Optimal Finite-Length and Asymptotic Index Codes for Five or Fewer Receivers

Lawrence Ong

***Abstract*—Index coding models broadcast networks in which a sender sends different messages to different receivers simultaneously, where each receiver may know some of the messages a priori. The aim is to find the minimum (normalised) index codelength that the sender sends. This paper considers unicast index coding, where each receiver requests exactly one message, and each message is requested by exactly one receiver. Each unicast index-coding instances can be fully described by a directed graph and vice versa, where each vertex corresponds to one receiver. For any directed graph representing a unicast index-coding instance, we show that if a maximum acyclic induced subgraph (MAIS) is obtained by removing two or fewer vertices from the graph, then the minimum index codelength equals the number of vertices in the MAIS, and linear codes are optimal for the corresponding index-coding instance. Using this result, we solved all unicast index-coding instances with up to five receivers, which correspond to all graphs with up to five vertices. For 9819 non-isomorphic graphs among all graphs up to five vertices, we obtained the minimum index codelength for all message alphabet sizes; for the remaining 28 graphs, we obtained the minimum index codelength if each message can be bijectively mapped to a vector of even length. This work complements the result by Arbabjolfaei et al. (ISIT 2013), who solved all unicast index-coding instances with up to five receivers in the asymptotic regime, where the message alphabet size tends to infinity.**

***Index Terms*—Index coding, broadcast with side information, graph theory, finite-length codes**

I. Introduction

Index coding [1], [2] studies noiseless one-hop broadcast networks, with one sender and multiple receivers. The sender has a set of messages, and each receiver wants a message subset, while knowing another message subset a priori. To this end, the sender encodes the messages into an index codeword and presents the codeword to all the receivers. The index codeword must enable each receiver to decode its requested message subset. In majority of the work on index coding, the aim is to minimise the normalised index codelength. Index coding have been receiving much attention lately, partly due to its equivalence to network coding [3]–[5].

To date, different index code construction techniques have been proposed [2], [6]–[11], but none are optimal in general. Among them, composite coding [10] have been shown to achieve the optimal (i.e., minimum) normalised codelength asymptotically (as the message size tends to infinity) for *unicast* index coding—where each receiver requests only one message, and each message is requested by only one receiver—if there are five or fewer receivers. In a more general setting (not necessarily unicast), Unal and Wagner [11] solved all index-coding instances with three receivers in the asymptotic regime.

In this paper, we consider unicast index coding where the message alphabet size is finite, and derive the optimal index codelength for all instances in this class with five or fewer receivers. Our result uses combinatorics and is derived based on our graph-theoretic result that shows that for any directed graph in which no two cycles are disjoint, if the a maximum acyclic induced subgraph (MAIS) is obtained by removing two or fewer vertices from the graph, then there must exist a subgraph of a certain form (see Figure 1). We incidentally showed that linear index codes are optimal for all unicast index-coding instances with up to and including five receivers.

The rest of the paper is organised as follows: We formally define unicast index coding in Section II. We survey existing results and summarise our contributions in this paper in Section III. We present our results in two parts: Section IV for graphs with specific MAIS values, and Section V for graphs with five or fewer vertices.

II. Index Coding: Definition and Notation

A. Unicast index coding and information-flow graph

A unicast index-coding instance consists of a single sender and multiple receivers $[n] \triangleq \{1, 2, \dots, n\}$. The sender has n messages, denoted by $\boldsymbol{X} = [X_1 X_2 \cdots X_n]$, where X_i for each $i \in [n]$ is independent and uniformly distributed over a finite alphabet $\mathcal{X}$. For a subset of integers $I = \{i_1, i_2, \dots, i_{|I|}\}$ where $i_1 < i_2 < \cdots < i_{|I|}$, let $\boldsymbol{X}_I \triangleq [X_{i_1} X_{i_2} \cdots X_{i_{|I|}}]$. Each receiver $i \in [n]$ has a priori knowledge of $\boldsymbol{X}_{K_i}$ for some $K_i \subseteq [m] \setminus \{i\}$, and needs to decode X_i. The sender is to encode $\boldsymbol{X}$ and present the coded symbols to all receivers, such that each receiver $i \in [n]$ uses the messages $\boldsymbol{X}_{K_i}$ it already knows to decode $\boldsymbol{X}_i$. The aim is for the sender to minimise its transmitted information through the channel so that each receiver can recover its requested message. Each unicast index-coding instance is completely defined by $\{K_i\}_{i=1}^n$ and $\mathcal{X}$.

A unicast index-coding instance can be represented by a directed graph G with a set of vertices, $V(G) = [n]$, and a set of arcs, $A(G)$. An arc from vertex i to vertex j, denoted by $(i \rightarrow j) \in A(G)$, exists if and only if receiver i knows x_j a priori. This means the side information of receiver i is $K_i = N^+_G(i)$, where $N^+_G(i)$ is the out-neighbourhood of i in G. By definition, there is no self loop or parallel arcs. This representation is known as the *side-information graph* [2].

This research was supported under Australian Research Council's Future Fellowship (project number FT140100219) and Discovery Project (project number DP150100903) funding schemes.

This paper was presented in part at the 2014 IEEE International Symposium on Network Coding (NetCod) and the 2014 IEEE International Symposium on Information Theory (ISIT).

B. Index codes

Let $d_G^+(i)$ denote the out-degree of the vertex i in graph G.

Definition 1: An index code $(\phi, \{\psi_i\})$ for an index-coding instance G with message alphabet $\mathcal{X}$ consists of the following:

1) A sender encoding function $\phi : \mathcal{X}^n \mapsto \mathcal{Y}^p$, for some finite alphabet $\mathcal{Y}$ and a positive integer $p \in \mathbb{Z}^+$; and
2) A receiver decoding function $\psi_i : \mathcal{Y}^p \times \mathcal{X}^{d_G^+(i)} \mapsto \mathcal{X}$, each for a receiver $i \in [n]$,

such that $X_i = \psi_i(\phi(\boldsymbol{X}), \boldsymbol{X}_{N_G^+(i)})$.

C. Asymptotic vs finite-length index codes

Fix G and $|\mathcal{X}| = m^t$, for some integers $m \geq 2$ and $t \geq 1$. The index codelength, in bits, for an index code $(\phi, \{\psi_i\})$ is $p \log_2 |\mathcal{Y}|$ bits (need not be an integer); the *normalised* codelength, or commonly referred to as the broadcast rate, is denoted as

$$\ell_{m^t}(G) \triangleq \frac{p \log_2 |\mathcal{Y}|}{\log_2 |\mathcal{X}|} = p \log_{|\mathcal{X}|} |\mathcal{Y}| \tag{1}$$

transmitted bits per message bit per receiver. For the rest of this paper, unless otherwise stated, we refer to the normalised codelength simply as codelength.

So, for a given message alphabet size $|\mathcal{X}| = m^t$, the minimum[1] codelength, over all possible index codes, is given by

$$r_{m^t}(G) \triangleq \min_{\phi, \{\psi_i\}} \ell_{m^t}(G). \tag{2}$$

Furthermore, we define the *optimal* index codelength (or the optimal broadcast rate) for an index-coding instance G, over all message alphabet sizes and all index codes, as

$$r(G) \triangleq \inf_{m,t} \min_{\phi, \{\psi_i\}} \ell_{m^t}(G) = \inf_{m,t} r_{m^t}(G). \tag{3}$$

The optimal index codelength is also known as the *beta capacity* $\beta(G)$.

We now show that the optimal index codelength can be obtained by taking the limit of $r_{m^t}(G)$ as $t \to \infty$ for any m, stated in the following proposition:

Proposition 1: For any m,

$$\lim_{t \to \infty} r_{m^t}(G) = \inf_t r_{m^t}(G) = r(G). \tag{4}$$

Proof: We fix m and vary t. Denote the minimum codelength, in bits, by $r^{\text{b}}_{m^t} \triangleq \min_{\phi, \{\psi_i\}} p \log_2 |\mathcal{Y}|$. Note that $r^{\text{b}}_{m^{t_1+t_2}} \leq r^{\text{b}}_{m^{t_1}} + r^{\text{b}}_{m^{t_2}}$, i.e., the sequence $\{r^{\text{b}}_{m^t}\}_{t=1}^{\infty}$ is subadditive.[2] By Fekete's Subadditive Lemma,

$$\lim_{t \to \infty} r_{m^t}(G) = \frac{1}{\log_2 m} \lim_{t \to \infty} \frac{r^{\text{b}}_{m^t}(G)}{t} \tag{5a}$$

$$= \frac{1}{\log_2 m} \inf_t \frac{r^{\text{b}}_{m^t}(G)}{t} \tag{5b}$$

$$= \inf_t r_{m^t}(G), \tag{5c}$$

for any fixed G and m. This proves the first equality in (4).

From the definition (3), for any $\epsilon > 0$, we can always find some m' and t' such that $r_{(m')^{t'}}(G) < r(G) + \epsilon$, using some index code $(\phi, \{\psi_i\})$ with codewords on $\mathcal{Y}^p$. This means $\frac{p \log_2 |\mathcal{Y}|}{t' \log_2 m'} < r(G) + \epsilon$. By concatenating this index code $b \in \mathbb{Z}^+$ times, we get codewords on $\mathcal{Y}^{bp}$ for the message alphabet size $(m')^{bt'}$, with a length of $\frac{bp \log_2 |\mathcal{Y}|}{bt' \log_2 m'} < r(G) + \epsilon$. Note that this concatenated code can be used as an index code for any message alphabet of size $|\mathcal{X}| = m^t$ as long as $m^t < (m')^{bt'}$ with zero padding, giving an index code of length $\frac{bp \log_2 |\mathcal{Y}|}{t \log_2 m}$. For any fixed m', t', p, and $|\mathcal{Y}|$, we can choose **any** m and sufficiently large integers t and b, such that $\frac{p \log_2 |\mathcal{Y}|}{\frac{t}{b} \log_2 m} - \frac{p \log_2 |\mathcal{Y}|}{t' \log_2 m'} \triangleq \eta > 0$ can be made as small as desired. Noting that $r(G) \leq r^t_m(G)$ by definition, and that $r_{m^t}(G) \leq \frac{bp \log_2 |\mathcal{Y}|}{t \log_2 m} < r(G) + \epsilon + \eta$ for any arbitrarily small $\epsilon, \eta > 0$, we have the second equality in (4) for any m. ■

It follows from Proposition 1 and subadditivity of the sequence $\{r^{\text{b}}_{m^t}\}_{t=1}^{\infty}$ that, for any m and t,

$$r(G) \leq r_{m^t}(G) \leq r_{m^1}(G). \tag{6}$$

We say that $r(G)$ is the (normalised) optimal *asymptotic* index codelength for G, when the length of the message vector, t, tends to infinity, and $r_{m^t}(G)$ is the optimal *finite-length* index codelength, where the messages are each a length-t vector over an alphabet of size m. The latter is also known as the *one-shot* index codelength [12].

Remark 1: We will see in Section V-A later that choosing $|\mathcal{Y}| = m$ for the finite-length case, i.e., finite t, may give a suboptimal index codelength.

D. Linear codes

Definition 2: (Linear codes) Re-write the encoding function as $\phi = [\phi_1 \phi_2 \cdots \phi_p]$, where $\phi_i : \mathcal{X}^n \mapsto \mathcal{Y}$, and consider the following three cases:

1) $\mathcal{X} = \mathcal{Y} = \mathbb{F}_q$, where $\mathbb{F}_q$ is a q-element finite field for some prime power q: If each ϕ_i is a linear function over the field $\mathbb{F}_q$, i.e., $\phi_i(\boldsymbol{X}) = \sum_{j=1}^n k_{ij} X_j \in \mathbb{F}_q$, for some $k_{ij} \in \mathbb{F}_q$, the index code is *scalar linear over the field* $\mathbb{F}_q$.
2) $\mathcal{X} = \mathbb{F}_q^t$ and $\mathcal{Y} = \mathbb{F}_q$: If $\phi_i : \mathbb{F}_q^{tn} \mapsto \mathbb{F}_q$ is a linear function over $\mathbb{F}_q$, then the index code is *vector linear over the field* $\mathbb{F}_q$.
3) $\mathcal{X} = \mathcal{Y} = \mathcal{F}$, for any finite alphabet $\mathcal{F}$: Without loss of generality, let $\mathcal{F} = \{0, 1, \ldots, |\mathcal{F}| - 1\}$. If $\phi_i : \mathcal{F}^n \mapsto \mathcal{F}$ is linear, meaning that $\phi_i(\boldsymbol{X}) = \sum_{j=1}^n k_{ij} X_j$, where the addition and the multiplication are defined over integer modulo-$|\mathcal{F}|$, then the index code is *scalar linear over the ring* $\mathcal{F}$.
4) $\mathcal{X} = \mathcal{F}^t$ and $\mathcal{Y} = \mathcal{F}$, where $\mathcal{F} = \{0, 1, \ldots, |\mathcal{F}| - 1\}$: If $\phi_i : \mathcal{F}^{tn} \mapsto \mathcal{F}$ is a linear function over integer modulo-$|\mathcal{F}|$, then the index codes is *vector linear over the ring* $\mathcal{F}$.

III. Related Results and Main Contributions

A. Existing lower bounds

Bar-Yossef, Birk, Jayram, and Kol [2] proposed a graph-theoretic lower bound on $r_2(G)$, by considering its acyclic

[1] The minimum exists because $1 \leq \ell_{m^t}(G) \leq n$, where the lower bound follows as each receiver must decode one message $X_i \in \mathcal{X}$ (which is independent of all its side information) from the codeword $\phi(\boldsymbol{X}) \in \mathcal{Y}^p$; the upper bound is obtained by sending all messages uncoded $\phi(\boldsymbol{X}) = \boldsymbol{X}$. So, $r_{m^t}(G)$ is obtained by minimising $\ell_{m^t}(G)$ over $|\mathcal{Y}| \in \{2, \ldots, |\mathcal{X}|^n\}$ and $p \in [\lceil n \log_2 |\mathcal{X}| \rceil]$.

[2] To see this, we can always concatenate the index codes for the index-coding instances with message alphabet sizes m^{t_1} and m^{t_2} to get an index code for the instance with message alphabet size $m^{t_1+t_2}$.

subgraph. Denote the number of vertices in a *maximum acyclic induced subgraph*[3] (MAIS) of G by $\mathsf{mais}(G)$. The lower bound is readily extended to any message size, as follows:

Lemma 1: For any m and t,

$$\mathsf{mais}(G) \leq r(G) \leq r_{m^t}(G). \tag{7}$$

Let the random variables of an index codeword be defined as $[Y_1 Y_2 \cdots Y_p] = \boldsymbol{Y} = \phi(\boldsymbol{X})$. Blasiak, Kleinberg, and Lubetzky [13] proposed a lower bound by showing that the joint entropies of $\{\boldsymbol{X}, \boldsymbol{Y}\}$ must satisfy the following constraints:

1) *Decodability:* Consider any receiver $i \in [n]$. Knowing $\boldsymbol{Y}$ and $\boldsymbol{X}_{N^+_G(i)}$, it can decode X_i. This means $H(X_i, \boldsymbol{X}_{N^+_G(i)}, \boldsymbol{Y}) = H(\boldsymbol{X}_{N^+_G(i)}, \boldsymbol{Y})$, for each $i \in [n]$.
2) *Submodularity of entropy:* For two subsets of random variables $\mathcal{S}$ and $\mathcal{T}$, we have $H(\mathcal{S}) + H(\mathcal{T}) \geq H(\mathcal{S} \cup \mathcal{T}) + H(\mathcal{S} \cap \mathcal{T})$.
3) *Non-Shannon-type information inequalities:* See Zhang and Yeung [14] for example.

By considering these (in)equalities, linear programs can be formed to obtain lower bounds to $H(\boldsymbol{Y})$, which in turn lower bounds the index codelength as $H(\boldsymbol{Y})/\log_2|\mathcal{X}| \leq \sum_{i=1}^{p} H(Y_i)/\log_2|\mathcal{X}| \leq \ell_{m^t}(G)$, for any index code and any choice of m and t.

There are infinitely many non-Shannon inequalities, and invoking all these inequalities in the linear program gives $r(G)$ [15, page 37].

In fact, noting that $r_{m^t}(G') \leq r_{m^t}(G)$, for any vertex-induced subgraph G' of G [2, Proposition 9], the MAIS lower bound (7) can be obtained by solving a linear program with decodability constraints.

In this paper, we will construct MAISs and linear programs invoking the first two types of constraints (i.e., decodability and submodularity) to obtain lower bounds to $r_{m^t}(G)$. Non-Shannon-type information inequalities are not required for the class of index-coding instances considered in this paper.

B. Existing upper bounds (achievability)

By choosing $\mathcal{Y} = \mathcal{X}$, and sending the messages uncoded, we get an index code of length $n = |V(G)|$. This gives the following trivial upper bound on the optimal index codelength:

$$r(G) \leq r_{m^t}(G) \leq n. \tag{8}$$

Consider the special case where each message is a binary bit, i.e., $\mathcal{X} = \mathbb{F}_2$. A scalar linear code can be formed by solving a graph function *minrank*. Consider a matrix A with binary elements. We say that a binary n-by-n matrix M *fits* G if

$$m_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } (i \rightarrow j) \notin A(G), \end{cases} \tag{9}$$

where $m_{i,j}$ is the element in M on the i-th row and j-th column. The rest of the elements can be either 0 or 1. Denote the rank of M over $\mathbb{F}_2$ by $\mathsf{rk}_2(M)$, the minrank of M over $\mathbb{F}_2$ is defined as

$$\mathsf{minrk}_2(G) \triangleq \min\{\mathsf{rk}_2(M) : M \text{ fits } G\}. \tag{10}$$

[3]It is defined as an induced subgraph with the largest number of vertices.

Bar-Yossef el al. [2] proved the following lemma:

Lemma 2:

$$r_2(G) \leq \mathsf{minrk}_2(G). \tag{11}$$

Furthermore, if we restrict the encoding function $\phi(\boldsymbol{X})$ to be scalar linear, then

$$\min_{\phi, \{\psi_i\}: \phi \text{ is scalar linear}} \ell_2(G) = \mathsf{minrk}_2(G). \tag{12}$$

Blasiak et al. extended minrk_2 to higher field sizes, $\mathcal{X} = \mathbb{F}_q^t$, obtained a similar upper bound, and showed that the bound is tight if the encoding function is restricted to be scalar or vector linear.

Both the MAIS upper bound and the minrank lower bound are NP-hard to compute [16], [17], and both have been shown to be loose in some instances [2], [18]. This implies that linear or vector-linear index codes, though having practical advantages of simplifying encoding and decoding, are not necessarily optimal.

There are other upper bounds obtained by finding the number of disjoint cycles [6] and the number of cliques [1], a special structure of interlinked cycles in the graph [8], [9], the local chromatic number of the graph [7], and partitioning the graph and finding the maximum out-degree in the partition [1]. Some of these approaches require $\mathcal{X}$ to be a finite field of a sufficiently large size.

Some approaches uses Shannon random coding [10] and rate-distortion theory [11]. As expected, these approaches are non-constructive, and requires the message alphabet size $|\mathcal{X}|$ to be infinitely large. Consequently, these results are upper bounds to $r(G)$, and not to $r_{m^t}(G)$ for any finite m and t.

We will show that, for most cases considered in this paper, the interlinked-cycle cover can be used to obtain optimal scalar linear codes. Here, we briefly describe the scheme:

Definition 3: (Interlinked cycle [9]) A directed subgraph G is an interlinked cycle if and only if we can find a vertex subset $V_\mathrm{I} \subseteq V(G)$, called an inner-vertex set, such that

1) there is no directed cycle in G that contains one and only one inner vertex, and
2) for any ordered pair of inner vertices (i, j), there is one and only one path from i to j, where all other vertices in the path, if exists, are not in V_I.

Definition 4: (Interlinked-cycle cover [9]) Given an interlinked cycle G with an inner-vertex set V_I and a message alphabet $\mathcal{X}$, a scalar linear code of length $|V(G)| - |V_\mathrm{I}| + 1$ over a ring with $|\mathcal{X}|$ elements can formed as follows:

$$\sum_{i \in V_\mathrm{I}} X_i, \tag{13}$$

$$X_j + \sum_{k \in N^+_G(j)} X_k, \quad \text{for each } j \in V(G) \setminus V_\mathrm{I}. \tag{14}$$

Note that a cycle C of length L, for any $L \geq 2$, with vertices and arcs $c_1 \rightarrow c_2 \rightarrow \cdots \rightarrow c_L \rightarrow c_1$ is a special case of interlinked cycles, by choosing any two vertices therein to be the inner-vertex set. For example, by choosing $\{c_{L-1}, c_L\}$ to be the inner-vertex set, we have the following scalar linear index code of length $|V(C)| - 1$ for C:

$$X_1 + X_2,\ X_2 + X_3,\ \ldots,\ X_{L-1} + X_L. \tag{15}$$

The above code (15), also known as cyclic codes, was used by Neely et el. [6] and Ong and Ho [19].

Also note that a clique Q (a subgraph in which each vertex has an outgoing arc to every other vertex) is an interlinked cycle with all its vertices in the inner-vertex set. The interlinked-cycle cover gives an index code of length-1: $\sum_{i \in V(Q)} X_i$. This is also known as the clique cover [2].

Recall that $N^+_G(i)$ is the out-neigbourhood of i in G. Let $N^-_G(i)$ denote the in-neighbourhood of i in G.

Definition 5: (Interlinked cycle with super vertices [20]) Consider a vertex set V_s in a graph G satisfying the following conditions: For all distinct pairs $i, j \in V_\text{s}$, we have

- $(i \rightarrow j) \in A(G)$, i.e., all vertices in V_s have arcs to each other, and
- $N^+_G(i) \setminus V_\text{s} = N^+_G(j) \setminus V_\text{s}$ and $N^-_G(i) \setminus V_\text{s} = N^-_G(j) \setminus V_\text{s}$, i.e., all vertices in V_s have the same incoming and outgoing connection to vertices outside V_s in G.

We can define a new graph G' by replacing V_s (and all arcs to and from these vertices) by a *super vertex*, say p, with $N^+_{G'}(p) = N^+_G(i) \setminus V_\text{s}$ and $N^-_{G'}(p) = N^-_G(i) \setminus V_\text{s}$, for any arbitrarily chosen $i \in V_s$. If G' is an interlinked cycle with an inner-vertex set V_I, where $p \notin V_I$, then we say that G is an interlinked cycle with an inner-vertex set V_I and a super-vertex set V_s. The index code formed by the interlinked-cycle cover for G' is an index code (of the same length) for G with X_p replaced by $\sum_{i \in V_\text{s}} X_i$.

C. Existing capacity results

Although there are several different approaches to computing upper bounds to $r(G)$ and $r_{m^t}(G)$, it is not easy to determine when these bounds are tight (or not). We now present a few classes of graphs where the bounds have been shown to be tight.

Bar-Yossef et al. [2] showed that if G is acyclic, then $r(G) = r_{m^t}(G) = |G| = \mathsf{mais}(G)$.

Consider a special class of graphs G where $(i \rightarrow j) \in A(G)$ if and only if $(j \rightarrow i) \in A(G)$. This models index-coding instances with symmetrical knowledge, i.e., if receiver i knows x_j, then receiver j knows x_i. Any graph G of this type can be mapped to a corresponding undirected graph G_u with the same vertex set as G, and an edge $(i, j) \in E(G_\text{u})$ exists if and only if $(i \rightarrow j) \in A(G)$. Bar-Yossef et al. [2] found $r_2(G_\text{u})$ for the following classes of undirected side-information graphs:

- G_u is a perfect graph,
- G_u is an odd hole where $|V(G)| \geq 5$, or
- G_u is an odd anti-hole where $|V(G)| \geq 5$.

Neely, Tehrani, and Zhang [6] and Tehrani, Dimakis, and Neely [21] showed that if G consists of disjoint cycles, then $r(G) = r_{m^t}(G) = |V(G)| - N_\text{cycle}$ for all m and t, where N_cycles is the number of cycles (all being disjoint) in G. This is commonly known as *cycle cover*. Yu and Neely [22] represented index-coding instances using bipartite graphs, and found $r(G)$ for all planar bipartite graphs.

It has been verified by intensive computer calculations that composite coding [10] (derived using Shannon random-coding arguments) is optimal for all G with $|V(G)| \leq 5$, giving $r(G)$.

Unal and Wagner [11] derived the asymptotic optimal index codelength $r(\cdot)$ for all general (in the sense that each message can be requested by several receivers) index-coding instances up to three receivers. Their method is based on rate-distortion theory, which also uses Shannon random-coding arguments.

To find $r_{m^t}(G)$ by brute force, one can form the *confusion graph* [2] of G with m^{tn} vertices, and calculate the chromatic number (which is NP-complete) of the confusion graph. This method is, however, intractable as the order of the confusion graph grows exponentially with tn.

D. Main results of this paper

The main results of this paper are as follows: We find the optimal index codelength and the minimum message alphabet size required to achieve the optimal index codelength for the following classes of index-coding instances:

1) (In Section IV) For any G that can be made acyclic after removing two or fewer arcs: We derive $r(G)$, and show that $r(G) = r_{m^t}(G)$ for all integers $m \geq 2$ and $t \geq 1$.
2) (In Section V) For any G of up to five vertices (there are 9847 non-isomorphic graphs in total):
   a) For 9819 non-isomorphic graphs, we derive $r(G)$, and show that $r(G) = r_{m^t}(G)$ for all integers $m \geq 2$ and $t \geq 1$.
   b) For the remaining 28 non-isomorphic graphs, we derive $r_{m^{2k}}(G)$, and show that $r(G) = r_{m^{2k}}(G)$, for all integers $m \geq 2$ and $k \geq 1$.

Furthermore, for all the above cases, we show that linear index codes (over a ring) are optimal.

Recall that $r_2(G)$ is the solution for an index-coding instance G where each message consists of a single binary bit. The above result of $r(G) = r_2(G)$, together with linear codes in $\mathbb{F}_2$ being optimal, means that the encoding can be done bit-by-bit without loss of optimality. The advantages are of this are that (i) the encoding is simple (bit-wise XOR of the messages), and that (ii) the decoding is instantaneous. For cases where $r(G) = r_{2^2}(G)$, we can achieve the optimal broadcast rate by encoding (and decoding) two bits of messages at a time.

IV. Optimal Index Codelength when $\mathsf{mais}(G) \geq |V(G)| - 2$

A. Main result

In this section, we show the following theorem:

Theorem 1: If $\mathsf{mais}(G) \geq |V(G)| - 2$, then

$$r(G) = r_{m^t}(G) = \mathsf{minrk}_2(G) = \mathsf{mais}(G), \tag{16}$$

for any integers $m \geq 2$ and $t \geq 1$, and the minimum index codelength is achievable using scalar linear codes over a ring with m^t elements.

It follows from Theorem 1 that the minimum alphabet size required to achieve $r(G)$ is $|\mathcal{X}| = 2$, i.e., binary messages.

This theorem will be used to establish the result for all graphs up to five vertices in Section V.

Remark 2: Characterising graphs having a certain $\mathsf{minrk}_2(G)$ value is hard. Dau et al. [23] managed to characterise all *undirected* graphs whose $\mathsf{minrk}_2(G_\text{u})$ is $|V(G_\text{u})| - 2$ or $|V(G_\text{u})| - 1$, and all *directed* graphs whose $\mathsf{minrk}_2(G)$ is 2 or $|V(G)|$. They are, however, unable to characterise directed graphs whose $\mathsf{minrk}_2(G)$ is $|V(G)| - 1$ or $|V(G)| - 2$. For any directed graph G whose $\mathsf{mais}(G)$ equals $|V(G)|-1$ or $|V(G)|-2$, we show in this paper that linear index codes are optimal, meaning that $\mathsf{mais}(G) = \mathsf{minrk}_2(G)$. So, we have incidentally characterised a subset of directed graphs whose $\mathsf{minrk}_2(G)$ equals $|V(G)| - 1$ or $|V(G)| - 2$.

Proof of Theorem 1: As we know from Lemma 1 that $\mathsf{mais}(G)$ is a lower bound on $r_{m^t}(G)$, we only need to prove achievability.

Without loss of generality, let $\mathcal{X} = \{0, 1, \ldots, |\mathcal{X}| - 1\}$. We will show that scalar linear codes over the ring $\mathcal{X}$ is optimal. To this end, we choose $\mathcal{Y} = \mathcal{X}^p$, and therefore the normalised codelength is given by $\ell_{m^t}(G) = p$.

1) $\mathsf{mais}(G) = |V(G)|$: For this case, G is acyclic. As mentioned in the previous section, sending all messages uncoded (i.e., $\phi(\boldsymbol{X}) = \boldsymbol{X}$, and hence we have a linear code of length $\ell_{m^t}(G) = p = |V(G)| = n$) achieves the MAIS lower bound, and we have (16).

2) $\mathsf{mais}(G) = |V(G)|-1$: For this case, the directed graph G must contain at least one cycle; otherwise, $\mathsf{mais}(G) = |V(G)|$. Let the cycle be $C \subseteq G$.

We send a cyclic code for C and the rest of the messages $\boldsymbol{X}_{V(G)\setminus V(C)}$ uncoded, forming an index code with a codelength of $|V(G)| - 1$. The cyclic code allows all receivers $i \in V(C)$ can decode X_i. In addition, all receivers $j \in V(G) \setminus V(C)$ can decode X_j as the messages were sent uncoded.

3) $\mathsf{mais}(G) = |V(G)|-2$: There are two possibilities for G:

(3.i) There are two vertex-disjoint cycles, or

(3.ii) There are no two vertex-disjoint cycles.

For case (3.i), we code the two disjoint cycles each with a cyclic code, and send the rest of the messages in G uncoded. This achieves a codelength of $|V(G)| - 2$.[4]

For case (3.ii), we will derive Lemmas 3 (stated next), which says that if $\mathsf{mais}(G) = |V(G)| - 2$ and there is no two vertex-disjoint cycles, then G contains a subgraph G' of the form depicted in Figure 1, in which each arrow represents a path.

Now, note that G' is an interlinked cycle with inner-vertex set $\{i_1, u_1, w_1\}$. Here, for path U, we label the vertices in the path as $u_1 \to u_2 \to \cdots u_{\text{last}}$. Using the interlinked-cycle cover, we obtain a scalar linear codes of length $|V(G')| - 2$ over the ring $\mathcal{X}$ for G'. Combining this with sending the remaining messages $\boldsymbol{X}_{V(G\setminus G')}$ uncoded gives an index code with a total codelength of $|V(G)| - 2$. ■

Remark 3: In a conference version of this paper [24], we presented an alternative coding scheme that constructs a scalar linear code of length $|V(G')| - 2$ for G'.

B. Existence of a special structure: Figure 1

It is easy to obtain a saving of one for each vertex-disjoint cycle using a simple cyclic code. The main challenge of Theorem 1 is to show that for case (3.ii), even though we cannot find two vertex-disjoint cycles, we can achieve a saving of two. The following lemma is a key step.

[4] Let the two disjoint cycles be C_1 and C_2. The two cyclic codes, each for one cycle, are of length $|V(C_1)|-1$ and $|V(C_2)|-1$ respectively. Together with uncoded messages with a total length $|V(G)| - |V(C_1)| - |V(C_2)|$, we get an overall codelength of $|V(G)| - 2$.

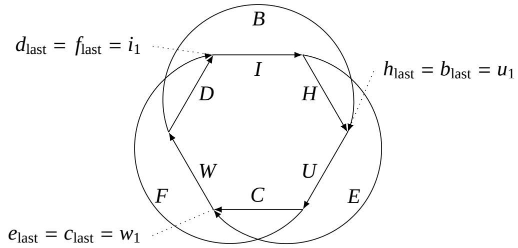

Fig. 1. An important element in proving Theorem 1 is to show that if $\mathsf{mais}(G) = |V(G)| - 2$ and condition (3.ii) is true, then G must contain a subgraph G' shown above. Here, every arrow represents a path, which is denoted by a capital letter. The paths do not share common vertices except the end points. Vertices in each path is denoted by the corresponding small letter, indexed in the direction of the arcs, e.g., path C is $c_1 \to c_2 \to \cdots \to c_{\text{last}}$. All paths except I, W, and U must contain one or more arcs.

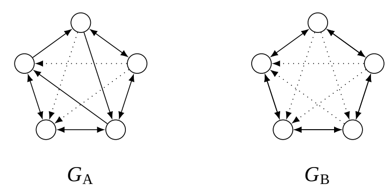

Fig. 2. Let $\mathcal{G}_{\text{s}}$ be a set of 28 non-isomorphic five-vertex graphs, formed by by removing any number (zero to three inclusive) of dotted arcs from G_{A} (this gives eight non-isomorphic graphs) and removing any number of dotted arcs from G_{B} (this gives 20 non-isomorphic graphs).

Lemma 3: If $\mathsf{mais}(G) = |V(G)| - 2$, and there are no two vertex-disjoint cycles, then G must contain a subgraph (not necessarily an induced subgraph) shown in Figure 1.

Proof: See Appendix A. ■

V. Optimal Index Codelength for All Graphs up to Five Vertices

In this section, we use Theorem 1 to obtain the optimal index codelength for graphs up to five vertices. First, we define $\mathcal{G}_{\text{s}}$ to be a set of 28 non-isomorphic five-vertex subgraphs of the two graphs in Figure 2. More specifically, $\mathcal{G}_{\text{s}}$ consist of

- all eight non-isomorphic graphs formed by removing any number (zero to three inclusive) of dotted arcs of G_{A}, and
- all 20 non-isomorphic graphs formed by removing any number (zero to five inclusive) of dotted arcs of G_{B}.

Also, let $\mathcal{G}_{1:5}$ be the set of all non-isomorphic graphs up to and including five vertices. $\mathcal{G}_{1:5}$ contains 9608 non-isomorphic graphs [25].

We now state our main results for $\mathcal{G}_{1:5} \setminus \mathcal{G}_{\text{s}}$ and for $\mathcal{G}_{\text{s}}$.

Theorem 2: For any $G \in \mathcal{G}_{1:5} \setminus \mathcal{G}_{\text{s}}$,

$$r(G) = r_{m^t}(G) = \mathsf{minrk}_2(G) = \mathsf{mais}(G), \tag{17}$$

for any integers $m \geq 2$ and $t \geq 1$. The optimal index codelength is achievable using *scalar* linear codes over a ring with m^t elements.

It follows from Theorem 2 that for any $G \in G_{1:5} \setminus G_{\text{s}}$, the minimum message alphabet size required to achieve $r(G)$ is $|\mathcal{X}| = 2$.

Theorem 3: For any $G \in \mathcal{G}_{\text{s}}$, we have that

$$2 = \mathsf{mais}(G) < r(G) = 2.5. \tag{18}$$

Number of receivers, $\|V(G)\|$		1	2	3		4		5			
Number of non-isomorphic G		1	3	16		218		9608			
		1	3	9	7	41	177	334	($\mathcal{G}_s$) 1	27	9246
Binary messages, i.e., $m = 2$, $t = 1$									*		
Messages of size m^t	$m \geq 3, t = 1$								(28)		
	$m \geq 2, t = 2$										
	⋮								⋮		
	$m \geq 2$, odd t								(28)		
	$m \geq 2$, even t										
	⋮								⋮		
	$m \geq 2, t = \infty$										

Note: The column width is not indicative of the number of non-isomorphic graphs.

Legend: * Solved by Bar-Yossef et al. [2] Solved by Arbabjolfaei et al. [10] Solved in this paper

TABLE I

Graphs for which the optimal index codelength is found for message of size m^t

In addition, if $m \geq 2$ and $t = 2k$ for some integer $k \geq 1$, then

$$r(G) = r_{m^{2k}}(G) = 2.5, \tag{19}$$

and the optimal index codelength is achievable using *vector* linear codes over a ring with m^k elements.

It follows from Theorem 3 that for any $G \in \mathcal{G}_s$, the minimum message alphabet size required to achieve $r(G)$ is $|\mathcal{X}| = 4$.

Proof of Theorem 2: Note that for any graph, we must have that

$$1 \leq \mathsf{mais}(G) \leq |V(G)|. \tag{20}$$

We now prove Theorem 2 by considering graphs of different orders. For $|V(G)| \in \{1, 2, 3\}$, we have

$$|V(G)| - 2 \leq 1 \leq \mathsf{mais}(G), \tag{21}$$

where the second inequality follows from (20). Invoking Theorem 1, we get (16) in Theorem 2.

For $|V(G)| = 4$, if $\mathsf{mais}(G) \in \{2, 3, 4\}$, then $|V(G)| - \mathsf{mais}(G) \leq 2$. We again use Theorem 1 to get (16) in Theorem 2. For the remaining case where $\mathsf{mais}(G) = 1$, any two-vertex induced subgraph is a cycle (i.e., there are arcs in both directions between any two vertices); otherwise $\mathsf{mais}(G) \geq 2$. In other words, each receiver i know all other messages $\boldsymbol{X}_{[4]\setminus\{i\}}$. So, sending a length-1 index code, $X_1 + X_2 + X_3 + X_4 \mod |\mathcal{X}|$, satisfies all receivers' requirements, and achieves the MAIS lower bound. So, we get (16), where the last equality is follows by observing that scalar linear codes are optimal.

For $|V(G)| = 5$, if $\mathsf{mais}(G) \in \{3, 4, 5\}$, then again we have (16) in Theorem 2. Also, if $\mathsf{mais}(G) = 1$, we can use the same argument for $|V(G)| = 4$ to show that the length-1 index code of $X_1 + X_2 + X_3 + X_4 + X_5 \mod |\mathcal{X}|$ is achievable and is hence optimal.

For all the above cases, scalar linear codes over the ring $\mathcal{X}$ are optimal, and the MAIS lower bound is tight. The proof of Theorem 2 is complete with Lemma 4 below, addressing the remaining case. ■

The main challenge in proving Theorem 2 is to show the following:

Lemma 4: If $|V(G)| = 5$, $G \notin \mathcal{G}_s$, and $\mathsf{mais}(G) = 2$, then

$$r(G) = r_{m^t}(G) = \mathsf{minrk}_2(G) = \mathsf{mais}(G), \tag{22}$$

for any integers $m \geq 2$ and $t \geq 1$. The optimal index codelength is achievable using *scalar* linear codes over a ring with m^t elements.

Proof of Lemma 4: See Appendix B. ■

Proof of Theorem 3: See Appendix C. ■

The results of Theorems 2 and 3 in comparison with existing results are summarised in Table I. In Table I, we consider all non-isomorphic directed graphs up to and including five vertices. The column denotes distinct non-isomorphic graphs. For example, there are 218 non-isomorphic graphs with four vertices. A cell is coloured if the optimal index codelength of the corresponding graph has been found. We have used different colours to indicate different research groups that found the optimal index codelength. The rows represent the message size, given by m^t.

For example, out of the 218 non-isomorphic graphs with four vertices, Bar-Yossef et al. have found the optimal index codelength for 41 of them, for all message sizes m^t. The 41 non-isomorphic graphs consists of the following:

- The empty graph,[5] which is both acyclic and perfect[6].
- 30 of them that are non-empty and acyclic [26].
- 10 of them that are non-empty and perfect [27]. (Note that if a graph is not empty, it cannot be both acyclic and perfect)

All the yellow cells correspond to acyclic and/or perfect graphs, except for the graph marked with an asterisk, which correspond to the (undirected) 5-cycle. For the 5-cycle, Bar-Yossef et al. showed that $r_2(G) = 3$, i.e., when the messages are binary. The lower bound was found by a brute-force exhaustive search.

Also shown in the table, Arbabjolfaei et al. found $r(G) = \lim_{r\to\infty} r_{m^t}(G)$ for all graphs up to five vertices.

Theorems 2 and 3 cover all coloured cells in the table, except the asterisked cell.

[5]An empty graph contains no arc.

[6]For a directed graph to be considered prefect (in the context of this paper), it must be a symmetric, and the corresponding undirected graph is a perfect graph.

A. Optimal codelength for $\mathcal{G}_s$ with binary messages via the confusion-graph technique

Recall that $\mathcal{G}_s$ contains all (non-strict) subgraphs of G_A and G_B in Figure 2 with none or some dotted arcs removed. Although, in Theorem 3, we have derived the optimal codelength for all $\mathcal{G} \in \mathcal{G}_s$ when m^t when t is any even integer, we do not have results for odd t.

In this section, we discuss the optimal codelength for the members of $\mathcal{G}_s$ specifically when each message is a binary bit, i.e., when $m = 2$ and $t = 1$. This corresponds to the cells in the 28 columns marked $\mathcal{G}_s$ and in the top coloured row in Table I.

1) Confusion graphs: One can use a brute-force technique of confusion graph (see Bar-Yossef et al. [2] for example) to determine the optimal codelength. We first describe confusion graphs:

Definition 6: For an index coding instance G and a message alphabet $\mathcal{X}$, its undirected confusion graph $G_{\text{confusion}}$ has $|\mathcal{X}|^{|V(G)|}$ vertices. The vertices are labelled with distinct realisations of the message tuples, i.e., $\{[x_1 x_2 \cdots x_n] \in \mathcal{X}^n\}$, where $n = |V(G)|$. An edge exists between two vertices, say $\boldsymbol{x}$ and $\boldsymbol{x}'$, if and only if there exists a receiver $j \in [n]$ such that

$$x_j \neq x'_j, \tag{23}$$

$$\text{and} \qquad \boldsymbol{x}_{N^+_G(j)} = \boldsymbol{x}'_{N^+_G(j)}. \tag{24}$$

Since message tuples corresponding to adjacent vertices cannot be mapped to the same codeword (otherwise, some receiver j cannot decode X_j due to (23) and (24)), any proper colouring scheme gives an index code (where the colours map to distinct index codewords), and vice versa. Hence, the total number of distinct codewords required for encoding equals the number of colours in the colouring scheme. Consequently,

$$r_{m^t}(G) = \frac{\log_2 \chi(G_{\text{confusion}})}{\log_2 |\mathcal{X}|}, \tag{25}$$

where $\chi(G)$ denotes the chromatic number of the undirected graph G. Note that the code here can be non-linear.

Remark 4: Using the method of confusion graph to determine $r_{m^t}(G)$ is intractable when the message alphabet size or the number of messages grows. Furthermore, this method alone cannot be used to determine $r(G)$.

2) 5-cycle with binary messages: If $\mathcal{G}$ is a 5-cycle (a member of $\mathcal{G}_s$) and the messages are binary, i.e., $|\mathcal{X}| = 2$, its confusion graph $G_{\text{confusion}}$ contains 32 vertices and 240 edges. One can use a brute-force search to find that $\chi(G_{\text{confusion}}) = 8$. This gives $r_2(\mathcal{G}) = 3$ [2]. This corresponds to the yellow cell marked with an asterisk in Table I. For this case, it turns out that scalar linear codes are optimal.

3) Other members in $\mathcal{G}_s$: For other members in $\mathcal{G}_s$, we first consider G_A and G_B in Figure 2. For these two graphs, we find that $\chi(G_{\text{confusion}}) = 7$. This means $r_2(G) = 2.8074$. The optimal codelength can be achieved by non-linear codes that map $\{0,1\}^5 \mapsto \{0,1,\ldots,6\}$, where we choose the output alphabet size to be $|\mathcal{Y}|^p = 7$.

For the rest of the members in $\mathcal{G}_s$, one can repeat this procedure to calculate $r_2(G)$.

4) Restricting the output alphabet to be a binary vector: Now, if we restrict the output alphabet to be a binary vector, we have the following:

Theorem 4: For any $G \in \mathcal{G}_s$,

$$r_{m^t}(G) \leq 3, \tag{26}$$

for any integers $m \geq 2$ and $t \geq 1$. Furthermore, if $m = 2$, $t = 1$, and $|\mathcal{Y}| = 2$, then

$$r_2(G) = 3, \tag{27}$$

and the optimal index codelength is achievable using binary scalar linear codes.

Proof: (Achievability): From Theorem 3, for any $G \in \mathcal{G}_s$, $\mathsf{mais}(G) = 2$. We can always remove some arc(s) (dotted or solid) from G to obtain a subgraph G^- where $\mathsf{mais}(G^-) = 3$ and $|V(G)^-| = 5$. With this, we have

$$r_{m^t}(G) \leq r_{m^t}(G^-) = 3, \tag{28}$$

for any $m \geq 2$, $t \geq 1$. Here, the inequality is due to Lemma 5 in Appendix B, and the equality follows from Theorem 1. So, a scalar linear code of length 3 exists for G for any m and t.

(Lower bound): We have manually found that $\chi(G'_{\text{confusion}}) = 7$, where $G'_{\text{confusion}}$ is the confusion graph of any $G' \in \{G_A, G_B\}$ for $m = 2$ and $t = 1$. From the proof of Lemma 5 in Appendix B, for any $G \in \mathcal{G}_s$ with the corresponding confusion graph $G_{\text{confusion}}$, we have that

$$\chi(G_{\text{confusion}}) \geq \chi(G'_{\text{confusion}}) \geq 7. \tag{29}$$

By definition,

$$r_{m^t}(G) = \frac{\log_2 \chi(G_{\text{confusion}})}{\log_2 |\mathcal{X}|} = \min_{\phi,\{\psi_i\}} \frac{p \log_2 |\mathcal{Y}|}{\log_2 2}, \tag{30}$$

where p is the length of the codewords.

If we restrict the codeword to be binary vectors, i.e., $|\mathcal{Y}| = 2$, we have

$$p \geq \log_2 \chi(G_{\text{confusion}}) = \log_2 7 = 2.8074. \tag{31}$$

Since p must be an integer, we have $p \geq 3$. We complete the proof by noting the existence of length-3 scalar linear codes. ■

VI. Conclusion

In this paper, we have studied unicast index coding, a special class of index coding where each receiver requests only one message, and each message is requested by only one receiver. To find the optimal index codelength and optimal index codes, we have used a graphical approach of representing each index-coding instance by a directed graph. We first derived the optimal index codelength for all graphs whose order at most two more than that of its maximum acyclic induced subgraph. We then use this result, combined with a combinatoric approach, to derive the optimal index codelength for all graphs with five or fewer vertices. We also showed that linear codes are optimal for all graphs in these two classes. While existing results give the optimal index codelength for all graphs with five or fewer vertices when the message alphabet size tends to infinity, in this work, we find the optimal codelength when the message alphabet sizes are finite.

Appendix A
Proof of Lemma 3: A Special Configuration

Recall that G must satisfy these two conditions:

- (C1) $\mathsf{mais}(G) = 2$.
- (C2) There are no two disjoint cycles in G.

We first give an intuition for Lemma 3, by showing that there must exists three joint cycles[7] in G, in Subsection A-A. In Subsections A-B to A-F, we prove that these three joint cycles must assume the configuration in Figure 1.

A. The existence of three joint cycles

Since $\mathsf{mais}(G) = 2$, let $V_r = \{u, v\}$ be the vertex set removed from from G to get an MAIS. We first show the following:

Proposition 2: There exist three cycles in G, each containing either u, v, or both u and v.

Proof: Every cycle must contain u, v, or both. Otherwise, removing u and v will not give an acyclic induced subgraph.

Suppose that there is only one cycle in G. Removing any vertex from the cycle gives an acyclic induced subgraph. Hence, $|V(G)| - \mathsf{mais}(G) = 1$. (Contradiction)

Suppose that there are only two cycles in G. Note that these two cycles cannot be vertex-disjoint, as per condition (C2) above. So, these two cycles must shared at least one vertex, and removing only this shared vertex gives an acyclic induced subgraph, i.e., $|V(G)| - \mathsf{mais}(G) = 1$. (Contradiction)

So, there must exist at least three cycles. ■

We further show some properties of these three cycles:

Proposition 3: There exist three cycles in G, where

1) any two cycles must have at least one common vertex, and
2) the three cycles do not have any common vertex.

Proof: It follows from Proposition 2 that there are at least three cycles. As no two cycles are vertex-disjoint, we have property 1. Arbitrarily select one cycle, say C'. Consider every other cycle $C_k \neq C'$, and denote the set of common vertices between C_k and C' as $V_{\text{common}}(k) \triangleq V(C_k) \cap V(C')$. Since every C_k shares some vertex with C', we have $V_{\text{common}}(k) \neq \emptyset$.

Now suppose that $\bigcap_{\text{all } C_k \neq C'} V_{\text{common}}(k) \neq \emptyset$, meaning that some vertex is shared among all cycles. Then removing only this vertex from G would have resulted in an acyclic subgraph (contradiction). So, there must exist two cycles, say C_1 and C_2, where $V_{\text{common}}(1) \cap V_{\text{common}}(2) = \emptyset$. Selecting C', C_1, and C_2 gives property 2. ■

Denote the subgraph formed by the three cycles in Proposition 3 by G_{sub}. We have the following:

Proposition 4: The subgraph G_{sub}, formed by the three cycles in Proposition 3, satisfies both conditions (C1) and (C2).

Proof: Since G cannot contain two vertex-disjoint cycles, so does any of its subgraphs. We have condition (C2). Denote by N the minimum number of vertices we need to remove to make G_{sub} acyclic. From Proposition 3, there is no common vertex among the three cycles. So, removing any one vertex will not disconnect all three cycles simultaneously, i.e., $N \geq 2$. On the other hand, we only need to remove two vertices, V_r, to make G acyclic. So, removing $V_r \cap V(G_{\text{sub}})$ from G_{sub} will definitely make it acyclic, i.e., $N \leq 2$. So, we have condition (C1). ■

Note that these three cycles, G_{sub}, capture all the constraints we impose on G in Lemma 3.

B. The three joint cycles must assume Figure 1

We will proceed to show that G_{sub} must assume the configuration in Figure 1. We will build the configuration from a cycle, say C_1, in G_{sub}. We call it the *centre cycle*. We re-label the vertices in G_{sub} such that the vertices in C_1 are in ascending order in the direction of the arcs, i.e., $1 \rightarrow 2 \rightarrow \cdots \rightarrow (|V(C_1)| - 1) \rightarrow |V(C_1)| \rightarrow 1$, where the choice of vertex 1 is arbitrary.

For any path P that originates from vertex b and terminates at vertex c, i.e., $b \rightarrow \cdots \rightarrow c$, we refer to all $\{z : z \in V(P) \setminus \{b, c\}\}$ as *internal vertices*. Here, we allow $b = c$; in such a case, P is a cycle.

We first show the following:

Proposition 5: Consider the subgraph G_{sub} and the cycle C_1 in the subgraph. Every arc not in C_1 belongs to some *outer path*, defined as a path that originates from a vertex in C_1 and terminates at a vertex (which can be the same vertex) in C_1, but with all arcs and all internal vertices (if exists) not in C_1.

Proof: Since G_{sub} is constructed by three cycles, any arc, say $(i \rightarrow j)$, not in C_1 must belong to either C_2 or C_3 (or both). Furthermore, from Proposition 3, C_2 and C_3 must each share some vertex with C_1. Hence, $(i \rightarrow j)$ must belong to an outer path that originates from C_1 and terminates at C_1. ■

Note that the outer paths cannot form any cycle outside C_1. Otherwise, we have two vertex-disjoint cycles, and this violates condition (C2).

It follows from Proposition 5 that G_{sub} consists of only a cycle C_1 and outer paths (from C_1 and back to C_1). Figure 4(a) shows an example of G_{sub} where C_1 is marked with thick arrows and all outer paths thin arrows.

We now prove a key proposition for proving Lemma 3.

Proposition 6: Remove vertex 1 in C_1. There exists another cycle in G_{sub} if and only if there is an outer path from some $b \in V(C_1) \setminus \{1\}$ to some $c \in V(C_1) \setminus \{1\}$, where $b \geq c$.

Proof: [The converse:] We remove vertex 1. If there is another cycle, then there is a vertex (not vertex 1) in C_1 that has a path back to itself (this is because any cycle must share some vertex with C_1). This cannot happen if every outer path terminates at a higher-indexed vertex (we can ignore all outer paths that originate or terminate at vertex 1 as the vertex has been removed). So, there must exist an outer path with $b \geq c$.

[The forward part:] Clearly, if $b = c$, we have another cycle formed by the outer path. Otherwise, i.e., $b > c$, the outer path and the path along C_1 from c to b form a cycle. See Figure 3(a) for an example. ■

Next, we define a *looping* outer path as an outer path that originates and terminates at the same vertex in C_1. The graph G_{sub} can be categorised as follows:

- there exists at least one looping outer path (Case 1), or

[7] Here, by joint cycles, we mean cycles that are not disjoint. We avoid using the term interlinked cycles, as they refer to a specific configuration in this paper (see Definition 3).

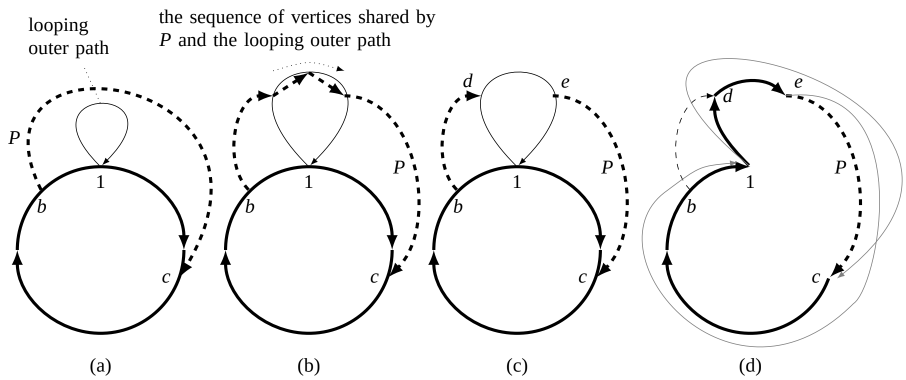

Fig. 3. Case 1 where there exists a looping outer path (drawn with thin solid lines) that starts and ends at vertex 1. The centre cycle C_1 is drawn with thick lines, and the second outer path (denoted as P) from b to c, dashed lines. To get another cycle after removing vertex 1, we must have that $1 < c \leq b \leq |V(C_1)|$, as shown in subfigure (a). However, there are two vertex-disjoint cycles in subfigure (a). So, P must touch the looping outer path, as shown in subfigure (b). Taking the segment of P from C_1 to the looping outer path, and that from the looping outer path back to C_1, we have subfigure (c). We can re-draw the path from 1 to c and that from e to 1 in subfigure (c) to get subfigure (d), where we have drawn the new centre cycle with thick lines.

- there is no looping outer path (which we will further divide into Cases 2 and 3).

We will show that in any case, we have Figure 1.

C. Case 1: There exists a looping outer path

Suppose that there exists a looping outer path from and to vertex $1 \in V(C_1)$. This incurs no loss of generality as the choice of vertex 1 is arbitrary. Removing vertex 1 disconnects both cycle C_1 and the cycle formed by the looping outer path. Recall that we need to remove two vertices to disconnect all cycles in G_{sub}. So, there must exist another cycle in G_{sub}.

From Proposition 6, there exists another outer path P from $b \in V(C_1) \setminus \{1\}$ to $c \in V(C_1) \setminus \{1\}$, where $b \geq c$. The outer path P must share some vertex with the looping outer path; otherwise there exist two cycles as shown in Figure 3(a).

Re-label the internal vertices of the looping outer path in ascending order, as follows: $1 \rightarrow (|V(C_1)| + 1) \rightarrow (|V(C_1)| + 2) \rightarrow \cdots \rightarrow (|V(C_1)| + L) \rightarrow 1$, where L is the number of internal vertices. It follows that the sequence of vertices shared by P and the looping outer path (in the order of the direction of P) must be in ascending order (see Figure 3(b)); otherwise, a cycle forms outside C_1.

See Figure 3(c). Consider only the following segments of P: (i) from b to the vertex where P first touches the looping outer path, denoted by d; and (ii) the vertex where P leaves the looping outer path, denoted by e, to c. It follows that $d \leq e$. By construction, all paths in Figure 3(c) do not share internal vertices, i.e., they touch only at end points. Finally, re-draw Figure 3(c) to get Figure 3(d), which is isomorphic to Figure 1 (where the thick lines in Figure 3(d) correspond to paths I, H, U, C, W, and D in Figure 1).

Note that vertices 1, b, and d must be unique. We have shown that if there is a looping outer path, then we have the configuration in Figure 1, where path I has zero arc, paths W and U possibly have zero arc (if $b = c$ and/or $d = e$), and all other paths must contain at least one arc.

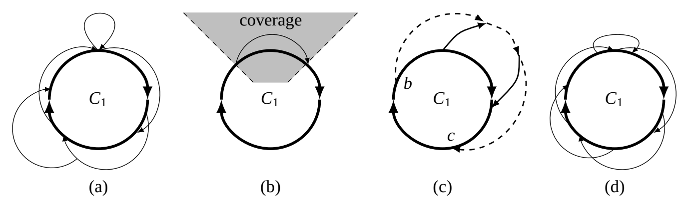

Fig. 4. We can always draw G_{sub} as in subfigure (a), i.e., a centre cycle C_1 and outer paths from C_1 and back to C_1. Subfigure (b) shows the coverage of an outer path, i.e., vertices in C_1 in the grey area *excluding* the two end points. Subfigure (c) shows that when multiple outer paths originate from one vertex, we consider only the outer path with the largest coverage, i.e., the dotted path from b to c. The outer paths in subfigure (d) provide full coverage.

D. No looping outer path

For a non-looping outer path from vertex $b \in V(C_1)$ to $c \in V(C_1) \setminus \{b\}$, we say that the vertices in C_1 from b to c (in the direction of the arcs in C_1) but excluding b and c is *covered* by this outer path. See Figure 4(b) for an example.

For the purpose of this paper, we exclude outer paths with strictly smaller coverage, or multiple outer paths with equal coverage. Referring to Figure 4(c), consider an outer path that originates from b. Suppose that it has multiple paths back to C_1. We consider only the path (back to C_1) that has the *largest coverage*. Similarly, for any path that terminates at c, we consider only the path (leaving C_1) that has the largest coverage. By doing this, each path that we consider has a unique originating vertex and a unique terminating vertex.

We now show the following property:

Proposition 7: If there is no looping outer paths in G_{sub}, then all largest-covering outer paths must, together, provide full coverage for the cycle C_1. In other words, every vertex in C_1 must be covered by at least one outer path.

Proof: Consider any vertex $a \in V(C_1)$. Re-label a as vertex 1, and other vertices $V(C_1)$ in ascending order in the arc direction. Remove vertex 1 from G_{sub}. There must exist another cycle. It follows from Proposition 6 that an outer path P from b to c must exist, where $1 < c < b \leq |V(C_1)|$ ($c \neq b$ since there is no looping path), meaning that this outer path

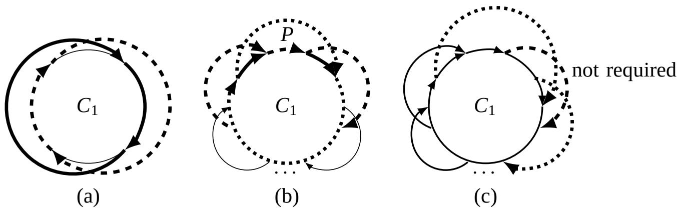

Fig. 5. (a) If two outer paths provide full coverage, we can always form two disjoint cycles, one formed by the thick solid path, and the other the dashed path. (b) G_{sub} with K outer paths, where $K \geq 4$, providing full coverage can be converted to $K-2$ outer paths providing full coverage. (c) If two non-adjacent outer paths give overlapping coverage (e.g., the two dotted paths), then the paths in between are redundant (the dashed path), i.e., $K-1$ outer paths are sufficient to give full coverage, instead of K.

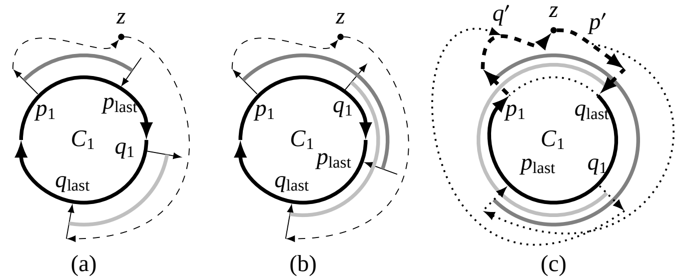

Fig. 6. The overlapping of the coverage of two outer paths, where the dark grey lines represent the coverage of the outer path P ($p_1 \to \cdots \to p_{\text{last}}$), and the light grey lines that of the outer path Q ($q_1 \to \cdots \to q_{\text{last}}$)

must cover vertex 1. We can safely ignore other outer paths that provide smaller or equal coverage, because if P does not cover vertex 1, then none of the ignored outer paths does. Since the choice of a is arbitrary, we have Proposition 7. ■

For example, the outer paths in Figure 4(d) provide full coverage for C_1, but the outer paths in Figures 4(a)–(c) do not. Removing one uncovered vertex from C_1 makes G_{sub} acyclic.

Now, we consider G_{sub} that consists of cycle C_1 and all outer paths that provide the largest coverage (i.e., we remove all other arcs have gives smaller or equal coverage). We are ready to proceed with Cases 2 and 3, defined as follows:

- (Case 2) There is no looping outer path, and no two outer paths have any common internal vertex.
- (Case 3) There is no looping outer path, and there exist two outer paths sharing the same internal vertex.

E. Case 2: No looping outer path, and all outer paths do not share internal vertices

We will show that we can always find three outer paths that provide full coverage.

First, note that one outer path cannot provide full coverage. Suppose that we can find two outer paths providing full coverage. We illustrate in Figure 5(a) that we can always form two vertex-disjoint cycles. So, this scenario cannot happen.

Next, suppose that we can find three outer paths providing full coverage, we have exactly Figure 1. As there is no looping outer path, the nine paths in Figure 1 each have one or more arcs.

Finally, we show that if we can find $K \geq 4$ outer paths providing full coverage, we can always modify the cycles such that $(K-2)$ outer paths provide full coverage. We illustrate this in Figure 5(b). We do the following:

1) Combine the dotted arrows to be the new C_1.
2) Combine the two adjacent dashed paths, and the dashed arc in C_1 that connects the two dashed outer paths (denoted by P, which can be of length 0) into a new outer path.
3) Remove all arcs and internal vertices in the the thick solid paths in C_1. Each thick solid path must contain at least one arc; otherwise, the outer paths cannot provide full coverage.

Note that by doing this, the new graph still retains the structure of a cycle with outer paths covering it. The new graph has $K-2$ outer paths providing full coverage. This reduction is always possible as the coverage of two non-adjacent outer paths does not overlap, illustrated in Figure 5(c).

By repeating this step, starting from any $K \geq 4$ outer paths, we can find a graph with $K = 2$ or $K = 3$ outer paths. As $K = 2$ is not possible, we will always get a graph with $K = 3$ outer paths providing full coverage, which is in the form of Figure 1.

F. Case 3: No looping outer path and two outer paths share some internal vertices

Let the two outer paths that share some common internal vertex be P and Q, and one of the shared internal vertices be z. Further, let the originating and terminating vertices of P be p_1 and p_{last} respectively, and those of Q be q_1 and q_{last}. Here, $p_1 \neq p_{\text{last}}$ and $q_1 \neq q_{\text{last}}$ as there is no looping outer path, and $p_1 \neq q_1$ and $p_{\text{last}} \neq q_{\text{last}}$ as no two outer paths have the same originating or terminating vertices.

Now, the coverage of P and Q can be either (a) non-overlapping, (b) overlapping once, or (c) overlapping twice, as shown in Figure 6. The dark grey line shows the coverage of P, and the light grey line that of Q. By definition, there is a subpath from p_1 to z along P and another subpath from z to p_{last} along P. The two subpaths must be vertex-disjoint, except z, as there is no cycle in P. Similarly, we have two vertex-disjoint paths from q_1 to z, and from z to q_{last}, both along Q. This means, there is an subpath from p_1 to q_{last} through z, and another from q_1 to p_{last} through z. So, $p_1 \neq q_{\text{last}}$, $q_1 \neq p_{\text{last}}$, as there is no looping outer path, and hence p_1, p_{last}, q_1, and q_{last} are distinct.

Suppose that we have Figure 6(a). The largest-covering outer path from p_1 should terminate at q_{last}, and that from q_1 at p_{last}. The outer path from p_1 to q_{last} and that from q_1 to p_{last} should have been chosen. This means the largest-covering paths actually overlap twice, i.e., we should have Figure 6(c).

Suppose that we have Figure 6(b). The outer path from p_1 to q_{last}, through z, gives the largest coverage, and it would have been chosen.

So, we can only have the configuration in Figure 6(c), where the coverage overlaps twice. The coverage from p_1 to q_{last} is smaller than that from p_1 to p_{last}. So, the largest-covering outer path from p_1 was correctly identified. Similarly, the largest-covering outer path from q_1 terminates at q_{last}.

We will now show that we can always get Figure 1 from Figure 6(c). Recall that there is a subpath from p_1 to z and another subpath from z to q_{last}, and these two subpaths are vertex-disjoint, except z. Otherwise, we get a cycle disjoint from C_1. We denote the outer path from p_1 to q_{last} (through z) by Z (drawn with a thick dashed line).

Next, recall that there is a subpath from q_1 to z, and another from z to p_{last}. So, the subpath from q_1 to z must meet Z. Denote the vertex it first meets Z as q'. Similarly, the subpath from z to p_{last} must share some common vertices with Z (at least vertex z). Let the last shared vertex be p'. With this construction, Z, the subpath from q_1 to q', and the subpath from p' to p_{last} are vertex-disjoint, except at p' and q'.

We now re-draw Figure 6(c) as follows: Let the path from q_{last} to p_1 along C_1 (drawn with a thick solid line) plus path Z (drawn with a thick dashed line) be the centre cycle, and let the subpaths (drawn with dotted arrows) (i) from p_1 to q_{last} along C_1, (ii) from p' to p_{last}, and (iii) from q_1 to q' be the three outer paths. Note that only p' and q' can co-locate. The resultant graph is isomorphic to Figure 1, with path I possibly having zero arc (if $p' = q' = z$).

Combining the Cases 1–3, we have Lemma 3. ■

Appendix B
Proof of Lemma 4

We first note the following:

Observation 1: If $\mathsf{mais}(G) = 2$, then any induced subgraph of G with three vertices must contain a cycle.[8] Otherwise, $\mathsf{mais}(G) \geq 3$ by considering the 3-vertex induced subgraph without a cycle.

We define edges in directed graphs as follows:

Definition 7: Consider a directed graph G with vertex set $V(G)$ and arc set $A(G)$. For a pair of vertices $i, j \in V(G)$, we say that there is an *edge* between these two vertices if and only if $(i \to j) \in A(G)$ and $(j \to i) \in A(G)$. A cycle formed by edges is called an *undirected cycle*.

As the proof of the lemma is rather involved, we divide the set of all graphs to be considered in this lemma, i.e., all G with $|V(G)| = 5$, $G \notin \mathcal{G}_s$, and $\mathsf{mais}(G) = 2$, into four categories according to the number of undirected cycles in G:

1) There is no undirected cycle.
2) There exists an undirected cycle of length 3.
3) There is no undirected cycle of length 3, but there exists an undirected cycle of length 4.
4) There is no undirected cycle of length 3 or 4, but there exists an undirected cycle of length 5.

Note that, by definition, there cannot be any undirected cycle of length 2 or less.

A. Two useful lemmas

We say that G^- is an *arc-deleted* subgraph of G if $V(G^-) = V(G)$ and $A(G') \subseteq A(G)$, i.e., removing zero or some arc(s) from G but retaining all the vertices.

We first prove two lemmas to be used subsequently:

[8]Recall that, unless stated otherwise, cycles refer to directed cycles.

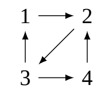

Fig. 7. If there is no edge in the graph, then $\{1, 3, 4\}$ cannot contain a cycle.

Lemma 5: Let G be an arc-deleted subgraph of G^+, and G^- be an arc-deleted subgraph of G. Then,

$$r_{m^t}(G^+) \leq r_{m^t}(G) \leq r_{m^t}(G^-), \tag{32}$$

and an index code for G^- is an index code for G and G^+.

Proof: Each receiver in G^+ has prior messages of at least what it has in G, and it requests the same message (i.e., receiver i requests X_i). So, any index code for G satisfies all decoding requirements for G^+ and hence is an index code for G^+. This proves $r_{m^t}(G^+) \leq r_{m^t}(G)$. By repeating the same argument, we have $r_{m^t}(G) \leq r_{m^t}(G^-)$. ■

Lemma 6: If $|V(G)| = 5$ and $\mathsf{mais}(G) = 2$, then the induced subgraph of any four vertices must contain an edge.

Proof: We will prove the lemma by contradiction. Suppose that there is an induced subgraph of four vertices without an edge. Recall that any induced subgraph of three vertices must contain a cycle. Referring to Figure 7, there must be a directed cycle in $\{1, 2, 3\}$. Since there is no edge, there cannot be any 2-cycle. Without loss of generality, let the cycle be $1 \to 2 \to 3 \to 1$. Again, as there cannot be any edge, the cycle in $\{2, 3, 4\}$ must be $2 \to 3 \to 4 \to 2$. Now, for $\{1, 3, 4\}$ to contain a cycle, it must contain an an edge (contradiction). We would have obtained the same result had we started by choosing the cycle in $\{1, 2, 3\}$ to be $1 \to 3 \to 2 \to 1$. ■

B. Basic ideas

We will prove Lemma 4 using the following ideas: For each category, we will show that any G must contain some a arc-deleted subgraph, say G_{sub}. We then show that there exists a scalar linear index code of length 2 (over the ring $\mathcal{X}$) for G_{sub}, thereby establishing $r_{m^t}(G_{\text{sub}}) \leq 2$. Since $G_{\text{sub}} = G^-$, from Lemma 5, we must have that $r_{m^t}(G) \leq 2$, where the 2-bit achievability uses the same linear code for G_{sub}. As $\mathsf{mais}(G) = 2$ is a lower bound on $r_{m^t}(G)$, we establish $r_{m^t}(G) = 2$. We will use a combinatoric approach.

C. Category 1: No undirected cycle

We start with the first category where there cannot be any undirected cycle in G. We have the following subcategories:

1) There is one or no edge: If there is no edge or only one edge, we can always find an induced subgraph of four vertices with no edge. It follows from Lemma 6 that $\mathsf{mais}(G) \neq 2$ (contradiction). Figure 8 shows an example where the graph G0.1 contains only one edge $1 - 2$, and the subgraph induced by $\{2, 3, 4, 5\}$ cannot contain any edge.

Here, we use the notation G$x.y$, where x is the length of the shortest undirected cycle in G, and y is the number of edges.

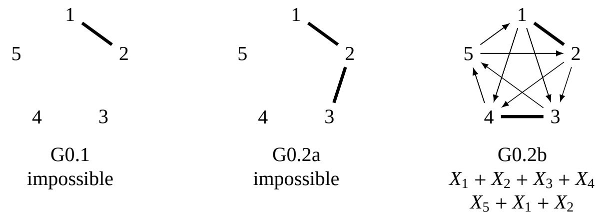

Fig. 8. G_{sub} where there is one or two edges. The first two graphs are impossible for $\mathsf{mais}(G) = 2$. For G0.2b, the length-2 index code shown here is also an index code for any G (with five vertices) containing this graph.

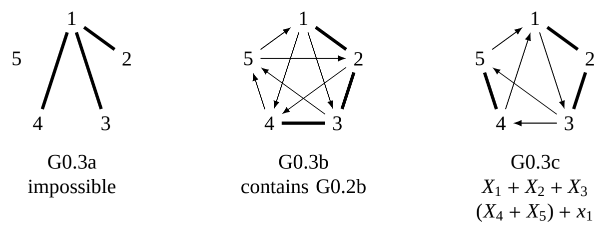

Fig. 9. G_{sub} where there are three edges and no undirected cycle. The first graph is impossible for $\mathsf{mais}(G) = 2$, and there exists two-bit linear codes for the second and the third graphs.

2) There are only two edges: The two edges in G can either be connected (see G0.2a in Figure 8) or disconnected (see G0.2b). We need to consider only non-isomorphic graphs, as the labelling of indices are arbitrary.

For G0.2a, the subgraph induced by vertices $\{1, 3, 4, 5\}$ contains no edge. By Lemma 6, this cannot happen.

For G0.2b, since there is no edge in $\{1, 4, 5\}$, there must be a length-3 cycle. Without loss of generality (due to symmetry), let the cycle by $1 \to 4 \to 5 \to 1$. This necessitates the cycle in $\{1, 3, 5\}$ to be $1 \to 3 \to 5 \to 1$. The cycles in $\{2, 3, 5\}$ and $\{2, 4, 5\}$ must also take the forms shown in the figure.

Note that G0.2b is an interlinked cycle with *inner vertices* $\{1, 2, 3, 4\}$. The interlinked-cycle cover gives an index code of length 2: $[(X_1 + X_2 + X_3 + X_4)\ (X_5 + X_1 + X_2)]$ (see Definitions 3 and 4). Here, it is understood that the addition is performed over the ring $\mathcal{X}$.

So, any G with 5 vertices, no undirected cycle, $\mathsf{mais}(G) = 2$, and only two edges must contain an arc-deleted subgraph isomorphic to G0.2b. By Lemma 5, $r_{m^t}(G) \leq r_{m^t}(\text{G0.2b}) \leq 2$. Since $2 = \mathsf{mais}(G) \leq r_{m^t}(G)$, we have $r_{m^t}(G) = 2$.

3) There are only three edges: Without any undirected cycle, three edges can form only three non-isomorphic configurations as depicted in Figure 9.

If the three edges form a star, we have G0.3a. By Lemma 6, it is impossible as the induced subgraph $\{2, 3, 4, 5\}$ has no edge.

If the three edges form a path, we have G0.3b. The vertex set $\{1, 4, 5\}$ must contain a cycle. Without loss of generality (due to symmetry), let it be $1 \to 4 \to 5 \to 1$. The rest of the cycles for subgraphs with three vertices are then fixed. Since G0.3b contains G0.2b as an arc-deleted subgraph, invoking Lemma 5, $r_{m^t}(\text{G0.3b}) \leq r_{m^t}(\text{G0.2b}) \leq 2$, and the length-2 index code for G0.2b also an index code for G0.3b.

If one of the three edges is disjoint from the other two, we have G0.3c. By symmetry and adding arcs to form cycles in $\{1, 3, 5\}$ and $\{1, 3, 4\}$, we have the configuration in the figure.

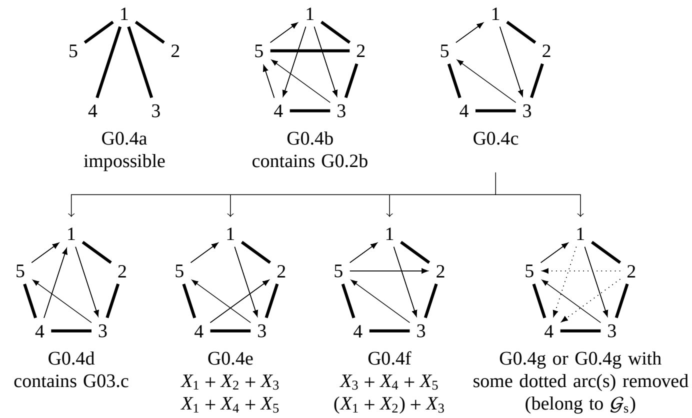

Fig. 10. G_{sub} where there are four edges and no undirected cycle.

G0.3c is an interlinked cycle with inner vertices $\{1, 2, 3\}$ and a super-vertex set $\{4, 5\}$ (see Definition 5). For an interlinked cycle of this type, the interlinked-cycle cover gives a index code $[(X_1 + X_2 + X_3)\ ((X_4 + X_5) + X_1)]$ of length 2.

4) There are only four edges: Without any undirected cycle, four edges can form only three non-isomorphic configurations G0.4a, G0.4b, or G0.4c in Figure 10.

If the four edges form a star, i.e., G0.4a, it is an impossible subgraph as $\{2, 3, 4, 5\}$ does not contain any edge.

For configuration G0.4b, the vertex set $\{1, 4, 5\}$ must contain a length-3 cycle. Without loss of generality (due to symmetric), let an arc in the cycle be $5 \to 1$, and so the cycle is $1 \to 4 \to 5 \to 1$. With this, the cycles for $\{1, 3, 5\}$ is also fixed. We see that this graph contains G0.2b as an arc-deleted subgraph, and hence $r_{m^t}(\text{G0.4b}) \leq r_{m^t}(\text{G0.2b}) \leq 2$, and the length-2 linear code for G0.2b is also an index code for G0.4b.

If the four edges form a path, we need to further categorise all G that contain G0.4c. Since the positions of edges in G0.4c are fixed, and we can only add arcs. The only positions to add arcs are within the pairs $\{(1, 4), (2, 4), (2, 5)\}$, and we can only add at most one arc in each pair (adding arcs in both directions forms an edge). So, any G in this category must satisfy either of the following:

- If there is an additional arc within any pair in $\{(1, 4), (2, 4), (2, 5)\}$ from a larger index to a smaller index, i.e., $4 \to 1$, $4 \to 2$, or $5 \to 2$, we get a graph that contains G0.4d, G0.4e, or G0.4f, respectively, as an arc-deleted subgraph. Note that a graph can also simultaneosly contain more than one of these graphs as subgraphs. Note that
  - G0.4d contains G0.3c as a subgraph;
  - G0.4e has a length-2 index code $[(X_1 + X_2 + X_3)\ (X_1 + X_4 + X_5)]$;
  - G0.4f contains an interlinked cycle with inner vertices $\{3, 4, 5\}$ and a super-vertex set $\{1, 2\}$.
- Otherwise, we must get G0.4g or G0.4g with some of the dotted arcs removed. These graphs belong to $\mathcal{G}_s$, and we will deal it Theorem 3.

A length-2 linear code exists for each of G0.4d, -e, or -f.

5) There are five of more edges: This configuration is impossible as it is known to contain an undirected cycle.

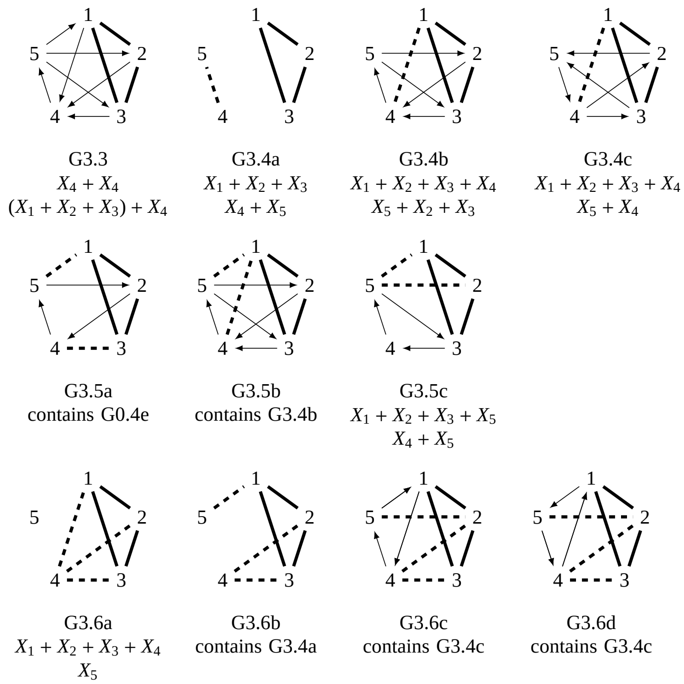

Fig. 11. G_{sub} where there is a length-3 undirected cycle (among vertices 1, 2, and 3; marked with continuous lines). Additional edges are marked with dashed lines. Arcs are then added so that every three vertices must contain at least one cycle.

So, we have shown that for any $G \notin \mathcal{G}_s$ such that $|V(G)| = 5$, $\mathsf{mais}(G) = 2$, and G contains no undirected cycle, then it must contain either G0.2b, G0.3c, G0.4e, or G0.4f as an arc-deleted subgraph. For any case, $r(G) = r_{m^t}(G) = 2$ for any m and t.

D. Category 2: An undirected cycle of length 3

Without loss of generality, let the undirected cycle be $1 - 2 - 3 - 1$ (depicted as solid lines in Figure 11). First, if there is an additional edge $4 - 5$ (denoted by G3.4a in Figure 11), there exists a length-2 index code using the clique cover $[(X_1 + X_2)\ (X_3 + X_4 + X_5)]$.

Otherwise (i.e., no edge between 4 and 5), any additional edge (in addition to $1 - 2 - 3 - 1$) must be between $\{1, 2, 3\}$ and $\{4, 5\}$. For this, we have the following categories, grouped by the number of additional edge (dashed lines in Figure 11):

1) No edge between the groups $\{1, 2, 3\}$ *and* $\{4, 5\}$*:* The only non-isomorphic graph where every three vertices contain a cycle is depicted in G3.3. This is an interlinked cycle with inner vertices $\{4, 5\}$ and a super-vertex set $\{1, 2, 3\}$. An index code for this graph is $[(X_4 + X_5)\ ((X_1 + X_2 + X_3) + X_4)]$.

2) One edge between the groups: Without loss of generality, let the additional edge be $1-4$. Two non-isomorphic graphs with different arc positions are possible: G3.4b and G3.4c. They are interlinked cycles with inner vertices $\{1, 2, 3, 4\}$.

3) Two edges between the groups: If the two edges connect four different vertices, we have G3.5a. If the two edges connect between the same vertex in $\{1, 2, 3\}$ to two different vertices in $\{4, 5\}$, we have G3.5b. Otherwise, if the two edges connect between different vertices in $\{1, 2, 3\}$ to the same vertex in $\{4, 5\}$, we have G3.5c, which is an interlinked cycle with inner vertices $\{1, 2, 3, 5\}$.

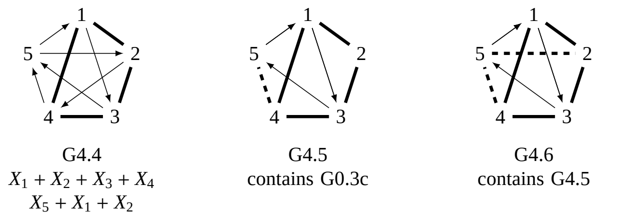

Fig. 12. G_{sub} where there is a length-4 undirected cycle (marked with thick solid lines) and no length-3 undirected cycle. Additional edges are marked with dashed lines. Arcs are then added so that every three vertices must contain at least one cycle. There are only three non-isomorphic graphs.

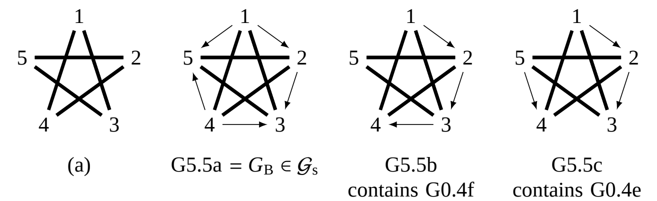

Fig. 13. G_{sub} where there is a length-5 undirected cycle (marked with thick lines) and no length-3 or -4 undirected cycle.

4) Three edges between the groups: The three edges can be placed in three non-isomorphic positions: (i) Between three vertices in $\{1, 2, 3\}$ and one vertex in $\{4, 5\}$, we have G3.6a; (ii) Between three vertices in $\{1, 2, 3\}$ and two vertices in $\{4, 5\}$, we have G3.6b; (iii) Between two vertices in $\{1, 2, 3\}$ and two vertices in $\{4, 5\}$, we have G3.6c and G3.6d. For G3.6a, the clique cover gives a linear index code $[(X_1 + X_2 + X_3 + X_4)\ X_5]$.

5) Four or more edges between the groups: We can show that the graph will always contain G3.4a with vertex relabelling.

So, we have shown that for any $G \notin \mathcal{G}_s$ such that $|V(G)| = 5$, $\mathsf{mais}(G) = 2$, and G contains an undirected cycle of length 3, then there exists a linear index code of length 2, which can be constructed using the interlinked-cycle cover (which includes the clique cover as a special case).

E. Category 3: An undirected cycle of length 4 and no undirected cycle of length 3

Next, we consider the category where there is an undirected cycle of length 4; without loss of generality, let the cycle be $1-2-3-4-1$. We find graphs when there is (i) no additional edge, (ii) one additional edge, or (iii) two additional edges. Note that there cannot be three additional edge, as it will create a length-3 undirected cycle. For each graph here, there exists a length-2 linear index code, as shown in Figure 12. Note that F4.4 is an interlinked cycle with inner vertices $\{1, 2, 3, 4\}$.

F. Category 4: An undirected cycle of length 5 and no undirected cycle of length 3 or 4

Without loss of generality, let the undirected cycle be $1 - 3 - 5 - 2 - 4 - 1$. With this, there cannot be any additional edge; otherwise, we get a length-3 or -4 cycle. Also, any additional arc must be between adjacent vertices on the "circumference", i.e., within any pair in $\{(1, 2), (2, 3), (3, 4), (4, 5), (5, 1)\}$.

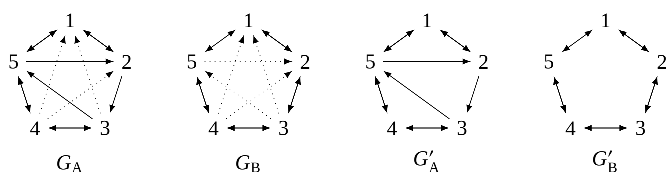

Fig. 14. G_{A} and G_{B} with vertices labelled.

If we add arcs in the way to obtain G5.5a, we get a graph in $\mathcal{G}_{\mathrm{s}}$. We will deal with this in the next section.

We now show that for any graph in Category 4 that is not a subgraph of G5.5a, there exists a two-bit linear index code. First, if we add (i) zero, (ii) one, or (iii) two arcs to Figure 13(a), we must get an isomorphic arc-deleted subgraph of G5.5a, and they are members in $\mathcal{G}_{\mathrm{s}}$.

If we add three arcs, the only graphs that are not isomorphic arc-deleted subgraphs of G5.5a are G5.5b and G5.5c. By relabelling the vertices, G5.5b contains G0.4f, and G5.5c contains G0.4e.

If we add four arcs to Figure 13(a), they must form a string (i.e., a path where the direction of the arcs can be arbitrary) on the circumference (dashed lines on Figure 13(a)). The only non-isomorphic combinations of length-4 strings along the circumference are: (i) →→→→, (ii) →→→←, (iii) ←←←→, (iv) →→←←, (v) ←←→→, (vi) →→←→, (vii) ←←→←, (viii) →←←→, (ix) →←→←, and (x) ←→←→. Configurations (i)–(iii) each contain G5.5b, (iv)–(v) each contain G5.5c, (iv)–(x) each are subgraphs of G5.5a (i.e., members of $\mathcal{G}_{\mathrm{s}}$).

Lastly, we add five arcs, i.e., one arc within any pair in $\{(1,2),(2,3),(3,4),(4,5),(5,1)\}$. We will now show that the graph must be G5.5a, G5.5b$^+$, or G5.5c$^+$.[9] We can easily show that there must be a two adjacent arc in the same direction. Without loss of generality, let them be $1 \to 2 \to 3$. For arcs between $\{3,4\}$, $\{4,5\}$, and $\{1,5\}$, if any of them does not follow the direction as that in G5.5a, we have either G5.5b$^+$ or G5.5c$^+$.

We have shown that for any $G \notin \mathcal{G}_{\mathrm{s}}$ such that $|V(G)| = 5$, $\mathsf{mais}(G) = 2$, and G contains an undirected cycle of length 5, and no undirected cycle of length 3 or 4, then it must contain G5.5b or G5.5c as an arc-deleted subgraph. So, $r(G) = r_{m^t}(G) = 2$.

This completes the proof of Lemma 4. ■

Remark 5: For all graph $G \notin \mathcal{G}_{\mathrm{s}}$ such that $|V(G)| = 5$ and $\mathsf{mais}(G) = 2$, except those that contain G0.4e, an optimal scalar linear index code of length 2 can be constructed using the interlinked-cycle cover.

Appendix C
Proof of Theorem 3

Refer to G_{A} and G_{B} in Figure 2. Denote G'_{A} and G'_{B} as the subgraphs formed by removed all dotted arcs in G_{A} and G_{B} respectively. Blasiak et al. [13] found $r(G)$ for all undirected cycles, which include the 5-cycle G'_{B} as a special case. Here, we need to further find $r_{m^t}(G)$ for all $G \in \mathcal{G}_{\mathrm{s}}$.

[9]Recall that G^+ contains G as an arc-deleted subgraph.

We first label the vertices of G_{A}, G_{B}, G'_{A}, and G'_{B} as in Figure 14.

(Achievability): Let each message be a vector of length 2, which can be written as $X_i = (X_i^{(1)}, X_i^{(2)}) \in \mathcal{X}' \times \mathcal{X}'$, where $\mathcal{X}' = \{0, 1, \ldots, m^k - 1\}$. Let $\mathcal{Y} = (\mathcal{X}')^5$, $\phi = [\phi_1 \cdots \phi_5]$ such that $\phi_i : (\mathcal{X}')^{10} \mapsto \mathcal{X}'$ is vector linear over the ring $\mathcal{X}'$. The codelength here is 2.5.

For the graph G'_{B}, by time-sharing the cycle cover over the cycles $\{(i, i{+}1 \mod 5) : i \in [5]\}$, we obtain the following index code: $\phi_1 = X_1^{(1)} + X_2^{(1)}$, $\phi_2 = X_2^{(2)} + X_3^{(2)}$, $\phi_3 = X_3^{(1)} + X_4^{(1)}$, $\phi_4 = X_4^{(2)} + X_5^{(2)}$, $\phi_5 = X_5^{(1)} + X_1^{(2)}$, where the addition is performed over modulo-m^k.

For the graph G'_{A}, by time-sharing the index codes for the interlinked cycle $\{1,2,3,5\}$ (with inner vertices $\{1,2,3\}$) and cycles $\{1,2\}$, $\{3,4\}$, $\{4,5\}$, we obtain the following index code: $\phi_1 = X_1^{(1)} + X_2^{(1)} + X_3^{(1)}$, $\phi_2 = X_5^{(1)} + X_1^{(1)}$, $\phi_3 = X_1^{(2)} + X_2^{(2)}$, $\phi_4 = X_3^{(2)} + X_4^{(1)}$, $\phi_5 = X_4^{(2)} + X_5^{(2)}$.

Note that any $G \in \mathcal{G}_{\mathrm{s}}$ must contain either G'_{A} or G'_{B} as an arc-deleted subgraph. Invoking Lemma 5, we have that $r_{m^{2k}}(G) \leq 2.5$, and the upper bounds can be attained by vector linear codes over the ring $\mathcal{X}'$.

(Lower bound): While, upper bounds found for G'_{A} and G'_{B} is applicable to all $G \in \mathcal{G}_{\mathrm{s}}$, for lower bounds, we need to consider G_{A} and G_{B}.

We will now use the following tools to find lower bounds for G_{A} and G_{B}:

1) *Submodularity of entropy:* Entropy is a submodular function, i.e., for any sets of random variables S and T, we must have that
$$H(S) + H(T) \geq H(S \cup T) + H(S \cap T). \tag{33}$$
2) *Decodability:* Given G. For any vertex $i \in V(G)$ with out-neighbourhood $N_G^+(i)$, receiver i must be able to decode X_i given the index code, denoted by Y, and all the messages it knows a priori, $\boldsymbol{X}_{N_G^+(i)}$, i.e.,
$$H(X_i | Y, \boldsymbol{X}_{N_G^+(i)}) = 0 \tag{34}$$
$$\Rightarrow \quad H(\boldsymbol{X}_{N_G^+(i)}, Y) = H(\boldsymbol{X}_{\{i\} \cup N_G^+(i)}, Y) \tag{35}$$

Note that while the submodularity inequality (33) is universal in the sense it does not depend on specific graphs, the decodability equality (35) does depend on G.

We now derive submodularity and decodability conditions can be applied to both G_{A} and G_{B}, which are based on those for undirected cycles [13]. Let $q = \log_2 |\mathcal{X}|$.

$$H(Y) + 2q = H(Y) + H(\boldsymbol{X}_{\{2,5\}}) \geq H(\boldsymbol{X}_{\{2,5\}}, Y) \tag{36}$$
$$H(Y) + 2q = H(Y) + H(\boldsymbol{X}_{\{1,3\}}) \geq H(\boldsymbol{X}_{\{1,3\}}, Y) \tag{37}$$
$$H(Y) + q = H(Y) + H(X_4) \geq H(X_4, Y) \tag{38}$$
$$H(\boldsymbol{X}_{\{1,2,5\}}, Y) + H(\boldsymbol{X}_{\{1,2,3\}}, Y)$$
$$\geq H(\boldsymbol{X}_{\{1,2,3,5\}}, Y) + H(\boldsymbol{X}_{\{1,2\}}, Y) \tag{39}$$
$$= H(\boldsymbol{X}_{\{1,2,3,4,5\}}, Y) + H(\boldsymbol{X}_{\{1,2\}}, Y) \tag{40}$$
$$H(\boldsymbol{X}_{\{1,2\}}, Y) + H(X_4, Y)$$
$$\geq H(\boldsymbol{X}_{\{1,2,4\}}, Y) + H(Y) \tag{41}$$
$$= H(\boldsymbol{X}_{\{1,2,4,5\}}, Y) + H(Y) \tag{42}$$

$$= H(\boldsymbol{X}_{\{1,2,3,4,5\}}, Y) + H(Y) \quad (43)$$

$$H(\boldsymbol{X}_{\{2,5\}}, Y) = H(\boldsymbol{X}_{\{1,2,5\}}, Y) \quad (44)$$

$$H(\boldsymbol{X}_{\{1,3\}}, Y) = H(\boldsymbol{X}_{\{1,2,3\}}, Y) \quad (45)$$

Here, inequalities (36)–(39), (41) follow from submodularity of entropy, and equalities (40), (42)–(45) from decodability.

Now,

$$\begin{aligned} 10q &= 2H(\boldsymbol{X}_{[5]}) = 2H(\boldsymbol{X}_{[5]}, Y) \\ &\leq (H(\boldsymbol{X}_{\{1,2,5\}}, Y) + H(\boldsymbol{X}_{\{1,2,3\}}, Y) - H(\boldsymbol{X}_{\{1,2\}}, Y)) \\ &\quad + (H(\boldsymbol{X}_{\{1,2\}}, Y) + H(X_4, Y) - H(Y)) \quad (46a) \\ &= H(\boldsymbol{X}_{\{2,5\}}, Y) + H(\boldsymbol{X}_{\{1,3\}}, Y) + H(X_4, Y) - H(Y) \quad (46b) \\ &\leq H(Y) + 2q + H(Y) + 2q + H(Y) + q + -H(Y), \quad (46c) \end{aligned}$$

where $[5] \triangleq \{1, 2, \ldots, 5\}$, (46a) follows from (40)–(43); (46b) follows from (44)–(45); (46c) follows from (36)–(38). This gives

$$p \log_2 |\mathcal{Y}| \geq H(Y) \geq 2.5q = 2.5 \log_2 |\mathcal{X}|, \quad (47)$$

for any index code C, and hence $r_{m^{2k}}(G_\mathrm{A}) \geq 2.5$, and $r_{m^{2k}}(G_\mathrm{B}) \geq 2.5$.

Note that any $G \in \mathcal{G}_\mathrm{s}$ must be either G_A or G_B, or its arc-deleted subgraph. Invoking Lemma 5, we have that $r_{m^{2k}}(G) \geq 2.5$.

Combining this lower bound and the achievability results, we have $r_{m^{2k}}(G) = 2.5$ for all m and k, and hence $r(G) = 2.5$.

Now, note that $\mathsf{mais}(G) \leq \mathsf{mais}(G^-)$, as removing arcs can only reduce the number of cycles. We can manually verify that $\mathsf{mais}(G_\mathrm{A}) = \mathsf{mais}(G_\mathrm{B}) = \mathsf{mais}(G'_\mathrm{A}) = \mathsf{mais}(G'_\mathrm{B}) = 2$. Since any $G \in \mathcal{G}_\mathrm{s}$ must satisfy (i) $G = (G_\mathrm{A})^-$ or $G = (G_\mathrm{B})^-$, and (ii) $G = (G'_\mathrm{A})^+$ or $G = (G'_\mathrm{B})^+$, we have $\mathsf{mais}(G) = 2$.

This completes the proof of Theorem 3. ■

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," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2825–2830, June 2006.

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

[3] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3187–3195, July 2010.

[4] M. Effros, S. El Rouayheb, and M. Langberg, "An equivalence between network coding and index coding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, July 7–12 2013, pp. 967–971.

[5] M. F. Wong, M. Langberg, and M. Effros, "On a capacity equivalence between network and index coding and the edge removal problem," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, July 7–12 2013, pp. 972–976.

[6] M. J. Neely, A. S. Tehrani, and Z. Zhang, "Dynamic index coding for wireless broadcast networks," in *Proc. 31st IEEE Conf. Comput. Commun. (INFOCOM)*, Orlando, USA, Mar. 25–30 2012, pp. 316–324.

[7] K. Shanmugam, A. G. Dimakis, and M. Langberg, "Local graph coloring and index coding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, July 7–12 2013, pp. 1152–1156.

[8] C. Thapa, L. Ong, and S. J. Johnson, "A new index coding scheme exploiting interlinked cycles," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, China, June 14–19 2015, pp. 1024–1028.

[9] ——, "Generalized interlinked cycle cover for index coding," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Jeju, Korea, Oct. 11–15 2015, pp. 4–8.

[10] F. Arbabjolfaei, B. Bandemer, Y.-H. Kim, E. Şaşoğlu, and L. Wang, "On the capacity region for index coding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, July 7–12 2013, pp. 962–966.

[11] S. Unal and A. B. Wagner, "General index coding with side information: Three decoder case," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, July 7–12 2013, pp. 1137–1141.

[12] M. Tahmasbi, A. Shahrasbi, and A. Gohari, "Critical graphs in index coding," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 2, pp. 225–235, Feb. 2015.

[13] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Broadcasting with side information: Bounding and approximating the broadcast rate," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 292–298, Sept. 2013.

[14] Z. Zhang and R. W. Yeung, "A non-shannon-type conditional inequality of information quantities," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, pp. 1982–1986, Nov. 1997.

[15] A. Blasiak, "Infora graph-theoretic approach to network coding," Ph.D. dissertation, Cornell Univ., USA, Aud. 2013.

[16] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum Press, New York-London, 1972, pp. 85–104.

[17] R. Peeters, "Orthogonal representations over finite fields and the chromatic number of graphs," *Combinatorica*, vol. 16, no. 3, pp. 417–431, Sept. 1996.

[18] E. Lubetzky and U. Stav, "Nonlinear index coding outperforming the linear optimum," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3544–3551, Aug. 2009.

[19] L. Ong and C. K. Ho, "Optimal index codes for a class of multicast networks with receiver side information," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, Canada, June 10–15 2012, pp. 2223–2228.

[20] C. Thapa, L. Ong, and S. J. Johnson. (2016, Feb. 29) Interlinked cycles for index coding: Generalizing cycles and cliques. [Online]. Available: http://arxiv.org/abs/1603.00092v1

[21] A. S. Tehrani, A. G. Dimakis, and M. J. Neely, "Bipartite index coding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, USA, July 1–6 2012, pp. 2256–2260.

[22] H. Yu and M. J. Neely, "Duality codes and the integrality gap bound for index coding," in *Proc. 51st Allerton Conf. Commun. Control Comput. (Allerton Conf.)*, Monticello, USA, Oct. 2–4 2013.

[23] S. H. Dau, V. Skachek, and Y. M. Chee, "Optimal index codes with near-extreme rates," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, USA, July 1–6 2012, pp. 2241–2245.

[24] L. Ong, "A new class of index coding instances where linear coding is optimal," in *Proc. Int. Symp. on Netw. Coding (NetCod)*, Aalborg, Denmark, June 27–28 2014.

[25] N. J. A. Sloane, "Number of directed graphs (or digraphs) with *n* nodes," in *The On-Line Encyclopedia of Integer Sequences*, 2010. [Online]. Available: http://oeis.org/A000273

[26] ——, "Number of acyclic digraphs with *n* unlabeled nodes," in *The On-Line Encyclopedia of Integer Sequences*, 2010. [Online]. Available: http://oeis.org/A003087

[27] ——, "Number of strongly perfect perfect graphs on *n* nodes," in *The On-Line Encyclopedia of Integer Sequences*, 2010. [Online]. Available: http://oeis.org/A123461