, B)}\) by \(\mathcal{A}_{(R, B)} = \{A' : \exists A \text{ s.t. } \phi_{(R, B)}(A) = A'\}\).

Remark 5. The function \(\phi_{(R, B)}\) preserves the rank of a matrix, since it just inserts duplicates of columns. That is, \(\text{rk}_2(\phi_{(R, B)}(A)) = \text{rk}_2(A)\).

For an EIC problem \((R, B)\), we will use the set \(\mathcal{A}_{(R, B)}\) to restrict the domain of minrank, resulting in the restricted-minrank:

Definition 15. The restricted minrank of a graph \(G = (V, E)\) in the field \(\mathbb{F}_2\) over set of matrices \(\mathcal{A} \subseteq \mathbb{F}_2^{V \times |V|}\), denoted \(\text{r-minrk}_2(G, \mathcal{A})\), is the rank of the lowest-rank matrix \(A' \in \mathcal{A}\) which fits \(G\):

\[
\text{r-minrk}_2(G, \mathcal{A}) = \min\{\text{rk}_2(A') : A' \in \mathcal{A} \land A' \text{ fits } G\}.
\]

Theorem 1. (Theorem 5 of [3]) Given a single unicast EIC \((R, B)\) and the corresponding problem graph \(G\), 
\[(C)_{(R, B)} = \text{minrk}_2(G)\]
We next generalize Theorem 1 to EIC problems which are not single unicast:

**Theorem 2.** Given an EIC problem \((R, B)\), corresponding problem graph \(G\), and column repetition function \(\phi(R, B)\) with range \(A(R, B)\),

\[(C)(R, B) = r\text{-minrk}_2(G, A(R, B)).\]

**Proof.** Let \(A' \in A(R, B)\) be the matrix of lowest rank in \(A(R, B)\) that fits \(G\) and let \(r := \text{rk}_2(A') = r\text{-minrk}_2(G, A(R, B))\). Let \(A \in \mathbb{F}_2^{V \times m}\) such that \(\phi(R, B)(A) = A'\). By Lemma 1 a matrix \(A^{(r)}\) composed of \(r\) linearly independent rows of \(A\) is a centralized linear solution to \((R, B)\) of length \(r\) (by the choice of \(r\), the rowspan of \(A^{(r)}\) equals the rowspan of \(A\)). Since a centralized source is able to construct each of these messages for this solution (that is, the rows of matrix \(A^{(r)}D\) we conclude that

\[(C)(R, B) \leq r = r\text{-minrk}_2(G, A(R, B)).\]

For the other direction, suppose that \(Z \in \mathbb{F}_2^{s \times m}\) is a linear solution to \((R, B)\) for some \(s \in \mathbb{Z}^+\). Let \(z_i \in \mathbb{F}_2^m\) denote the \(i\)th row of \(Z\). We will show the row span of \(Z\) contains the row span of some matrix \(A\) such that \(A' := \phi(R, B)(A)\) fits \(G\). Consider some \((u, a) \in P\). By the definition of a linear solution, there exists some vector \(\alpha^{(u,a)}\) such that

\[e_a = \alpha^{(u,a)} \cdot \begin{bmatrix} Z \\text{diag}(B_a) \end{bmatrix}.\]

Write

\[\alpha^{(u,a)} = [\lambda_1^{(u,a)} \ldots \lambda_s^{(u,a)} \mu_1 \ldots \mu_m]\]

for some \(\lambda_j^{(u,a)}, \mu_j \in \mathbb{F}_2\), so that

\[e_a = \sum_{i=1}^s \lambda_i^{(u,a)} z_i + \sum_{j=1}^m \mu_j B_{uj} e_j.\]

Let \(A_{(u,a)} \in \mathbb{F}_2^m\) be the vector

\[A_{(u,a)} = e_a - \sum_{j=1}^m \mu_j B_{uj} e_j = \sum_{j=1}^s \lambda_j^{(u,a)} z_j.\]

Then \(A_{(u,a)}\) is in the row span of \(Z\), and moreover the \(a\)'th entry of \(A_{(u,a)}\) satisfies

\[A_{(u,a),a} = 1.\]

Additionally, for any block \(b \in [m]\) with \(b \neq a\) such that \(B_{ab} = 0\),

\[A_{(u,a),b} = (e_a)_b - \mu_b B_{ab} = 0.\]

Let \(A \in \mathbb{F}_2^{P \times m}\) be the matrix whose rows are given by \(A_{(u,a)}\) for \((u, a) \in P\). Let \(A' := \phi(R, B)(A)\). We claim that \(A'\) fits \(G\). Indeed, we have for all \((u, a) \in P\) that

\[A'_{(u,a),(u,a)} = A_{(u,a),a} = 1\]

by the above, and so the first requirement of Definition 5 is met.

To see the second requirement of Definition 5, first note that for all \(b \neq a\), we have

\[A'_{(u,a),(u,b)} = A_{(u,a),b}\]

which by the above is non-zero only if \(B_{ab} = 1\); that is only if there is an edge (of the “first type”) from \(v_{(u,a)}\) to \(v_{(u,b)}\) in \(G\). Second, there is always an edge (of the “second type”) from \(v_{(u,a)}\) to \(v_{(w,a)}\). Thus, the only non-zero off-diagonal entries of \(A'\) correspond to edges in \(G\), and the second requirement of Definition 5 is satisfied.
Thus, for any linear solution $Z$ of length $s$, there is a matrix $A$ so that row span of $Z$ contains the row span of $A$ and so that $A' = \phi(R,B)(A)$ fits $G$. Thus,

\[(C)'(R,B) \geq s \geq \text{rk}_2(A) = \text{rk}_2(A') \geq \text{r-minrk}_2(G,A(R,B)).\]

This completes the proof.

Because the minrank gives the optimal linear solution for a centralized sender with all data blocks, our definition of the problem graph is a natural extension of index coding and the side information graph to the embedded index coding model. In the following it will be helpful to use the following theorem from \[3\] relating minrank to some other standard graph properties. For a graph $G$, the chromatic number $\chi(G)$ is the minimum number of colors required to color the vertices of $G$ so that no neighboring vertices have the same color. The clique number $\omega(G)$ is the size of the largest clique in $G$. The independence number, denoted $\alpha(G)$, is the set of the largest independent set in $G$, so $\alpha(G) = \omega(\overline{G})$.

**Theorem 3** (\[3\]). $\omega(\overline{G}) \leq \text{minrk}_2(G) \leq \chi(\overline{G})$.

These bounds also apply to our restricted version of minrank:

**Corollary 1.** Given an EIC problem $(R,B)$, a corresponding problem graph $G = (V,E)$, and the column repetition function $\phi(R,B)$ with range $A(R,B)$, we have:

\[\omega(\overline{G}) \leq \text{minrk}_2(G) \leq \text{r-minrk}_2(G,A(R,B)) \leq \chi(\overline{G}).\]

**Proof.** Since $\text{r-minrk}_2$ is a minimization over a smaller set of matrices than $\text{minrk}_2$, clearly $\text{minrk}_2(G) \leq \text{r-minrk}_2(G,A)$. Thus

\[\omega(\overline{G}) \leq \text{minrk}_2(G) \leq \text{r-minrk}_2(G,A)\]

follows from Theorem \[3\]. Using a similar approach as in \[3\], we show the final inequality by describing a matrix $A' \in A$ such that $\text{rk}_2(A') \leq \chi(\overline{G})$.

By the definition of chromatic number, there is a partition of $V$ into sets $C_1, \ldots, C_{\chi(\overline{G})}$ so that each $C_i$ forms a clique in $G$. Let $C \subseteq V$ be a clique from such a partition. Define a vector $c^{(C)} \in \mathbb{F}_2^n$ so that the $a$'th entry of $c^{(C)}$ is given by

\[c_a^{(C)} = \begin{cases} 1 & \exists u \in [n] \text{ so that } v_{(u,a)} \in C \\ 0 & \text{else} \end{cases}\]

Now, define a matrix $A \in \mathbb{F}_2^{[P] \times m}$ with rows indexed by elements of $P$, so that if $v_{(u,a)}$ is in the clique $C$, then

\[A_{(u,a)} = c^{(C)}.\]

Let $A' = \phi(R,B)(A)$. Thus

\[\text{rk}_2(A') = \text{rk}_2(A) \leq \chi(\overline{G})\]

since there are only $\chi(\overline{G})$ distinct rows of $A$.

Now we just need to show that $A'$ fits $G$. Consider some row $A'_{(u,a)}$, where $A_{(u,a)} = c^{(C)}$ for some clique $C$, and choose some $\langle w,b \rangle \neq (u,a)$ such that $A_{(u,a)}(w,b) = 1$. If $a = b$, then $(v_{(u,a)}, v_{(w,b)}) \in E$ is an edge of the “second type.” On the other hand, if $a \neq b$, then by the definition of $\phi(R,B)$, $c^{(C)}_b = 1$, so there exists some $x \in [n]$ so that $v_{(x,b)} \in C$. Since $C$ is a clique, $(v_{(u,a)}, v_{(x,b)}) \in E$. By the definition of problem graph this is an edge of the “first type,” so $B_{ab} = 1$, so we also have $(v_{(u,a)}, v_{(w,b)}) \in E$. Thus any non-zero off-diagonal entry of $A$ corresponds to an edge in $G$. Moreover, the diagonal entries of $A'$ are

\[A'_{(u,a),(u,a)} = A_{(u,a),(u,a)} = c_a^{(C)}\]

where $v_{(u,a)} \in C$, so this is 1 from the definition of $c_a^{(C)}$. Thus, $A'$ fits $G$. ☐

11
4.2 Cost of Decentralization: \((D) \leq 2(C)\)

It can easily be seen that \((C) \leq (D)\), that is, that the minimum length of a decentralized embedded index code is at least the minimum length of a centralized solution. Indeed, the \((D)\) messages transmitted in the decentralized solution can all be constructed by a centralized source which has access to all data blocks. Thus we are interested in how much larger \((D)\) can be than \((C)\). In fact, we show that it is no more than a factor of 2 worse:

**Theorem 4.** Given an EIC problem \((R, B)\), \((D)_{(R, B)} \leq 2 \cdot (C)_{(R, B)}\).

**Proof.** Let \(P\) be the set of requirement pairs for \((R, B)\). Let \(G = (V, E)\) be the problem graph for \((R, B)\) and let \(\phi_{(R, B)}\) be the corresponding column expansion function, with image \(A_{(R, B)}\). By Theorem 2 \((C)_{(R, B)} = r\text{-minrk}_2(G, A_{(R, B)})\). Let \(A' \in A_{(R, B)}\) be a matrix with \(A' = \phi_{(R, B)}(A)\) for some \(A \in \mathbb{F}_2^{V \times m}\) so that

\[
\text{rk}_2(A') = \text{rk}_2(a) = (C)_{(R, B)} =: r
\]

and so that \(A'\) fits \(G\). By Lemma 5 there is a matrix \(A^{(r)}\) with rows \(A_1, \ldots, A_r\) that is a centralized linear broadcast solution to \((R, B)\). We will show how to simulate this centralized solution using only 2r messages.

Since \(A_1, \ldots, A_r\) are rows of \(A\), they correspond to requirement pairs in \(P\). Fix \(\ell \in [r]\) and suppose that \(A_\ell\) corresponds to \((u, a) \in P\). Since \(A'\) fits \(G\), the diagonal entries of \(A'\) are non-zero. This means that \(A_\ell(a) \neq 0\). Further, for \(b \neq a\), if \(A_\ell(b) \neq 0\) then there is an edge of the “first type” in \(G\): that is, \(B_{ab} = 1\), which means that node \(u\) has block \(b\). Thus, node \(u\) is able to compute

\[
\sum_{b : B_{ub} = 1} A_\ell b \cdot \mathcal{D} = A_\ell \cdot \mathcal{D} - e_a \cdot \mathcal{D}.
\]

The decentralized scheme is then as follows. For each \(\ell \in [r]\) corresponding to \((u, a)\), we have two broadcasts:

1. Node \(u\) broadcasts \(\sum_{b : B_{ub} = 1} A_\ell b \cdot \mathcal{D}\). That is, \(A_\ell\) is a row of \(\beta^{(u)}\).

2. Fix any other node \(w\) so that \(B_{wa} = 1\). Then node \(w\) broadcasts \(e_a \cdot \mathcal{D}\). That is, \(e_a\) is a row of \(\beta^{(w)}\).

Now every node can add together the two broadcasts corresponding to \(\ell \in [r]\) to obtain \(A_\ell \cdot \mathcal{D}\). Since \(A^{(r)}\) is a linear centralized solution to \((R, B)\), this scheme is a linear centralized solution to \((R, B)\). \(\square\)

We note that the proof of Theorem 4 crucially uses the EIC formulation; this shows why considering EIC separately as a special case of multi-sender index coding can be valuable.

4.3 Cost of Task-Based Solutions: Upper Bound for \((T)\)

We first show how the minrank can be used to re-formulate the length \((T)\) of the optimal task-based solution. Let \((R, B)\) be an EIC problem, with problem graph \(G = (V, E)\). Recall from Definition 10 that, given a task-based solution \(T\), the neighborhood partition \(\{\hat{N}_1, \ldots, \hat{N}_n\}\) is a partition of \(V\) so that \(\hat{N}_w \subseteq N_w\) is the set of vertices \(v_{(u, a)}\) so that \(w\) is responsible for \((u, a)\) in \(T\).

For \(\hat{N}_w \subseteq N_w\) corresponding to a task-based solution \(T\), let \(G|_{\hat{N}_w}\) denote the induced subgraph of \(G\) on the vertices \(\hat{N}_w\). As per Remark 5 each \(\hat{N}_w\) corresponds to an EIC problem \((R^{(w)}, B^{(w)})\), over the set of blocks \(\{a \in [m] : \exists u \in [n] \text{ s.t. } v_{(u, a)} \in \hat{N}_w\}\). Thus by definition of \(\hat{N}_w\), node \(w\) has all blocks used in problem \((R^{(w)}, B^{(w)})\) and any centralized solution to \((R^{(w)}, B^{(w)})\) can be broadcast by \(w\). Note that such a solution can easily be used as a self-contained part of a solution to the problem \((R, B)\) with the full set of \(m\) blocks. To do so, we just insert zeros in encoding and decoding vectors for blocks in \([m]\) not used in \((R^{(w)}, B^{(w)})\).

Then the messages of the solution to \((R^{(w)}, B^{(w)})\) can be used by vertices of \(\hat{N}_w\) as in the subproblem.

We first show how solutions to these subproblems can be used as building blocks for task-based solutions.
Lemma 2. Let \( G \) be a problem graph for EIC problem \((R, B)\). Let \( \{\tilde{N}_1, \ldots, \tilde{N}_n\} \) be a neighborhood partition. Let \( G|_{\tilde{N}_w} \) be the subgraph of \( G \) induced by \( \tilde{N}_w \) and let \((R^{(w)}, B^{(w)})\) be the problem with problem graph \( G|_{\tilde{N}_w} \) for all \( w \in [n] \). Then any set of solutions \( \{\beta^{(w)} \in \mathbb{F}^{h_w \times m} : w \in [n], h_w \in \mathbb{Z}^+\} \) for problems \((R^{(w)}, B^{(w)}) : w \in [n]\) forms a task-based solution to \((R, B)\) with length \( \sum_{i=1}^{n} h_i \).

Proof. For each vertex \( v_{(u,a)} \) of \( G \) there is some \( \tilde{N}_w \) such that \( v_{(u,a)} \in \tilde{N}_w \). Let \( \beta^{(w)} \) be the centralized linear broadcast solution to EIC problem \((R^{(w)}, B^{(w)})\), where \( \beta^{(w)} \in \mathbb{F}^{h_w \times m} \) for some \( h_w \in \mathbb{Z}^+ \). Then there exists some \( \alpha^{(u,a)} \) such that \( e_a = \alpha^{(u,a)} \cdot \left[ \frac{\beta^{(w)}}{\text{diag}(B_u)} \right] \). Since there is such a vertex \( v_{(u,a)} \) for each \((u, a) \in P\), all requests in \( P \) are satisfied in this way by some \( \beta^{(w)} \cdot D \). By definition of \((R^{(w)}, B^{(w)})\), each \( \beta^{(w)} : D \) for \( w \in [n] \) can be broadcast by node \( w \). Thus \( \beta^{(1)}, \ldots, \beta^{(n)} \) forms a task-based solution to \((R, B)\) with length \( \sum_{i=1}^{n} h_i \).

We can then compute the length of an optimal task-based solution, \((T)\), in terms of neighborhood partitions.

Lemma 3. Given an EIC problem \((R, B)\), let \( \mathcal{N} \) be the set of all possible neighborhood partitions (as in Definition 10). For \( \{\tilde{N}_1, \ldots, \tilde{N}_n\} \in \mathcal{N} \), let \((R^{(w)}, B^{(w)})\) be the EIC problem induced by \( \tilde{N}_w \). Then

\[
(T)_{(R, B)} = \min_{\{\tilde{N}_1, \ldots, \tilde{N}_n\} \in \mathcal{N}} \sum_{i=1}^{n} r \cdot \text{minrk}_2(\tilde{G}_i, \tilde{A}(R^{(i)}, B^{(i)})).
\]

Proof. We first show that \((T)_{(R, B)} \leq \min_{\{\tilde{N}_1, \ldots, \tilde{N}_n\} \in \mathcal{N}} \sum_{i=1}^{n} r \cdot \text{minrk}_2(\tilde{G}_i, \tilde{A}(R^{(i)}, B^{(i)}))\). Consider the neighborhood partition \( \{\tilde{N}_1, \ldots, \tilde{N}_n\} \) which minimizes \( \sum_{i=1}^{n} r \cdot \text{minrk}_2(\tilde{G}_i, \tilde{A}(R^{(i)}, B^{(i)})) \). A possible task based solution \( T \) can be constructed by optimally solving the centralized index coding problem \((R^{(w)}, B^{(w)})\) defined by each \( \tilde{G}_i \) with sending node \( w \), as shown in Lemma 2. By Theorem 2 each centralized subproblem solution has length \( r \cdot \text{minrk}_2(\tilde{G}_i, \tilde{A}(R^{(i)}, B^{(i)})) \), so the total length of \( T \) is \( \sum_{i=1}^{n} r \cdot \text{minrk}_2(\tilde{G}_i, \tilde{A}(R^{(i)}, B^{(i)})) \).

Next we show \( \min_{\{\tilde{N}_1, \ldots, \tilde{N}_n\} \in \mathcal{N}} \sum_{i=1}^{n} r \cdot \text{minrk}_2(\tilde{G}_i, \tilde{A}(R^{(i)}, B^{(i)})) \leq (T)_{(R, B)} \). Let \( T \) be the optimal task-based solution, with length \( (T)_{(R, B)} \). Construct \( \{\tilde{N}_1, \ldots, \tilde{N}_n\} \) so that

\[
\tilde{N}_w = \{v_{(u,a)} : w \text{ is responsible for } (u, a) \in T \}.
\]

By the definition of a task-based solution, each vertex is assigned to exactly one such \( \tilde{N}_w \), so we have a neighborhood partition \( \{\tilde{N}_1, \ldots, \tilde{N}_n\} \in \mathcal{N} \). The centralized index coding problems \((R^{(w)}, B^{(w)})\) for each \( w \in [n] \) have problem graphs \( \tilde{G}_i \) and optimal solutions of length \( r \cdot \text{minrk}_2(\tilde{G}_i, \tilde{A}(R^{(i)}, B^{(i)})) \) (Theorem 2). If \((T)_{(R, B)} < \sum_{i=1}^{n} r \cdot \text{minrk}_2(\tilde{G}_i, \tilde{A}(R^{(i)}, B^{(i)}))\), then the solution to \((R^{(w)}, B^{(w)})\) for some \( w \in [n] \) must have length strictly less than \( r \cdot \text{minrk}_2(\tilde{G}_i, \tilde{A}(R^{(i)}, B^{(i)})) \). This contradicts Theorem 2. Thus \((T)_{(R, B)} \geq \min_{\{\tilde{N}_1, \ldots, \tilde{N}_n\} \in \mathcal{N}} \sum_{i=1}^{n} r \cdot \text{minrk}_2(\tilde{G}_i, \tilde{A}(R^{(i)}, B^{(i)}))\).

The following upper bound on \((T)_{(R, B)}\) follows from Lemma 3 and Theorem 3.

Lemma 4. Given an EIC problem defined by \((R, B)\), let \( \mathcal{N} \) be the set of all possible neighborhood partitions (Definition 10). Then

\[
(T)_{(R, B)} \leq \min_{\{\tilde{N}_1, \ldots, \tilde{N}_n\} \in \mathcal{N}} \sum_{i=1}^{n} \chi(\tilde{G}_i).
\]

In Section 5.2 we will use Lemma 4 to develop algorithms to approximate the optimal neighborhood partition (in the sense that the right hand side of 1 is minimized), by reducing the problem to the minimum cover problem.
Figure 4: (a) Example of a problem graph for a single unicast EIC problem that has \( (C) < (D) < (T) \). Each pair of boxes is a vertex in the problem graph, where black boxes contain indices of requested data blocks and the white boxes contain indices of side information blocks; the label \( u, w, x, y, z \) indicates which node the vertex corresponds to. (b) The subgraphs induced by each node along with its sender neighborhood.

### 4.4 An example where \((C) \neq (D) \neq (T)\)

Figure 4a is an example of an EIC problem \((R, B)\) for which \((C) < (D) < (T)\). First consider \((C)\): by inspection, a central source with all data blocks could send messages \(D_2 \oplus D_4, D_3, D_1 \oplus D_5\) so that all five nodes can decode their requested block, but no combination of fewer messages suffices. Thus, \((C) = 3\).

Next consider \((D)\). A solution of minimum-length is: node \(y\) broadcasts \(D_2 \oplus D_3\), node \(z\) broadcasts \(D_3 \oplus D_1\), node \(u\) broadcasts \(D_4\), and node \(x\) broadcasts \(D_5\). It can be checked that this is indeed a minimum-length solution. Thus, \((D) = 4\).

Finally, consider \((T)\). Then out-neighborhoods of each node, as shown in Figure 4b, are the subgraphs over which we can apply index coding (Remark 3). In particular, we construct the neighborhood partition from these subgraphs (Definition 10). Since the graph induced by each neighborhood is acyclic, as shown in [31] there is no way to do any non-trivial coding in any subgraph to a code shorter than the uncoded solution. Thus any task based solution requires that all blocks be broadcast uncoded. Since there are five blocks that need to be sent, we have \((T) = 5\).

### 4.5 Separations between \((T)\) and \((C)\)

We next give a condition on the problem graph \(G\) which guarantees that \((T)\) is strictly larger than \((C)\), with a gap as big as the gaps from the graph-theoretic minrank bounds.

**Lemma 5.** Given an EIC problem defined by \((R, B)\) and corresponding problem graph \(G\). If \((\chi(G) - 1)\chi(G) < |V|\) then there is an optimal task-based solution with neighborhood partition \(\tilde{N}_1, ..., \tilde{N}_n\) so that

\[
(C)_{(R,B)} \leq \chi(G) \leq \sum_{i=1}^{n} \frac{|V(G)|}{\chi(G)} \leq \sum_{i=1}^{n} \frac{|V(G)|}{\chi(G)} = |V| - 1
\]

**Proof.** Let \(V := V(G)\). First note that for any graph \(G\), \(\alpha(G) = \omega(G)\) and \(\frac{|V|}{\chi(G)} \leq \alpha(G)\) (see, e.g., [33]). Consider coloring each graph \(G|_{\tilde{N}_i}\) (induced by an element of the neighborhood partition) individually, compared to coloring all of \(G\) at once. Since every node in \(\tilde{N}_i\) shares a neighbor in \(G\) (i.e. any of the vertices \(v_{(i,a)} \in V\) for some \(a \in [m]\)) there is a color in the minimum coloring of \(G\) not necessary to color \(G|_{\tilde{N}_i}\). Thus \(\chi(\tilde{N}_i) \leq \chi(G) - 1\). Putting these steps together:

\[
\sum_{i=1}^{n} \omega(G|_{\tilde{N}_i}) 
\geq \sum_{i=1}^{n} \frac{|V(G)|}{\chi(G)} 
\geq \sum_{i=1}^{n} \frac{|V(G)|}{\chi(G)} = \frac{|V|}{\chi(G) - 1} = \frac{|V|}{\chi(G) - 1}
\]
Since by assumption we have \[ \frac{|V|}{\chi(G)} > \chi(G) \], this proves the claim. Note that \( \sum_{i=1}^{n} \omega(G|_{\tilde{N}_i}) \geq \sum_{i=1}^{n} \frac{|V(G|_{\tilde{N}_i})|}{\chi(G|_{\tilde{N}_i})} \) follows from applying \( \omega(G) \geq \frac{|V|}{\chi(G)} \) to each induced graph \( G|_{\tilde{N}_i} \). Additionally, \( \sum_{i=1}^{n} |V(G|_{\tilde{N}_i})| = |V| \) leads to the equality on the right because the graphs in the set \( \{ G|_{\tilde{N}_i} : i \in [n] \} \) are induced by corresponding elements of the vertex partition \( \tilde{N}_1, \ldots, \tilde{N}_n \).

Lemma 5 establishes a gap between \( (C)_{(R,B)} \) and \( (T)_{(R,B)} \) whenever \( |V| > (\chi(G) - 1)(\chi(G)) \), so we note here a few small graphs which illustrate how this may or may not be the case. First, we note that for all graphs it is true that \( |V| \leq \chi(G)\chi(G) \). For cliques and graphs consisting of multiple disconnected cliques, \( |V| = \chi(G)\chi(G) > (\chi(G) - 1)(\chi(G)) \) so Lemma 5 establishes a gap. On the other hand, for (directed) cycles, Lemma 5 does not establish a gap: \( \chi(G) = 2 \) or 3 and \( \chi(G) = n \) so \( (\chi(G) - 1)(\chi(G)) \leq n \).

5 Algorithms

In this section, we use results from the previous section to design two heuristics for finding good EIC solutions. We also demonstrate empirically that our algorithms perform well.

First, we use Theorem 4 to give an algorithm which produces an EIC solution that is optimal within a factor of two. We show empirically that our algorithm is faster (more precisely, has a smaller search space) than the algorithm of [17]. We note that our algorithm is tailored for EIC while the approach of [17] works more generally in the multi-sender model. This demonstrates the value of focusing on EIC as a special case.

Second, we use Lemma 5 to give a heuristic algorithm to design a task-based scheme for an EIC problem. We show empirically that the quality of solution returned by our algorithm is within a small constant factor (at most 1.4 in our experiments) of the optimal centralized scheme.

We describe both of these in more detail below.

5.1 Approximating \( (D) \)

The proof of Theorem 4 gives an algorithm to approximate the optimal decentralized solution to an EIC problem, which we detail in Algorithm 1. Algorithm 1 first computes the exact optimal centralized solution with length \( (C)_{(R,B)} \) and then uses the transformation outlined in the proof of Theorem 4 to arrive at a decentralized solution with length at most \( 2(C)_{(R,B)} \). We note that in practice the optimal centralized solution could also be approximated, leading to a decentralized solution of length at most twice the cost of the approximation.

Algorithm 1. Given an EIC problem \( (R,B) \):

1. Construct the problem graph \( G \)
2. Find \( A \in \mathbb{F}_2^{[V] \times m} \) such that \( \text{rk}_2(A) = \text{rk}_2(\phi_{(R,B)}(A)) = \text{r-minrk}_2(G, A_{(R,B)}) \)
   (a) Let \( r := \text{rk}_2(A) \)
   (b) Let \( A_1, \ldots, A_r \) be linearly independent rows of \( A \)
3. For each \( A_{\ell} \in \{ A_1 \ldots A_r \} \):
   (a) Let \( \ell = (u, a) \)
   (b) Node \( u \): compute and broadcast \( \sum_{k=(w,a):k\in|V|;B_{w,a};k\in|V|} A_{tk}D_a \)
   (c) For each node \( w \) s.t. \( B_{(w,a)} = 1 \): compute and broadcast \( \sum_{k=(w,a):w\in[n]} A_{tk}D_a \)

We compare Algorithm 1 to the algorithms in previous work [13, 17], which apply more generally to any multi-sender index coding problem but can in particular be applied to EIC. Since the algorithms of [13, 17] only apply to EIC problems in which each node is requesting a single block, we restrict our analysis to
that case. In this case, applying \( \phi_{(R,B)} \) has no effect (that is, \( A = \phi_{(R,B)}(A) \)), so the computation of \( r\text{-mrkr}_2(G,A_{(R,B)}) \) in step 2 of Algorithm 1 is equivalent to computing \( \text{mrkr}_2(G) \).

To compare the complexity of these algorithms, we observe that the main computational task of both methods is computing the minrank of a graph, by searching over a set of possible fitting matrices. In practice, we may wish to use a heuristic to approximate the minrank; however, one way to compare the speed of these algorithms is to compute the size of the search space that would be required to compute the minrank exactly. As we describe below, the search space for Algorithm 1 is much smaller than that for the other algorithms. (We note that if the minrank is computed exactly, then Combined LT-CMAR approach of previous work becomes an exact algorithm, while our algorithm is a two-approximation.)

In [13], the set of possible matrices are those that fit a constraint matrix \( C \): a matrix \( M \) of the same dimensions as \( F \) is a possible solution if \( F_{ij} = 0 \rightarrow M_{ij} = 0 \). The number of linearly independent rows of such a solution \( M \) is the corresponding solution size, so the goal is to find the \( M \) of minimum rank. Let \( n_k = |B_k| \), i.e. the number of blocks node \( k \) has. In this approach, the search space of matrices that fit the requirements to decode is of size

\[
2\sum_{k=1}^{n} n_k \cdot n_k.
\]

In [17], the minimization problem is over a smaller search space of matrices, but with additional constraints. As in other work, a solution matrix \( M \) represents the requested blocks and side information of each node and the goal is to minimize \( rk_2(M) \). Additionally, \( M \) must be in the rowspan of a matrix \( C \), which represents the blocks available at each node to use in constructing messages. Using Gaussian elimination, the submatrices of \( C \) corresponding to each sender node are altered to maximize repeated rows in \( C \) as a whole. Then letting \( n_k^{(c)} \) be the number of such redundant rows, the search space for solution matrices is of size

\[
2\sum_{k=1}^{n} (n_k^2 + n_k) / 2 - \sum_{k=1}^{n} ((n_k^{(c)})^2 + n_k^{(c)}) / 2.
\]

A heuristic method, the Combined LT-CMAR procedure, for finding some of such redundancies is provided in [17].

In contrast, Algorithm 1 in particular step 2 requires searching over \( 2\sum_{k=1}^{n_k} n_k \) matrices.

Figure 4 shows how the base-2 logarithm of the search space of Algorithm 1 compares to that of the Combined LT-CMAR procedure of [17]. The larger the ratio, the more costly the Combined LT-CMAR procedure is relative to our EIC heuristic. Values are computed on Erdős-Rényi graphs randomly generated

Combined LT-CMAR procedure of [17]. The larger the ratio, the more costly the Combined LT-CMAR

5.2 Approximating \((T)\)

Computing a task-based solution consists of two main steps: finding a neighborhood partition (Definition 10) and finding an index coding solution to the task defined by each \( N_i \) for sender node \( i \). Our heuristic uses Lemma 4 to approximate an optimal choice for a neighborhood partition. In order to see how, we define the neighborhood-cliques associated with an EIC problem:

Definition 16. Given a problem graph \( G \) for some EIC problem \((R,B)\) with sender node neighborhoods \( N_1,...,N_n \), let the set of neighborhood-cliques be \( \mathcal{C} := \{ V(C) : C \) is a maximal clique in \( G|_{N_i} \) for some \( N_i \} \)

\[
\mathcal{C} := \{ V(C) : C \) is a maximal clique in \( G|_{N_i} \) for some \( N_i \}.
\]

16
Figure 5: $S_{LT-CMAR}$ is the value $\log_2$ of the search space size for a given $(R, B)$ using the method of [17]. $S_{EIC}$ is the value $\log_2$ of the search space size using our method of approximating with the corresponding centralized solution. Ratios $S_{LT-CMAR}/S_{EIC}$ are plotted for the averages over sets of 20 Erdős-Renyi graphs with the given number of vertices $n$ and probability $p$ for each directed edge. When $S_{LT-CMAR}/S_{EIC} > 1$ our algorithm has a strictly smaller search space.

We first define the min-cover and min-clique-cover problems. Let $U = \{u_1, ..., u_n\}$ be a set of $n$ elements. Let $S = \{X_1, ..., X_m\}$ be subsets of $U$, i.e. $X_j \subseteq U$ for each $j \in [m]$, such that for all $i \in [n]$, $u_i$ is in some $X_j$. The min-cover problem over $(U, S)$ is to find the smallest $\{X_{j_1}, ..., X_{j_k}\} \subseteq S$, such that $\bigcup_{\ell=1}^k X_{j_\ell} = U$.

The min-clique-cover problem over a graph $G$ is an instance $(U, S)$ of min-cover in which $U = V(G)$ and $S$ is the set of all cliques in $G$, including non-maximal cliques and single vertices.

Using neighborhood-cliques, solving for the chromatic numbers used to upper bound the length of a task-based solution reduces to min-cover:

**Theorem 5.** Given an EIC problem $(R, B)$ and the corresponding problem graph $G$, solving for the neighborhood partition $\tilde{N}_1, ..., \tilde{N}_n$ to minimize $\sum_{i=1}^n \chi(G_{|\tilde{N}_i})$ is exactly equivalent to the min-cover problem over vertices of $G$ with sets $\mathcal{C} = \{V(C_i) : C_i$ is a maximal clique in $G_{|\tilde{N}_i}\}$.

**Lemma 6.** Given the neighborhood partition $\{\tilde{N}_i : i \in [n]\}$ for some $(R, B)$ with problem graph $G$, there exists a cover $\mathcal{C}$ of $V(G)$ chosen from elements of $\mathcal{C}$ (the set of neighborhood-cliques) such that

$$\sum_{C_j \in \mathcal{C}} \chi(C_j) = \sum_{i=1}^n \chi(G_{|\tilde{N}_i}).$$

**Proof.** Take some $\tilde{N}_i$ and consider a minimum coloring using $\chi(G_{|\tilde{N}_i})$ colors. Each set of vertices with a shared color is by definition a clique in $G_{|\tilde{N}_i}$, call such a clique $C_j$. Since $\tilde{N}_i \subseteq N_i$, $C_j$ (or a larger clique containing $C_j$ if $C_j$ is not maximal) is in $\mathcal{C}$. We can apply this to all $\tilde{N}_i$ to get a cover of $V(G)$. Since we create such a $C_j$ for each color used for $\tilde{N}_i$, and the complement of a clique is 1-colorable,

$$\sum_{j : C_j \text{used for } \tilde{N}_i} \chi(C_j) = |\{j \text{ used for } \tilde{N}_i\}| = \chi(G_{|\tilde{N}_i}).$$

**Lemma 7.** Given an EIC problem $(R, B)$ with problem graph $G$ and clique cover $\mathcal{C} := \{C_j\} \subseteq \mathcal{C}$, there is a corresponding choice of neighborhood partition $\tilde{N}_1, ..., \tilde{N}_n$ such that

$$\sum \chi(C_j) \geq \sum \chi(G_{|\tilde{N}_i}).$$
Proof. For each $i$, let $\tilde{N}_i := \bigcup \{C_j \in \mathcal{C} : C_j \subseteq N_i\}$. Now letting $c := |\{C_j \in \mathcal{C} : C_j \subseteq N_i\}|$, we can color $G|_{\tilde{N}_i}$ with $c$ colors, so $\chi(G|_{\tilde{N}_i}) \leq c = \sum \chi(C_j)$.

The proof of Theorem 4 follows immediately from Lemmas 6 and 7. This also gives us an algorithm for $\tilde{N}_1, ..., \tilde{N}_n$. Since min-cover is NP-hard solving for these will be as well, but we can use existing min-cover approximation algorithms. Below, Algorithm 2 computes the neighborhood partition which minimizes $\sum_{i=1}^{n} \chi(G|_{\tilde{N}_i})$ and the length of the minimum task-based solution using that partition.

**Algorithm 2.** Given an EIC problem $(R, B)$:

1. Construct the problem graph $G$
2. Let $\mathcal{C} = \emptyset$
3. For each $v_i \in V(G)$
   (a) Let $N_i \subset V(G)$ be the out-neighborhood of $v_i$
   (b) Compute $G|_{N_i}$, the subgraph induced by $N_i$
   (c) Compute the set of maximal cliques in $G|_{N_i}$ and add each to $\mathcal{C}$
4. Compute min clique cover $\mathcal{C}$ of $\mathcal{C}$
5. Let $T = 0$
6. For each $v_i \in V(G)$
   (a) Let $\tilde{N}_i = \bigcup \{C_j \in \mathcal{C} : C_j \subseteq N_i\}$
   (b) Compute $G|_{\tilde{N}_i}$, the subgraph induced by $\tilde{N}_i$
      i. Let $(R^{(i)}, B^{(i)})$ be a EIC problem with problem graph $G|_{\tilde{N}_i}$, keeping vertex labels from $(R, B)$
      (c) $T = T + r\text{-minrk}_2(G|_{\tilde{N}_i}, A_{(R^{(i)}, B^{(i)})})$
7. $T$ is the total cost of the optimal task-based solution given neighborhood partition $\{\tilde{N}_1, ..., \tilde{N}_n\}$.

Figure 6 shows the ratio of the length of our approximately optimal task based solution compared to the length of the optimal centralized solution. This ratio upper bounds the ratio of a true optimal task based solution to the corresponding centralized solution. In all of our experiments this approximation ratio is upper-bounded by 1.4. As in the experiments in Figure 5, Erdős-Rényi graphs are randomly generated for a variety of values for $n$, the number of nodes, and $p$, the directed edge probability. As the size of the graph increases for a fixed edge probability, the ratio appears to converge. For a fixed number of nodes, there also appears to be some upper bound on the ratio even as the probability of each edge goes to 1.

### 6 Conclusion

In this paper we defined embedded index coding, a special case of multi-sender index coding in which each node of the network is both a broadcast sender and a receiver. We characterized an EIC problem using a problem graph, and we used this formulation to show that the optimal length of a solution to an EIC problem is bounded by twice the length of the optimal centralized index coding solution. We also defined task-based solutions to EIC problems, in which the set messages broadcast by each node can be decoded independently of messages from other senders, and we proved characterizations and bounds for task-based solutions. Finally, we used these bounds to develop heuristics for finding good solutions to EIC problems, and showed empirically that these heuristics perform well.

---

2Sample sizes in these experiments are 10 random graphs, except $p = 0.9, n = 6$ which only uses 5, since the search space for the brute-force minrank algorithm explodes, increasing exponentially in the number of graph edges.
We end with some open questions and future directions. Since this work first appeared, it was shown by [12] that for any integer $k$, there exists an index coding problem with problem graph $G$ and $\text{minrk}_2(G) = k$, such that the task-based solution cost is $\Theta(k^2)$. Since we’ve shown a decentralized solution has cost within a constant factor of the centralized solution cost, i.e. $\text{minrk}_2(G)$, this result also shows a gap between general decentralized and task-based solutions. However, the exact relationship between decentralized solutions and centralized solutions to embedded problems remains open.

It is also an interesting question to improve on algorithms for finding task-based solutions. Our current approach uses an upper bound on minrank, given by the chromatic number of the complement of the problem graph. This bound is known to be quite loose in some settings. The fractional chromatic number of the complement of the problem graph, $\chi_f(G)$, has been used to tighten the upper on minrank of $G$ [5], and it was also shown by [24] that the optimal centralized index coding solution size with locality of one is $\chi_f(G)$. Thus the fractional chromatic number may be a useful approach in this direction.

References

[1] Fatemeh Arbabjolfaei, Bernd Bandemer, Young-Han Kim, Eren Şaşoğlu, and Lele Wang. On the capacity region for index coding. In 2013 IEEE International Symposium on Information Theory, pages 962–966. IEEE, 2013.

[2] Fatemeh Arbabjolfaei and Young-Han Kim. Three stories on a two-sided coin: Index coding, locally recoverable distributed storage, and guessing games on graphs. arXiv preprint arXiv:1511.01050, 2015.

[3] Ziv Bar-Yossef, Yitzhak Birk, T Jayram, and Tomer Kol. Index coding with side information. In 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06).

[4] Yitzhak Birk and Tomer Kol. Coding on demand by an informed source (iscod) for efficient broadcast of different supplemental data to caching clients. IEEE Transactions on Information Theory, 52(6):2825–2830, 2006.

[5] Anna Blasiak, Robert Kleinberg, and Eyal Lubetzky. Index coding via linear programming. arXiv preprint arXiv:1004.1379, 2010.

[6] Mohammad Asad R Chaudhry, Zakia Asad, Alex Sprintson, and Michael Langberg. On the complementary index coding problem. In Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on, pages 244–248. IEEE, 2011.

[7] Mohammad Asad R Chaudhry and Alex Sprintson. Efficient algorithms for index coding. In INFOCOM Workshops 2008, IEEE, pages 1–4. IEEE, 2008.
[8] Michelle Effros, Salim El Rouayheb, and Michael Langberg. An equivalence between network coding and index coding. *IEEE Transactions on Information Theory*, 61(5):2478–2487, 2015.

[9] Salim El Rouayheb, Alex Sprintson, and Costas Georgiades. On the relation between the index coding and the network coding problems. In *2008 IEEE International Symposium on Information Theory*, pages 1823–1827. IEEE, 2008.

[10] Salim El Rouayheb, Alex Sprintson, and Parastoo Sadeghi. On coding for cooperative data exchange. In *2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo)*, pages 1–5. IEEE, 2010.

[11] Hooshang Ghasemi and Aditya Ramamoorthy. Improved lower bounds for coded caching. *IEEE Transactions on Information Theory*, 63(7):4388–4413, 2017.

[12] Ishay Haviv. Task-based solutions to embedded index coding. *arXiv preprint arXiv:1906.09794*, 2019.

[13] Jae-Won Kim and Jong-Seon No. Linear index coding with multiple senders and extension to a cellular network. *arXiv preprint arXiv:1901.07136*, 2019.

[14] Anh Le, Arash S Tehrani, Alexandros G Dimakis, and Athina Markopoulou. Instantly decodable network codes for real-time applications. In *2013 international symposium on network coding (NetCod)*, pages 1–6. IEEE, 2013.

[15] Min Li, Lawrence Ong, and Sarah J Johnson. Improved bounds for multi-sender index coding. In *Information Theory (ISIT), 2017 IEEE International Symposium on*, pages 3060–3064. IEEE, 2017.

[16] Min Li, Lawrence Ong, and Sarah J Johnson. Cooperative multi-sender index coding. *IEEE Transactions on Information Theory*, 2018.

[17] Min Li, Lawrence Ong, and Sarah J Johnson. Multi-sender index coding for collaborative broadcasting: A rank-minimization approach. *IEEE Transactions on Communications*, 2018.

[18] S-YR Li, Raymond W Yeung, and Ning Cai. Linear network coding. *IEEE transactions on information theory*, 49(2):371–381, 2003.

[19] Songze Li, Mohammad Ali Maddah-Ali, and A Salman Avestimehr. Fundamental tradeoff between computation and communication in distributed computing. In *Information Theory (ISIT), 2016 IEEE International Symposium on*, pages 1814–1818. IEEE, 2016.

[20] Eyal Lubetzky and Uri Stav. Nonlinear index coding outperforming the linear optimum. *IEEE Transactions on Information Theory*, 55(8):3544–3551, 2009.

[21] Mohammad Ali Maddah-Ali and Urs Niesen. Fundamental limits of caching. *IEEE Transactions on Information Theory*, 60(5):2856–2867, 2014.

[22] Mohammad Ali Maddah-Ali and Urs Niesen. Decentralized coded caching attains order-optimal memory-rate tradeoff. *IEEE/ACM Transactions on Networking (TON)*, 23(4):1029–1040, 2015.

[23] Arya Mazumdar. On a duality between recoverable distributed storage and index coding. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 1977–1981. IEEE, 2014.

[24] Lakshmi Natarajan, Prasad Krishnan, and V Lalitha. On locally decodable index codes. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 446–450. IEEE, 2018.

[25] Michael J Neely, Arash Saber Tehrani, and Zhen Zhang. Dynamic index coding for wireless broadcast networks. *IEEE Transactions on Information Theory*, 59(11):7525–7540, 2013.
[26] Lawrence Ong, Chin Keong Ho, and Fabian Lim. The single-uniprior index-coding problem: The single-sender case and the multi-sender extension. *IEEE Transactions on Information Theory*, 62(6):3165–3182, 2016.

[27] Parastoo Sadeghi, Fatemeh Arbabjolfaei, and Young-Han Kim. Distributed index coding. In *Information Theory Workshop (ITW), 2016 IEEE*, pages 330–334. IEEE, 2016.

[28] Karthikeyan Shanmugam and Alexandros G Dimakis. Bounding multiple unicasts through index coding and locally repairable codes. *arXiv preprint arXiv:1402.3895*, 2014.

[29] Karthikeyan Shanmugam, Alexandros G Dimakis, and Michael Langberg. Local graph coloring and index coding. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 1152–1156. IEEE, 2013.

[30] Karthikeyan Shanmugam, Alexandros G Dimakis, and Michael Langberg. Graph theory versus minimum rank for index coding. In *2014 IEEE International Symposium on Information Theory*, pages 291–295. IEEE, 2014.

[31] Mehrdad Tahmasbi, Amirbehshad Shahrasbi, and Amin Gohari. Critical graphs in index coding. *IEEE Journal on Selected Areas in Communications*, 33(2):225–235, 2015.

[32] Chandra Thapa, Lawrence Ong, and Sarah J Johnson. Interlinked cycles for index coding: Generalizing cycles and cliques. *IEEE Transactions on Information Theory*, 63(6):3692–3711, 2017.

[33] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, NJ, 1996.