# A Field-Size Independent Code Construction for Groupcast Index Coding Problems

Mahesh Babu Vaddi and B. Sundar Rajan
Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru 560012, KA, India
E-mail: {vaddi, bsrajan}@iisc.ac.in

***Abstract*—The length of an optimal scalar linear index code of a groupcast index coding problem is equal to the minrank of its side-information hypergraph. The side-information hypergraph becomes a side-information graph for a special class of groupcast index coding problems known as unicast index coding problems. The number of computations required to find the minrank of a side-information graph depends on the number of edges present in the side-information graph. In this paper, we define the notion of minrank-critical edges in a side-information graph and derive some properties of minrank, which identifies minrank-non-critical edges. Using these properties we present a method to reduce the number of computations required to compute minrank. Apart from this, we give a heuristic method to compute minrank. Also, we give an heuristic algorithm to find a clique cover of the side-information graph by using some binary operations on the adjacency matrix of the side-information graph. We also give a method to convert a groupcast index coding problem into a single unicast index coding problem. Combining all these results, we give a method to construct index codes (with not necessarily optimal length) for groupcast index coding problems. The construction technique is independent of field size and hence can be used to construct index codes over the binary field. In some cases the constructed index codes are better than the best known in the literature both in terms of the length of the code and the minimum field size required.**

## I. Introduction

An index coding problem [1], comprises of a transmitter that has a set of $K$ messages $\{x_1, x_2, \ldots, x_K\}$, and a set of $m$ receivers $\{R_1, R_2, \ldots, R_m\}$. Each receiver, $R_k = (\mathcal{K}_k, \mathcal{W}_k)$, knows a subset of messages, $\mathcal{K}_k \subset X$, called its *side-information*, and demands another subset of messages, $\mathcal{W}_k \subseteq \mathcal{K}_k^{\mathsf{c}}$, called its *Want-set*. The transmitter can take cognizance of the side-information of the receivers and broadcast coded messages, called the index code. The objective is to minimize the number of coded transmissions, called the length of the index code.

An index coding problem with no restrictions on want-set and side-information is called a groupcast index coding problem. Without loss of generality a groupcast index coding problem with $m$ receivers and want-set $\mathcal{W}_k$ for $k \in [1:m] = \{1, 2, \ldots, m\}$ can be converted into another groupcast index coding problem with $\sum_{k\in[1:m]} |\mathcal{W}_k|$ receivers such that every receiver wants exactly one message. A groupcast index coding problem with $K$ messages $\{x_1, x_2, \ldots, x_K\}$ can be represented by a hypergraph $\mathcal{H}$ with $K$ vertices $\{x_1, x_2, \ldots, x_K\}$ and $\sum_{k\in[1:m]} |\mathcal{W}_k|$ number of hyperedges [4].

An index coding problem is unicast [2] if the demand sets of the receivers are disjoint. It is called single unicast if the demand sets of the receivers are disjoint and every receiver wants only one message. Any unicast index problem can be equivalently reduced to a single unicast index coding problem (SUICP). In an SUICP, the number of messages is equal to the number of receivers.

Any SUICP with $K$ messages $\{x_1, x_2, \ldots, x_K\}$ can be expressed as a side-information graph $G$ with $K$ vertices $\{x_1, x_2, \ldots, x_K\}$. In $G$, there exists an edge from $x_i$ to $x_j$ if the receiver wanting $x_i$ knows $x_j$. In a unicast index coding problem with $K$ messages and $K$ receivers, the side-information graph has $\sum_{k\in[1:K]} |\mathcal{K}_k|$ number of edges. A matrix $\mathbf{A} = (a_{i,j})$ fits $G$ if $a_{i,i} = 1$ for all $i$ and $a_{i,j}$=0 whenever $(i, j)$ is not an edge of $G$. Let $\text{rk}_q(\mathbf{A})$ denote the rank of this matrix over $\mathbb{F}_q$. The $\text{minrank}_q(G)$ is defined as

$$\text{minrank}_q(G) \triangleq \min\{\text{rk}_q(\mathbf{A}) : \mathbf{A} \text{ fits } G\}.$$

In [3] and [4], it was shown that for any given index coding problem, the length of an optimal scalar linear index code over $\mathbb{F}_q$ is equal to the $\text{minrank}_q(G)$ of its side-information graph. However, finding the minrank for any arbitrary side-information graph is NP-hard [4]. There exists a low rank matrix completion method to find the rank of a binary matrix which is also NP-hard [5].

A directed graph $G$ with $K$ vertices is called a $\kappa(G)$-partial clique [1] iff every vertex in $G$ has at least $(K - 1 - \kappa)$ outgoing edges and there exits at least one vertex in $G$ which has exactly $(K - 1 - \kappa)$ outgoing edges. For an index coding problem whose side-information graph is a $\kappa(G)$-partial clique, a maximum distance separable (MDS) code of length $K$ and dimension $\kappa+1$, over a finite field $\mathbb{F}_q$ for $q \geq K$, can be used as an index code. The $\kappa(G)$-partial clique method provides a savings of $K-\kappa-1$ transmissions when compared with the naive technique of transmitting all $K$ messages.

Tehrani *et. al* in [7] proposed a partition multicast technique to address the groupcast index coding problem. In this technique one divides the messages into partitions and consider each partition as a partial clique. The messages are partitioned in such a way that the sum of savings of all the partitions are maximized. However, the proposed partition multicast technique is suboptimal and computing it is NP-hard. The required field size in partition multicast depends on the number of messages in the partition and the number of messages known to each receiver in the partition.

In this paper, the vertices $x_i$ and $x_j$ in a side-information graph are connected with an undirected edge if receiver $R_i$

978-1-5386-6900-6/19/$31.00 ©2019 IEEE

knows $x_j$ and receiver $R_j$ knows $x_i$. Throughout we assume a finite field with characteristic 2 and use the XOR operation for convenience. However the results are easily extendable to finite fields with any characteristic.

### A. Contributions

The main contributions of this paper are summarized as follows.

- We give a method to construct index codes for groupcast index coding problems which is independent of field size. Partition multicast index codes is the best known in the literature for groupcast index coding problems and they do not exist for all fields. We give instances of groupcast index coding problems where the length of index code obtained by using the proposed method is less than that of partition multicast.
- To give a method to construct index codes for groupcast index coding problem, we develop many tools to address single unicast index coding problems. We define the notion of minrank-critical edges in a side-information graph and derive some properties of minrank, which identify minrank-non-critical edges in a side-information graph. Using these properties we present a method to reduce the number of computations required to compute the minrank. We also give a heuristic method to compute the same along with a heuristic algorithm to find a clique cover of the side-information graph. We propose a sub-optimal method to convert a groupcast index coding problem into a single unicast index coding problem.

The remaining part of this paper is organized as follows. In Section II, we derive some properties of the minrank of a side-information graph and give a method to reduce the complexity of the minrank computation problem. In Section III, we give a heuristic method for minrank computation. In Section IV, we give a method to construct index codes for groupcast index coding problems which works over every finite field. We conclude the paper in Section V. In the Appendix we give a heuristic algorithm to find the clique cover of a side-information graph.

The proofs of all the theorems and lemmas in this paper have been omitted due to space constraints. These and more examples can be found in [10].

## II. Properties of minrank of a side-information graph

In this section, we derive some properties of the minrank of the index coding problem. By using the derived properties, we provide a method to identify minrank-non-critical edges of a side-information graph. As the number of computations required to find exact value of the minrank is exponential in the number of edges present in the side-information graph, identification of every minrank-non-critical edge can reduce the number of computations required to compute the minrank by half.

In a given index coding problem with side-information graph $G$, an edge $e$ is said to be critical if the removal of $e$ from $G$ strictly reduces the capacity region. The index coding problem $G$ is critical if every edge $e$ is critical. Tahmasbi, Shahrasbi and Gohari [8] studied critical graphs and analyzed properties of critical graphs with respect to capacity region.

In this paper, we analyze properties of minrank by defining the notion of minrank-critical edges.

*Definition* 1. In a given index coding problem with side-information graph $G$, an edge $e$ is said to be minrank-critical if the removal of $e$ from $G$ strictly increases the minrank of the graph $G$. An edge $e \in E$ is said to be minrank-non-critical if the removal of $e$ from $G$ does not change the minrank of the graph $G$.

In Lemma 1-Lemma 4 and in Theorem 1, we establish some properties of minrank that would be useful to identify minrank non-critical edges in a side-information graph.

*Lemma* 1. Let $G$ be the side-information graph of an SUICP with $K$ messages. Let $G^{(k)}$ be the side-information graph after removing all the incoming and outgoing edges associated with a vertex $x_k$ for any $x_k \in V(G)$. Then, the minrank of $G^{(k)}$ is at most one greater than the minrank of $G$.

*Lemma* 2. Consider the side-information graph $G$ in Fig. 1 in which $V(G) = V(G_1) \cup V(G_2) \cup V(G_3)$ and there are no edges between $V(G_1)$ and $V(G_3)$. Then, we have

$$\begin{aligned}&\text{minrank}(G_1) + \text{minrank}(G_3) \leq \text{minrank}(G)\\ &\leq \text{minrank}(G_1) + \text{minrank}(G_2) + \text{minrank}(G_3).\end{aligned}$$

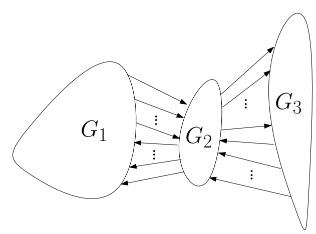


Fig. 1

*Theorem* 1. Let $G$ be a side-information graph and $G_k$ be the induced subgraph of $G$ with the vertex set $V(G) \setminus \{x_k\}$ for any $x_k \in V(G)$. If $x_k$ is not present in any directed cycle in $G$ and the minrank of $G_k$ is $n-1$ (for some positive integer $n$), then the minrank of $G$ is $n$.

*Lemma* 3. In the side-information graph $G$, if $x_k$ is not present in any directed cycle in $G$, then all the incoming and outgoing edges from $x_k$ are minrank-non-critical.

*Note* 1. A condition for critical graphs was obtained in [8] for linear index coding (one-shot or asymptotic) and for asymptotic non-linear index coding. Taking the equal rate line in capacity region, and the inverse when it touches the boundary, we can obtain the corresponding optimal broadcast rate, which in the context of scalar linear codes is minrank. By using this observation, we can obtain Theorem 1 and Lemma 3 by using the results in [8]. However, Theorem 1 and Lemma 3 are proved in [10] by using simple arguments on the fitting matrix of a side-information graph.

*Lemma* 4. Let $\mathfrak{C}$ be the set of vertices in any clique of size $t$ in the graph $G$. Let $G_C$ be the side-information graph after removing all the incoming and outgoing edges associated with the $t$ vertices in $\mathfrak{C}$, i.e., $G_C = V(G) \setminus \mathfrak{C}$. Then, the minrank of $G_C$ is at most one greater than the minrank of $G$.

*Definition* 2. Let $G$ be a side-information graph. Let $C_i = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{|C_i|}}\}$ and $C_j = \{x_{j_1}, x_{j_2}, \ldots, x_{j_{|C_j|}}\}$ be two cliques in $G$. Let $V_R = V(G) \setminus (\{x_{i_1}, x_{i_2}, \ldots, x_{i_{|C_i|}}\} \cup \{x_{j_1}, x_{j_2}, \ldots, x_{j_{|C_j|}}\})$. We say that the cliques $C_i$ and $C_j$ are cycle-free if there exist at least two vertices $x_k \in C_i$ and $x_{k'} \in C_j$ such that there is no cycle consisting of vertices only from a non-trivial subset of $\{x_k, x_{k'}\}$ and any subset of $V_R$.

The example given below illustrate Definition 2.

*Example* 1. Consider the side-information graph $G$ given in Fig. 2. In $G$, there exist two cliques $\{x_1, x_2, x_3\}$ and $\{x_4, x_5\}$ and every vertex in the clique $\{x_1, x_2, x_3\}$ is having an outgoing edge with every vertex of the clique $\{x_4, x_5\}$. In the graph $G$, the vertex $x_3$ in the clique $\{x_1, x_2, x_3\}$ is not present in any cycle comprising of vertices only from the set $V_R$. The vertex $x_5$ in the clique $\{x_4, x_5\}$ is not present in any directed cycle which comprises of vertices only from the set $V_R$. There also does not exist a directed cycle comprising of $x_3$ from the clique $\{x_1, x_2, x_3\}$ along with $x_5$ from the clique $\{x_4, x_5\}$ and vertices only from the set $V_R$. Hence, according to Definition 2, the clique $\{x_1, x_2, x_3\}$ and the clique $\{x_4, x_5\}$ are cycle-free.

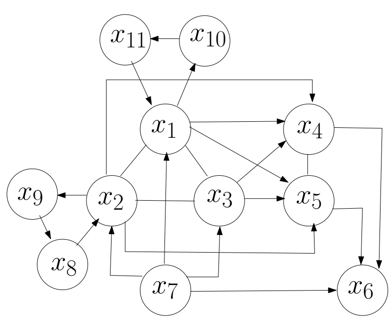


Fig. 2: Side-information graph $G$

Theorem 2 given below identifies the minrank non-critical edges between cliques.

*Theorem* 2. Let $G$ be a side-information graph. Let $C_i = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{|C_i|}}\}$ and $C_j = \{x_{j_1}, x_{j_2} \ldots, x_{j_{|C_j|}}\}$ be any two cliques in $G$ that are cycle-free. Then all the edges between $C_i$ and $C_j$ (the incoming and outgoing edges from any vertex in $C_i$ to any vertex in $C_j$) are minrank-non-critical.

Note that Definition 2 and Theorem 2 are also applicable if clique $C_i$ is a single vertex (trivial clique) or $C_j$ is a single vertex or both $C_i$ and $C_j$ are single vertices.

*Example* 2. Consider the side-information graph $G$ given in Fig. 2. According to Definition 2, the clique $\{x_1, x_2, x_3\}$ and the clique $\{x_4, x_5\}$ are cycle-free. Hence, from Theorem 2, the six edges from the clique $\{x_1, x_2, x_3\}$ to the clique $\{x_4, x_5\}$ are minrank-non-critical.

*Theorem* 3. Let $G$ be a side-information graph with $K$ vertices $\{x_1, x_2, \ldots, x_K\}$. Let $\tilde{G}$ be the graph obtained from $G$ by the following reduction procedure:

- Find a set of cliques $\{C_1, C_2, \ldots, C_t\}$ in $G$ such that all the $t$ cliques partition $V(G)$. Note that any vertex is also a trivial clique of size one. If the cliques $C_i$ and $C_j$ are cycle-free, delete all the edges between $C_i$ and $C_j$ for every $i, j \in [1:t]$.

Then, minrank$(G) =$ minrank$(\tilde{G})$.

Theorem 3 reduces the minrank computation problem into a smaller problem in terms of number of edges (number of vertices remain the same after reduction). Construction I given in next section reduces the minrank computation problem into a smaller problem in terms of both the number of vertices and the number of edges.

## III. A HEURISTIC METHOD TO REDUCE THE MINRANK COMPUTATION PROBLEM

In the following three steps, we give a heuristic approach to reduce the given minrank computation problem into another minrank computation problem which requires lesser computational complexity. We refer these three steps as Construction I in the rest of the paper.

**Construction I**

*Step 1:* Let $\{C_1, C_2, \ldots, C_t\}$ be the set of $t$ cliques in $G$. These $t$ cliques partition $V(G) = \{x_1, x_2, \ldots, x_K\}$. Note that any vertex is also a trivial clique of size one. Let $G_R$ be the graph obtained from $G$ after the following two steps:

*Step 2:* Let $\{x_{i_1}, x_{i_2}, \ldots, x_{i_{|C_i|}}\}$ be the vertices in the $i$th clique for $i \in [1:t]$. If $|\,C_i\,| \geq 2$, combine these $|C_i|$ vertices into one new vertex $y_i$. Else, leave the vertex in $C_i$ as it is.

*Step 3:* Now the number of vertices is equal to the number of cliques in $G$, that is $t$. If the number of directed edges from $C_i$ to $C_j$ in $G$ are $|C_i| \cdot |C_j|$, then introduce a directed edge from $y_i$ to $y_j$ for $i, j \in [1:t]$. Otherwise, there does not exist a directed edge from $y_i$ to $y_j$ for $i, j \in [1:t]$.

By using the procedure given in Construction I, we obtain the graph $G_R$ from graph $G$. We have minrank$(G) \leq$ minrank$(G_R)$.

In Lemma 5, we give a sufficient condition when the minrank of the graphs $G$ and $G_R$ are equal. The necessary and sufficient conditions that the side-information graph $G$ need to satisfy such that the minrank of $G$ is equal to the minrank of $G_R$ needs further investigation.

*Lemma* 5. Let $G$ be a side-information graph. Let $\{C_1, C_2, \ldots, C_t\}$ be a set of $t$ cliques in $G$. If every pair of these $t$ cliques are cycle-free, then the minrank of $G$ is equal to the minrank of $G_R$.

The lemma given below (also available in [9] in different form) establishes a relation between the index code for $G_R$ and the index code for $G$.

*Lemma* 6. Let $G$ be the side-information graph of a single unicast ICP. Let $G_R$ be the graph obtained from $G$ by using

Construction I. An index code $\mathfrak{C}$ for the ICP represented by $G_R$ can be used as an index code for the ICP represented by $G$ after replacing $y_i$ with the XOR of vertices present in $C_i$ for $i \in [1:t]$ ($y_i$ and $C_i$ are defined in Construction I).

Construction I along with Lemma 6 gives a simple index code construction procedure for unicast index coding problems. The following example illustrates Construction I.

*Example* 3. Consider the index coding problem represented by the side-information graph $G$ ($|V(G)| = 7$) given in Fig. 3. From Construction I, the reduced side-information graph $G_R$ is given in Fig. 3. In this example, the minrank of the graph $G$ and also of $G_R$ is three. The index code for the index coding problem represented by $G_R$ is $\{y_1+y_4, y_4+x_6, x_6+x_7\}$ and the index code for the index coding problem represented by $G$ is $\{\underbrace{x_1+x_2+x_3}_{y_1}+\underbrace{x_4+x_5}_{y_4}, \quad \underbrace{x_4+x_5}_{y_4}+x_6, \quad x_6+x_7\}$.

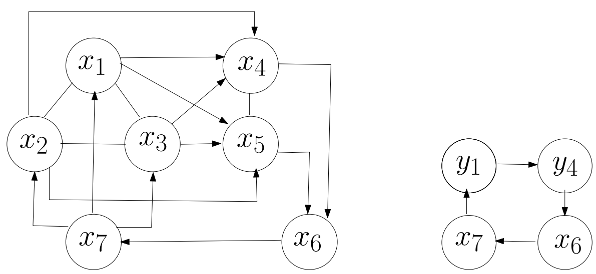


Fig. 3: Side-information graph $G$ and its reduced side-information graph $G_R$

*Note* 2. Construction I starts with $G$ and obtains $G_R$. It can also be applied on the graph $\tilde{G}$ obtained in Theorem 3. Let $\tilde{G}_R$ be the graph obtained by applying Construction I on $\tilde{G}$. In this case, the reduction of minrank computation problem by using Theorem 3 and Construction I is summarized below.

$$G \quad \underbrace{\Rightarrow}_{\text{Theorem 3}} \quad \tilde{G} \quad \underbrace{\Rightarrow}_{\text{Construction I}} \quad \tilde{G}_R$$
$$\text{minrank(G) = minrank}(\tilde{G}) \ \leq \ \text{minrank}(\tilde{G}_R).$$

The relation between the minrank of $G_R$ and $\tilde{G}_R$ requires further investigation.

In Theorem 3 and Construction I, it is assumed that the cliques in the graph are known. Note that finding a clique cover of a graph is an NP-hard problem. There exist various heuristic algorithms to find clique covers. In the Appendix, we give a heuristic algorithm to find the cliques by using the binary operations on the adjacency matrix.

## IV. Code construction for groupcast index coding problems

In this section, we give a method to convert a groupcast index coding problem into a single unicast index coding problem. This method, along with the other techniques given in this paper leads to a construction of index codes for groupcast index coding problems.

### A. Converting a groupcast ICP into a single unicast ICP

Consider a groupcast index coding problem with $K$ messages $\{x_1, x_2, \ldots, x_K\}$ and a set of $m$ receivers $\{R_1, R_2, \ldots, R_m\}$. Let $\mathcal{W}_k$ be the want-set and $\mathcal{K}_k$ be the side-information of receiver $R_k$ for $k \in [1:m]$.

*Theorem* 4. Consider a groupcast index coding problem with $K$ messages and $m$ receivers. Let $\gamma_k$ be the set of receivers wanting the message $x_k$ for $k \in [1:K]$ and $\tilde{\mathcal{K}}_k = \bigcap_{\forall R_j \in \gamma_k} \mathcal{K}_j$. Consider a single unicast index coding problem with $K$ messages $\{x_1, x_2, \ldots, x_K\}$ and $K$ receivers $\{\tilde{R}_1, \tilde{R}_2, \ldots, \tilde{R}_K\}$. The $k$th receiver $\tilde{R}_k$ wanting $x_k$ and having the side-information $\tilde{\mathcal{K}}_k$. Then, any index code for this single unicast ICP is also an index code for the groupcast ICP.

### B. Steps to construct index code for groupcast index coding problems

In the following four steps, we give a heuristic approach to construct an index code for groupcast index coding problems. We refer the following four steps as Construction II in the rest of the paper.

**Construction II**

*Step* 1. Convert the given groupcast index coding problem into a single unicast index coding problem by using the construction in Theorem 4.

*Step* 2. Find the clique cover by using any clique cover algorithm (algorithm described in appendix can be used to find clique cover in polynomial time by using binary operations on adjacency operation). Reduce the given minrank problem into a smaller problem by using Construction I.

*Step* 3. Find the cycle cover in the reduced minrank problem by using any cycle cover algorithm.

*Step* 4. Construct the index code by using the clique cover and cycle cover found in Step 2 and Step 3.

The other method that can be used to construct index codes for groupcast problems is partition multicast [7]. However, computing partition multicast is NP-hard and requires higher field size. The field size required in partition multicast depends on the number of messages in a partition and the number of messages known to each receiver in the partition. Whereas, Construction II can be used to construct index code in polynomial time and this method is independent of field size. Note that both partition multicast and Construction II are suboptimal in the length of index code.

*Example* 4. Some of the groupcast index coding problems in which the length of the index code given by Construction II is less than the length obtained from partition multicast are given in Table I. In Table I, we use $K$,$m$,$l^*$ and $l_{PM}$ to denote number of messages, number of receivers, length of index code by using Construction II and length of index code by using partition multicast respectively. The minimum field size required to construct the index code is mentioned with the length of index code in both the methods. For the groupcast index coding problem given in S. No. 3 of Table I,

| S.No | $K$ | $m$ | $\mathcal{W}_k$ | $\mathcal{K}_k$ | Index Code | $l^*$ | $l_{PM}$ |
|---|---|---|---|---|---|---|---|
| 1 | 12 | 6 | $\mathcal{W}_1=\{x_1,x_7,x_9\}$,<br>$\mathcal{W}_2=\{x_1,x_{11},x_{12}\}$,<br>$\mathcal{W}_3=\{x_2,x_{12}\}$,<br>$\mathcal{W}_4=\{x_3,x_5,x_{10}\}$,<br>$\mathcal{W}_5=\{x_4,x_6\},\mathcal{W}_6=\{x_3,x_8\}$. | $\mathcal{K}_1=\{x_2,x_3,x_{10},x_{11},x_{12}\}$<br>$\mathcal{K}_2=\{x_2,x_3,x_5,x_6,x_8,x_{10}\}$.<br>$\mathcal{K}_3=\{x_1,x_3,x_5,x_6\}$<br>$\mathcal{K}_4=\{x_4,x_6,x_7,x_8,x_{11}\}$.<br>$\mathcal{K}_5=\{x_1,x_2,x_5,x_7\},\mathcal{K}_6=\{x_4,x_9\}$. | $\mathfrak{C}=\{x_1+x_2+x_3$,<br>$x_5+x_6+x_{12}$,<br>$x_5+x_6+x_7$,<br>$x_3+x_4,x_8+x_9$,<br>$x_9+x_{10}+x_{11}\}$ | 6<br>$(\mathbb{F}_2)$ | 9<br>$(\mathbb{F}_2)$ |
| 2 | 12 | 6 | $\mathcal{W}_1=\{x_1,x_7,x_9\}$,<br>$\mathcal{W}_2=\{x_1,x_{11},x_{12}\}$,<br>$\mathcal{W}_3=\{x_2,x_{12}\}$,<br>$\mathcal{W}_4=\{x_3,x_5,x_{10}\}$,<br>$\mathcal{W}_5=\{x_4,x_6\},\mathcal{W}_6=\{x_3,x_8\}$. | $\mathcal{K}_1=\{x_2,x_3,x_{10},x_{11},x_{12}\}$<br>$\mathcal{K}_2=\{x_2,x_3,x_5,x_6,x_8,x_{10}\}$.<br>$\mathcal{K}_3=\{x_1,x_3,x_5,x_6\}$<br>$\mathcal{K}_4=\{x_4,x_6,x_7,x_8,x_{11}\}$.<br>$\mathcal{K}_5=\{x_1,x_2,x_5,x_7\}$,<br>$\mathcal{K}_6=\{x_4,x_5,x_6,x_9\}$. | $\mathfrak{C}=\{x_1+x_2+x_3$,<br>$x_5+x_6+x_{12}$,<br>$x_5+x_6+x_7$,<br>$x_3+x_4,x_8+x_9$,<br>$x_9+x_{10}+x_{11}\}$ | 6<br>$(\mathbb{F}_2)$ | 8<br>$(\mathbb{F}_{13})$ |
| 3 | 10 | 8 | $\mathcal{W}_1=\{x_1,x_2,x_{10}\}$,<br>$\mathcal{W}_2=\{x_3,x_5,x_{10}\}$,<br>$\mathcal{W}_3=\{x_4,x_9\},\mathcal{W}_4=\{x_7\}$,<br>$\mathcal{W}_5=\{x_4,x_8\},\mathcal{W}_6=\{x_6\}$,<br>$\mathcal{W}_7=\{x_1,x_4\},\mathcal{W}_8=\{x_6,x_9\}$. | $\mathcal{K}_1=\{x_3,x_4,x_5\}$,<br>$\mathcal{K}_2=\{x_2,x_4,x_7,x_8\}$,<br>$\mathcal{K}_3=\{x_1,x_6,x_{10}\},\mathcal{K}_4=\{x_8,x_9,x_{10}\}$,<br>$\mathcal{K}_5=\{x_6,x_7,x_9\},\mathcal{K}_6=\{x_2,x_3,x_{10}\}$,<br>$\mathcal{K}_7=\{x_5,x_6,x_{10}\},\mathcal{K}_8=\{x_1,x_2,x_3\}$, | $\mathfrak{C}=\{x_1+x_5$,<br>$x_5+x_7+x_8$,<br>$x_7+x_8+x_9$,<br>$x_2+x_3+x_4$,<br>$x_4+x_6,\ x_{10}\}$ | 6<br>$(\mathbb{F}_2)$ | 7<br>$(\mathbb{F}_{11})$ |

TABLE I: Some instances of the groupcast index coding problem where the length of the index code given by Construction II is less than the length obtained from partition multicast.

the minimum field size required in partition multicast is $\mathbb{F}_{11}$, whereas Construction II gives the index code in $\mathbb{F}_2$ and the length of index code given by Construction II is one less than that of partition multicast.

## V. Discussion

By developing various tools for the most general groupcast index coding problem, we give a method to construct index codes for groupcast index coding problems. The construction technique is independent of field size and hence can be used to construct index codes over the binary field.


## Acknowledgement

This work was supported partly by the Science and Engineering Research Board (SERB) of Department of Science and Technology (DST), Government of India, through J.C. Bose National Fellowship to B. Sundar Rajan.


## References


[1] Y. Birk and T. Kol, "Coding-on-demand by an informed-source (ISCOD) for efficient broadcast of different supplemental data to caching clients", in IEEE *Trans. Inf. Theory,*, pp.2825-2830, June 2006.

[2] L Ong and C K Ho, "Optimal index codes for a class of multicast networks with receiver side information", in *Proc. IEEE ICC*, 2012, pp. 2213-2218.

[3] Z. Bar-Yossef, Z. Birk, T. S. Jayram, and T. Kol, "Index coding with side information", in IEEE *Trans. Inf. Theory,*, vol. 57, no.3, pp.1479-1494, Mar. 2011.

[4] S. H. Dau, V. Skachek, and Y. M. Chee, "Error correction for index coding with side information", in IEEE *Trans. Inf. Theory,*, vol. 59, no.3, pp.1517-1531, Mar. 2013.

[5] B. Recht. M. Fazel and P. Parrilo, "Guaranteed minimum-rank solutions on linear matrix equations via nuclear norm minimization," SIAM Review, vol. 52, no. 3, pp. 471-501, 2010.

[6] S. Kwak, J. So, and Y. Sung, "An extended least difference greedy clique-cover algorithm for index coding," ISIT 2014, Australia, 2014.

[7] A. S. Tehrani, A. G. Dimakis, and M. J. Neely, "Bipartite index coding," in *Proc. IEEE ISIT 2012*, pp. 2246–2250.

[8] M. Tahmasbi, A. Shahrasbi and A. Gohari, "Critical graphs in index coding", in *Journal in selected areas of Communications*, vol. 33, no.2, pp.225-235, Feb. 2015.

[9] C. Thapa, L. Ong and J. Johnson, "Interlinked cycles for index coding: Generalizing cycles and cliques", *in IEEE Trans. Inf. Theory,*, vol. 63, no.9, pp.3692-3711, Jun. 2017.

[10] M. B. Vaddi and B. S. Rajan, "A Field-size independent code construction for groupcast index coding problems," in arXiv: 1901.00164v1 [cs.IT] 01 Jan 2019.


## Appendix

## A HEURISTIC ALGORITHM TO FIND A CLIQUE COVER

Birk *et al.* [1] proposed least difference greedy (LDG) clique cover algorithm to find the cliques in a side-information graph. Kwak *et al.* [6] improved the LDG algorithm by proposing extended least difference greedy (ELDG) clique cover algorithm to find the cliques in a side-information graph. However, the way the cliques are found in both the LDG and ELDG algorithms depends both on undirected and directed edges in the side-information graph, whereas directed edges do not contribute to cliques.

In [10], we give a method for the heuristic search of the cliques in a side-information graph by using binary operations on the adjacency matrix $\mathbf{A}$. This algorithm computes the Hadamard product of $\mathbf{A}^{\mathsf{T}}$ with $\mathbf{A}$ and this operation gives all cliques of size two in $G$. Then the algorithm combines the rows and columns corresponding to all cliques of size two. Let the resultant matrix be $\mathbf{A}_1$ and let the corresponding side-information graph whose adjacency matrix $\mathbf{A}_1$ be $G_1$. Then, the algorithm computes the Hadamard product of $\mathbf{A}_1^{\mathsf{T}}$ with $\mathbf{A}_1$ and this operation gives all cliques of size two in $G_1$. Then, the algorithm combines the rows and columns corresponding to all cliques of size two in $G_1$. This procedure is continued till the Hadamard product gives all zero matrix and the algorithm outputs the set of all cliques.

LDG and ELDG algorithms use fitting matrix to find cliques, whereas the algorithm presented in [10] uses the adjacency matrix to find cliques.