Three Stories on a Two-sided Coin: Index Coding, Locally Recoverable Distributed Storage, and Guessing Games on Graphs

Fatemeh Arbabjolfaei and Young-Han Kim
Department of Electrical and Computer Engineering
University of California, San Diego
Email: {farbabjo, yhk}@ucsd.edu

Abstract—Three science and engineering problems of recent interests—index coding, locally recoverable distributed storage, and guessing games on graphs—are discussed and the connection between their optimal solutions is elucidated. By generalizing recent results by Shanmugam and Dimakis and by Mazumdar on the complementarity between the optimal broadcast rate of an index coding problem on a directed graph and the normalized rate of a locally recoverable distributed storage problem on the same graph, it is shown that the capacity region and the optimal rate region of these two problems are complementary. The main ingredients in establishing this result are the notion of confusion graph introduced by Alon et al. (2008), the vertex transitivity of a confusion graph, the characterization of the index coding capacity region via the fractional chromatic number of confusion graphs, and the characterization of the optimal rate region of the locally recoverable distributed storage via the independence number of confusion graphs. As the third and final facet of the complementarity, guessing games on graphs by Riis are discussed as special cases of the locally recoverable distributed storage problem, and it is shown that the winning probability of the optimal strategy for a guessing game and the ratio between the winning probabilities of the optimal strategy and a random guess can be characterized, respectively, by the capacity region for index coding and the optimal rate region for distributed storage.

I. Introduction

This paper discusses index coding, locally recoverable distributed storage, and guessing games on directed graphs, with the goal of elucidating the relationship between the optimal solutions of these three problems.

Index coding. A server has multiple messages and wishes to send them to their corresponding receivers. Each receiver has some side information about a subset of messages (not including the desired message). The goal is to exploit the side information of the receivers to minimize the number of required transmissions. This problem arises in many contexts such as intra-cell communication, content broadcasting and coded caching [1], as well as satellite communication as originally introduced by Birk and Kol [2].

It is easy to see that index coding is a special case of network coding which is a key open problem in network information theory. Effros, El Rouayheb, and Langberg [3] showed that for any network coding problem, there exists an equivalent index coding problem. This equivalence makes index coding even more intriguing. So far the index coding problem has been attacked using tools from various disciplines such as graph theory [2], [4]–[10], algebra [11], [12], interference alignment [13]–[15], source coding [16], and random coding [17].

Locally recoverable distributed storage. A set of servers collectively store data such that if a server fails, its contents can be efficiently reconstructed from the contents of the other servers (among many others, see [18]–[20]). The goal is to design a distributed storage code that maximizes the amount of data that can be stored while satisfying the single-failure recovery constraint. In the mentioned references, it is assumed that each server is able to connect to all the other servers of the system. However, in a real system, due to some constraints, a server may only have access to a subset of the other servers. In [21], Mazumdar took this point into consideration by assuming that the topology of the system is given by a directed graph and established a duality between the normalized rate of such a distributed storage system and the broadcast rate of the index coding problem. In particular, Mazumdar showed that for any directed graph on n nodes, the broadcast rate of the index coding problem and the normalized rate of the locally recoverable distributed storage problem sum up to n. In an independent concurrent work, Shanmugam and Dimakis [22] established a similar result for vector linear codes and showed that the dual code of a linear index code is a linear code for the locally recoverable distributed storage problem and vice versa.

Guessing games on graphs. Consider a cooperative game among multiple players, in which a value (hat color) is assigned to each player independently. The value of each player is not revealed to her, but she knows the values assigned to a subset of the other players she can observe. The players should simultaneously guess their own values and they win if all of them guess

correctly. After the values are assigned, the players are not allowed to communicate, but they can come up with a strategy beforehand to maximize their probability of winning. What is the best strategy and how much improvement over random guesses can we make by employing the optimal strategy? This interesting puzzle, which apparently seems to be in the realm of recreational mathematics, was introduced originally by Riss [23] for studying multiple unicast network coding problems. Its relationship to index coding was observed firstly by Yi, Sun, Jafar, and Gesbert [9].

In this paper, we define the $\boldsymbol{\lambda}$-directed capacity $C(\boldsymbol{\lambda})$ of the index coding problem and the $\boldsymbol{\lambda}$-directed optimal rate $R(\boldsymbol{\lambda})$ of the distributed storage problem and rewrite the capacity region of the index coding problem and the optimal rate region of the distributed storage problem in terms of the set of $\boldsymbol{\lambda}$-directed capacities, and the set of $\boldsymbol{\lambda}$-directed optimal rates respectively. Denoting the all-ones vector by $\mathbf{1}$, $C(\mathbf{1})$ is equal to the reciprocal of the broadcast rate of the index coding problem and $R(\mathbf{1})$ equals the reciprocal of the normalized rate of the distributed storage. The complementarity between the capacity region and the optimal rate region of the two problems is demonstrated by generalizing the complementarity result by Mazumdar [21], and by Shanmugam and Dimakis [22], to a complementarity relationship between $C(\boldsymbol{\lambda})$ and $R(\boldsymbol{\lambda})$ for any nonnegative real n-tuple $\boldsymbol{\lambda}$. The same technique is used to make a complementary connection between the sum-capacity of the index coding problem and the optimal sum-rate of the distributed storage. Next, we will discuss that the optimal guessing number of guessing games on graphs is inversely related to the optimal sum-rate of the distributed storage problem which together with the complementarity connection between the sum-capacity and the optimal sum-rate makes the inverse relationship between the optimal complementary guessing number and the sum-capacity of the index coding problem clear (see Fig. 1). As a result of the clear connection between the optimal solutions of these three problems, the known results for one of these problems can be easily extended to the other problems. We will conclude the paper by one such example.

The rest of the paper is organized as follows. We first review some mathematical preliminaries in Section II. In Section III, formal definitions of the problems are presented. In Section IV, we overview the characterization of the capacity region of the index coding problem, the optimal rate region of the distributed storage problem, and the guessing number of the guessing games via the notion of confusion graph. Section V is the main part of the paper in which the complementarity between the index coding problem and the distributed storage problem is presented and the guessing game on a graph is shown to be a special case of the distributed storage
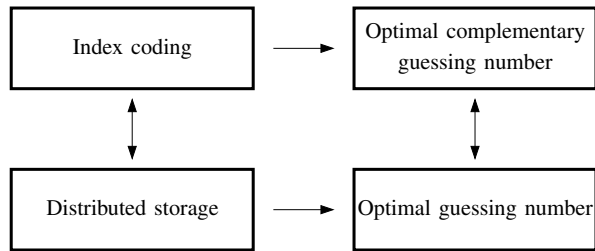

Fig. 1. The relationship between index coding, distributed storage, optimal guessing number, and optimal complementary guessing number.

problem on the same graph. Finally, in Section VI, we conclude by mentioning some results for the guessing games that due to the established connections, also hold for the index coding problem.

II. MATHEMATICAL PRELIMINARIES

Throughout the paper, $\mathbf{1}$ denotes the $n \times 1$ vector of all ones. The notations $V(\Gamma)$ and $E(\Gamma)$ mean the vertex set and the edge set of a graph Γ respectively. For $v_1, v_2 \in V(\Gamma)$, $v_1 \sim v_2$ denotes that there exists an edge between v_1 and v_2. The set of positive integers that are less than or equal to n, $\{1, 2, \ldots, n\}$, is denoted by $[n]$.

A. Graph Coloring

A (vertex) coloring of an undirected graph Γ is a mapping that assigns a color to each vertex such that no two adjacent vertices share the same color. The *chromatic number* $\chi(\Gamma)$ is the minimum number of colors such that a coloring of the graph exists. More generally, a b-fold coloring assigns a set of b colors to each vertex such that no two adjacent vertices share the same color. The b-fold chromatic number $\chi^{(b)}(\Gamma)$ is the minimum number of colors such that a b-fold coloring exists. The *fractional chromatic number* of the graph is defined as

$$\chi_f(\Gamma) = \lim_{b \to \infty} \frac{\chi^{(b)}(\Gamma)}{b} = \inf_b \frac{\chi^{(b)}(\Gamma)}{b},$$

where the limit exists since $\chi^{(b)}(\Gamma)$ is subadditive. Consequently, $\chi_f(\Gamma) \leq \chi(\Gamma)$.

B. Vertex Transitive Graphs

Definition 1. An automorphism of a graph Γ is a bijective function $\sigma : V(\Gamma) \to V(\Gamma)$, such that the pair of vertices $(\sigma(u), \sigma(v))$ form an edge iff (if and only if) the pair (u, v) form an edge.

Definition 2. A graph Γ is said to be vertex transitive if for any two vertices u and v of Γ, there exists some automorphism $\sigma : V(\Gamma) \to V(\Gamma)$ such that $\sigma(u) = v$.

Definition 3. The independence number of the undirected graph Γ denoted $\alpha(\Gamma)$ is the cardinality of the largest independent set of the graph.
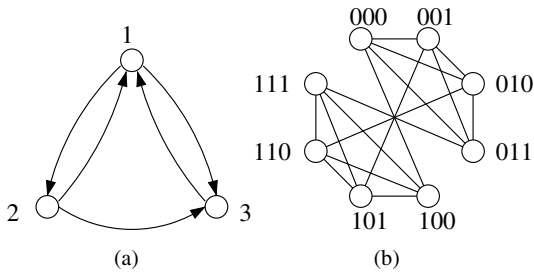
Fig. 2. (a) A 3-node directed graph. (b) The confusion graph corresponding to the integer tuple $(t_1, t_2, t_3) = (1, 1, 1)$.

Lemma 1 (Scheinerman and Ullman [24, Prop. 3.1.1]). *For any vertex transitive graph Γ we have*

$$\chi_f(\Gamma) = |V(\Gamma)|/\alpha(\Gamma).$$

C. Disjunctive Graph Product

Given two (undirected) graphs Γ_1 and Γ_2, the disjunctive product $\Gamma = \Gamma_1 * \Gamma_2$ produces a graph Γ on the Cartesian product of the original vertex sets, i.e., $V(\Gamma) = V(\Gamma_1) \times V(\Gamma_2)$ and the edge set is constructed from the original edge sets according to the following rule. $(u_1, u_2) \sim (v_1, v_2)$ iff $u_1 \sim v_1$ or $u_2 \sim v_2$. The fractional chromatic number of the disjunctive product is multiplicative.

Lemma 2 (Scheinerman and Ullman [24, Cor. 3.4.2]).

$$\chi_f(\Gamma_1 * \Gamma_2) = \chi_f(\Gamma_1)\chi_f(\Gamma_2).$$

D. Confusion Graphs

The notion of confusion graph is originally introduced by Alon, Hassidim, Lubetzky, Stav, and Weinstein [25]. Consider a directed graph $G = (V, E)$ with $V = [n]$. Let $A_j = \{i \in V : (i, j) \in E\}$, $j \in [n]$ and let $\mathbf{t} = (t_1, \dots, t_n)$ be a length-n integer tuple. Two binary n-tuples $x^n, z^n \in \prod_{i=1}^n \{0,1\}^{t_i}$ are said to be *confusable at node $j \in [n]$* if $x_j \neq z_j$ and $x_i = z_i$ for all $i \in A_j$. We say x^n and z^n are *confusable* if they are confusable at some node j.

Given a directed graph G and a length-n integer tuple $\mathbf{t} = (t_1, \dots, t_n)$, the *confusion graph* $\Gamma_{\mathbf{t}}(G)$ is an undirected graph with $\prod_{i=1}^n 2^{t_i}$ vertices such that every vertex corresponds to a binary tuple x^n and two vertices are connected iff the corresponding binary tuples are confusable. As an example, the confusion graph of the directed graph in Fig. 2(a) corresponding to $(t_1, t_2, t_3) = (1, 1, 1)$ is depicted in Fig. 2(b).

Remark 1. Every confusion graph is vertex transitive.

The following lemma shows that given any directed graph G, confusion graph corresponding to a larger integer tuple has a larger fractional chromatic number. The proof is presented in Appendix A.

Lemma 3. *Let $\mathbf{s} = (s_1, \dots, s_n)$ and $\mathbf{t} = (t_1, \dots, t_n)$ be two integer tuples such that $\mathbf{s} \leq \mathbf{t}$. Then for any directed graph G with n vertices, we have*

$$\chi_f(\Gamma_{\mathbf{s}}(G)) \leq \chi_f(\Gamma_{\mathbf{t}}(G)). \qquad (1)$$

III. PROBLEM DEFINITIONS

A. Index Coding

In the index coding problem, a sender wishes to communicate a tuple of n messages, $x^n = (x_1, \dots, x_n)$, $x_j \in \{0,1\}^{t_j}$, to their corresponding receivers using a shared noiseless channel. Receiver $j \in [n]$ has prior knowledge of a subset $x(A_j) := (x_i : i \in A_j)$, $A_j \subseteq [n] \setminus \{j\}$, of the messages and wishes to recover x_j. It is assumed that the sender is aware of $A_1, \dots, A_n$. The goal is to minimize the amount of information that should be broadcast from the sender to the receivers so that every receiver can recover its desired message.

Any instance of this problem, referred to collectively as the *index coding problem*, is fully specified by the side information sets $A_1, \dots, A_n$. Equivalently, it can be specified by a side information graph G with n nodes, in which a directed edge $i \to j$ represents that receiver j has message i as side information ($i \in A_j$). For instance, Fig. 2(a) shows the directed graph representing the index coding problem with $A_1 = \{2, 3\}$, $A_2 = \{1\}$, and $A_3 = \{1, 2\}$. We often identify an index coding problem with its side information graph and simply write "index coding problem G."

A $(t_1, \dots, t_n, r)$ index code is defined by
- an encoder $\phi : \prod_{i=1}^n \{0,1\}^{t_i} \to \{0,1\}^r$ that maps n-tuple of messages x^n to an r-bit index and
- n decoders $\psi_j : \{0,1\}^r \times \prod_{k \in A_j} \{0,1\}^{t_k} \to \{0,1\}^{t_j}$ that maps the received index $\phi(x^n)$ and the side information $x(A_j)$ back to x_j for $j \in [n]$.

Thus, for every $x^n \in \prod_{i=1}^n \{0,1\}^{t_i}$,

$$\psi_j(\phi(x^n), x(A_j)) = x_j, \quad j \in [n].$$

A rate tuple $(R_1, \dots, R_n)$ is said to be *achievable* for the index coding problem G if there exists a $(t_1, \dots, t_n, r)$ index code such that

$$R_j \leq \frac{t_j}{r}, \quad j \in [n].$$

The *capacity region* $\mathscr{C}$ of the index coding problem is defined as the closure of the set of achievable rate tuples. Let $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n)$ be a non-negative real tuple. Define the $\boldsymbol{\lambda}$-directed capacity $C(\boldsymbol{\lambda})$ of the index coding problem G as

$$C(\boldsymbol{\lambda}) = \max\{R : R\boldsymbol{\lambda} \in \mathscr{C}\}. \qquad (2)$$

Remark 2. The capacity region can be written in terms of $\boldsymbol{\lambda}$-directed capacities.

$$\mathscr{C} = \bigcup_{\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^n : \sum_{i=1}^n \lambda_i = n} \left\{ \mathbf{R} \in \mathbb{R}_{\geq 0}^n : \mathbf{R} \leq C(\boldsymbol{\lambda})\boldsymbol{\lambda} \right\}. \qquad (3)$$

If $\boldsymbol{\lambda}_1 = c\boldsymbol{\lambda}_2$ for some constant c, then $C(\boldsymbol{\lambda}_1)\boldsymbol{\lambda}_1 = C(\boldsymbol{\lambda}_2)\boldsymbol{\lambda}_2$ and thus in (3), it suffices to do the union over vectors $\boldsymbol{\lambda}$ such that $\sum_{i=1}^{n} \lambda_i = n$. The $\mathbf{1}$-directed capacity of the index coding problem G is referred to as the symmetric capacity,

$$C_{\text{sym}} = C(\mathbf{1}) = \max\{R \colon (R, \ldots, R) \in \mathscr{C}\}.$$

The reciprocal of the symmetric capacity is referred to as the broadcast rate β of the index coding problem G. For any nonnegative real vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$, the $\boldsymbol{\mu}$-weighted sum-capacity $\bar{C}(\boldsymbol{\mu})$ of the index coding problem G is defined as

$$\bar{C}(\boldsymbol{\mu}) = \max\left\{\sum_{i=1}^{n} \mu_i R_i \colon (R_1, \ldots, R_n) \in \mathscr{C}\right\}.$$

The $\mathbf{1}$-weighted sum-capacity is simply referred to as sum-capacity,

$$C_{\text{sum}} = \bar{C}(\mathbf{1}) = \max\left\{\sum_{i=1}^{n} R_i \colon (R_1, \ldots, R_n) \in \mathscr{C}\right\}.$$

B. Locally Recoverable Distributed Storage

In the locally recoverable distributed storage problem, data is to be stored in a network of n servers such that by having access to all of the servers the stored data can be exactly recovered. Let $x_j \in \{0, 1\}^{t_j}$ denote the content of server j, $j \in [n]$. It is assumed that each server has access to the contents of a subset of the other servers $x(A_j)$, where $A_j \subseteq [n] \setminus \{j\}$ and is referred to as the recoverability set of node j. The goal is to find the maximum amount of data that can be stored in the network such that if any of the servers fails, its content can be recovered from the contents of its recoverability set. Any instance of the distributed storage problem is fully represented by the storage recovery graph $G = (V, E)$ in which each node represents a server and there exists a directed edge $i \rightarrow j$ iff server i is in the recoverability set of server j.

A $(t_1, \ldots, t_n, r)$ distributed storage code is defined by

- a message set $[2^r]$,
- a one-to-one encoding function $x^n \colon [2^r] \rightarrow \prod_{i=1}^{n} \{0, 1\}^{t_i}$ that assigns a codeword $x^n(m)$ to each message $m \in [2^r]$, (the set $\mathcal{C} = \{x^n(1), \ldots, x^n(2^r)\}$ is referred to as the codebook), and
- n recovery functions $f_j \colon \prod_{k \in A_j} \{0, 1\}^{t_k} \rightarrow \{0, 1\}^{t_j}$ that maps the contents of the recoverability set $x(A_j)$ to x_j for $j \in [n]$.

Thus, for every $x^n \in \mathcal{C}$,

$$f(x(A_j)) = x_j, \quad j \in [n].$$

A rate tuple $(R'_1, \ldots, R'_n)$ is said to be *achievable* for the distributed storage problem G if there exists a $(t_1, \ldots, t_n, r)$ distributed storage code such that

$$R'_j \geq \frac{t_j}{r}, \quad j \in [n].$$

The *optimal rate region* $\mathscr{R}$ of the distributed storage problem is defined as the closure of the set of achievable rate tuples. For any non-negative real tuple $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$, the $\boldsymbol{\lambda}$-directed optimal rate $R(\boldsymbol{\lambda})$ of the distributed storage problem G is defined as

$$R(\boldsymbol{\lambda}) = \min\{R' \colon R'\boldsymbol{\lambda} \in \mathscr{R}\}. \tag{4}$$

Remark 3. The optimal rate region can be written in terms of $\boldsymbol{\lambda}$-directed optimal rates.

$$\mathscr{R} = \bigcup_{\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^n \colon \sum_{i=1}^{n} \lambda_i = n} \{\mathbf{R}' \in \mathbb{R}^n \colon \mathbf{R}' \geq R(\boldsymbol{\lambda})\boldsymbol{\lambda}\}. \tag{5}$$

The $\mathbf{1}$-directed optimal rate of the distributed storage problem G is referred to as the symmetric coding rate,

$$R_{\text{sym}} = R(\mathbf{1}) = \min\{R' \colon (R', \ldots, R') \in \mathscr{R}\}.$$

The reciprocal of the symmetric coding rate is sometimes referred to as the normalized rate. For any nonnegative real vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$, the $\boldsymbol{\mu}$-weighted optimal sum-rate $\bar{R}(\boldsymbol{\mu})$ of the distributed storage problem G is defined as

$$\bar{R}(\boldsymbol{\mu}) = \min\left\{\sum_{i=1}^{n} \mu_i R'_i \colon (R'_1, \ldots, R'_n) \in \mathscr{R}\right\}.$$

The $\mathbf{1}$-weighted optimal sum-rate $\bar{R}(\mathbf{1})$ is simply referred to as optimal sum-rate

$$R_{\text{sum}} = \bar{R}(\mathbf{1}) = \min\left\{\sum_{i=1}^{n} R'_i \colon (R'_1, \ldots, R'_n) \in \mathscr{R}\right\}.$$

C. Guessing Games on Directed Graphs

Given a directed graph $G = (V, E)$, $V = [n]$, consider the following cooperative game among n players. Player $j \in [n]$ is associated to node j and is assigned a value $x_j \in \{0, 1\}^{t_j}$ independently from the other players. It is assumed that player j can observe the values assigned to a subset of the other players $x(A_j) := (x_i \colon i \in A_j)$, $A_j \subseteq [n] \setminus \{j\}$. A_j is referred to as the set of neighbors of player (node) j. The players should simultaneously guess their own value and the goal is to maximize the probability that all players guess their value correctly. No communication is allowed between the players, but they can agree on a strategy beforehand. Note that the setting that we presented here is slightly different from the original one defined by Riis [23] in that the range of the values assigned to the players can be different.

A $(t_1, \ldots, t_n, W)$ guessing strategy is defined by

- n functions $h_j : \prod_{k \in A_j} \{0,1\}^{t_k} \to \{0,1\}^{t_j}$ that maps the values of the neighbors $x(A_j)$ to x_j, for $j \in [n]$, and
- a set W of n-tuples that can be guessed correctly using these functions,

$$W := \bigcap_{j=1}^n \left\{ x^n \in \prod_{i=1}^n \{0,1\}^{t_i} : h_j(x(A_j)) = x_j \right\}.$$

Let P_{win} be the probability of winning, namely, the probability that everyone guesses her value correctly. If the players adopt a $(t_1, \ldots, t_n, W)$ strategy, i.e., player j, $j \in [n]$ uses function h_j to guess her value based on the values of her neighbors, then

$$P_{\text{win}} = |W| / \prod_{i=1}^n 2^{t_i}.$$

Let P_{rand} be the probability of winning if every player guesses her value randomly. As player j, $j \in [n]$, is correct with probability $\frac{1}{2^{t_j}}$ independent of others, we have

$$P_{\text{rand}} = 1 / \prod_{i=1}^n 2^{t_i}.$$

To measure the performance of adopting a strategy, in [23] guessing number is defined as the logarithm of the ratio between the winning probabilities of an strategy and a random guess. In the following, we define the guessing number for the general case where the range of the values assigned to the players are different.

Definition 4. Given a directed graph G, the guessing number of a $(t_1, \ldots, t_n, W)$ guessing strategy is defined as

$$k(G, \mathbf{t}) = \log_s (P_{\text{win}}/P_{\text{rand}}), \quad (6)$$

where $s = 2^{\frac{1}{n} \sum_{i=1}^n t_i}$, and P_{win} is the probability that the players win if they adopt that strategy.

Note that for the case where $t_j = t, \forall j$, we have $k(G, t) = \log(|W|)/t$. The optimal guessing number $k(G)$ of a directed graph G is defined as

$$k(G) = \sup_{\mathbf{t}} \sup k(G, \mathbf{t}), \quad (7)$$

where the second supremum is over all $(t_1, \ldots, t_n, W)$ strategies. The following is an alternative way to measure the performance of adopting a strategy.

Definition 5. Given a directed graph G, the complementary guessing number of a $(t_1, \ldots, t_n, W)$ guessing strategy is defined as

$$k'(G, \mathbf{t}) = \log_s (1/P_{\text{win}}), \quad (8)$$

where $s = 2^{\frac{1}{n} \sum_{i=1}^n t_i}$ and P_{win} is the probability that the players win if they adopt that strategy.

The optimal complementary guessing number is defined in a similar way.

$$k'(G) = \inf_{\mathbf{t}} \inf k'(G, \mathbf{t}), \quad (9)$$

where the second infimum is over all $(t_1, \ldots, t_n, W)$ guessing strategies.

Remark 4. For any $(t_1, \ldots, t_n, W)$ guessing strategy, we have $k'(G, \mathbf{t}) = n - k(G, \mathbf{t})$, and thus

$$k'(G) = n - k(G). \quad (10)$$

Based on the definitions, it is easy to see the following relationship between the guessing games and the distributed storage problem.

Remark 5. Given any directed graph G, a $(t_1, \ldots, t_n, r)$ distributed storage code exists iff a $(t_1, \ldots, t_n, W)$ guessing strategy with $|W| \geq 2^r$ exists.

We will return to the relationship between these two problems at the and of Section V.

IV. CAPACITY REGION, OPTIMAL RATE REGION, AND GUESSING NUMBER VIA CONFUSION GRAPH

In order to shed light on how these three problems are related, we will review the characterization of the capacity region of the index coding problem, the optimal rate region of the distributed storage problem, and the guessing number of guessing games on graphs via the notion of confusion graph.

A. Index Coding

In [8], the capacity region is characterized through the fractional chromatic number of the confusion graph.

Proposition 1. *The capacity region $\mathscr{C}$ of the index coding problem G is the closure of all rate tuples $(R_1, \ldots, R_n)$ such that*

$$R_j \leq \frac{t_j}{\log(\chi_f(\Gamma_{\mathbf{t}}(G)))}, \quad j \in [n], \quad (11)$$

for some $\mathbf{t} = (t_1, \ldots, t_n)$.

B. Distributed Storage and Guessing Game

In [21], Mazumdar characterized the normalized rate of the distributed storage problem through the independence number of the confusion graph. The following is a simple generalization of Mazumdar's result.

Proposition 2. *A rate tuple $(R'_1, \ldots, R'_n)$ is achievable for the distributed storage problem G iff there exists an integer tuple $\mathbf{t} = (t_1, \ldots, t_n)$ such that*

$$R'_j \geq \frac{t_j}{\lfloor \log(\alpha(\Gamma_{\mathbf{t}}(G))) \rfloor}, \quad j \in [n].$$

Proof: Sufficiency (achievability). For a given tuple $\mathbf{t} = (t_1, \ldots, t_n)$, consider a maximal independent set of

the confusion graph $\Gamma = \Gamma_{\mathbf{t}}(G)$. By the definition of the confusion graph, no two n-tuples in this independent set are confusable and therefore for these $\alpha(\Gamma)$ n-tuples, contents of each server is a function of the contents of its recoverability set. Therefore, it is possible to use these $\alpha(\Gamma)$ n-tuples to store $r = \lfloor \log(\alpha(\Gamma)) \rfloor$ bits in the distributed network. This proves the existence of a $(t_1, \ldots, t_n, \lfloor \log(\alpha(\Gamma_{\mathbf{t}}(G))) \rfloor)$ distributed storage code.

Necessity (converse). Consider any $(t_1, \ldots, t_n, r)$ distributed storage code, which has at least 2^r distinct n-tuples that satisfy the required function relationship. By definition, these n-tuples form an independent set of the confusion graph $\Gamma = \Gamma_{\mathbf{t}}(G)$. Thus, $\alpha(\Gamma) \geq 2^r$, or equivalently, $r \leq \lfloor \log(\alpha(\Gamma)) \rfloor$. Therefore, any achievable $(R'_1, \ldots, R'_n)$ must satisfy

$$R'_j \geq \frac{t_j}{\lfloor \log(\alpha(\Gamma_{\mathbf{t}}(G))) \rfloor}, \quad j \in [n],$$

for some $\mathbf{t} = (t_1, \ldots, t_n)$. ∎

Since

$$\frac{k t_j}{\lfloor k \log(\alpha(\Gamma_{\mathbf{t}}(G))) \rfloor} \leq \frac{k t_j}{k \log(\alpha(\Gamma_{\mathbf{t}}(G))) - 1},$$

letting $k \to \infty$ establishes the following.

Proposition 3. *The optimal rate region $\mathscr{R}$ of the distributed storage problem G is the closure of all rate tuples $(R'_1, \ldots, R'_n)$ such that*

$$R'_j \geq \frac{t_j}{\log(\alpha(\Gamma_{\mathbf{t}}))}, \quad j \in [n],$$

for some $\mathbf{t} = (t_1, \ldots, t_n)$.

Using an argument similar to the proof of Proposition 2, the optimal guessing number of the guessing game on directed graph G can be characterized via the independence number of confusion graphs.

Proposition 4. *For the guessing game on directed graph G on n nodes we have*

$$k(G) = \sup_{\mathbf{t} \in \mathbb{Z}_{\geq 0}^n} \log(\alpha(\Gamma_{\mathbf{t}})) / (\frac{1}{n} \sum_{i=1}^n t_i).$$

V. RELATIONSHIPS

In this section, we first discuss the complementarity between the index coding and the distributed storage and then clarify the relationship between guessing games and the distributed storage problem.

A. Index Coding Versus Distributed Storage

For any length-n integer tuple $\mathbf{t}$, the confusion graph $\Gamma_{\mathbf{t}}$ is vertex transitive. Therefore, Lemma 1 yields

$$\log(\chi_f(\Gamma_{\mathbf{t}})) = \sum_{i=1}^n t_i - \log(\alpha(\Gamma_{\mathbf{t}})), \quad (12)$$

which is the key to establishing the complementarity between the index coding problem and the distributed storage problem. We start by stating the following proposition. The proof of the proposition is relegated to Appendix B.

Proposition 5. *For any directed graph G on n nodes and any $\boldsymbol{\lambda} \in \mathbb{Q}_{\geq 0}^n$,*

$$C(\boldsymbol{\lambda}) = \sup_{r : r\boldsymbol{\lambda} \in \mathbb{Z}_{\geq 0}^n} \frac{r}{\log(\chi_f(\Gamma_{r\boldsymbol{\lambda}}(G)))}, \quad (13)$$

$$R(\boldsymbol{\lambda}) = \inf_{r : r\boldsymbol{\lambda} \in \mathbb{Z}_{\geq 0}^n} \frac{r}{\log(\alpha(\Gamma_{r\boldsymbol{\lambda}}(G)))}. \quad (14)$$

The following theorem is the main result of this paper which establishes the complementarity between the $\boldsymbol{\lambda}$-directed capacity $C(\boldsymbol{\lambda})$ and the $\boldsymbol{\lambda}$-directed optimal rate $R(\boldsymbol{\lambda})$, for any nonnegative real tuple $\boldsymbol{\lambda}$.

Theorem 1. *For any directed graph G on n nodes and any $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^n$, we have*

$$\frac{1}{C(\boldsymbol{\lambda})} = \sum_{i=1}^n \lambda_i - \frac{1}{R(\boldsymbol{\lambda})}. \quad (15)$$

Proof: For $\boldsymbol{\lambda} \in \mathbb{Q}_{\geq 0}^n$, we have

$$C(\boldsymbol{\lambda}) = \sup_{r : r\boldsymbol{\lambda} \in \mathbb{Z}_{\geq 0}^n} \frac{r}{\log(\chi_f(\Gamma_{r\boldsymbol{\lambda}}(G)))} \quad (16)$$

$$= \sup_{r : r\boldsymbol{\lambda} \in \mathbb{Z}_{\geq 0}^n} \frac{r}{r \sum_{i=1}^n \lambda_i - \log(\alpha(\Gamma_{r\boldsymbol{\lambda}}(G)))} \quad (17)$$

$$= \frac{1}{\sum_{i=1}^n \lambda_i - \frac{1}{R(\boldsymbol{\lambda})}}, \quad (18)$$

where (16) and (18) follow from Proposition 5, and (17) follows from (12).

For $\boldsymbol{\lambda} \notin \mathbb{Q}^n$, (15) holds due to the continuity of the functions $C(\boldsymbol{\lambda})$ and $R(\boldsymbol{\lambda})$ and $\mathbb{Q}$ being dense in $\mathbb{R}$. ∎

Due to equations (3) and (5), the above theorem establishes the complementarity between the two problems in the strong sense that given the capacity region of the index coding problem G (more precisely, given the boundary points of the capacity region), Theorem 1 completely determines the optimal rate region for the distributed storage problem G and vice versa. This includes as an special case the complementarity between the symmetric capacity of the index coding problem and the symmetric coding rate of the distributed storage established by Mazumdar [21], and by Shanmugam and Dimakis [22].

Corollary 1. *Setting $\boldsymbol{\lambda} = \mathbf{1}$ in Theorem 1 yields*

$$\frac{1}{C_{\text{sym}}} = n - \frac{1}{R_{\text{sym}}}. \quad (19)$$

Equation (12) can also be used to show how the sum-capacity of the index coding problem is related to the optimal sum-rate of the distributed storage problem. The proof is straightforward and is omitted.

Theorem 2.

$$\frac{1}{C_{\text{sum}}} = 1 - \frac{1}{R_{\text{sum}}}. \qquad (20)$$

B. Guessing Games Versus Distributed Storage

The following proposition shows that for any directed graph G, the optimal guessing number of the guessing game on G is inversely related to the optimal sum-rate of the distributed storage problem G.

Theorem 3. *For any directed graph G on n nodes*

$$k(G) = \frac{n}{R_{\text{sum}}}.$$

Proof: For any tuple $\mathbf{t}$, let $s(\mathbf{t}) = 2^{\frac{1}{n}\sum_{i=1}^{n} t_i}$, then

$$R_{\text{sum}} = \min_{R' \in \mathscr{R}} \sum_{i=1}^{n} R_i'$$

$$= \inf_{\mathbf{t}} \frac{\sum_{i=1}^{n} t_i}{\log(\alpha(\Gamma_{\mathbf{t}}))} \qquad (21)$$

$$= \frac{n}{\sup_{\mathbf{t}} \frac{\log(\alpha(\Gamma_{\mathbf{t}}))}{\log(s(\mathbf{t}))}} = \frac{n}{k(G)}, \qquad (22)$$

where (21) follows from Proposition 3, and (22) follows from Proposition 4. ∎

Combining Theorems 2 and 3, and (10) yields the inverse relationship between the optimal complementary guessing number and the index coding sum-capacity.

Corollary 2. For any directed graph G on n nodes

$$k'(G) = \frac{n}{C_{\text{sum}}}.$$

VI. Concluding Remarks

In this paper, we surveyed three problems and elaborated on the connections between their optimal solutions. This can be used in directly translating any achievability scheme or converse result for one problem to the other problems. For example, in [26], it is shown that even for undirected graphs, the guessing strategy based on the fractional clique covering [27] is not optimal. It is also shown via an example that non-Shannon inequalities provide a better bound on the guessing number of undirected graphs than Shannon inequalities. The established connections between the problems answer the corresponding open problems in the context of index coding, namely, even when restricted to undirected graphs, neither the fractional clique covering inner bound nor the polymatroidal outer bound is tight for the index coding problem.

VII. Acknowledgments

Appendix A
Proof of Lemma 3

First assume that $s_j + k = t_j$ for some $j \in [n]$ and some positive integer k and $s_i = t_i, \forall i \neq j$. In this case, we will prove the lemma by contradiction. Assume that (1) does not hold. Then as any confusion graph is vertex transitive, by Lemma 1, we have

$$\alpha(\Gamma_{\mathbf{t}}(G)) > 2^k \alpha(\Gamma_{\mathbf{s}}(G)). \qquad (23)$$

Each vertex of $\Gamma_{\mathbf{t}}$ is associated with an n-tuple that has t_i bits for user $i \in [n]$. Consider the $\alpha(\Gamma_{\mathbf{t}})$ n-tuples in a maximal independent set of $\Gamma_{\mathbf{t}}$ and partition them into (at most 2^k) subsets based on the first k bits of user j. As these k bits are the same for all the members of each partition, after removing these k bits from all the n-tuples, each partition will correspond to an independent set of $\Gamma_{\mathbf{s}}$. However, there are at most 2^k partitions and hence if (23) holds, due to the pigeonhole principle, there exists a partition with more than $\alpha(\Gamma_{\mathbf{s}})$ members, i.e., there exists an independent set of size more than $\alpha(\Gamma_{\mathbf{s}})$ in $\Gamma_{\mathbf{s}}$, which contradicts the definition of the independence number of a graph. Therefore, (1) holds if the two integer tuples differ only at one element. Applying this (at most n times) to length-n tuples that differ only at one element, completes the proof of the lemma.

Appendix B
Proof of Proposition 5

Due to Proposition 1, $\mathscr{C} = cl(\mathscr{C}^{\circ})$, where

$$\mathscr{C}^{\circ} = \{\mathbf{R} \in \mathbb{R}_{\geq 0}^{n} \colon \mathbf{R} \leq \frac{\mathbf{t}}{\log(\chi_f(\Gamma_{\mathbf{t}}))} \text{ for some } \mathbf{t} \in \mathbb{Z}_{\geq 0}^{n}\}.$$

The following lemma shows that the $\boldsymbol{\lambda}$-directed capacity defined in (2), can also be defined in terms of $\mathscr{C}^{\circ}$.

Lemma 4. *For any non-negative real tuple $\boldsymbol{\lambda}$,*

$$C(\boldsymbol{\lambda}) = \sup\{R \colon R\boldsymbol{\lambda} \in \mathscr{C}^{\circ}\}. \qquad (24)$$

Proof: Let $R^* = \sup\{R \colon R\boldsymbol{\lambda} \in \mathscr{C}^{\circ}\}$, then $R^*\boldsymbol{\lambda} \in \mathscr{C}$ and by the definition of $C(\boldsymbol{\lambda})$ we have $R^* \leq C(\boldsymbol{\lambda})$.

Assume that $R^* < C(\boldsymbol{\lambda})$. Let

$$\epsilon = \frac{1}{2}(C(\boldsymbol{\lambda}) - R^*) \min_{i \colon \lambda_i > 0} \lambda_i,$$

and define the ϵ-neighborhood $N_{\epsilon}(C(\boldsymbol{\lambda})\boldsymbol{\lambda})$ as

$$N_{\epsilon}(C(\boldsymbol{\lambda})\boldsymbol{\lambda}) = \bigcap_{i \colon \lambda_i > 0} \{\mathbf{R} \in \mathbb{R}^n \colon e_i^T(C(\boldsymbol{\lambda})\boldsymbol{\lambda} - \mathbf{R}) < \epsilon\},$$

where all of the components of the $n \times 1$ vector e_i are zero, except the i-th component, which is one. If $N_{\epsilon}(C(\boldsymbol{\lambda})\boldsymbol{\lambda}) \cap \mathscr{C}^{\circ} = \emptyset$, then it contradicts the fact that $C(\boldsymbol{\lambda})\boldsymbol{\lambda}$ belongs to $\mathscr{C}$. Alternatively, if $N_{\epsilon}(C(\boldsymbol{\lambda})\boldsymbol{\lambda}) \cap \mathscr{C}^{\circ} \neq \emptyset$, then there exists $R > R^*$ such that $R\boldsymbol{\lambda} \in$

$\mathscr{C}^{\circ}$, which contradicts the definition of R^*. Therefore, $R^* = C(\boldsymbol{\lambda})$ and the proof is complete. ∎

Proof of Proposition 5: Let $\boldsymbol{\lambda} = \left(\frac{a_1}{b}, \ldots, \frac{a_n}{b}\right)^T$, $b \in \mathbb{N}$, and $a_1, \ldots, a_n \in \mathbb{Z}_{\geq 0}$. If $r\boldsymbol{\lambda} \in \mathbb{Z}_{\geq 0}^n$, then due to Proposition 1, we have

$$\frac{r\boldsymbol{\lambda}}{\log(\chi_f(\Gamma_{r\boldsymbol{\lambda}}(G)))} \in \mathscr{C}.$$

Therefore, we have

$$C(\boldsymbol{\lambda}) \geq \sup_{r : r\boldsymbol{\lambda} \in \mathbb{Z}_{\geq 0}^n} \frac{r}{\log(\chi_f(\Gamma_{r\boldsymbol{\lambda}}(G)))}.$$

Next, let R be any real number such that $R\lambda \in \mathscr{C}^0$. Then, there exists integer tuple $\mathbf{t}$ such that $R\boldsymbol{\lambda} \leq \mathbf{t}/\log(\chi_f(\Gamma_{\mathbf{t}}))$, and hence $R \leq \frac{t_i}{\lambda_i}/\log(\chi_f(\Gamma_{\mathbf{t}}))$, for all i such that $\lambda_i > 0$. Let

$$j = \arg \min_{i : \lambda_i > 0} \frac{t_i}{\lambda_i}, \tag{25}$$

then we have

$$R \leq \frac{q}{a_j \log(\chi_f(\Gamma_{\mathbf{t}}))}, \tag{26}$$

where $q = t_j b$. Due to (25), $a_j \mathbf{t} \geq q\boldsymbol{\lambda} = t_j(a_1, \ldots, a_n)^T \in \mathbb{Z}_{\geq 0}^n$ and we have

$$\log(\chi_f(\Gamma_{q\lambda})) \leq \log(\chi_f(\Gamma_{a_j \mathbf{t}})) \tag{27}$$

$$\leq \log(\chi_f(\Gamma_{\mathbf{t}}^{a_j})) \tag{28}$$

$$= a_j \log(\chi_f(\Gamma_{\mathbf{t}})), \tag{29}$$

where (27) follows from Lemma 3, (28) follows from the fact that the set of edges of $\Gamma_{a_j \mathbf{t}}$ is a subset of the set of edges of $\Gamma_{\mathbf{t}}^{a_j}$, and (29) follows from Lemma 2. Combining (26) and (29), we have

$$R \leq \frac{q}{\log(\chi_f(\Gamma_{q\lambda}))} \leq \sup_{r : r\boldsymbol{\lambda} \in \mathbb{Z}_{\geq 0}^n} \frac{r}{\log(\chi_f(\Gamma_{r\boldsymbol{\lambda}}(G)))},$$

which together with Lemma 4 yields

$$C(\boldsymbol{\lambda}) \leq \sup_{r : r\boldsymbol{\lambda} \in \mathbb{Z}_{\geq 0}^n} \frac{r}{\log(\chi_f(\Gamma_{r\boldsymbol{\lambda}}(G)))},$$

and hence (13) holds. Following similar steps as above, one can show that (14) also holds. ∎

REFERENCES

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[2] 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, Jun. 2006.

[3] M. Effros, S. El Rouayheb, and M. Langberg, "An equivalence between network coding and index coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 967–971.

[4] N. Alon, A. Hassidim, E. Lubetzky, U. Stav, and A. Weinstein, "Broadcasting with side information," in *49th Ann. IEEE Symp. Found. Comput. Sci.*, Philadelphia, PA, Oct. 2008, pp. 823–832.

[5] 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. 5811–5823, Sep. 2013.

[6] K. Shanmugam, A. Dimakis, and M. Langberg, "Local graph coloring and index coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 1152–1156.

[7] M. Tahmasbi, A. Shahrasbi, and A. Gohari, "Critical graphs in index coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, Jul. 2014, pp. 281–285.

[8] F. Arbabjolfaei and Y.-H. Kim, "Structural properties of index coding capacity using fractional graph theory," *Accepted to IEEE Int. Symp. Inf. Theory*, 2015.

[9] X. Yi, H. Sun, S. A. Jafar, and D. Gesbert, "Fractional coloring (orthogonal access) achieves all-unicast capacity (DoF) region of index coding (TIM) if and only if network topology is chordal," 2015. [Online]. Available: http://arxiv.org/abs/1501.07870

[10] L. Ong, F. Lim, and C. K. Ho, "The multi-sender multicast index coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 1147–1151.

[11] 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.

[12] X. Huang and S. El Rouayheb, "Index coding and network coding via rank minimization," 2015. [Online]. Available: http://arxiv.org/abs/1504.06844

[13] H. Maleki, V. R. Cadambe, and S. A. Jafar, "Index coding an interference alignment perspective," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5402–5432, Sep. 2014.

[14] S. A. Jafar, "Topological interference management through index coding," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 529–468, Jan. 2014.

[15] H. Sun and S. A. Jafar, "Index coding capacity: How far can one go with only shannon inequalities?" *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3041–3055, Jun. 2015.

[16] S. Unal and A. Wagner, "A rate-distortion approach to index coding," in *Proc. UCSD Inf. Theory Appl. Workshop*, San Diego, CA, Jul. 2014, pp. 1–5.

[17] F. Arbabjolfaei, B. Bandemer, Y.-H. Kim, E. Sasoglu, and L. Wang, "On the capacity region for index coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 962–966.

[18] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1837–1852, Mar. 2012.

[19] D. Papailiopoulos, A. Dimakis, and V. Cadambe, "Repair optimal erasure codes through hadamard designs," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 3021–3037, May 2013.

[20] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh, "Asymptotic interference alignment for optimal repair of mds codes in distributed storage," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2974–2987, May 2013.

[21] A. Mazumdar, "On a duality between recoverable distributed storage and index coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, Jul. 2014, pp. 1977–1981.

[22] K. Shanmugam and A. G. Dimakis, "Bounding multiple unicasts through index coding and locally repairable codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, Jul. 2014, pp. 296–300.

[23] S. Riis, "Information flows, graphs and their guessing numbers," *Electronic journal of Combinatorics*, vol. 14, no. R44, 2007.

[24] E. R. Scheinerman and D. H. Ullman, *Fractional Graph Theory, A Rational Approach to the Theory of Graphs*. New York: Dover Publications, 2011.

[25] N. Alon, A. Hassidim, E. Lubetzky, U. Stav, and A. Weinstein, "Broadcasting with side information," in *49th Ann. IEEE Symp. Found. Comput. Sci.*, Philadelphia, PA, Oct. 2008, pp. 823–832.

[26] R. Baber, D. Christofides, A. N. Dang, S. Riis, and E. R. Vaughan, "Multiple unicasts, graph guessing games, and non-shannon inequalities," in *Proc. IEEE Int. Symp. on Network Coding*, Calgary, AB, June 2013, pp. 1–6.

[27] D. Christofides and K. Markstrm, "The guessing number of undirected graphs," *Electronic journal of Combinatorics*, vol. 18, no. P192, Sep. 2011.