Duality Codes and the Integrality Gap Bound for Index Coding

Hao Yu and Michael J. Neely
University of Southern California

Abstract—This paper considers a base station that delivers packets to multiple receivers through a sequence of coded transmissions. All receivers overhear the same transmissions. Each receiver may already have some of the packets as side information, and requests another subset of the packets. This problem is known as the index coding problem and can be represented by a bipartite digraph. An integer linear program is developed that provides a lower bound on the minimum number of transmissions required for any coding algorithm. Conversely, its linear programming relaxation is shown to provide an upper bound that is achievable by a simple form of vector linear coding. Thus, the information theoretic optimum is bounded by the *integrality gap* between the integer program and its linear relaxation. In the special case when the digraph has a planar structure, the integrality gap is shown to be zero, so that exact optimality is achieved. This work illuminates the relationship between index coding, duality, and integrality gaps between integer programs and their linear relaxations.

I. INTRODUCTION

Consider a noiseless wireless system with N receivers, W independent packets of the same size, and a single broadcast station. The broadcast station has all packets. Each receiver has a subset of the packets as side information, but desires another (disjoint) subset of the packets. The broadcast station must deliver the packets to their intended receivers. To this end, it makes a sequence of (possibly coded) transmissions that are overheard by all receivers. The goal is to find a coding scheme with the minimum number of transmissions (clearance time) such that each user is able to decode its demanded packets. This problem was introduced by Birk and Kol in [1], [2] and is known as the index coding problem.

The formulation of the index coding problem is simple, elegant and captures the essence of broadcasting with side information. However, it still seems to be intractable. The first index coding problem investigated by Birk and Kol considers only the case of unicast packets and can be represented as a directed side information graph. Work by Bar-Yossef et. al. in [3] shows that the performance of the best *scalar linear code* is equal to the graph parameter *minrank* of the side information graph. However, computing the minrank of a given graph is NP-hard [4]. Further, it is known that restricting to scalar linear codes is generally sub-optimal [5], [6].

One branch of research on index coding aims to find tight performance bounds. Work in [3] shows that if the index coding problem has an *undirected* side information graph

(such as when it has symmetric demands) then the minrank is lower-bounded by the independence number of the graph, and upper-bounded by the clique cover number. For the unicast index coding problem, work in [3] shows that the optimal clearance time (with respect to any scalar, vector or non-linear code) is lower-bounded by the maximum acyclic subgraph of the side information graph. Work in [7] generalizes this to the multi-cast case using a directed bipartite graph. It shows that the optimum of the general problem is lower-bounded by the maximum acyclic subgraph induced by deletions of packet vertices, user-vertices and packet-to-user arcs. In [8], a sequence of linear programs is proposed to bound the optimal clearance time.

Another branch of research on index coding focuses on studying the performance of specific codes and specific graph structures. Work in [5] shows that vector linear codes can have strictly better performance compared with scalar linear codes. Work in [6] demonstrates that non-linear codes can outperform both scalar and vector linear codes. Instead of finding the minimum clearance time, Chaudhry et. al. in [9] consider the problem of maximizing the total number of saved transmissions by exploiting a specific code structure together with graph theory algorithms.

This paper studies index coding from a perspective of optimization and duality. It illustrates the inherent duality between the information theoretical lower bound in [7] and the performance of specific codes. Section II extends the bipartite digraph representation of the problem to a weighted bipartite digraph. Section III uses this new graph structure to develop an integer linear program that finds the tightest lower bound given by [7]. Section IV considers the linear programming (LP) relaxation of the integer program, and shows that the dual problem of this relaxation corresponds to a simple form of vector linear codes, called vector cyclic codes. It follows that the information theoretic optimum is bounded by the *integrality gap* between the integer program and its LP relaxation. Section V shows that in the special case when the bipartite digraph is planar, the integrality gap is zero. In this case, optimality is achieved by a scalar cyclic code. Our technical report [10] further considers a different representation of the original integer program that yields a smaller integrality gap. The dual problem of its LP relaxation leads to a more sophisticated *partial clique coding strategy* that time shares between Reed-Solomon erasure codes. Th smaller integrality gap ensures that these codes are closer to the lower bound.

II. THE WEIGHTED BIPARTITE DIGRAPH

There are N receivers, also called *users*. Let $\mathcal{U} = \{u_1, \ldots, u_N\}$ be the set of users. Assume there are W total packets, labeled $\{q_1, \ldots, q_W\}$. For each $m \in \{1, \ldots, W\}$, define $\mathcal{S}_m$ as the set of users in $\mathcal{U}$ that already have packet q_m as side information, and define $\mathcal{D}_m$ as the set of users in $\mathcal{U}$ that demand packet q_m. Without loss of generality, assume that each packet is demanded by at least one user (else, that packet can be eliminated). Thus, the demand set $\mathcal{D}_m$ is non-empty for all $m \in \{1, \ldots, W\}$. On the other hand, the side information sets $\mathcal{S}_m$ can be empty. Indeed, the set $\mathcal{S}_m$ is empty if and only if no user has packet q_m as side information. It is reasonable to assume that the set of users that demand a packet is disjoint from the set of users that already have that packet as side information, so that $\mathcal{S}_m \cap \mathcal{D}_m = \emptyset$ for all $m \in \{1, \ldots, W\}$.

This index coding problem is represented by a bipartite directed graph in [7], [11], where user vertices are on the left of the graph, packet vertices are on the right, and the $\mathcal{S}_m$ and $\mathcal{D}_m$ sets are represented by directed arcs. A directed graph is also called a *digraph*. It is useful to extend this representation to a *weighted* bipartite digraph as follows: Two packets q_k and q_m are said to have the same *type* if $\mathcal{S}_k = \mathcal{S}_m$ and $\mathcal{D}_k = \mathcal{D}_m$. That is, two packets have the same type if they have the same side information and demand sets. Note that if a user demands one packet of a certain type, then it must demand *all* packets of that type. Likewise, if a user has one packet of a certain type as side information, then it must have *all* packets of that type.

Let M be the number of packet types, and let $\mathcal{P} = \{p_1, \ldots, p_M\}$ be the set of types. The index coding problem can be represented by a weighted bipartite digraph $\mathcal{G} = (\mathcal{U}, \mathcal{P}, \mathcal{A}, \mathcal{W}_{\mathcal{P}})$ as follows: Let $\mathcal{U}$ be the set of vertices on the left side of the graph and let $\mathcal{P}$ be the set of vertices on the right side of the graph (see Fig. 1). The arc set $\mathcal{A}$ has a user-to-packet arc (u_n, p_m) if and only if user $u_n \in \mathcal{U}$ has all packets of type p_m. The arc set $\mathcal{A}$ has a packet-to-user arc (p_m, u_n) if and only if user $u_n \in \mathcal{U}$ demands all packets of type p_m. Finally, define $\mathcal{W}_{\mathcal{P}}$ as the set of integral weights associated with packet vertices in $\mathcal{P}$. The weight $w_{p_m} \in \mathcal{W}_{\mathcal{P}}$ of packet vertex $p_m \in \mathcal{P}$ is equal to the number of packets of type p_m. Thus, the total number of packets W satisfies $W = \sum_{m=1}^{M} w_{p_m}$.

A packet is said to be a *unicast packet* if it is demanded by only one user, and is said to be a *multicast packet* if it is demanded by two or more users. An index coding problem is said to be *unicast* if all packets are unicast packets. The first index coding problem introduced by Birk and Kol in [1] was a unicast problem. The current paper also focuses exclusively on the unicast case. Figure 1 shows an example of the weighted bipartite digraph representation for a unicast index coding problem with 3 user vertices and 3 packet types. In this example, packet types p_1, p_2, p_3 are demanded by users u_1, u_2, u_3, respectively, so that $\mathcal{D}_1 = \{u_1\}$, $\mathcal{D}_2 = \{u_2\}$, $\mathcal{D}_3 = \{u_3\}$. Furthermore, the side information sets are as follows:

- Packet type p_1 is contained as side information by users in the set $\mathcal{S}_1 = \{u_2, u_3\}$.

- Packet type p_2 is contained as side information by the user in the set $\mathcal{S}_2 = \{u_3\}$.
- Packet type p_3 is contained as side information by the user in the set $\mathcal{S}_3 = \{u_1\}$.
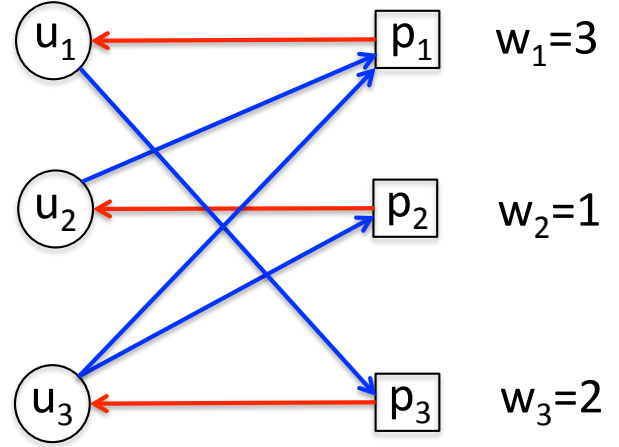

Fig. 1.　The bipartite digraph representation of a unicast index coding problem with 3 user vertices and 3 packet type vertices.

The index coding problem with graph $\mathcal{G} = (\mathcal{U}, \mathcal{P}, \mathcal{A}, \mathcal{W}_{\mathcal{P}})$ can equally represent a system with M variable size packets, where w_{p_m} is the (integer) size of packet p_m. With this interpretation, each packet type represents a single packet. Thus, this paper often refers to packet type p_m as *packet* p_m.

III. THE ACYCLIC SUBGRAPH BOUND AND ITS LP RELAXATION

The following definitions from graph theory are useful. A sequence of vertices $\{s_1, s_2, \ldots, s_K\}$ of a general digraph is defined as a *cycle* if $(s_i, s_{i+1}) \in \mathcal{A}$ for all $i \in \{1, 2, \ldots, K-1\}$, all vertices in $\{s_1, s_2, \ldots, s_{K-1}\}$ are distinct, and $s_1 = s_K$. A digraph is *acyclic* if it contains no cycle. A set of vertices is called a *feedback vertex set* if the removal of vertices in this set leaves an acyclic digraph. In a vertex-weighted digraph, the feedback vertex set with the minimum sum weight is called the *minimum feedback vertex set*.

For the weighted bipartite digraph $\mathcal{G} = (\mathcal{U}, \mathcal{P}, \mathcal{A}, \mathcal{W}_{\mathcal{P}})$ (as defined in the previous section), there exists a subset $\mathcal{P}_{\text{fd}} \subseteq \mathcal{P}$ such that the removal of vertices in $\mathcal{P}_{\text{fd}}$ and all the associated packet-to-user arcs and user-to-packet arcs leaves an acyclic subgraph. In this case, $\mathcal{P}_{\text{fd}}$ is called a *feedback packet vertex set*. A trivial feedback packet vertex set is $\mathcal{P}_{\text{fd}} = \mathcal{P}$ and the corresponding acyclic subgraph has no packet vertex. This trivial feedback packet vertex set has *weight* W, since the sum weight of all packet vertices is W. It is often possible to find a feedback packet vertex set with sum weight smaller than W. The feedback packet vertex set with the minimum sum weight is called the *minimum feedback packet vertex set*. The acyclic subgraph induced by the deletion of the minimum feedback packet vertex set is called the *maximum acyclic subgraph*.

Assume that each transmission from the base station sends a number of bits equal to the number of bits in each of the fixed length packets. It is trivial to satisfy all demands with

W transmissions, where each of the W packets is successively transmitted without coding. However, coding can often be used to reduce the number of transmissions. Let $T_{\min}(\mathcal{G})$ represent the minimum number of transmissions required to deliver all packets to their intended users for an index coding problem defined by the weighted bipartite digraph $\mathcal{G}$. The value $T_{\min}(\mathcal{G})$ considers all possible coding strategies. A theorem in [7] provides an information theoretic lower bound on $T_{\min}(\mathcal{G})$. While the theorem holds for general (possibly multicast) index coding problems, this paper uses it in the unicast case.

Theorem 1 (Theorem 1 and Lemma 1 in [7]): Consider an index coding problem $\mathcal{G} = (\mathcal{U}, \mathcal{P}, \mathcal{A}, \mathcal{W}_{\mathcal{P}})$. Let $\mathcal{P}_{\mathrm{fd}} \subseteq \mathcal{P}$ be a feedback packet vertex set and let $\mathcal{G}'$ be the acyclic subgraph induced by the deletion of $\mathcal{P}_{\mathrm{fd}}$. If $\sum_{p_m \in \mathcal{G}'} w_{p_m} = W'$, then $T_{\min}(\mathcal{G}) \geq W'$.

Suppose the largest cycle in digraph $\mathcal{G}$ involves L packet vertices. Define the set of all cycles in $\mathcal{G}$ as $\mathcal{C} = \bigcup_{i=1}^{L} \mathcal{C}_i$, where $\mathcal{C}_i, i = 2, \ldots, L$ is the set of all cycles involving i packet vertices. These cycles can possibly overlap, i.e., some of them can share common vertices. The number of cycles can possibly be exponential in the number of vertices of the graph. The problem of identifying the tightest lower bound provided by Theorem 1 can be formulated as an integer linear programming (ILP) problem as below:

$$
\begin{aligned}
\max_{x_m, m=1, \ldots, M} \quad & \sum_{m=1}^{M} x_m w_{p_m} \\
(\text{P1}) \qquad \text{s.t.} \quad & \sum_{m=1}^{M} x_m \mathbf{1}_{\{p_m \in C_i\}} \leq i - 1, \\
& \forall C_i \in \mathcal{C}_i, i = 2, \ldots, L \\
& x_m \in \{0, 1\}, \quad m = 1, \ldots, M
\end{aligned}
$$

where $x_m \in \{0, 1\}, m = 1, \ldots, M$ indicates if packet vertex p_m remains in the acyclic subgraph, objective function $\sum_{m=1}^{M} x_m w_{p_m}$ is the sum weight of the acyclic subgraph, $\mathbf{1}_{\{p_m \in C_i\}}$ is the indicator function which equals one if and only if packet vertex p_m participates in cycle $C_i \in \mathcal{C}_i$, and $\sum_{m=1}^{M} x_m \mathbf{1}_{\{p_m \in C_i\}} \leq i - 1$ is the constraint that for each cycle $C_i \in \mathcal{C}_i$, at most $i - 1$ packet vertices remains in the acyclic subgraph. This problem finds the maximum packet weighted acyclic subgraph formed by packet vertex deletion.

The integer constraints of the above problem can be convexified to form the following *linear programming (LP) relaxation*:

$$
\begin{aligned}
\max_{x_m, m=1, \ldots, M} \quad & \sum_{m=1}^{M} x_m w_{p_m} \\
(\text{P1}') \qquad \text{s.t.} \quad & \sum_{m=1}^{M} x_m \mathbf{1}_{\{p_m \in C_i\}} \leq i - 1, \\
& \forall C_i \in \mathcal{C}_i, i = 2, \ldots, L \\
& 0 \leq x_m \leq 1, \quad m = 1, \ldots, M
\end{aligned}
$$

The only difference between problem (P1) and its relaxation (P1') is that the constraints $x_m \in \{0, 1\}$ are changed to $0 \leq x_m \leq 1$.

Define $val(\text{P1})$ as the optimal objective function value of the integer program (P1), being the size of the maximum

acyclic subgraph. Theorem 1 implies that $val(\text{P1}) \leq T_{\min}(\mathcal{G})$. The optimal objective function value for the relaxation (P1') can be written as $val(\text{P1}) + gap(\text{P1}', \text{P1})$, where $gap(\text{P1}', \text{P1})$ is the *integrality gap* between the LP relaxation (P1') and the integer program (P1). Since the relaxation (P1') has less restrictive constraints, the value of $gap(\text{P1}', \text{P1})$ is always non-negative. The next section proves constructively that:

$$val(\text{P1}) \leq T_{\min}(\mathcal{G}) \leq val(\text{P1}) + gap(\text{P1}', \text{P1})$$

Thus, the difference between the minimum clearance time and the maximum acyclic subgraph bound is bounded by the integrality gap $gap(\text{P1}', \text{P1})$. Furthermore, Section V shows that $gap(\text{P1}', \text{P1}) = 0$ in special cases when the digraph $\mathcal{G}$ is planar.

IV. CYCLIC CODES AND LINEAR PROGRAMMING DUALITY

Inspired by the observation that the lower bound in Theorem 1 is closely connected with cycles in graph $\mathcal{G}$, this section considers *cyclic codes* that exploit cycles in $\mathcal{G}$. It is shown that the problem of finding the optimal cyclic code is the dual problem of the LP relaxation (P1'). Thus, the performance gap between the optimal cyclic code and the optimal index code is ultimately bounded by the integrality gap $gap(\text{P1}', \text{P1})$.

A. Cyclic Codes

Suppose there exists a cycle in $\mathcal{G}$ that involves K users $\{u_1, u_2, \ldots, u_K\}$ and K packets of the same size $\{q_1, q_2, \ldots, q_K\}$. In this cycle, user u_1 has q_K and demands q_1, user u_2 has q_1 and demands q_2, user u_3 has q_2 and demands q_3, and so on. If the weight of each packet node is identically one, a *K-cycle coding action* can deliver all K packets by transmitting $Z_i = q_i + q_{i+1}, i = 1, \ldots, K - 1$ with $K - 1$ transmissions, where addition is the mod-2 summation of each bit in both packets. After transmissions, user $u_i \in \{u_2, \ldots, u_K\}$ can decode packet q_i by performing $q_{i-1} + Z_{i-1} = q_{i-1} + (q_{i-1} + q_i) = q_i$. At the same time, user u_1 can decode packet q_1 by performing:

$$
\begin{aligned}
& Z_1 + \ldots + Z_{K-1} + q_K \\
= {}& (q_1 + q_2) + (q_2 + q_3) + \ldots + (q_{K-1} + q_K) + q_K \\
= {}& q_1.
\end{aligned}
$$

The linear index code of $\mathcal{G}$ is said to be cyclic if it uses a sequence of coding actions that involve only cyclic coding actions and direct broadcasts without coding. Linear codes can be further categorized into *scalar linear codes* and *vector linear codes* according to whether the transmitted message is a linear combination of the original packets or the subpackets obtained by subdivisions. In scalar linear codes, each packet is considered as an element of a finite field and the transmitted message is a linear combination of packets over that field. In vector linear codes, each packet is assumed to be sufficiently large and can be divided into many smaller subpackets and the transmitted message is a linear combination of these subpackets instead of the original packets. The problem of

finding the optimal scalar cyclic code to clear $\mathcal{G}$ can be formulated as an ILP problem as below:

$$\min_{\substack{y_{C_i},\forall C_i \in \mathcal{C}_i, \\ i=2,\ldots,L; \\ y_m, m=1,\ldots,M}} \sum_{i=2}^{L}\sum_{C_i \in \mathcal{C}_i} y_{C_i}(i-1) + \sum_{m=1}^{M} y_m$$

(P2)   s.t. $y_m + \sum_{i=2}^{L}\sum_{C_i \in \mathcal{C}_i} y_{C_i}\mathbf{1}_{\{p_m \in C_i\}} \geq w_{p_m},$

$$m = 1,\ldots,M$$
$$y_{C_i} \in \mathbb{Z}^+, \quad \forall C_i \in \mathcal{C}_i, i=2,\ldots,L$$
$$y_m \in \mathbb{Z}^+, \quad m=1,\ldots,M$$

where y_{C_i} is the number of cycle codes over each cycle $C_i \in \mathcal{C}_i, i=2,\ldots,L$, y_m is the number of direct broadcasts over each packet vertex $p_m, m=1,\ldots,M$, objective function $\sum_{i=2}^{L}\sum_{C_i \in \mathcal{C}_i} y_{C_i}(i-1) + \sum_{m=1}^{M} y_m$ is the total number of transmissions, and $y_m + \sum_{i=2}^{L}\sum_{C_i \in \mathcal{C}_i} y_{C_i}\mathbf{1}_{\{p_m \in C_i\}} \geq w_{p_m}$ is the constraint that all the w_{p_m} packets represented by packet vertex p_m are cleared by either cycle codes or direct broadcasts.

The LP relaxation of integer program (P2) is below:

$$\min_{\substack{y_{C_i},\forall C_i \in \mathcal{C}_i, \\ i=2,\ldots,L; \\ y_m, m=1,\ldots,M}} \sum_{i=2}^{L}\sum_{C_i \in \mathcal{C}_i} y_{C_i}(i-1) + \sum_{m=1}^{M} y_m$$

(P2′)   s.t. $y_m + \sum_{i=2}^{L}\sum_{C_i \in \mathcal{C}_i} y_{C_i}\mathbf{1}_{\{p_m \in C_i\}} \geq w_{p_m},$

$$m = 1,\ldots,M$$
$$y_{C_i} \geq 0, \quad \forall C_i \in \mathcal{C}_i, i=2,\ldots,L$$
$$y_m \geq 0, \quad m=1,\ldots,M$$

The only difference between the above problem and the original problem (P2) is that the constraints that y_{C_i} and y_m are non-negative integral are replaced by the relaxed constraints that $y_{C_i} \geq 0$ and $y_m \geq 0$.

Since all the parameters in the linear constraints of (P2′) are integers, an optimal solution can be found that has all variables equal to rational numbers. Let an optimal solution of (P2′) be $y_{C_i}^*, \forall C_i \in \mathcal{C}_i, i=2,\ldots,L; y_m^*, m=1,\ldots,M$, and assume these values are all rational numbers. The optimal vector cyclic code can be constructed as follows. First, one can find an integer θ such that $\theta y_{C_i}^*, \forall C_i \in \mathcal{C}_i, i=2,\ldots,L; \theta y_m^*, m=1,\ldots,M$ are all integers. Next, divide each packet into θ subpackets. After the subdivision, a single cyclic coding action over a cycle C_i is no longer the linear combination of packets but a linear combination of subpackets. Further, a single (uncoded) direct broadcast from a packet vertex p_m is no longer the broadcast of one packet but one subpacket. Then, the optimal vector cyclic code performs $\theta y_{C_i}^*$ cyclic coding actions over each cycle $\forall C_i \in \mathcal{C}_i, i=2,\ldots,L$ and broadcasts θy_m^* subpackets over each packet vertex $p_m, m=1,\ldots,M$.

Define $gap(P2, P2')$ as the non-negative integrality gap between integer program (P2) and its LP relaxation (P2′). Let $T_{\text{cyclic}}(\mathcal{G})$ and $T'_{\text{cyclic}}(\mathcal{G})$ be the clearance time attained by the optimal vector cyclic code and the optimal scalar cyclic code, respectively. Then $T_{\text{cyclic}}(\mathcal{G}) - T'_{\text{cyclic}}(\mathcal{G}) = gap(P2, P2')$.

B. Duality Between Lower Bounds and Cyclic Codes

Lemma 1: The LP relaxations (P1′) and (P2′) form a primal-dual linear programming pair. In particular, the vector cyclic code[1] associated with problem (P2′) achieves a clearance time of $val(P1) + gap(P1', P1)$.

Proof: The Lagrangian function of (P2′) can be written as

$$L(y_{C_i}, y_m, \lambda_m, \mu_{C_i}, \mu_m)$$

$$= \sum_{i=2}^{L}\sum_{C_i \in \mathcal{C}_i} y_{C_i}(i-1) + \sum_{m=1}^{M} y_m - \sum_{i=2}^{L}\sum_{C_i \in \mathcal{C}_i} \mu_{C_i} y_{C_i}$$

$$- \sum_{m=1}^{M} \mu_m y_m + \sum_{m=1}^{M} \lambda_m \left[w_{p_m} - y_m - \sum_{i=2}^{L}\sum_{C_i \in \mathcal{C}_i} y_{C_i}\mathbf{1}_{\{p_m \in C_i\}} \right]$$

$$= \sum_{m=1}^{M} \lambda_m w_{p_m} + \sum_{m=1}^{M} y_m[1 - \lambda_m - \mu_m]$$

$$+ \sum_{i=2}^{L}\sum_{C_i \in \mathcal{C}_i} y_{C_i}\left[(i-1) - \sum_{m=1}^{M} \lambda_m \mathbf{1}_{\{p_m \in C_i\}} - \mu_{C_i} \right]$$

where $\lambda_m \geq 0, m=1,\ldots,M$; $\mu_{C_i} \geq 0, \forall C_i \in \mathcal{C}_i, i=2,\ldots,L$ and $\mu_m \geq 0, m=1,\ldots,M$. The dual problem of (P2′) is defined as:

$$\max_{\substack{\lambda_m \geq 0, m=1,\ldots,M; \\ \mu_{C_i} \geq 0, \\ \forall C_i \in \mathcal{C}_i, i=2,\ldots,L; \\ \mu_m \geq 0, m=1,\ldots,M}} \min_{\substack{y_{C_i} \in \mathbb{R}, \\ \forall C_i \in \mathcal{C}_i, i=2,\ldots,L; \\ y_m \in \mathbb{R}, m=1,\ldots,M}} L(y_{C_i}, y_m, \lambda_m, \mu_{C_i}, \mu_m)$$

Note that,

$$\min_{\substack{y_{C_i} \in \mathbb{R}, \forall C_i \in \mathcal{C}_i, i=2,\ldots,L \\ y_m \in \mathbb{R}, m=1,\ldots,M}} L(y_{C_i}, y_m, \lambda_m, \mu_{C_i}, \mu_m)$$

$$= \begin{cases} \sum_{m=1}^{M} \lambda_m w_{p_m} & \text{if } \begin{array}{l} (i-1) - \sum_{m=1}^{M} \lambda_m \mathbf{1}_{\{p_m \in C_i\}} - \mu_{C_i} = 0, \\ \forall C_i \in \mathcal{C}_i, i=2,\ldots,L; \\ 1 - \lambda_m - \mu_m = 0, m=1,\ldots,M \end{array} \\ -\infty & \text{otherwise} \end{cases}$$

Then, the dual problem of (P2′) can be written as,

$$\max_{\substack{\lambda_m, m=1,\ldots,M; \\ \mu_{C_i}, \forall C_i \in \mathcal{C}_i, \\ i=2,\ldots,L; \\ \mu_m, m=1,\ldots,M}} \sum_{m=1}^{M} \lambda_m w_{p_m}$$

s.t. $(i-1) - \sum_{m=1}^{M} \lambda_m \mathbf{1}_{\{p_m \in C_i\}} - \mu_{C_i} = 0,$

$$\forall C_i \in \mathcal{C}_i, i=2,\ldots,L$$
$$1 - \lambda_m - \mu_m = 0, \quad m=1,\ldots,M$$
$$\lambda_m \geq 0, \quad m=1,\ldots,M$$
$$\mu_{C_i} \geq 0, \quad \forall C_i \in \mathcal{C}_i, i=2,\ldots,L$$
$$\mu_m \geq 0, \quad m=1,\ldots,M$$

Eliminating variables $\mu_{C_i}, \forall C_i \in \mathcal{C}_i, i=2,\ldots,L$ and $\mu_m, m=1,\ldots,M$, we obtain the same problem as (P1′). Thus, the clearance time of the vector cyclic code associated with problem (P2′) is equal to the value of the optimal objective function in problem (P1′), which is $val(P1) +$

---

[1] Similarly, the scalar cyclic code associated with problem (P2) achieves a clearance time of $val(P1) + gap(P1', P1) + gap(P2, P2')$.

$gap(\mathrm{P1}', \mathrm{P1})$. Then, the clearance time of the scalar cyclic code associated with problem (P2) is equal to $val(\mathrm{P1}) + gap(\mathrm{P1}', \mathrm{P1}) + gap(\mathrm{P2}, \mathrm{P2}')$. ∎

Thus far, we have proven the following lower and upper bound for the minimum clearance time of an index coding problem.

$$val(\mathrm{P1}) \leq T_{\min}(\mathcal{G}) \leq val(\mathrm{P1}) + gap(\mathrm{P1}', \mathrm{P1})$$

where the first inequality follows from Theorem 1 and the second inequality follows from Lemma 1. Hence, the performance gap between the optimal index code and the optimal vector cyclic code is ultimately bounded by the integrality gap between integer program (P1) and its LP relaxation (P1′).

There are various techniques for bounding the integrality gaps of integer linear programs, such as the random rounding methods in [12], [13]. Rather than explore this direction, the next section provides a special case where the gap is equal to zero.

V. OPTIMALITY OF CYCLIC CODES IN PLANAR BIPARTITE GRAPHS

In graph theory, a *planar graph* is a graph that can be drawn as a picture on a 2-dimensional plane in a way so that no two arcs meet at a point other than a common vertex. The main result in this section is the following theorem:

Theorem 2: If the bipartite digraph $\mathcal{G}$ for a (unicast) index coding problem is planar, then $val(\mathrm{P1}) = val(\mathrm{P2})$, i.e., $gap(\mathrm{P1}', \mathrm{P1}) = 0$ and $gap(\mathrm{P2}, \mathrm{P2}') = 0$. Hence, the (scalar) cyclic code given by (P2) is an optimal index code.

The proof of Theorem 2 relies on the cycle-packing and feedback arc set duality in arc-weighted planar graphs, which is summarized in the following theorem.

Theorem 3 (Theorem 2.1 in [14]): Let $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{W}_{\mathcal{A}})$ be an arc-weighed planar digraph where $\mathcal{V}$ is the set of vertices, $\mathcal{A}$ is the set of arcs and $\mathcal{W}_{\mathcal{A}}$ is an integral arc weight assignment which assigns each arc $a \in \mathcal{A}$ a non-negative integral weight $w_a \in \mathbb{Z}^+$. Let $\mathcal{C}$ be the set of cycles in $\mathcal{G}$. We have

$$\min \left\{ \sum_{a \in \mathcal{A}} x_a w_a : \textstyle\sum_{a \in \mathcal{A}} x_a \mathbf{1}_{\{a \in C\}} \geq 1, \forall C \in \mathcal{C}; x_a \in \{0,1\}, \forall a \in \mathcal{A} \right\}$$
$$= \max \left\{ \sum_{C \in \mathcal{C}} y_C : \textstyle\sum_{C \in \mathcal{C}} y_C \mathbf{1}_{\{a \in C\}} \leq w_a, \forall a \in \mathcal{A}; y_C \in \mathbb{Z}^+, \forall C \in \mathcal{C} \right\}.$$
$$(1)$$

The integer program on the left-hand-side of (1) is a minimum feedback arc set problem, while the integer program on the right-hand-side of (1) is a cycle packing problem. Both problems are associated with *arc weighted digraphs*. To apply this theorem, we introduce the respective complementary problems of (P1) and (P2). The complementary problem of (P1) is a minimum feedback *packet vertex* set problem and the complementary problem (P2) is a cycle packing problem. However, both complementary problems are associated with *packet vertex-weighted digraphs*. To settle this issue, we modify the bipartite digraph $\mathcal{G}$ to produce an arc-weighted digraph $\mathcal{G}^s$, which is planar if and only if $\mathcal{G}$ is planar. We then show that the minimum feedback packet vertex set problem and the cycle packing problem in $\mathcal{G}$ can be reduced to

the minimum feedback arc set problem and the cycle packing problem in $\mathcal{G}^s$, respectively. The following subsections develop the proof of Theorem 2.

A. Complementary Problems

The integer program (P1) finds the maximum packet weighted acyclic subgraph. This is equivalent to finding the minimum weight feedback packet vertex set. Indeed, this is the set of packets whose deletion induce the maximum packet weighted acyclic subgraph. Thus, an equivalent problem to (P1) is:

$$\begin{aligned} &\min_{x_m, m=1,\ldots,M} \quad \sum_{m=1}^{M} x_m w_{p_m} \\ (\mathrm{P3}) \quad &\text{s.t.} \quad \sum_{m=1}^{M} x_m \mathbf{1}_{\{p_m \in C_i\}} \geq 1, \\ &\qquad\qquad \forall C_i \in \mathcal{C}_i, i = 2,\ldots,L \\ &\qquad x_m \in \{0,1\}, \quad m = 1,\ldots,M \end{aligned}$$

where $x_m \in \{0,1\}, m = 1,\ldots,M$ indicates if packet vertex p_m is selected into the feedback vertex set, objective function $\sum_{m=1}^{M} x_m w_{p_m}$ is the sum weight of the feedback vertex set, $\mathbf{1}_{\{p_m \in C_i\}}$ is the indicator function which equals one only if packet vertex p_m participates in cycle $C_i, \forall C_i \in \mathcal{C}_i, i = 2,\ldots,L$, and $\sum_{m=1}^{M} x_m \mathbf{1}_{\{p_m \in C_i\}} \geq 1$ is the constraint that at least one packet vertex in each cycle is selected into the feedback vertex set. If $x_m^*, m = 1,\ldots,M$ is the optimal solution of (P3) and attains the optimal value W_0, then $\bar{x}_m^* = 1 - x_m^*, m = 1,\ldots,M$ is the optimal solution of (P1) and attains the optimal value $W - W_0$.

In [9], Chaudhry et. al. introduced the concept of complementary index coding problems. Instead of trying to find the minimum number of transmissions to clear the problem, the complementary index coding problem is formulated to maximize the number of saved transmissions by exploiting a specific code structure. Recall that any K-cycle code can deliver K packets in $K - 1$ transmissions and hence one transmission is saved in each K-cycle code.

The complementary index coding problem which aims to maximize the number of saved transmissions by exploiting scalar cycles in $\mathcal{G}$ can be formulated as an ILP problem as below:

$$\begin{aligned} &\max_{\substack{y_{C_i}, \forall C_i \in \mathcal{C}_i, \\ i=2,\ldots,L}} \quad \sum_{i=2}^{L} \sum_{C_i \in \mathcal{C}_i} y_{C_i} \\ (\mathrm{P4}) \quad &\text{s.t.} \quad \sum_{i=2}^{L} \sum_{C_i \in \mathcal{C}_i} y_{C_i} \mathbf{1}_{\{p_m \in C_i\}} \leq w_{p_m}, \\ &\qquad\qquad m = 1,\ldots,M \\ &\qquad y_{C_i} \in \mathbb{Z}^+, \quad \forall C_i \in \mathcal{C}_i, i = 2,\ldots,L \end{aligned}$$

where y_{C_i} is the number of cycle codes over each cycle $C_i, \forall C_i \in \mathcal{C}_i, i = 2,\ldots,L$, objective function $\sum_{i=2}^{L} \sum_{C_i \in \mathcal{C}_i} y_{C_i}$ is the total number of cycle codes, i.e., total number of saved transmissions, and $\sum_{i=2}^{L} \sum_{C_i \in \mathcal{C}_i} y_{C_i} \mathbf{1}_{\{p_m \in C_i\}} \leq w_{p_m}$ is the constraint that each

packet vertex p_m can participate in at most w_{p_m} cycle codes. This is important because if packet p_m has already participated w_{p_m} times in cyclic coding actions, then all of its packets have been delivered and new cyclic coding actions that involve this packet vertex can no longer save any transmissions. If the optimal solution of (P4) is $y^*_{C_i}, \forall C_i \in \mathcal{C}_i, i = 2, \ldots, L$ and attains the optimal value W_0, then the optimal solution of (P2) is $\overline{y}^*_{C_i} = y^*_{C_i}, \forall C_i \in \mathcal{C}_i, i = 2, \ldots, L; \overline{y}^*_m = w_{p_m} - \sum_{i=2}^{L} \sum_{C_i \in \mathcal{C}_i} y^*_{C_i} \mathbf{1}_{\{p_m \in C_i\}}, m = 1, \ldots, M$ and attains the optimal value $W - W_0$.

B. Packet Split Digraphs

Definition 1 (Packet Split Digraphs): Given a graph $\mathcal{G} = (\mathcal{U}, \mathcal{P}, \mathcal{A}, \mathcal{W_P})$, we construct the corresponding packet split digraph $\mathcal{G}^s = (\mathcal{V}^s, \mathcal{A}^s, \mathcal{W}^s)$ as follows:

1) For each packet vertex $p_m \in \mathcal{P}, m = 1, \ldots, M$, we create two packet vertices p_m^{in} and p_m^{out}. Let $\mathcal{V}^s = \mathcal{U} \cup \{p_1^{in}, p_1^{out}, p_2^{in}, p_2^{out}, \ldots, p_M^{in}, p_M^{out}\}$.

2) For each packet vertex $p_m \in \mathcal{P}, m = 1, \ldots, M$, we create a packet-to-packet arc (p_m^{in}, p_m^{out}) in $\mathcal{A}^s$. For each arc $(u_n, p_m) \in \mathcal{A}$, we create a user-to-packet arc (u_n, p_m^{in}) in $\mathcal{A}^s$. For each arc $(p_m, u_n) \in \mathcal{A}$, we create a packet-to-user arc (p_m^{out}, u_n) in $\mathcal{A}^s$.

3) For each arc (p_m^{in}, p_m^{out}) in $\mathcal{A}^s$, we assign a weight which is equal to $w_{p_m} \in \mathcal{W_P}$. For each arc (u_n, p_m^{in}) or (p_m^{out}, u_n) in $\mathcal{A}^s$, we assign an integral weight which is larger than $\sum_{m=1}^{M} w_{p_m}$.

For any bipartite digraph $\mathcal{G}$, the packet split digraph $\mathcal{G}^s$, which is an arc-weighted digraph, can always be constructed. Figure 2 shows the packet split digraph constructed from the bipartite digraph in Figure 1. In any digraph, a set of arcs is called a feedback arc set if the removal of arcs in this set leaves an acyclic digraph. If the digraph is arc-weighted, the feedback arc set with the minimum sum weight is called the minimum feedback arc set.

The following facts summarize the connections between the packet split digraph and the original digraph.

Fact 1: There is a bijection between $\mathcal{G}$ and $\mathcal{G}^s$. This bijection maps user vertices, user-to-packet arcs, packet vertices, and packet-to-user arcs in $\mathcal{G}$ to user vertices, user-to-packet arcs, packet-to-packet arcs, and packet-to-user arcs in $\mathcal{G}^s$, respectively. Thus, this bijection also maps cycles in $\mathcal{G}$ to cycles in $\mathcal{G}^s$.

Fact 2: Every minimum feedback arc set of packet split graph $\mathcal{G}^s$ contains only packet-to-packet arcs and no packet-to-user arcs or user-to-packet arcs.

Proof: In digraph $\mathcal{G}$, each cycle contains at least one packet vertex. By the bipartite property of $\mathcal{G}$ and Fact 1, each cycle $\mathcal{G}^s$ contains at least one packet-to-packet arc. As such, the arc set composed of all packet-to-packet arcs is a feedback arc set of $\mathcal{G}^s$ and this feedback arc set contains no packet-to-user arcs or user-to-packet arcs. Note that the sum weight of this arc set is strictly less than the weight of any single packet-to-user or user-to-packet arc. Any feedback arc set with a packet-to-user arc or user-to-packet arc has a sum weight strictly larger than that of this one and hence can not be a minimum feedback arc set. ∎
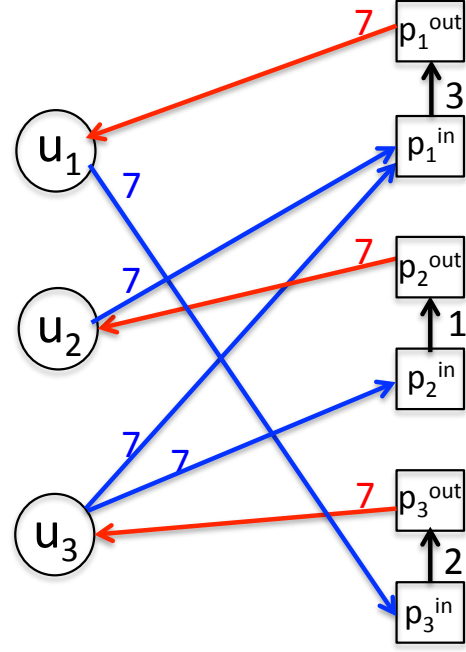

Fig. 2. The packet split digraph constructed from the bipartite digraph given in Figure 1

Fact 3: If $\mathcal{A}^s_{fd} \subseteq \mathcal{A}^s$ is a minimum feedback arc set of the packet split digraph $\mathcal{G}^s$, then a minimum feedback packet vertex set $\mathcal{P}_{fd} \subseteq \mathcal{P}$ of $\mathcal{G}$ is immediate. In addition, the sum weight of $\mathcal{P}_{fd}$ is equal to the sum weight of $\mathcal{A}^s_{fd}$.

Proof: Let $\mathcal{A}^s_{fd}$ be a minimum feedback arc set of $\mathcal{G}^s$ and the sum weight of $\mathcal{A}^s_{fd}$ be W_{fd}. By Fact 2, $\mathcal{A}^s_{fd}$ contains only packet-to-packet arcs. By Fact 1, the packet vertex set $\mathcal{P}_{fd} \subseteq \mathcal{P}$ composed by packet vertices corresponding to arcs in $\mathcal{A}^s_{fd}$ is a feedback packet vertex set of $\mathcal{G}$ and the sum weight of $\mathcal{P}_{fd}$ is equal to W_{fd}. If $\mathcal{P}_{fd}$ is not a minimum feedback packet vertex set, there must exist a feedback packet vertex set, say $\mathcal{P}'_{fd}$, whose sum weight $W'_{fd} < W_{fd}$. By Fact 1, the counterpart of $\mathcal{P}'_{fd}$ in $\mathcal{G}^s$ is a feedback arc set and the sum weight of this feedback arc set is equal to W'_{fd}. Denote this feedback arc set as $\mathcal{A}^{s,\prime}_{fd}$, then $\mathcal{A}^{s,\prime}_{fd}$ has a sum weight strictly less than W_{fd}. This contradicts the fact that $\mathcal{A}^s_{fd}$ is a minimum feedback arc set of $\mathcal{G}^s$. Hence, $\mathcal{P}_{fd}$ must be a minimum feedback packet set of $\mathcal{G}$. ∎

C. Optimality of Cyclic Codes in Planar Graphs

The planarity of a digraph is not affected by arc directions, so that a digraph is planar if and only if its undirected counterpart, where all directed arcs are turned into undirected edges, is planar. In an undirected graph, subdividing an edge (v_1, v_2) is the operation of deleting edge (v_1, v_2), adding a vertex v_0, and adding edges (v_1, v_0) and (v_0, v_2) (see Figure 3a); contracting/shrinking an edge (v_1, v_2) is the operation of deleting edge(v_1, v_2), adding a vertex v_0, replacing any edge (v, v_1) with (v, v_0), and replacing any edge (v_2, v) with (v_0, v) (see Figure 3b). If a graph $\mathcal{G}$ is planar, subdividing and contracting operations preserve the planarity.

In the index coding problem, a packet is said to be a *uniprior packet* if it is contained as side information by only one user.
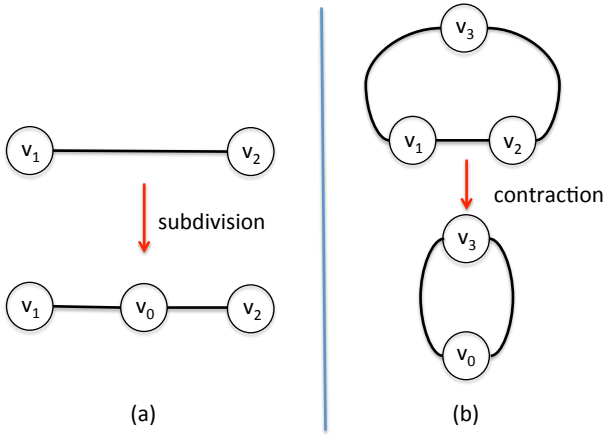
Fig. 3. (a) Subdivision of arc (v_1, v_2). (b) Contraction of arc (v_1, v_2).

The following lemma is proposed to characterize the planarity of the packet split graph $\mathcal{G}^s$.

Lemma 2: Let $\mathcal{G}$ be an index coding problem where each packet vertex is either unicast or uniprior and let $\mathcal{G}^s$ be the packet split digraph of $\mathcal{G}$. $\mathcal{G}^s$ is planar if and only if $\mathcal{G}$ is planar.

Proof:

- "only if" part: This part is relatively easy. Assume $\mathcal{G}^s$ is planar and is drawn in a plane. A planar drawing of $\mathcal{G}$ can be obtained by contracting all the packet-to-packet arcs of $\mathcal{G}^s$ into packet vertices. This part holds for any $\mathcal{G}$ even if some packet vertex is neither unicast nor uniprior.

- "if" part: Assume $\mathcal{G}$ is planar and is drawn in a plane. A planar drawing of $\mathcal{G}^s$ can be obtained by subdividing all packet-to-user arcs and user-to-packet arcs in $\mathcal{G}$. Specifically, for each unicast packet vertex p_m with one single outgoing link, we can subdivide the outgoing link into two parts; add a new node p_m^{out} in the middle and reindex the node p_m as p_m^{in}. Similarly, for each uniprior packet vertex p_m with one single incoming link, we can subdivide the incoming link into two parts; add a new node p_m^{in} in the middle and reindex the node p_m as p_m^{out}. The subdivision operations as above yield a planar drawing of $\mathcal{G}^s$. ∎

Corollary 1: For any unicast index coding problem $\mathcal{G}$, $\mathcal{G}^s$ is planar if and only if $\mathcal{G}$ is planar.

Now we are ready to present the main proof of Theorem 2.

Proof of Theorem 2: Let $\mathcal{G}^s = (\mathcal{V}^s, \mathcal{A}^s, \mathcal{W}^s)$ be the packet spit digraph of $\mathcal{G} = (\mathcal{U}, \mathcal{P}, \mathcal{A}, \mathcal{W}_\mathcal{P})$. Since $\mathcal{G}$ is a planar graph, $\mathcal{G}^s$ is also planar by Corollary 1. Let $\mathcal{C}^s$ be the set of cycles in $\mathcal{G}^s$. The minimum feedback arc set problem in $\mathcal{G}^s$ can be formulated as an ILP problem as follows:

$$
\begin{aligned}
(\text{P3}^*) \quad &\min_{x_a, a \in \mathcal{A}} \sum_{a \in \mathcal{A}}^{M} x_a w_a \\
&\text{s.t.} \quad \sum_{a \in \mathcal{A}} x_a \mathbf{1}_{\{a \in C\}} \geq 1, \quad \forall C \in \mathcal{C}^s \\
&\quad x_a \in \{0, 1\}, \quad a \in \mathcal{A}
\end{aligned}
$$

Similarly, the cycle-packing problem in $\mathcal{G}^s$ can formulated as another integer linear programming as follows:

$$
\begin{aligned}
(\text{P4}^*) \quad &\max_{y_C, C \in \mathcal{C}^s} \sum_{C \in \mathcal{C}^s} y_C \\
&\text{s.t.} \quad \sum_{C \in \mathcal{C}^s}^{L} y_C \mathbf{1}_{\{a \in C\}} \leq w_a, \quad \forall a \in \mathcal{A}^s \\
&\quad y_C \in \mathbb{Z}^+, \forall C \in \mathcal{C}^s
\end{aligned}
$$

By Theorem 3, if $\mathcal{G}^s$ is a planar graph, then (P3*) and (P4*) have the same optimal value. In what follows, we show that the optimal value of (P3) is equal to that of (P3*) and the optimal value of (P4) is equal to that of (P4*).

- **(P3) and (P3*) have the same optimal value:** By Fact 3, the minimum feedback arc set corresponding to the solution of (P3*) can be converted to a minimum feedback packet set solution of (P3) which attains the same optimal objective function value as that of (P3*). On the other hand, by Fact 1, the optimal solution of (P3) can be converted to a solution of (P3*) which attains the same objective value as that of (P3).

- **(P4) and (P4*) have the same optimal value:** By Fact 1, there is a bijection from $\mathcal{C}$ to $\mathcal{C}^s$. This is equivalent to say, there is a bijection from variables in (P4) to those in (P4*). Let $\mathcal{A}_1^s$ be the set of packet-to-packet arcs and $\mathcal{A}_2^s$ be the set of packet-to-user and user-to-packet arcs. So $\mathcal{A}_1^s \cup \mathcal{A}_2^s = \mathcal{A}^s$ and $\mathcal{A}_1^s \cap \mathcal{A}_2^s = \emptyset$. The constraints $\sum_{C \in \mathcal{C}^s} y_C \mathbf{1}_{\{a \in C\}} \leq w_a, \forall a \in \mathcal{A}_1^s$ in (P4*) are essentially the same as the constraints $\sum_{i=2}^{L} \sum_{C_i \in \mathcal{C}_i} y_{C_i} \mathbf{1}_{\{p_m \in C_i\}} \leq w_{p_m}, m = 1, \ldots, M$ in (P4). The other inequality constraints $\sum_{C \in \mathcal{C}^s} y_C \mathbf{1}_{\{a \in C\}} \leq w_a$ over $a \in \mathcal{A}_2^s$ can be shown to be redundant as follows. Let $y_C, C \in \mathcal{C}^s$ be an arbitrary non-negative integral vector which satisfies all the constraints $\sum_{C \in \mathcal{C}^s} y_C \mathbf{1}_{\{a \in C\}} \leq w_a$ over $a \in \mathcal{A}_1^s$. Due to the bipartite property, each cycle in $\mathcal{G}$ contains at least one packet vertex. By Fact 1, each cycle in $\mathcal{G}^s$ contains at least one packet-to-packet arc. Thus, for any $C \in \mathcal{C}^s$, there exists some $a \in \mathcal{A}_1^s$ such that $\mathbf{1}_{\{a \in C\}} = 1$. Then, for any $\bar{a} \in \mathcal{A}_2^s$ we have, $\sum_{C \in \mathcal{C}^s} y_C \mathbf{1}_{\{\bar{a} \in C\}} \leq \sum_{C \in \mathcal{C}^s} y_C \leq \sum_{C \in \mathcal{C}^s} \left[y_C \cdot \sum_{a \in \mathcal{A}_1^s} \mathbf{1}_{\{a \in C\}} \right] = \sum_{a \in \mathcal{A}_1^s} \left[\sum_{C \in \mathcal{C}^s} y_C \mathbf{1}_{\{a \in C\}} \right] \leq \sum_{a \in \mathcal{A}_1^s} w_a < w_{\bar{a}}$ where the first inequality follows from the fact that $0 \leq \mathbf{1}_{\{\bar{a} \in C\}} \leq 1$; the second inequality follows from the fact that for any $C \in \mathcal{C}^s$ there exists some $a \in \mathcal{A}_1^s$ such that $\mathbf{1}_{\{a \in C\}} = 1$; the third inequality follows from the fact that all the constraints $\sum_{C \in \mathcal{C}^s} y_C \mathbf{1}_{\{a \in C\}} \leq w_a$ over $a \in \mathcal{A}_1^s$ are satisfied; and the last inequality follows from the fact that the weight of any packet-to-user arc or user-to-packet-arc is strictly larger than the sum weight of all packet-to-packet arcs. This is to say all the constraints $\sum_{C \in \mathcal{C}^s} y_C \mathbf{1}_{\{a \in C\}} \leq w_a$ over $a \in \mathcal{A}_1^s$ are automatically satisfied and hence redundant. Hence, (P4) and (P4*) are two equivalent optimization problems.

Combining the above facts, we can conclude that the optimal value of (P3) is equal to that of (P4). Denote this value as

W_0. According to Theorem 1, $W - W_0$ is an lower bound of the clearance time of the index coding problem $\mathcal{G}$. On the other hand, $W - W_0$ is the clearance time achieved by the scalar cyclic code corresponding to the solution of (P4), or equivalently (P2). Hence, we can conclude that the cyclic code given by (P2) is the optimal index code. ∎

VI. Conclusion

This paper studies index coding from a perspective of optimization and duality. It illustrates the inherent duality between the information theoretical lower bound, defined by the maximum acyclic subgraph, and the optimal cyclic codes. The performance of cyclic codes is bounded by the integrality gap of the LP relaxation of the tightest lower bound problem. In the special case when the index coding problem has a planar digraph representation, the integrality gap associated with cyclic coding is shown to be zero. So the exact optimality is achieved by cyclic coding.

References

[1] Y. Birk and T. Kol, "Informed-source coding-on-demand (ISCOD) over broadcast channels," in *Proceedings of IEEE 17th Annual Joint Conference of the IEEE Computer and Communications Societies (IN-FOCOM98)*, 1998.

[2] ——, "Coding on demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2825 – 2830, June 2006.

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

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

[5] N. Alon, E. Lubetzky, U. Stav, A. Weinstein, and A. Hassidim, "Broadcasting with side information," in *Proceedings of IEEE 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2008, pp. 823–832.

[6] E. Lubetzky and U. Stav, "Nonlinear index coding outperforming the linear optimum," *IEEE Transactions on Information Theory*, vol. 55, no. 8, pp. 3544–3551, August 2009.

[7] M. J. Neely, A. S. Tehrani, and Z. Zhang, "Dynamic index coding for wireless broadcast networks," *IEEE Transactions on Information Theory*, to appear.

[8] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Index coding via linear programming," *arXiv preprint arXiv:1004.1379*, 2010.

[9] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg, "On the complementary index coding problem," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2011.

[10] H. Yu and M. J. Neely, "Duality codes and the integrality gap bound for index coding," *arXiv preprint arXiv:1307.2295*, 2013.

[11] A. S. Tehrani, A. G. Dimakis, and M. J. Neely, "Bipartite index coding," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2012.

[12] P. Raghavan and C. D. Tompson, "Randomized rounding: a technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.

[13] P. Raghavan, "Probabilistic construction of deterministic algorithms: approximating packing integer programs," *Journal of Computer and System Sciences*, vol. 37, no. 2, pp. 130–143, 1988.

[14] B. Guenin and R. Thomas, "Packing directed circuits exactly," *Combinatorica*, vol. 31, no. 4, pp. 397– 421, 2011.