Joint Index Coding and Incentive Design for Selfish Clients

Yu-Pin Hsu, *Member, IEEE*, I-Hong Hou, *Senior Member, IEEE*, and Alex Sprintson, *Senior Member, IEEE*

***Abstract*— The index coding problem includes a server, a group of clients, and a set of data chunks. While each client wants a subset of the data chunks and already has another subset as its side information, the server transmits some (uncoded or coded) chunks to the clients over a noiseless broadcast channel. The objective of the problem is to satisfy the demands of all clients with the minimum number of transmissions. This paper investigates the index coding setting from a game-theoretical perspective. We consider *selfish* clients, where each selfish client has *private side information* and a *private valuation* of each data chunk it wants. In this context, our objectives are following: 1) to motivate each selfish client to reveal the correct side information and true valuation of each data chunk it wants; 2) to maximize the social welfare, i.e., the total valuation of the data chunks recovered by the clients minus the total cost incurred by the transmissions from the server. Our main contribution is to jointly develop coding and incentive schemes for achieving the first objective perfectly and achieving the second objective optimally or approximately with guaranteed approximation ratios (potentially within some restricted sets of coding matrices).**

***Index Terms*— Network coding, game theory, incentive design.**

I. Introduction

THERE has been a dramatic proliferation of research on *wireless network coding* because it can substantially reduce the number of transmissions by the *broadcast* nature of a wireless medium. On one hand, the wireless medium allows a wireless sender node to broadcast data to all neighboring nodes with a single transmission. On the other hand, a wireless receiver node can overhear the wireless channel and store the overheard data for decoding future transmissions, which is referred to as *side information*. Take the wireless network in Fig. 1-(a) for example, where sender s_1 sends data chunk d_1 to receiver r_1 through a forwarder and sender s_2 sends data chunk d_2 to receiver r_2 also through the forwarder. While receiver r_1 can obtain data chunk d_2 destined to r_2 by overhearing the transmissions from s_2, receiver r_2 can also obtain data chunk d_1 destined to r_1 by overhearing the transmissions from s_1. Leveraging the side information, the forwarder can simply broadcast a single XOR-coded data chunk $d_1 + d_2$, and then both receivers can obtain the data chunks they want by subtracting the side information they have from the received data chunk $d_1 + d_2$. However, with the conventional communication approach (without coding), the forwarder has to transmit both data chunks d_1 and d_2 separately.

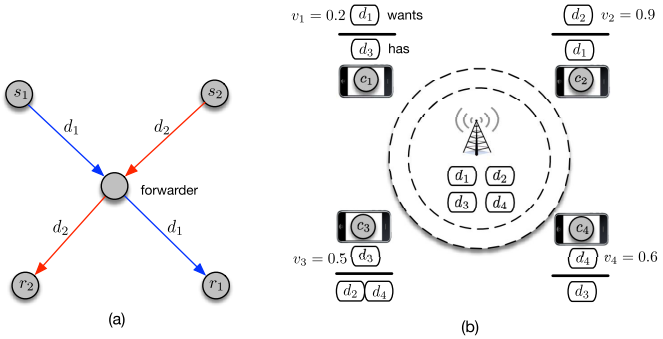

Fig. 1. (a) Senders s_1 and s_2 transmit data chunks d_1 and d_2 to receivers r_1 and r_2, respectively, through a forwarder; (b) A server has a set $\mathbf{D} = \{d_1, d_2, d_3, d_4\}$ of data chunks. Client c_i *wants* data chunk $w_i = d_i$ with its valuation v_i (as shown in the figure), and *has* set H_i (as shown in the figure) as its side information.

The index coding problem is one of fundamental problems in wireless network coding. An instance of the index coding problem includes a server (playing the role of the forwarder in Fig. 1-(a)), a set of wireless clients, and a set $\mathbf{D}$ of data chunks. Each client wants a subset of the data chunks in set $\mathbf{D}$ and has a different subset of the data chunks in set $\mathbf{D}$ given to it as side information. The server can transmit uncoded data chunks or coded data chunks (i.e., combinations of data chunks in set $\mathbf{D}$) to all clients over a noiseless broadcast channel. The goal of the problem is to identify a coding (transmission) scheme requiring the minimum number of transmissions to satisfy the demands of all clients. For example, Fig. 1-(b) depicts an instance of the index coding problem. With the assist of coding, broadcasting only three coded data chunks $d_1 + d_2$, $d_2 + d_3$, and d_4 (over $GF(2)$, i.e., using the bit-wise Exclusive-OR operation) can satisfy all clients.

Since transmitting an (uncoded or coded) data chunk can incur a significant transmission cost (like the energy

Manuscript received May 19, 2020; revised October 22, 2020; accepted December 25, 2020. Date of publication January 5, 2021; date of current version April 16, 2021. This work of Yu-Pin Hsu was supported by Ministry of Science and Technology, Taiwan, under grant number 105-2218-E-305-003- and 107-2221-E-305-007-MY3. This work of I-Hong Hou is supported in part by NSF and Intel under contract number CNS-1719384, in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number W911NF-18-1-0331, and in part by Office of Naval Research under contract N00014-18-1-2048. This work of Alex Sprintson is supported in part by NSF under grant number 1718658. This article was presented in part at the Proceedings of IEEE International Symposium on Information Theory (ISIT). The associate editor coordinating the review of this article and approving it for publication was L. Ong. *(Corresponding author: Yu-Pin Hsu.)*

Yu-Pin Hsu is with the Department of Communication Engineering, National Taipei University, New Taipei City 237303, Taiwan (e-mail: yupinhsu@mail.ntpu.edu.tw).

I-Hong Hou and Alex Sprintson are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: ihou@tamu.edu; spalex@tamu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2021.3049123.

Digital Object Identifier 10.1109/TCOMM.2021.3049123

0090-6778 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

consumption), the server transmits an (uncoded or coded) data chunk only when it is important enough to the clients. Precisely, the server transmits an (uncoded or coded) data chunk only when the overall valuation of the data chunks recovered by that transmitted data chunk can justify the transmission cost. In this context, a client has to evaluate the data chunks it wants. For example, while a client is watching an online video, it prefetches some subsequent data chunks and stores them in a prefetch queue for smoothly playing the video. The lower the size of the prefetch queue is, the higher the valuation of the next data chunk is. Once the prefetch queue is empty, the client would announce a high enough valuation to justify the transmission cost. In contrast, if the size of the prefetch queue is larger, the client would announce a lower valuation and seek an opportunity that the server would transmit a coded data chunk (that can recover the data chunks the client wants) because the total valuation of the recovered data chunks can justify the transmission cost.

Thus, unlike the original index coding problem (where the server has to satisfy all clients), this paper investigates the scenario where the server's transmissions strike a balance between the valuation of the data chunks and the cost of the transmissions. Instead of minimizing the number of transmissions, the first goal of this paper is to develop a coding scheme for maximizing the *social welfare*, i.e., the difference between the valuation of the data chunks that can be recovered by a client and the cost incurred by the transmissions from the server.

To maximize the social welfare, the server needs to know each client's required data chunks, each client's valuations of the data chunks it wants, and each client's side information. Motivated by Fig. 1-(a), the server can forward the data chunks according to the destinations' IP and MAC addresses; hence, it knows each client's required data chunks. However, the server does not know each client's valuations of the data chunks it wants because the data chunks are evaluated by the client itself. Moreover, the server also does not know each client's side information because it does not fully know the client's surrounding environment like Fig. 1-(a). Thus, the server has to ask all clients to submit the unknown information. All prior works on the index coding problem assumed that the server knows the side information, for example, because the clients honestly and periodically broadcast the their side information such as [2]. However, the server cannot expect a *selfish* client to reveal its true information. For example, suppose that the server's strategy is simply to calculate the social welfare according to the information submitted by all clients and then to construct a code for maximizing that social welfare. With that strategy, a selfish client would be reluctant to evaluate the data chunks it wants and would simply submit valuations as high as possible to get a higher chance of recovering the data chunks it wants. In particular, with the wrong information, the server cannot maximize the true social welfare.

To address that issue, an incentive for motivating each selfish client to reveal its true information is needed. This paper uses (money) payment adjustment between the server and the clients as an incentive (or a punishment). The pricing idea has been widely used in network design for motivating a selfish network user to behave in a prescribed way. For example, [3] developed a payment scheme for motivating each network user to submit its true utility function. According to [4, Theorem 9.36 or Lemma 11.9], a payment scheme can motivate a selfish client to submit the true valuation of a data chunk it wants (without considering the possibility that a client can lie about its side information) if and only if 1) there exists a threshold such that the client can recover the data chunk it wants if the valuation it submits is higher than the threshold, but cannot otherwise, and 2) the client is charged the value of the threshold if it can recover the data chunk it wants. Next, we consider a network consisting of a server and a single client that wants data chunk d_1 with valuation 0.6 and has data chunks d_2, d_3 as its side information. Following [4], the server might set some thresholds (satisfying the above theorem or lemma) as follows.

- If the client submits the side information $\{d_2, d_3\}$, then set the threshold to 0.5. If the client submits the valuation more than 0.5, the server transmits $d_1 + d_2 + d_3$ (over $GF(2)$); otherwise, the server transmits nothing.
- If the client submits the side information $\{d_2\}$, then set the threshold to 0.4. If the client submits the valuation more than 0.4, the server transmits $d_1 + d_2$ (over $GF(2)$); otherwise, the server transmits nothing.

Then, the client would submit the true valuation but the incomplete side information of $\{d_2\}$ so that it can recover the data chunk it wants with the minimum payment 0.4. That is, a selfish client in our problem can lie about *its valuations* or *the side information it has* to recover the data chunks it wants with the minimum payment. Thus, we need to revisit those conditions for the incentive design in our problem. The second goal of this paper is to develop a payment scheme for motivating all selfish clients to reveal their true information.

A. Contributions

We investigate the index coding setting in the presence of selfish clients, aiming to propose a joint coding and incentive design (called a mechanism) for 1) motivating each selfish client to truthfully reveal its side information and the valuation of each data chunk it wants and 2) maximizing the social welfare. Our first main contribution is to provide a sufficient condition for mechanisms that can motivate each selfish client to be truthful. Our second main contribution is to develop such mechanisms. With the proposed sufficient condition, we can establish their truthfulness. Moreover, we analyze their optimality or worst-case approximation ratios (potentially within some restricted sets of coding matrices) in terms of the social welfare.

B. Related Works

The index coding problem was introduced in [5] and has become a hot topic. Most related works characterized capacity regions (e.g., [6], [7]) for various network settings or developed computationally efficient coding schemes to (optimally or

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

approximately) achieve the regions (e.g., [8], [9]). In addition to the original index coding problem, some variants of the index coding problem have also been investigated, such as the pliable index coding problem (e.g., [10]) and the secure index coding problem (e.g., [11]). The reader is referred to [12] for extensive surveys. All the prior works on the index coding neglected potentially selfish clients. Thus, our work introduces another variant of the index coding problem by considering selfish clients. Moreover, our work is the first one to investigate the social welfare on the index coding setting.

Many prior works on network coding considered selfish clients. Most of those works (e.g., [13], [14]) analyzed *equilibrium* in the presence of selfish clients. Few works (e.g., [15], [16]) developed incentive schemes for network-coding-enabled networks. In particular, those works focused on incentive design for *fixed* coding schemes. For example, [15] and [16] used random linear codes. In contrast, our work considers a *joint* coding and incentive design problem.

Our problem is also related to combinatorial multi-item auction design (e.g., see [4]) for motivating an auction participant to reveal the true valuation of each item. However, our problem is fundamentally different from the traditional auction design. The traditional auction design can fully manage the item allocation. If an item is/isn't allocated to an auction participant, the auction participant does/doesn't obtain it. However, because a client in our problem can lie about its side information, the server cannot fully manage the recovery of data chunks for a client. While the server decides not to recover a data chunk that a client wants (based on the side information revealed by the client), the client might still recover the data chunk with the hidden side information. Moreover, as discussed in Section I, the results (such as [4, Theorem 9.36 or Lemma 11.9]) for the traditional auction design cannot immediately apply to our problem. Thus, this paper is exploring a new problem in the intersection of coding theory and game theory.

II. System Model

A. Network Model

Consider a wireless broadcast network consisting of a server and a set $c_1, \cdots, c_n$ of n wireless clients, as illustrated in Fig. 1-(b). The server has a set $\mathbf{D} = \{d_1, \cdots, d_m\}$ of m data chunks, where each data chunk d_i represents an element of the Galois field $GF(q)$ of order q. The server can transmit uncoded data chunks or coded data chunks (combined from data chunks in set $\mathbf{D}$) to all clients over a noiseless broadcast channel. Each client wants a single data chunk in set $\mathbf{D}$. Let $w_i \in \mathbf{D}$ be the data chunk client c_i wants. Multiple clients can request the same data chunk, i.e., $w_i = w_j$ for some i and j. Moreover, each client already has a subset of data chunks in set $\mathbf{D}$ as its side information. Let $H_i \subseteq \mathbf{D}$ be the side information client c_i has.

B. Coding Schemes

In this paper, we consider *scalar-linear coding schemes*, where every transmission made by the server is a linear combination of the data chunks in set $\mathbf{D}$. Precisely, the i-th transmission t_i made by the server can be expressed by $t_i = \sum_{j=1}^{m} g_{i,j} d_j$ with *coding coefficient* $g_{i,j} \in GF(q)$ Let $G_i = (g_{i,1}, \cdots, g_{i,m})$ be the *coding vector* of t_i. Moreover, let $G = [G_i]$ be the *coding matrix* whose i-th rows is the coding vector of t_i. For a given coding matrix G, let function $\eta(G)$ represent the total number of transmissions made by the server.

After receiving the transmissions $t_1, \cdots, t_{\eta(G)}$ from the server, client c_i can recover data chunk w_i it wants if and only if there exists a (decoding) function that maps received data chunks $t_1, \cdots, t_{\eta(G)}$ and its side information H_i to data chunk w_i. Note that the server does not need to satisfy all clients in our setting. For a given coding matrix G, let indicator function $\mathbf{1}_i(H_i, G)$ indicate if client c_i can recover data chunk w_i with side information H_i, where $\mathbf{1}_i(H_i, G) = 1$ if it can; $\mathbf{1}_i(H_i, G) = 0$ if it cannot.

Each client c_i has a valuation $v_i \geq 0$ representing the importance of data chunk w_i to it. Suppose that each transmission (from the server) incurs a transmission cost of one unit. The transmission cost can reflect, for example, the power consumption. To capture the tradeoff between the importance of the data chunks and the power consumption, we define a *social welfare* by

$$\sum_{i=1}^{n} v_i \cdot \mathbf{1}_i(H_i, G) - \eta(G), \tag{1}$$

where the first term $v_i \cdot \mathbf{1}_i(H_i, G)$ expresses the valuation of data chunk w_i that can be recovered by client c_i and the second term $\eta(G)$ expresses the cost of the total $\eta(G)$ transmissions made by the server. For example, the social welfare of transmitting $d_1 + d_2$, $d_2 + d_3$, and d_4 (the solution to the index coding problem) in Fig. 1-(b) is $0.2 + 0.9 + 0.5 + 0.6 - 3 = -0.8$. In contrast, the social welfare of transmitting $d_3 + d_4$ is $0.5 + 0.6 - 1 = 0.1$ (where only clients c_3 and c_4 can recover the data chunks they want). Thus, transmitting $d_3 + d_4$ is more valuable than transmitting $d_1 + d_2$, $d_2 + d_3$, and d_4 from the global view. In this paper, we aim to develop a *coding scheme* that identifies a coding matrix G for maximizing the social welfare.

C. Incentive Schemes

To maximize the social welfare, the server has to know data chunk w_i, valuation v_i and side information H_i about each client c_i. As discussed in Section I, we suppose that the server knows w_i for all i but asks each client c_i to submit its valuation and the *indices* (but not the content) of the data chunks in its side information.[1] Let $\hat{v}_i > 0$ and $\hat{H}_i \subseteq \mathbf{D}$ be the valuation and the side information,[2] respectively, revealed by client c_i. Each client c_i can tell a lie, so that $\hat{v}_i$ and $\hat{H}_i$ (obtained by the server) can be different from the true information v_i and H_i (owned by client c_i). Thus, in this paper, we also aim to

[1] Submitting the information would incur a slight transmission cost. The cost can be reflected in valuation v_i while client c_i evaluates data chunk w_i. Thus, the cost is not included in the social welfare in Eq. (1).

[2] If a client submits an index that is out of the indices of the data chunks in set $\mathbf{D}$ as its side information, then the server neglects it (because the server only manages the data chunks in set $\mathbf{D}$). Thus, we assume $\hat{H}_i \subseteq \mathbf{D}$ without loss of generality.

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

develop an incentive scheme for motivating each client c_i to tell the truth so that $\hat{v}_i = v_i$ and $\hat{H}_i = H_i$ for all i. Let $\hat{\mathbf{V}} = \{\hat{v}_1, \cdots, \hat{v}_n\}$ and $\hat{\mathbf{H}} = \{\hat{H}_1, \cdots, \hat{H}_n\}$ be the sets of all corresponding elements. Moreover, let $\hat{\mathbf{V}}_{-i} = \hat{\mathbf{V}} - \{\hat{v}_i\}$ and $\hat{\mathbf{H}}_{-i} = \hat{\mathbf{H}} - \{\hat{H}_i\}$ be the set of all corresponding elements except the one for client c_i.

In this paper, we consider money transfers between the server and the clients as an incentive. Each client c_i has to pay the server for data chunk w_i if the client can recover it. In this context, valuation v_i of data chunk w_i implies the maximum amount of money client c_i is willing to pay to obtain it. Let $p_i \geq 0$ be the payment of client c_i charged by the server. A scheme determining payment p_i for each client c_i is referred to as a *payment scheme*. In general, a payment scheme depends on valuation set $\hat{\mathbf{V}}$, side information set $\hat{\mathbf{H}}$, and coding matrix G (which determines indicator $\mathbf{1}_i(H_i, G)$ for each client c_i). The design of payment schemes and that of coding schemes depend on each other. Thus, we define a *mechanism* π by a joint coding and payment scheme.

The mechanism used by the server is given to all clients. For a given mechanism π, we define a *utility* for client c_i by $u_i(\hat{\mathbf{V}}, \hat{\mathbf{H}}, \pi) = (v_i - p_i) \cdot \mathbf{1}_i(H_i, G)$, which is the difference between the valuation of the data chunk and the money charged by the server if it can recover the data chunk it wants, but is zero otherwise. Though the utility of a client can be computed only when all clients' information is given (because the mechanism π needs all client's information to compute price p_i and coding matrix G), we consider non-cooperative clients where a client has no information about other clients and does not cooperate with other clients. In this context, we aim to develop a mechanism π for guaranteeing that a client can maximize its utility when submitting its true information, for any given information submitted by all other clients. Mathematically, for every $\hat{\mathbf{V}}_{-i}$ and $\hat{\mathbf{H}}_{-i}$, the mechanism π satisfies

$$u_i\big(\{v_i, \hat{\mathbf{V}}_{-i}\}, \{H_i, \hat{\mathbf{H}}_{-i}\}, \pi\big) \geq u_i\big(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}, \pi\big), \tag{2}$$

for all possible $\hat{v}_i$ and $\hat{H}_i$. In game theory, we call the mechanism a *dominant strategy* [17]. A dominant strategy equilibrium is always a Nash equilibrium.

D. Problem Formulation

A mechanism satisfying Eq. (2) is referred to as a *truthful* mechanism. Moreover, if a truthful mechanism yields the social welfare equal to the maximum value of Eq. (1) obtained by a coding scheme that knows the true information $\mathbf{V}$ and $\mathbf{H}$, it is referred to as an *optimal* truthful mechanism. We aim to develop an optimal truthful mechanism such that the social welfare and the utilities of all clients are simultaneously optimized. Our problem involves both the global and local optimization problems.

Note that local utility $u_i(\hat{\mathbf{V}}, \hat{\mathbf{H}}, \pi)$ involves more than one type of private information (i.e., valuation and side information). The traditional incentive design for a single type of private information might be insufficient to motivate a client in our problem to reveal the true information of both types. Thus, Section III characterizes truthful mechanisms for our problem. With the results in Section III, we will develop optimal or approximate truthful mechanisms for various scenarios of our problem.

III. Characterizing Truthful Mechanisms

This section provides a sufficient condition of truthful mechanisms for our problem. To that end, we introduce a type of coding schemes as follows, where we use indicator function $\mathbf{1}_i(\hat{H}_i, G)$ to indicate if client c_i can recover data chunk w_i with side information $\hat{H}_i$ and coding matrix G.

Definition 1: A coding scheme is a **threshold-type** coding scheme if, for every valuation set $\hat{\mathbf{V}}_{-i}$ and side information set $\hat{\mathbf{H}}$, there exists a threshold $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$ such that

- when $\hat{v}_i > \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$, it constructs a coding matrix G such that $\mathbf{1}_i(\hat{H}_i, G) = 1$;
- when $\hat{v}_i < \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$, it constructs a coding matrix G such that $\mathbf{1}_i(\hat{H}_i, G) = 0$,

for all i.

Note that threshold $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$ for client c_i is independent of valuation $\hat{v}_i$ submitted by client c_i, but is dependent on side information $\hat{H}_i$ submitted by client c_i. The next theorem provides a sufficient condition of truthful mechanisms for our problem.

Theorem 1: Suppose that $\hat{H}_i \subseteq H_i$ for all i. A mechanism is truthful if the following four conditions hold:

1) *The coding scheme is a threshold-type coding scheme.*
2) *The payment scheme determines payment $p_i = \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$ for client c_i if $\mathbf{1}_i(\hat{H}_i, G) = 1$, or $p_i = 0$ if $\mathbf{1}_i(\hat{H}_i, G) = 0$, for all coding matrices G constructed by the coding mechanism.*
3) *For every $\hat{\mathbf{V}}_{-i}$ and $\hat{\mathbf{H}}_{-i}$, $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\}) \leq \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\})$ for all $\hat{H}_i \subseteq H_i$.*
4) *For every $\hat{\mathbf{V}}_{-i}$ and $\hat{\mathbf{H}}_{-i}$, if the coding scheme can construct a coding matrix G such that $\mathbf{1}_i(\hat{H}_i, G) = 0$ but $\mathbf{1}_i(H_i, G) = 1$ for some $\hat{v}_i$ and $\hat{H}_i$, then $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\}) = 0$.*

Proof: See Appendix A. ■

The first two conditions in the above theorem claim that a client cannot affect the payment by lying about the valuation of the data chunk it wants, because its payment depends on the valuations submitted by other clients. The third condition claims that a client can minimize its payment when submitting its complete side information. The fourth condition further considers the case when the server cannot fully manage the recovery of the data chunk a client wants, i.e., the server decides not to recover w_i (based on $\hat{v}_i$ and $\hat{H}_i$) but client c_i can still recover it by the hidden side information $H_i - \hat{H}_i$. For that case, the fourth condition claims that the client can obtain the data chunk it wants for free when submitting its true side information. The theorem will be used later to establish the truthfulness of the proposed mechanisms. We remark that the theorem generalizes the sufficient condition in [4, Theorem 9.36 or Lemma 11.9] to the case when a client can lie about the side information it has.

Note that Theorem 1 assumes $\hat{H}_i \subseteq H_i$ for all i. To avoid the case when $\hat{H}_i \not\subseteq H_i$ (i.e., client c_i announces data chunks

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

it does not really have as its side information), the server can use a hash function for *validating* if a client really has the content of a data chunk. The hash function takes the index and the first few bits of a data chunk as an input, performs some operations on it, and returns a value as the output. For example, the hash function can be SHA-256, which has been widely used in the Bitcoin protocol for *validating* a transaction [18] An important property of the hash function is that a slightly different input can produce a completely different output value. That is, the hash output value for data chunk d_i is different for all i. The server can compute the mapping between the data chunks in set $\mathbf{D}$ and their respective hash output values in advance. Then, through a hash output value, the server can know the associated input data chunk.

Thus, the server asks each client to submit the hash output values, instead of submitting the indices of its side information. If a client wants to lie to the server that it has a data chunk in set $\mathbf{D} - H_i$, it could only take the index and some randomly guessed bits (as it does not really have the content of that data chunk) as an input of the hash function. However, because the random bits are unlikely to be exactly the same as those bits in a data chunk in set $\mathbf{D} - H_i$, the hash output value is unlikely to be that produced by a data chunk in set $\mathbf{D} - H_i$. Through the hash output values, the server can identify the side information that a client announces but it does not really have. Thus, with the assist of the hash function techniques, a client does not want to lie about a data chunk it does not really have as its side information. The rest of this paper assumes $\hat{H}_i \subseteq H_i$ for all i and aims to motivate each client to submit its *complete* side information so that $\hat{H}_i = H_i$ for all i.

IV. VCG-Based Mechanism Design

This section proposes an optimal truthful mechanism leveraging the celebrated Vickrey-Clarke-Groves (VCG) approach [4]. Note that the original VCG mechanism provides an auction participant with an incentive to reveal only the true valuation of each item. However, our Theorem 2 will show that the proposed VCG-based mechanism can motivate each selfish client to reveal not only the true valuation of the data chunk it wants but also its complete side information.

Our VCG-based mechanism uses the following function

$$w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G) = \sum_{i=1}^{n} \hat{v}_i \cdot \mathbf{1}_i(\hat{H}_i, G) - \eta(G). \tag{3}$$

The function $w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G)$ is the social welfare in Eq. (1) computed by the information $\hat{\mathbf{V}}$ and $\hat{\mathbf{H}}$ obtained by the server. Then, we propose our VCG-based mechanism as follows, including a VCG-based coding scheme and a VCG-based payment scheme, for computing a coding matrix and payments when obtaining valuation set $\hat{\mathbf{V}}$ and side information set $\hat{\mathbf{H}}$ from the clients.

VCG-based coding scheme: Identify a coding matrix G^* for maximizing function $w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G)$:

$$G^* \in \arg\max_{G \in \mathbf{G}} w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G), \tag{4}$$

where $\mathbf{G}$ is a set of the coding matrices that can be selected. If there is a tie in Eq. (4), it is broken arbitrarily.

VCG-based payment scheme: If $\mathbf{1}_i(\hat{H}_i, G^*) = 0$ for coding matrix G^* computed by Eq. (4), then client c_i is charged $p_i = 0$; otherwise, it is charged

$$p_i = \max_{G \in \mathbf{G}} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) - w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G^*), \tag{5}$$

where valuation set $\{0, \hat{\mathbf{V}}_{-i}\}$ is valuation set $\hat{\mathbf{V}}$ with valuation v_i being substituted by zero. Because of the optimality of the first term in Eq. (5), payment p_i is non-negative. The idea underlying Eq. (5) is to calculate threshold $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$. Suppose that client c_i submits valuation $\tilde{v}_i > \max_{G \in \mathbf{G}} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) - w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G^*)$. Then, we can obtain

$$\begin{aligned} w(\{\tilde{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G^*) &\overset{(a)}{=} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G^*) + \tilde{v}_i \\ &\overset{(b)}{>} \max_{G \in \mathbf{G}} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G), \end{aligned}$$

where (a) is from Eq. (3); (b) is from the assumption of $\tilde{v}_i$. In the above inequality, the term $\max_{G \in \mathbf{G}} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G)$ is the maximum function value among all possible coding matrices G such that $\mathbf{1}_i(\hat{H}_i, G) = 0$. Moreover, we want to emphasize that Eq. (5) calculates payment p_i for coding matrix G^* such that $\mathbf{1}_i(\hat{H}_i, G^*) = 1$. Thus, the inequality implies that the VCG-based coding scheme constructs a coding matrix G such that $\mathbf{1}_i(\hat{H}_i, G) = 1$ when client c_i submits a valuation greater than the value computed by Eq. (5), implying the first and second conditions in Theorem 1.

The idea behind why the VCG-based mechanism can satisfy the third in Theorem 1 is that for a fixed valuation $\tilde{v}_i$, if client c_i reveals more data chunks as its side information $\hat{H}_i$, the VCG-based coding scheme is more likely to construct a coding matrix G^* such that $\mathbf{1}_i(\hat{H}_i, G^*) = 1$. That is, the threshold decreases with the size $|\hat{H}_i|$, implying the third condition in Theorem 1. Moreover, if the VCG-based coding scheme can construct a coding matrix G^* such that $\mathbf{1}_i(\hat{H}_i, G^*) = 0$ but $\mathbf{1}_i(H_i, G^*) = 1$, then when client c_i submits zero valuation $\tilde{v}_i = 0$ and its complete side information $\hat{H}_i = H_i$, the coding matrix G^* is also a maximizer in Eq. (4) (and $\mathbf{1}_i(H_i, G^*) = 1$). That is, the threshold is zero, implying the fourth condition in Theorem 1.

The next theorem establishes the truthfulness and the optimality of the proposed VCG-based mechanism.

Theorem 2: The VCG-based mechanism is an optimal truthful mechanism.

Proof: Appendix B confirms that the VCG-based mechanism is truthful (by Theorem 1). Then, all clients submit the true valuations of the data chunks they want and their complete side information. Moreover, by Eq. (4), the VCG-coding scheme maximizes the social welfare. Thus, the VCG-based mechanism is an optimal truthful mechanism. ■

The proposed VCG-based mechanism involves the combinatorial optimization problems in both Eqs. (4) and (5). Our technical report in [19] shows (by a reduction from the original index coding problem) that the combinatorial optimization problems are NP-hard. To develop computationally efficient mechanisms, the rest of this paper focuses on *sparse coding schemes*, which construct coding matrices G (over $GF(2)$) such that at most two coding coefficients in each coding vector

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SUMMARY OF THE PROPOSED MECHANISMS

proposed coding schemes	**applicable scenarios**	**restrictions on set $\mathbf{G}$**	**optimality results in the restricted set $\mathbf{G}$**	**complexity results**	**associated payment scheme**
VCG-based coding scheme	multiple unicast, multiple multicast	general set	optimal	NP-hard, NP-hard to approximate in multiple multicast	VCG-based payment scheme
VCG-based coding scheme along with Alg. 1	multiple unicast	sparse and instantly decodable coding matrices	optimal	$O(n^3)$	VCG-based payment scheme along with Alg. 1
Alg. 2	multiple unicast	sparse coding matrices	$\max\|C\|$-approximate	$O(n^4)$	Alg. 3
revised Alg. 2 with Alg. 4	multiple unicast	sparse coding matrices	$\sqrt{n}$-approximate	$O(n^5)$	revised Alg. 3 with Alg. 4

G_i is nonzero. That is, a sparse coding scheme combines at most two data chunks based on a small field size $GF(2)$ for each transmission, resulting in a smaller packet size and a lower encoding/decoding complexity. Some sparse coding schemes have been also developed (e.g., [20]) for approximately minimizing the number of transmissions in the original index coding problem.

This paper will consider two different scenarios separately: the *multiple unicast scenario* and the *multiple multicast scenario*. While in the multiple unicast scenario each client wants a different data chunk (i.e., $n = m$), in the multiple multicast scenario many clients can request the same data chunk (i.e., $n \geq m$). Moreover, this paper will also consider two different decoding schemes separately: the *instant decoding scheme* and the *general decoding scheme*. While an instant decoding scheme can combine each *individual* transmission (from the server) with its side information but cannot combine multiple transmissions, a general decoding scheme can combine more than one transmission with its side information. For example, in Fig. 1, client c_2 can instantly decode data chunk d_2 by $d_1 + d_2$; however, it cannot instantly decode data chunk d_1 by $d_1 + d_2$ or $d_2 + d_3$ separately (but it can decode d_1 by combining both $d_1 + d_2$ and $d_2 + d_3$). The instant decoding scheme requires a client to store at most one transmission for recovering the data chunk it wants, resulting in a smaller buffer size and a lower decoding complexity. This paper has those partitions such that each section serves a certain set of clients.

Section V develops computationally efficient mechanisms for the multiple unicast scenario. While Section V-A proposes an algorithm optimally solving Eqs. (4) and (5) in polynomial time when the set $\mathbf{G}$ is restricted to sparse and instantly decodable coding matrices, Section V-B establishes that the combinatorial optimization problems in Eqs. (4) and (5) are still NP-hard when the set $\mathbf{G}$ is restricted to only sparse coding matrices. To cope with the NP-hardness, Sections V-B and V-C develop two approximate truthful mechanisms. Moreover, Section V-D analyzes the computational complexities of the proposed polynomial-time coding schemes. Subsequently, Section VI shows that the combinatorial optimization problems in Eqs. (4) and (5) for the multiple multicast scenario are not only NP-hard but also NP-hard to approximate even using those simple sparse coding schemes. Table I summarizes our main results where the symbol C will be defined soon.

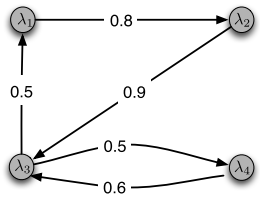

Fig. 2. Weighted dependency graph for the instance in Fig. 1 when $\hat{v}_1 = 0.8$, $\hat{v}_2 = 0.9$, $\hat{v}_3 = 0.5$, $\hat{v}_4 = 0.6$, and $\hat{H}_i = H_i$ for all i.

V. MECHANISM DESIGN FOR THE MULTIPLE UNICAST SCENARIO

This section develops computationally efficient truthful mechanisms for the multiple unicast scenario by proposing polynomial-time algorithms for (optimally or approximately) solving Eq. (4) within the set $\mathbf{G}$ of sparse coding matrices. We remark that an approximate solution to Eqs. (4) and (5) is no longer a truthful mechanism (see Example 1 later). Thus, we devise alternative payment schemes to substitute the previously proposed VCG-based payment scheme for guaranteeing the truthfulness (see Sections V-B and V-C later).

To solve Eq. (4), we introduce a *weighted dependency graph* constructed as follows: given valuation set $\hat{\mathbf{V}}$ and side information set $\hat{\mathbf{H}}$,

- for each client c_i, construct a vertex λ_i;
- for any two clients c_i and c_j such that $w_i \in \hat{H}_j$, construct a directed arc (λ_i, λ_j);
- associate each arc (λ_i, λ_j) with an arc weight $\gamma_{(\lambda_i,\lambda_j)} = \hat{v}_i$.

The weighted dependency graph generalizes the dependency graph in [5] to a weighted version. We denote the weighted dependency graph by $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$, where $\mathbf{\Lambda}$ is the vertex set, $\mathbf{A}$ is the arc set, and $\mathbf{\Gamma}$ is the arc weight set. Fig. 2 illustrates the weighted dependency graph for the instance in Fig. 1.

We make two observations about weighted dependency graphs:

- *for the general decoding scheme*, the server can satisfy all clients in a cycle C in graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ with $|C| - 1$ sparse coded data chunks;
- *for the instant decoding scheme*, the server can satisfy all clients in a cycle C with $|C| = 2$ in graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ with $|C| - 1 = 1$ sparse coded data chunk.

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

For example, with the general decoding scheme, clients c_1, c_2, and c_3 in cycle $(\lambda_1, \lambda_2, \lambda_3)$ of Fig. 2 can recover the data chunks they want with d_1+d_2 and d_2+d_3. In contrast, with the instant decoding scheme, clients c_1, c_2, and c_3 cannot recover the data chunks they want with any two transmissions among d_1+d_2, d_2+d_3, or d_3+d_1. However, clients c_3 and c_4 in cycle (λ_3, λ_4) of Fig. 2 can instantly decode the data chunks they want with d_3+d_4.

We say that a coding scheme *encodes along cycle* C in weighted dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ if it constructs $|C|-1$ sparse coded data chunks for satisfying all clients in the cycle. Note that, for the instant decoding scheme, a coding scheme can encode along cycle C with $|C|=2$ only, according to the above observations. While encoding along a set of (vertex) disjoint cycles can satisfy all clients in those cycles with fewer transmissions than the number of the satisfied clients, all other sparse codes with no cycle being involved cannot (see [20] for details). Those transmissions with no cycle being involved can be substituted by uncoded data chunks without changing the function value in Eq. (3). Thus, we can focus on sparse coding schemes that encodes along disjoint cycles and additionally transmits uncoded data chunk w_i if vertex λ_i is not in those cycles (i.e., client c_i cannot recover data chunk w_i with the coded data chunks along the cycles) but $\hat{v}_i \geq 1$.

We aim to identify a sparse coding matrix G including the coding vectors along a set $\mathbf{C}$ of disjoint cycles and those of uncoded data chunk w_i if λ_i in not in those cycles but $\hat{v}_i \geq 1$, for maximizing

$$w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G) = \sum_{C\in\mathbf{C}} \underbrace{\left(\sum_{\lambda_i\in C} \hat{v}_i - (|C|-1)\right)}_{(a)} + \underbrace{\sum_{\substack{\lambda_i\notin C, \forall C\in\mathbf{C} \\ \hat{v}_i\geq 1}} (\hat{v}_i - 1)}_{(b)}, \quad (6)$$

where (a) is because encoding along cycle C satisfies all clients in the cycle with $|C|-1$ transmissions; (b) considers uncoded data chunks for those clients that submit valuations of no less than one but cannot recover the data chunks they want with the coded data chunks along the cycles. Then, we use the notation $[x]_1^+ = \min\{x, 1\}$ to represent the truncation of x toward one; in particular, we can re-write Eq. (6) in terms of truncated valuations as follow:

$$w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G) = \sum_{C\in\mathbf{C}} \left(\sum_{\lambda_i\in C} [\hat{v}_i]_1^+ - (|C|-1)\right) + \sum_{\substack{\lambda_i\notin C, \forall C\in\mathbf{C} \\ \hat{v}_i\geq 1}} ([\hat{v}_i]_1^+ - 1) + \underbrace{\sum_{\hat{v}_i\geq 1} (\hat{v}_i - 1)}_{(a)}, \quad (7)$$

where (a) adds back the deducted value (caused by the truncation). Because the value of the term $\sum_{\substack{\lambda_i\notin C, \forall C\in\mathbf{C} \\ \hat{v}_i\geq 1}} ([\hat{v}_i]_1^+ - 1)$ is zero and the value of the term $\sum_{\hat{v}_i\geq 1}(\hat{v}_i - 1)$ is constant, it suffices to maximize $\sum_{C\in\mathbf{C}} (\sum_{\lambda_i\in C} [\hat{v}_i]_1^+ - (|C|-1))$.

To that end, we associate each cycle C in weighted dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ with a cycle weight $\gamma(C)$ defined by

$$\gamma(C) = \sum_{a\in C} [\gamma_a]_1^+ - (|C|-1), \quad (8)$$

which implies the difference between the total truncated valuation submitted by the clients in cycle C and the cost of encoding along cycle C. Note that, for the instant decoding scheme, we assign cycle weight $\gamma(C)$ to cycles C with $|C|=2$ only. Then, we can turn our attention to a maximum weight cycle packing problem: identifying a set $\mathbf{C}$ of disjoint cycles for maximizing the total cycle weight $\sum_{C\in\mathbf{C}} \gamma(C)$ in graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$.

Section V-A optimally solves our maximum weight cycle packing problem for the instant decoding scheme. For the general decoding scheme, Sections V-B and V-C propose two approximate solutions to our maximum weight cycle packing problem and their respective payment schemes as the incentives.

A. The Instant Decoding Scheme

This section develops Alg. 1 for optimally solving Eqs. (4) and (5) when set $\mathbf{G}$ is restricted to those sparse and instantly decodable coding matrices. Note that Theorem 2 holds for any set $\mathbf{G}$. Thus, the VCG-based mechanism along with Alg. 1 is an optimal truthful mechanism in the set $\mathbf{G}$ of sparse and instantly decodable coding matrices.

Algorithm 1: Polynomial-Time Algorithm for Solving Eqs. (4) and (5) in the Multiple Unicast Scenario

input : Valuation set $\hat{\mathbf{V}}$ and side information set $\hat{\mathbf{H}}$.
output: Sparse and instantly decodable coding matrix G^* for maximizing function $w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G)$.

1 $G^* \leftarrow \emptyset$;
2 Construct weighted dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$;
3 Construct undirected auxiliary graph $\mathcal{G}(\tilde{\mathbf{\Lambda}}, \tilde{\mathbf{E}}, \tilde{\mathbf{\Gamma}})$;
4 Find a maximum weight matching M^* in $\mathcal{G}(\tilde{\mathbf{\Lambda}}, \tilde{\mathbf{E}}, \tilde{\mathbf{\Gamma}})$;
5 For each edge $(\tilde{\lambda}_i, \tilde{\lambda}_j) \in M^*$, add the coding vector of $w_i + w_j$ to coding matrix G^*;
6 For each vertex $\lambda_i \notin M^*$ but $\hat{v}_i \geq 1$, add the coding vector of data chunk w_i to coding matrix G^*;

Given valuation set $\hat{\mathbf{V}}$ and side information set $\hat{\mathbf{H}}$, Alg. 1 aims to construct a sparse and instantly decodable coding matrix G for maximizing function $w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G)$ in Eq. (7) in polynomial time. To that end, Alg. 1 constructs weighted dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ in Line 2, aiming to identify a set $\mathbf{C}$ of disjoint cycles C with $|C|=2$ for maximizing total cycle weight $\sum_{C\in\mathbf{C}} \gamma(C)$ in set $\mathbf{C}$. To identify such a set of disjoint cycles, Alg. 1 constructs an undirected graph $\mathcal{G}(\tilde{\mathbf{\Lambda}}, \tilde{\mathbf{E}}, \tilde{\mathbf{\Lambda}})$ in Line 3 with the following procedure:

- for each vertex $\lambda \in \mathbf{\Lambda}$, construct a vertex $\tilde{\lambda} \in \tilde{\mathbf{\Lambda}}$;
- for any two vertices $\lambda_i, \lambda_j \in \mathbf{\Lambda}$ such that both arcs (λ_i, λ_j) and (λ_j, λ_i) are in set $\mathbf{A}$, construct an edge $(\tilde{\lambda}_i, \tilde{\lambda}_j) \in \tilde{\mathbf{E}}$;
- associate each edge $(\tilde{\lambda}_i, \tilde{\lambda}_j) \in \tilde{\mathbf{E}}$ with an edge weight $\tilde{\gamma} \in \tilde{\mathbf{\Gamma}}$ such that $\tilde{\gamma}_{(\tilde{\lambda}_i, \tilde{\lambda}_j)} = [\gamma_{(\lambda_i, \lambda_j)}]_1^+ + [\gamma_{(\lambda_j, \lambda_i)}]_1^+ - 1$.

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

With the construction, each cycle C with $|C| = 2$ in graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ corresponds to an edge in graph $\mathcal{G}(\tilde{\mathbf{\Lambda}}, \tilde{\mathbf{E}}, \tilde{\mathbf{\Gamma}})$; in particular, a set of disjoint cycles in graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ corresponds to a matching in graph $\mathcal{G}(\tilde{\mathbf{\Lambda}}, \tilde{\mathbf{E}}, \tilde{\mathbf{\Gamma}})$. Moreover, a cycle weight in graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ corresponds to the edge weight in graph $\mathcal{G}(\tilde{\mathbf{\Lambda}}, \tilde{\mathbf{E}}, \tilde{\mathbf{\Gamma}})$. Thus, a set of disjoint cycles in graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ for maximizing the total cycle weight corresponds to a maximum weight matching in graph $\mathcal{G}(\tilde{\mathbf{\Lambda}}, \tilde{\mathbf{E}}, \tilde{\mathbf{\Gamma}})$. Alg. 1 identifies a maximum weight matching in graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ in Line 4 (by some polynomial-time algorithms like the Edmonds's algorithm [21]). Subsequently, Alg. 1 adds the coding vectors of the coded data chunks (along those cycles corresponding to the maximum weight matching) to coding matrix G^* in Line 5. Finally, if client c_i submitting valuation $\hat{v}_i \geq 1$ is not satisfied by the coding matrix constructed by the maximum weight matching, then Alg. 1 adds the coding vector of data chunk w_i to coding matrix G^* in Line 6. The discussion in this paragraph leads to the following result: Alg. 1 can optimally solve the combinatorial optimization problems in Eqs. (4) and (5) in polynomial time in the set $\mathbf{G}$ of sparse and instantly decodable coding matrices.

B. General Decoding Scheme: $\max|C|$-Approximate Truthful Mechanism

Our technical report in [19] shows (by a reduction from the cycle packing problem [22]) that the combinatorial optimization problems in Eqs. (4) and (5) in the set $\mathbf{G}$ of sparse and general decodable coding matrices are still NP-hard. Thus, this section and next section develop two algorithms (Alg. 2 and its further modification) for approximately solving Eq. (4) in the set $\mathbf{G}$ of sparse coding matrices. To that end, Alg. 2 constructs weighted dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ in Line 2, aiming to approximately solving our maximum weight cycle packing problem.

The idea underlying Alg. 2 is to iteratively identify a maximum weight cycle in a greedy way. Note that, in general, identifying a maximum weight cycle in a graph is NP-hard [23]. However, for our problem, we can observe that cycle weight $\gamma(C)$ of cycle C in Eq. (8) can be written as

$$\gamma(C) = 1 - \sum_{a \in C} (1 - [\gamma_a]_1^+). \tag{9}$$

By associating each arc $a \in \mathbf{A}$ with an arc cost $\zeta_a = 1 - [\gamma_a]_1^+$, we can associate each cycle C with a cycle cost $\zeta(C) = \sum_{a \in C} \zeta_a$, which is the total arc cost in cycle C. Then, cycle weight $\gamma(C)$ in Eq. (9) becomes $\gamma(C) = 1 - \zeta(C)$. Removing the constant, a maximum weight cycle C minimizes cycle cost $\zeta(C)$. Thus, Alg. 2 identifies a minimum cost cycle in Line 5 (by some polynomial-time algorithms like the Floyd-Warshall algorithm [24]), followed by adding the coding vectors of the coded data chunks along the cycle to coding matrix G in Line 6. Subsequently, Alg. 2 removes the cycle from the present graph in Line 7. The condition in Line 4 guarantees that the maximum weight cycle in the present graph has a non-negative weight. Finally, if client c_i submitting valuation $\hat{v}_i \geq 1$ is not in those selected cycles, then Alg. 2 adds the coding vector of data chunk w_i to G in Line 11.

Algorithm 2: $\max|C|$-Approximate Coding Scheme for the Multiple Unicast Scenario

input : Valuation set $\hat{\mathbf{V}}$ and side information set $\hat{\mathbf{H}}$.
output: Sparse coding matrix G.

1 $G \leftarrow \emptyset$;
2 Construct weighted dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$;
3 Associate each arc $a \in \mathbf{A}$ with arc cost $\zeta_a = 1 - [\gamma_a]_1^+$;
4 **while** *there is a cycle in the present graph* $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ *whose cycle cost is less than or equal to one* **do**
5 Find a cycle C in the present graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ for minimizing cycle cost $\zeta(C)$;
6 Add the coding vectors of the $|C| - 1$ coded data chunks along cycle C to coding matrix G;
7 Remove all vertices in cycle C and their incident arcs from the present graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$;
8 **end**
9 **for** $i \leftarrow 1$ **to** n **do**
10 **if** λ_i *is not in the selected cycles but* $\hat{v}_i \geq 1$ **then**
11 Add the coding vector of data chunk w_i to coding matrix G;
12 **end**
13 **end**

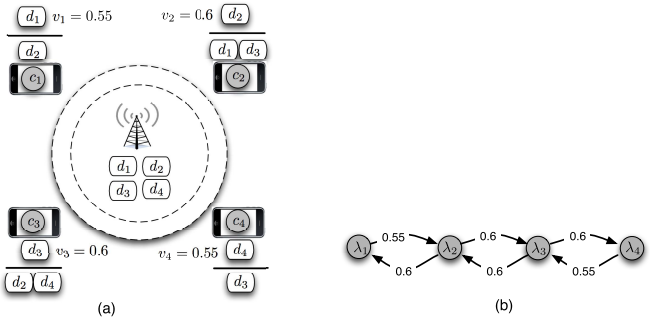

Fig. 3. (a) Counter-example of the truthfulness property when we use Alg. 2 to solve Eqs. (4) and (5); (b) The weighted dependency graph.

Let $G_{\text{alg 2}}$ be the coding matrix produced by Alg. 2 and let G^* be a sparse coding matrix maximizing function $w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G)$. The next theorem analyzes the *approximation ratio* $\frac{w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G^*)}{w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G_{\text{alg 2}})}$ of Alg. 2.

Theorem 3: The approximation ratio (with respect to an optimal sparse coding matrix) of Alg. 2 is the maximum cycle length in weighted dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$.

Proof: See Appendix C. ∎

Because of Theorem 3, we refer to Alg. 2 as $\max|C|$-approximate coding scheme. Next, we show that applying $\max|C|$-approximate coding scheme to solve Eqs. (4) and (5) is no longer a truthful mechanism.

Example 1: Look at Fig. 3. First, suppose that all clients submit the true valuations of the data chunks they want. Then, Alg. 2 produces $d_2 + d_3$ along cycle (λ_2, λ_3) in Fig. 3(b). In this case, client c_1 has zero utility. Second, suppose that client c_1 submits $\hat{v}_1 = 0.7$ but other clients submit the true valuations of the data chunks they want. Then, Alg. 2 produces $d_1 + d_2$ and $d_3 + d_4$ along cycles (λ_1, λ_2) and (λ_3, λ_4),

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

respectively. By solving Eq. (5) with Alg. 2, client c_1 is charged $p_1 = (0.6+0.6-1)-(0.6+0.6+0.55-2) = 0.45$. In this case, client c_1 has utility $0.55-0.45 = 0.1$. Client c_1 can obtain a higher utility by lying about the valuation of data chunk w_i.

Algorithm 3: Payment Scheme for Those Clients With $\mathbf{1}_i(\hat{H}_i, G) = 1$ Under Coding Matrix G Constructed by Alg. 2

input : Valuation set $\hat{\mathbf{V}}$, side information set $\hat{\mathbf{H}}$, and client c_i with $\mathbf{1}_i(\hat{H}_i, G) = 1$ under coding matrix G constructed by Alg. 2.
output: Payment p_i for client c_i.

1 Construct weighted dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$;
2 Associate each arc $a \in \mathbf{A}$ with arc cost ζ_a:

$$\zeta_a = \begin{cases} 1 & \text{if } a = (\lambda_i, \lambda_j) \text{ for some } j; \\ 1 - [\gamma_a]_1^+ & \text{if } a \neq (\lambda_i, \lambda_j) \text{ for all } j. \end{cases}$$

3 $p_i \leftarrow 1$;
4 **while** *there exists a cycle in the present graph* $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ *whose cost is less than or equal to one, and there exists a cycle containing vertex* λ_i **do**
5 Find a cycle $C_1 = \arg\min_C \zeta(C)$ in the present graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ for minimizing the cycle cost;
6 Find a cycle $C_2 = \arg\min_{C \cap \lambda_i \neq \emptyset} \zeta(C)$ in the present graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ that contains vertex λ_i and minimizes the cycle cost among those cycles containing vertex λ_i;
7 $p_i \leftarrow \min\{p_i, \zeta(C_2) - \zeta(C_1)\}$;
8 Remove all vertices in cycle C_1 and their incident arcs from the present graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$;
9 **end**

To address the issue in the above example, we propose a payment scheme in Alg. 3 so that the joint design of Algs. 2 and 3 is a truthful mechanism. The underlying idea of Alg. 3 is to calculate threshold $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$ for each client c_i (with $\mathbf{1}_i(\hat{H}_i, G) = 1$ under coding matrix G constructed by Alg. 2) as payment p_i. To that end, Alg. 3 constructs weighted dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$ in Line 1; moreover, Alg. 3 associates each arc $a \in \mathbf{A}$ with an arc cost ζ_a in Line 2. Note that Alg. 3 defines the arc costs in a different way from Alg. 2; precisely, Alg. 3 associates each outgoing arc from vertex λ_i with the cost of one unit (i.e., assuming valuation $\hat{v}_i = 0$). Then, Alg. 3 calculates the difference of the cycle costs between cycle C_1 (in Line 5) and cycle C_2 (in Line 6), where cycle C_1 has the *globally* maximum weight but cycle C_2 has the *locally* maximum weight among those cycles containing vertex λ_i. While the value of $1-\zeta(C_1)$ (see Eq. (9)) is analogous to the first term of Eq. (5), that of $1-\zeta(C_2)$ is analogous to the second term of Eq. (5). Thus, the difference $\zeta(C_2) - \zeta(C_1)$ of the cycle costs in Line 7 for each iteration is the minimum valuation $\hat{v}_i$ submitted by client c_i such that a cycle containing vertex λ_i can be selected by Line 5 of Alg. 2 in that iteration. Then, Alg. 3 identifies threshold $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$ by searching for the minimum among all iterations in Line 7 along with the initial valuation of p_i being 1 as in Line 3 (because each client c_i can recover the data chunk it wants when submitting $\hat{v}_i \geq 1$).

Following Appendix B, we can show shows that the joint Algs. 2 and 3 is a truthful mechanism.

Theorem 4: In the multiple unicast scenario, the mechanism consisting of the coding scheme in Alg. 2 and the payment scheme in Alg. 3 is a truthful mechanism.

Proof: See our technical report in [19]. ■

Algorithm 4: Identifying Cycle C for Maximizing $\frac{\gamma(C)}{\sqrt{|C|}}$

input : Weight dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$.
output: Cycle C maximizing $\frac{\gamma(C)}{\sqrt{|C|}}$.

1 $C \leftarrow \emptyset$;
2 **for** $i \leftarrow 2$ **to** n **do**
3 Find a cycle $C' = \arg\min_{C''} \zeta(C'')$ subject to $|C''| \leq i$;
4 $C \leftarrow \arg\max_{C' \text{ or } C} \{\frac{\gamma(C')}{\sqrt{|C'|}}, \frac{\gamma(C)}{\sqrt{|C|}}\}$;
5 **end**

C. $\sqrt{n}$-Approximate Truthful Mechanism

This section proposes another approximate sparse coding scheme and its corresponding payment scheme for guaranteeing the truthfulness. The approximate coding scheme modifies the previously proposed Alg. 2. The modified approximate coding scheme substitutes Line 5 of Alg. 2 (i.e., identifying a cycle for maximizing cycle weight $\gamma(C)$) by identifying a cycle C for maximizing $\frac{\gamma(C)}{\sqrt{|C|}}$. The underlying idea is to maximize cycle weight $\gamma(C)$ (as in Alg. 2) and at the same time to minimize the number of transmissions (because shorter cycle lengths $|C|$ can yield more cycles). To that end, we propose Alg. 4 for obtaining such a cycle in a weighted dependency graph. Line 3 of Alg. 4 searches for a cycle for maximizing cycle weight subject to the cycle length being no more than i. Then, Line 4 of Alg. 4 can identify cycle C for maximizing $\frac{\gamma(C)}{\sqrt{|C|}}$ subject to the cycle length being no more than i; in particular, Line 4 can identify cycle C for maximizing $\frac{\gamma(C)}{\sqrt{|C|}}$ in the last iteration. See our technical report in [19] for proving the correctness of Alg. 4. The next theorem provides the approximation ratio of the modified approximation algorithm.

Theorem 5: In the multiple unicast scenario, substituting Line 5 of Alg. 2 by identifying a cycle C for maximizing $\frac{\gamma(C)}{\sqrt{|C|}}$ yields the approximation ratio (with respect to the welfare yielded by an optimal sparse coding matrix) of $\sqrt{n}$.

Proof: See Appendix D. ■

Because of Theorem 5, we refer to the coding scheme modified from Alg. 2 as $\sqrt{n}$-approximate coding scheme. To the best of our knowledge, [22] developed the best approximation algorithm for a cycle packing problem (identifying the maximum number of disjoint cycles), which is a special case of our problem when $\hat{v}_i = 1$ for all i. That paper showed that the approximation ratio of that algorithm is $\sqrt{n}$; furthermore,

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

TABLE II
DIFFERENCES OF THE SOCIAL WELFARES

the number n of client	10	16	22	28	34	40	46
$\lvert H_i\rvert = 3$	0.03	0.1	0.12	0.17	0.17	0.2	0.21
$\lvert H_i\rvert = 6$	0	0.1	0.12	0.2	0.22	0.38	0.39

it conjectured that $\sqrt{n}$ is the best approximation ratio for that cycle packing problem.

Following Theorem 4, we can establish the truthfulness as follows.

Theorem 6: Substituting Line 5 of Alg. 2 and Lines 5 and 6 of Alg. 3 by identifying a cycle C for maximizing $\frac{\gamma(C)}{\sqrt{|C|}}$ yields a truthful mechanism.

D. Complexities of the Proposed Coding Schemes

This section investigates the computational complexities of the three proposed coding schemes for the multiple unicast scenario: 1) Alg. 1 for the set $\mathbf{G}$ of sparse and instantly decodable coding matrices; 2) $\max|C|$-approximate coding scheme (Alg. 2) for for the set $\mathbf{G}$ of sparse coding matrices; 3) $\sqrt{n}$-approximate coding scheme (modified Alg. 2 along with Alg. 4) for the set $\mathbf{G}$ of sparse coding matrices. All three schemes are based on a weighted dependency graph. Constructing a weighted dependency graph takes $O(n^2)$ steps to check all pairs of clients.

Regarding Alg. 1, we can apply the Edmond's maximum weight matching algorithm [21] to Line 4 of Alg. 1, whose complexity is $O(n^3)$. Then, the complexity of Alg. 1 is $O(n^3)$.

Regarding Alg. 2, we can apply the Floyd-Warshall algorithm [24] to Line 3 for each iteration, whose complexity is $O(n^3)$. Because there are at most $n/2$ cycles in a weighted dependency graph (i.e., at most $n/2$ iterations from Line 4), the complexity of $\max|C|$-approximate coding scheme is $O(n^4)$.

Regarding Alg. 4, we can apply the Bellman-Ford algorithm [24] to Line 3 for each iteration i, whose complexity is $O(i \cdot n^2)$. Hence, the complexity of Alg. 4 is $O(n^4)$; in turn, the complexity of the $\sqrt{n}$-approximate coding scheme is $O(n^5)$.

E. Numerical Results

This section numerically analyzes the proposed coding schemes for the multiple unicast scenario via computer simulations. Fig. 4 simulates Alg. 1 with the instant decoding scheme and the two approximate coding schemes with the general decoding scheme. The two sub-figures display the social welfare when each client has 3 and 6 data chunks, respectively, in its side information. The experiment setting is following: We simulate n clients (x-axle) and set $\mathbf{D} = \{d_1, \cdots, d_n\}$, where client c_i wants data chunk d_i. Valuation v_i of data chunk d_i is uniformly picked between 0 and 1. The data chunks in side information H_i of client c_i is randomly selected from set $\mathbf{D} - \{d_i\}$. As a result of the proposed payment schemes as the incentives, we can guarantee that all clients submit their true information. Thus, we simulate the proposed coding schemes along with the true information (like

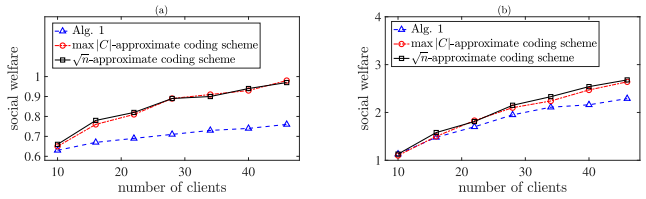

Fig. 4. Social welfare of Alg. 1, $\max|C|$-approximate coding scheme in Alg. 2, and $\sqrt{n}$-approximate coding scheme modified from Alg. 2: (a) each client has 3 data chunks in its side information; (b) each client has 6 data chunks in its side information.

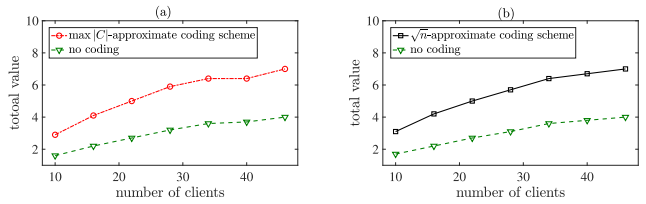

Fig. 5. Benefit from the approximate coding schemes when each client has 3 data chunks in its side information.

no selfish clients). Note that under that setting, the proposed approximate coding schemes can approximate the *true* social welfare in Eq. (1). All results are averaged over 500 simulation times.

From Fig. 4, we can observe that even though both approximate coding schemes cannot achieve the maximum social welfare (in the set $\mathbf{G}$ of sparse coding matrices), they still outperform Alg. 1 (that is optimal in the set of sparse and instantly decodable coding matrices). The result tells us that the proposed approximate coding schemes can take advantage of the general decoding scheme. Moreover, Table II shows the difference between the social welfare yielded by $\sqrt{n}$-approximate coding scheme in Alg. 2 and that yielded by Alg. 1 for various fixed numbers of clients and fixed sizes of side information. From the table, the difference when $|H_i| = 6$ is almost more than that when $|H_i| = 3$ (especially when the number of clients is larger like $n = 34$, 40, or 46), i.e., the advantage from the general decoding scheme becomes more obvious when the clients have more data chunks as their side information. That would be because when clients have more data chunks as their side information, there are more cycles in the weighted dependency graph whose lengths are more than two.

While the proposed approximate coding schemes cannot be optimal in the set $\mathbf{G}$ of sparse coding matrices, we want to further validate them via simulations. Figs. 5 and 6 display the total valuation $\sum_{i=1}^{n} v_i \cdot \mathbf{1}_i(H_i, G)$ of the data chunks that can be recovered. For a fair comparison of the total valuation, we have to fix the number of transmissions from the server for

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

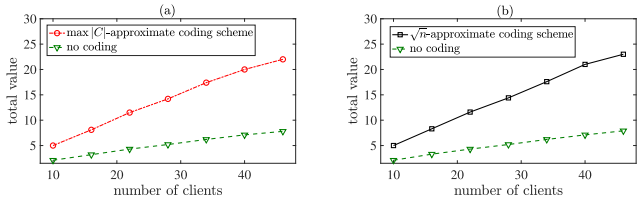

Fig. 6. Benefit from the approximate coding schemes when each client has 6 data chunks in its side information.

various coding schemes. Thus, we cannot compare the total valuation yielded by the approximate coding schemes with that yielded by Alg. 1 (as they cause different numbers of transmissions). Instead, Figs. 5 and 6 compare the approximate coding schemes with the best "no coding" scheme. The results for the "no coding" scheme in Figs. 5 and 6 are obtained in the following way: we first obtain the number $\eta(G)$ of transmissions incurred by a given approximate coding scheme, and then calculate the sum of the top $\eta(G)$ valuations of the data chunks in set $\mathbf{D}$ (which is the maximum total valuation when the server transmits $\eta(G)$ uncoded data chunks). From Figs. 4–6, we can observe that both approximate coding schemes can improve the total valuation over the best uncoded transmission scheme.

VI. Inapproximability Results for the Multiple Multicast Scenario

Thus far, we analyzed the multiple unicast scenario. This section analyzes the multiple multicast scenario. Given a graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{E})$ (with vertex set $\mathbf{\Lambda}$ and edge set $\mathbf{E}$) of the independent set problem, we construct an instance of our problem as follows. For each vertex $\lambda \in \mathbf{\Lambda}$ and edge $e \in \mathbf{E}$, we construct data chunks d_λ and d_e. The data chunk set $\mathbf{D}$ consists of d_λ and d_e for all $\lambda \in \mathbf{\Lambda}$ and $e \in \mathbf{E}$. For each edge $e = (x, y) \in \mathbf{E}$, we construct three clients $c_{e,1}, c_{e,2}, c_{e,3}$ such that

- $w_{e,1} = d_e$, $\hat{H}_{e,1} = \{d_x, d_y\}$, $\hat{v}_{e,1} = 1$,
- $w_{e,2} = d_x$, $\hat{H}_{e,2} = \{d_e\}$, $\hat{v}_{e,2} = \frac{1}{\deg(x)}$,
- $w_{e,3} = d_y$, $\hat{H}_{e,3} = \{d_e\}$, $\hat{v}_{e,3} = \frac{1}{\deg(y)}$,

where $\deg(\lambda)$ is the number of edges that are incident to vertex $\lambda \in \mathbf{\Lambda}$. With the reduction, our technical report in [19] shows that the combinatorial optimization problem in Eq. (4) equivalently becomes the independent set problem (for both the set $\mathbf{G}$ of sparse and instantly decodable coding matrices and the set $\mathbf{G}$ of sparse and general decodable coding matrices). Thus, the social welfare under our reduction is at most the number $|\mathbf{\Lambda}|$ of vertices (i.e., the approximation ratio is at most $|\mathbf{\Lambda}|$). Following the hardness result [25] of the independent set problem, the combinatorial optimization problem in Eq. (4) in the multiple multicast scenario is NP-hard and NP-hard to approximate within the factor of $|\mathbf{\Lambda}|^{1-\epsilon}$ for any constant $\epsilon > 0$.

VII. Concluding Remarks

This paper treated the index coding setting in the presence of selfish clients. We proposed a sufficient condition for truthful mechanisms (i.e., joint coding and payment schemes). Leveraging the proposed condition, we proposed truthful mechanisms, including the VCG-based mechanism and some polynomial-time mechanisms. While the VCG-based mechanism can maximize the social welfare, other polynomial-time mechanisms can either maximize the social welfare or approximate it with provable approximation ratios within some restricted sets of coding matrices. Following are some remarks on the assumptions made in the paper and possible extensions.

- This paper supposed that each client wants a single data chunk. Consider the case when a client wants a bundle of more than one data chunk. If the total valuation of different data chunks is the sum of their individual valuations (i.e., the client treats each data chunks in the bundle separately), then the client can be substituted by multiple clients that each wants a single data chunk in the bundle and has the same side information. In this context, the proposed mechanisms and their results hold for that case. The more general case when there are different valuations for different subsets of the bundle would be an interesting extension.
- This paper assumed scalar-linear coding schemes. We made that assumption for consistency. In fact, while those polynomial-time mechanisms (with sparse coding schemes) need that assumption, Theorem 1 and the proposed VCG-based mechanism do not.
- This paper developed polynomial-time mechanisms with sparse coding schemes. To develop a mechanism that can encode more than two data chunks is an interesting extension. A possible solution would be to construct an approximation algorithm that can identify a clique cover in a weighted dependency graph. Then, for each of the cliques, the server can broadcast a single XOR-coded data chunk combining all data chunks associated with the clique (like [5]).
- The inapproximability results for the multiple multicast scenario in the paper is based on the proposed VCG-based mechanism. It is a promising future work to develop a polynomial-time mechanism for the multiple multicast scenario beyond the scope of the VCG-based mechanism.
- The payment schemes proposed in the paper can motivate each client to submit its true information such that the server can maximize or approximate the true social welfare. However, the payment collected by the server do not always cover the transmission cost. For example, the proposed VCG-based mechanism satisfies the fourth condition in Theorem 1. Though that issue might be hard to avoid according to [4, Section 9.3.5.5], that is still an interesting future work.

Appendix A
Proof of Theorem 1

Consider a fixed valuation set $\hat{\mathbf{V}}_{-i}$ and side information set $\hat{\mathbf{H}}_{-i}$. Next, we consider two cases as follows.

1) *The coding scheme can construct a coding matrix G such that $\mathbf{1}_i(\hat{H}_i, G) = 0$ but $\mathbf{1}_i(H_i, G) = 1$ for some $\hat{v}_i$ and $\hat{H}_i$*: First, suppose that client c_i submits the true valuation v_i and the complete side information H_i. By the first and fourth conditions, client c_i can recover data chunk w_i if $v_i > 0$ and cannot if $v_i = 0$.

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

By the second and the fourth condition, client is charged 0 if it can recover w_i. Thus, client c_i has utility v_i. Second, suppose that client c_i submits a valuation $\hat{v}_i \neq v_i$ or side information $\hat{H}_i \subset H_i$. By the definition of the utility, client c_i has utility $(v_i - p_i) \cdot \mathbf{1}_i(H_i, G) \leq v_i$ or $(v_i - p_i) \cdot \mathbf{1}_i(\hat{H}_i, G) \leq v_i$. Thus, client c_i can maximize its utility by submitting both the true valuation and the complete side information.

2) *The coding scheme cannot construct a coding matrix G such that $\mathbf{1}_i(\hat{H}_i, G) = 0$ but $\mathbf{1}_i(H_i, G) = 1$ for all $\hat{v}_i$ and $\hat{H}_i$*: Note that, under the condition of this case, we have $\mathbf{1}_i(H_i, G) = 0$ if $\mathbf{1}_i(\hat{H}_i, G) = 0$. Moreover, we also have $\mathbf{1}_i(H_i, G) = 1$ if $\mathbf{1}_i(\hat{H}_i, G) = 1$, because of $\hat{H}_i \subseteq H_i$. In summary, this case has $\mathbf{1}_i(H_i, G) = \mathbf{1}_i(\hat{H}_i, G)$ for all $\hat{H}_i \subseteq H_i$ and all possible coding matrices G. Then, we claim that *for a fixed $\hat{H}_i$, client c_i can maximize its utility by submitting the true valuation v_i of data chunk w_i*. To prove that claim, we consider three cases as follows.
   a) $v_i > \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$: First, suppose that client c_i submits the true valuation v_i of data chunk w_i. By the first and the second conditions, client c_i can recover data chunk w_i (because $\mathbf{1}_i(H_i, G) = \mathbf{1}_i(\hat{H}_i, G) = 1$) and is charged threshold $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$. Thus, client c_i has utility $v_i - \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}}) > 0$. Second, suppose that client c_i submits valuation $\hat{v}_i \neq v_i$ of data chunk w_i. By the first and second conditions, if $\mathbf{1}_i(\hat{H}_i, G) = 1$ for some coding matrix G, then client c_i has utility $v_i - \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$; if $\mathbf{1}_i(\hat{H}_i, G) = 0$, then client c_i has zero utility. In summary, client c_i can maximize its utility by submitting the true valuation v_i of data chunk w_i.
   b) $v_i < \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$: First, suppose that client c_i submits the true valuation v_i of data chunk w_i. By the first condition, client c_i has zero utility. Second, suppose that client c_i submits valuation $\hat{v}_i \neq v_i$ of data chunk w_i. By the first and second conditions, if $\mathbf{1}_i(\hat{H}_i, G) = 1$ for some coding matrix G, then client c_i has utility $v_i - \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}}) < 0$; if $\mathbf{1}_i(\hat{H}_i, G) = 0$, then client c_i has zero utility. In summary, client c_i can maximize its utility by submitting the true valuation v_i of data chunk w_i.
   c) $v_i = \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$: First, suppose that client c_i submits the true valuation v_i of data chunk w_i. By the first and second conditions, client c_i has zero utility for either $\mathbf{1}_i(\hat{H}_i, G) = 1$ or $\mathbf{1}_i(\hat{H}_i, G) = 0$, for all possible coding matrices G. Second, suppose that client c_i submits valuation $\hat{v}_i \neq v_i$ of data chunk w_i. By the first and second conditions, if $\mathbf{1}_i(\hat{H}_i, G) = 1$ for some coding matrix G, then client c_i has utility $v_i - \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}}) = 0$; if $\mathbf{1}_i(\hat{H}_i, G) = 0$, then client c_i has zero utility. In summary, client c_i can maximize its utility by submitting the true valuation v_i of data chunk w_i.

According to the above claim, we can suppose that client c_i submits its true valuation v_i. It remains to show that, *client c_i can maximize its utility by submitting its complete side information H_i*. To prove that claim, we consider three cases as follows.

   a) $v_i > \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\})$: First, suppose that client c_i submits the complete side information H_i. Client c_i has utility $v_i - \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\}) > 0$. Second, suppose that client c_i submits incomplete side information $\hat{H}_i \subset H_i$. If $\mathbf{1}_i(\hat{H}_i, G) = 1$ for some coding matrix G, then client c_i has utility $v_i - \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}) \leq v_i - \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\})$ by the third condition; if $\mathbf{1}_i(\hat{H}_i, G) = 0$, then client c_i has zero utility. In summary, client c_i can maximize its utility by submitting the complete side information H_i.
   b) $v_i < \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\})$: First, suppose that client c_i submits the complete side information H_i. Client c_i has zero utility. Second, suppose that client c_i submits incomplete side information $\hat{H}_i \subset H_i$. By the third condition, we have $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}) \geq \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\}) > v_i$. Client c_i has zero utility. In summary, client c_i can maximize its utility by submitting the complete side information H_i.
   c) $v_i = \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\})$: First, suppose that client c_i submits the complete side information H_i. Client c_i has zero utility for either $\mathbf{1}_i(H_i, G) = 1$ or $\mathbf{1}_i(H_i, G) = 0$, for all possible coding matrices G. Second, suppose that client c_i submits incomplete side information $\hat{H}_i \subset H_i$. If $\mathbf{1}_i(\hat{H}_i, G) = 1$ for some coding matrix G, then client c_i has utility $v_i - \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}) \leq v_i - \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\}) = 0$ by the third condition; if $\mathbf{1}_i(\hat{H}_i, G) = 0$, then client c_i has zero utility. In summary, client c_i can maximize its utility by submitting the complete side information H_i.

APPENDIX B
PROOF OF THEOREM 2

Let $\hat{\mathbf{G}}_i \subseteq \mathbf{G}$ be the set of coding matrices such that $\mathbf{1}_i(\hat{H}_i, G) = 1$. Moreover, let $\hat{\mathbf{G}}_i^c \subseteq \mathbf{G}$ be the set of coding matrices such that $\mathbf{1}_i(\hat{H}_i, G) = 0$. Thus, we can express set $\mathbf{G}$ in Eq. (4) by $\mathbf{G} = \hat{\mathbf{G}}_i \cup \hat{\mathbf{G}}_i^c$. Moreover, let $\mathbf{G}_i \subseteq \mathbf{G}$ and $\mathbf{G}_i^c \subseteq \mathbf{G}$ be the sets of coding matrices G such that $\mathbf{1}_i(H_i, G) = 1$ and $\mathbf{1}_i(H_i, G) = 0$, respectively. In addition, let $\mathbf{G}^*$ be the set of coding matrices that are solutions to Eq. (4).

Then, we show that the VCG-based mechanism satisfies the four conditions in Theorem 1 as follows.

1) *The VCG-based coding scheme is a threshold-type coding scheme*: Consider a fixed valuation set $\hat{\mathbf{V}}_{-i}$ and a fixed side information set $\hat{\mathbf{H}}$. First, suppose that while client c_i submits a valuation $\hat{v}_i$ of data chunk w_i, the VCG-based coding scheme *can* construct a coding matrix G such that $\mathbf{1}_i(\hat{H}_i, G) = 1$. That is, the solution set $\mathbf{G}^*$ to Eq. (4) either belongs to set $\hat{\mathbf{G}}_i$ or[3] has a

[3]In the either-or, the first case corresponds to the case when $\hat{v}_i > \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$ and the second one corresponds to that when when $\hat{v}_i = \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$.

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

non-empty intersection with both sets $\hat{\mathbf{G}}_i$ and $\hat{\mathbf{G}}_i^c$ (with the arbitrary tie-breaking rule). Let $G^* \in \mathbf{G}^* \cap \hat{\mathbf{G}}_i$ be a solution to Eq. (4) such that $\mathbf{1}_i(\hat{H}_i, G^*) = 1$. Second, suppose that client c_i submits a valuation $\tilde{v}_i > \hat{v}_i$ of data chunk w_i. Because the number $\eta(G)$ of transmissions in Eq. (3) is independent of the valuation set for a fixed coding matrix G, we can express function $w(\{\tilde{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G)$ by

$$w(\{\tilde{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) = \begin{cases} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) + (\tilde{v}_i - \hat{v}_i) & \text{if } G \in \hat{\mathbf{G}}_i; \\ w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) & \text{if } G \in \hat{\mathbf{G}}_i^c, \end{cases} \tag{10}$$

Then, we can obtain

$$\begin{aligned} & w(\{\tilde{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G^*) \\ & \overset{(a)}{=} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G^*) + (\tilde{v}_i - \hat{v}_i) \\ & \overset{(b)}{\geq} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) + (\tilde{v}_i - \hat{v}_i) \text{ for all } G \in \hat{\mathbf{G}}_i^c \\ & \overset{(c)}{>} w(\{\tilde{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) \text{ for all } G \in \hat{\mathbf{G}}_i^c, \end{aligned}$$

where (a) is from Eq. (10) along with $G^* \in \hat{\mathbf{G}}_i$; (b) is because coding matrix G^* maximizes function $w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G)$; (c) is from Eq. (10) and $\tilde{v}_i > \hat{v}_i$. Thus, while client c_i submits valuation $\tilde{v}_i$ ($> \hat{v}_i$) of data chunk w_i, the set $\mathbf{G}^*$ of solutions to Eq. (4) belongs to $\hat{\mathbf{G}}_i$. That is, the VCG-based coding scheme constructs a coding matrix G such that $\mathbf{1}_i(\hat{H}_i, G) = 1$ for sure. Thus, the VCG-based coding scheme is a threshold-type coding scheme.

2) *The VCG-based payment scheme determines payment* $p_i = \bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$ *if* $\mathbf{1}_i(\hat{H}_i, G^*) = 1$: Consider a fixed valuation set $\hat{\mathbf{V}}_{-i}$ and a fixed side information set $\hat{\mathbf{H}}$. We claim that *threshold* $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}})$ *of the VCG-based coding scheme is*

$$\bar{v}_i(\hat{\mathbf{V}}_{-i}, \hat{\mathbf{H}}) = \max_{G \in \hat{\mathbf{G}}_i^c} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) - \max_{G \in \hat{\mathbf{G}}_i} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G). \tag{11}$$

Then, the first term of Eq. (5)

$$\max_{G \in \hat{\mathbf{G}}_i \cup \hat{\mathbf{G}}_i^c} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) = \max_{G \in \hat{\mathbf{G}}_i^c} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) \tag{12}$$

is the first term of Eq. (11) because of $\hat{v}_i = 0$. Moreover, the second term of Eq. (5)

$$\begin{aligned} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G^*) & \overset{(a)}{=} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G^*) - \hat{v}_i \\ & \overset{(b)}{=} \max_{G \in \hat{\mathbf{G}}_i} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) - \hat{v}_i \\ & \overset{(c)}{=} \max_{G \in \hat{\mathbf{G}}_i} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) \end{aligned}$$

is the second term of Eq. (11), where (a) is from Eqs. (10) along with $G^* \in \hat{\mathbf{G}}_i$ (because Eq. (5) calculates the price for the case when $\mathbf{1}_i(\hat{H}_i, G^*) = 1$); (b) is because G^* maximizes function $w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G^*)$ along with $G^* \in \hat{\mathbf{G}}_i$; (c) is from Eq. (10). Then, we complete the proof if the claim is true.

To establish that claim, we first consider the case when client c_i submits a valuation $\hat{v}_i > \max_{G \in \hat{\mathbf{G}}_i^c} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) - \max_{G \in \hat{\mathbf{G}}_i} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G)$ of data chunk w_i. Then, we can obtain

$$\begin{aligned} & \max_{G \in \hat{\mathbf{G}}_i} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) \\ & \overset{(a)}{=} \max_{G \in \hat{\mathbf{G}}_i} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) + \hat{v}_i \\ & \overset{(b)}{>} \max_{G \in \hat{\mathbf{G}}_i^c} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) \\ & \overset{(c)}{=} \max_{G \in \hat{\mathbf{G}}_i^c} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G), \end{aligned}$$

where (a) and (c) are from Eq. (10); (b) is by the assumption of $\hat{v}_i$. Thus, when $\hat{v}_i > \max_{G \in \hat{\mathbf{G}}_i^c} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) - \max_{G \in \hat{\mathbf{G}}_i} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G)$, the VCG-based coding scheme constructs a coding matrix G such that $\mathbf{1}_i(\hat{H}_i, G) = 1$. Second, similar to the above argument, when $\hat{v}_i < \max_{G \in \hat{\mathbf{G}}_i^c} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G) - \max_{G \in \hat{\mathbf{G}}_i} w(\{0, \hat{\mathbf{V}}_{-i}\}, \hat{\mathbf{H}}, G)$, the VCG-based coding scheme constructs a coding matrix G such that $\mathbf{1}_i(\hat{H}_i, G) = 0$. Fully considering both cases establishes Eq. (11).

3) $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\}) \leq \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\})$ *under the VCG-based coding scheme for all* $\hat{H}_i \subseteq H_i$: Consider a fixed valuation set $\hat{\mathbf{V}}_{-i}$ and a fixed side information set $\hat{\mathbf{H}}_{-i}$. Then, we can obtain

$$\begin{aligned} & \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\}) \\ & \overset{(a)}{=} \max_{G \in \mathbf{G}} w(\{0, \hat{\mathbf{V}}_{-i}\}, \{H_i, \hat{\mathbf{H}}_{-i}\}, G) - \max_{G \in \mathbf{G}_i} w(\{0, \hat{\mathbf{V}}_{-i}\}, \{H_i, \hat{\mathbf{H}}_{-i}\}, G) \\ & \overset{(b)}{=} \max_{G \in \mathbf{G}} w(\{0, \hat{\mathbf{V}}_{-i}\}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}, G) - \max_{G \in \mathbf{G}_i} w(\{0, \hat{\mathbf{V}}_{-i}\}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}, G) \\ & \overset{(c)}{\leq} \max_{G \in \mathbf{G}} w(\{0, \hat{\mathbf{V}}_{-i}\}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}, G) - \max_{G \in \hat{\mathbf{G}}_i} w(\{0, \hat{\mathbf{V}}_{-i}\}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}, G) \\ & \overset{(d)}{=} \bar{v}_i(\hat{\mathbf{V}}_{-i}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}), \end{aligned}$$

for all $\hat{H}_i \subseteq H_i$, where (a) and (d) are from Eqs. (11) and (12); (b) is because $\hat{v}_i = 0$; (c) is because $\hat{\mathbf{G}}_i \subseteq \mathbf{G}_i$.

4) *If the VCG-based coding scheme in Eq. (4) can construct a coding matrix* $G^* \in \mathbf{G}$ *such that* $\mathbf{1}_i(\hat{H}_i, G^*) = 0$ *but* $\mathbf{1}_i(H_i, G^*) = 1$ *for some* $\hat{v}_i$ *and* $\hat{H}_i$*, then* $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\}) = 0$: Consider a fixed valuation set $\hat{\mathbf{V}}_{-i}$ and a fixed side information set $\hat{\mathbf{H}}_{-i}$. Suppose that client c_i submits a valuation $\tilde{v}_i > 0$ and its complete side information H_i. Then, we can obtain

$$\begin{aligned} & w(\{\tilde{v}_i, \hat{\mathbf{V}}_{-i}\}, \{H_i, \hat{\mathbf{H}}_{-i}\}, G^*) \\ & \overset{(a)}{=} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}, G^*) + \tilde{v}_i \end{aligned}$$

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

$$\begin{aligned}
&\overset{(b)}{\geq} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}, G) + \tilde{v}_i \ \text{ for all } G \in \mathbf{G}_i^c \\
&\overset{(c)}{=} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \{H_i, \hat{\mathbf{H}}_{-i}\}, G) + \tilde{v}_i \ \text{ for all } G \in \mathbf{G}_i^c \\
&\overset{(d)}{>} w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \{H_i, \hat{\mathbf{H}}_{-i}\}, G) \ \text{ for all } G \in \mathbf{G}_i^c \\
&\overset{(e)}{=} w(\{\tilde{v}_i, \hat{\mathbf{V}}_{-i}\}, \{H_i, \hat{\mathbf{H}}_{-i}\}, G) \ \text{ for all } G \in \mathbf{G}_i^c,
\end{aligned}$$

where (a) is because $\mathbf{1}_i(\hat{H}_i, G^*) = 0$ but $\mathbf{1}_i(H_i, G^*) = 1$ by assumption; (b) is because G^* maximizes function $w(\{\hat{v}_i, \hat{\mathbf{V}}_{-i}\}, \{\hat{H}_i, \hat{\mathbf{H}}_{-i}\}, G)$; (c) is from Eq. (10) along with $\hat{\mathbf{G}}_i^c \subseteq \mathbf{G}_i^c$; (d) is because $\tilde{v}_i > 0$; (e) is from Eq. (10). Thus, when $\tilde{v}_i > 0$, the VCG-based coding scheme constructs a coding matrix G such that $\mathbf{1}_i(H_i, G) = 1$ for sure. Thus, the threshold is $\bar{v}_i(\hat{\mathbf{V}}_{-i}, \{H_i, \hat{\mathbf{H}}_{-i}\}) = 0$.

APPENDIX C
PROOF OF THEOREM 3

Let $\mathbf{C}^*$ be a set of those disjoint cycles that maximizes the total cycle weight in weighted dependency graph $\mathcal{G}(\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Gamma})$. Let graph $\mathcal{G}(\mathbf{\Lambda}_k, \mathbf{A}_k, \mathbf{\Gamma}_k)$ be the remaining graph at the beginning of iteration k. Let C_k be a cycle minimizing the cycle cost in graph $\mathcal{G}(\mathbf{\Lambda}_k, \mathbf{A}_k, \mathbf{\Gamma}_k)$. Let $\mathbf{C}_k^*$ be the set of those cycles that appear in set $\mathbf{C}^*$ and also in graph $\mathcal{G}(\mathbf{\Lambda}_k, \mathbf{A}_k, \mathbf{\Gamma}_k)$. By $C_k \cap \mathbf{C}_k^*$ we denote the set of those cycles in $\mathbf{C}_k^*$ that has a common vertex with cycle C_k.

Define $APX_k = \gamma(C_k)$ and $OPT_k = \sum_{C \in (C_k \cap \mathbf{C}_k^*)} \gamma(C)$. Because cycle C_k maximizes the cycle weight in iteration k, any cycle $C \in C_k \cap \mathbf{C}_k^*$ has cycle weight $\gamma(C)$ less than or equal to APX_k. Moreover, since there are $|C_k|$ vertices in cycle C_k, there are at most $|C_k|$ cycles in set $C_k \cap \mathbf{C}_k^*$ (because those cycles are disjoint). Thus, we can obtain $OPT_k \leq |C_k| \cdot APX_k$. Then, we can complete the proof (by Eq. (7)) as follow:

$$\begin{aligned}
&w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G^*) \\
&\quad = w([\hat{\mathbf{V}}]_1^+, \hat{\mathbf{H}}, G^*) + \sum_{\hat{v}_i \geq 1} (\hat{v}_i - 1) \\
&\quad \leq \sum_k OPT_k + \sum_{\hat{v}_i \geq 1} (\hat{v}_i - 1) \\
&\quad \leq \sum_k |C_k| \cdot APX_k + \sum_{\hat{v}_i \geq 1} (\hat{v}_i - 1) \\
&\quad \leq \max_k |C_k| \left(\sum_k APX_k + \sum_{\hat{v}_i \geq 1} (\hat{v}_i - 1) \right) \\
&\quad = \max_k |C_k| \left(w([\hat{\mathbf{V}}]_1^+, \hat{\mathbf{H}}, G_{\text{alg 2}}) + \sum_{\hat{v}_i \geq 1} (\hat{v}_i - 1) \right) \\
&\quad = \max_k |C_k| \cdot w(\hat{\mathbf{V}}, \hat{\mathbf{H}}, G_{\text{alg 2}}),
\end{aligned}$$

where $[\hat{\mathbf{V}}]_1^+ = ([\hat{v}_1]_1^+, \cdots, [\hat{v}_n]_1^+)$.

APPENDIX D
PROOF OF THEOREM 5

Follow the notation set in Appendix C; however, re-define C_k as a cycle maximizing $\frac{\gamma(C)}{\sqrt{|C|}}$ in graph $\mathcal{G}(\mathbf{\Lambda}_k, \mathbf{A}_k, \mathbf{\Gamma}_k)$, i.e., $\frac{\gamma(C_k)}{\sqrt{|C_k|}} \geq \frac{\gamma(C)}{\sqrt{|C|}}$ for all cycles C in graph $\mathcal{G}(\mathbf{\Lambda}_k, \mathbf{A}_k, \mathbf{\Gamma}_k)$. Then, we can bound OPT_k above by

$$\begin{aligned}
OPT_k &= \sum_{C \in C_k \cap \mathbf{C}_k^*} \gamma(C) \\
&\leq \frac{\gamma(C_k)}{\sqrt{|C_k|}} \sum_{C \in C_k \cap \mathbf{C}_k^*} \sqrt{|C|} \\
&\overset{(a)}{\leq} \frac{APT_k}{\sqrt{|C_k|}} \sqrt{|C_k \cap \mathbf{C}_k^*|} \sqrt{\sum_{C \in C_k \cap \mathbf{C}_k^*} |C|} \\
&\overset{(b)}{\leq} \frac{APT_k}{\sqrt{|C_k|}} \sqrt{|C_k|} \sqrt{n} \\
&= \sqrt{n} \cdot APT_k,
\end{aligned}$$

where (a) is from the Cauchy-Schwarz inequality; (b) is because all cycles $C \in C_k \cap \mathbf{C}_k^*$ are disjoint. Hence, $OPT_k \leq \sqrt{n} \cdot APX_k$, yielding the approximation ratio of $\sqrt{n}$.

REFERENCES

[1] Y.-P. Hsu, I.-H. Hou, and A. Sprintson, "The index coding problem: A game-theoretical perspective," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 977–981.

[2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 497–510, Jun. 2008.

[3] I.-H. Hou and P. R. Kumar, "Utility-optimal scheduling in time-varying wireless networks with delay constraints," in *Proc. 11th ACM Int. Symp. Mobile Netw. Comput.*, 2010, pp. 31–40.

[4] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[5] Y. Birk and T. Kol, "Informed-source coding-on-demand (ISCOD) over broadcast channels," in *Proc. 17th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Mar. 1998, pp. 1257–1264.

[6] F. Arbabjolfaei and Y.-H. Kim, "Generalized lexicographic products and the index coding capacity," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1520–1529, Mar. 2020.

[7] C. Arunachala, V. Aggarwal, and B. S. Rajan, "Optimal linear broadcast rates of some two-sender unicast index coding problems," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 3965–3977, Jun. 2019.

[8] A. Agarwal, L. Flodin, and A. Mazumdar, "Linear programming approximations for index coding," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5547–5564, Apr. 2019.

[9] M. Li, L. Ong, and S. J. Johnson, "Multi-sender index coding for collaborative broadcasting: A rank-minimization approach," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1452–1466, Feb. 2019.

[10] S. Brahma and C. Fragouli, "Pliable index coding," *IEEE Trans. Inf. Theory*, vol. 61, no. 11, pp. 6192–6203, Nov. 2015.

[11] M. M. Mojahedian, M. R. Aref, and A. Gohari, "Perfectly secure index coding," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7382–7395, Nov. 2017.

[12] F. Arbabjolfaei and Y.-H. Kim, *Fundamentals Index Coding*. New York, NY, USA: Now, 2018.

[13] S. Bhadra, S. Shakkottai, and P. Gupta, "Min-cost selfish multicast with network coding," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 5077–5087, Nov. 2006.

[14] H. Mohsenian-Rad, J. Huang, V. W. S. Wong, S. Jaggi, and R. Schober, "Inter-session network coding with strategic users: A game-theoretic analysis of the butterfly network," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1473–1484, Apr. 2013.

[15] C. Wu, M. Gerla, and M. van der Schaar, "Social norm incentives for network coding in manets," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1761–1774, Jun. 2017.

[16] T. Chen and S. Zhong, "INPAC: An enforceable incentive scheme for wireless networks using network coding," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[17] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[18] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.

[19] Y.-P. Hsu, I.-H. Hou, and A. Sprintson, "Joint index coding and incentive design for selfish clients," 2020, *arXiv:2005.08545*. [Online]. Available: http://arxiv.org/abs/2005.08545
[20] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg, "Finding sparse solutions for the index coding problem," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2011, pp. 1–5.
[21] Z. Galil, "Efficient algorithms for finding maximum matching in graphs," *ACM Comput. Surveys*, vol. 18, no. 1, pp. 23–38, Mar. 1986.
[22] M. Krivelevich, Z. Nutov, M. R. Salavatipour, J. Verstraete, and R. Yuster, "Approximation algorithms and hardness results for cycle packing problems," *ACM Trans. Algorithms*, vol. 3, no. 4, p. 48, Nov. 2007.
[23] A. Schrijver, *Combinatorial Optimization: Polyhedra Efficiency*. Cham, Switzerland: Springer, 2003.
[24] T. H. Cormen, C. E. Leiserson, R. L. Rivestand, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2014.
[25] J. Hastad, "Clique is hard to approximate within $n^1 - \epsilon$," *Acta Math.*, vol. 182, pp. 105–142, Dec. 1999.

Yu-Pin Hsu (Member, IEEE) received the B.S. and M.S. degrees from National Chiao Tung University, Taiwan, in 2005 and 2007 respectively, and the Ph.D. degree from Texas A&M University, College Station, TX, USA, in 2014. He was a Post-Doctoral Fellow with the Singapore University of Technology and Design, Singapore, in 2014, and with the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2015. In 2016, he joined National Taipei University, Taiwan, where he is currently an Assistant Professor. His research interests include center around communication networks with focus on algorithmic and control-theoretic aspects.

I-Hong Hou (Senior Member, IEEE) received the Ph.D. degree from the Computer Science Department, University of Illinois at Urbana–Champaign. He is currently an Associate Professor with the ECE Department, Texas A&M University. His research interests include wireless networks, real-time systems, and cloud computing. His work has received the Best Paper Award from ACM MobiHoc 2017 and ACM MobiHoc 2020, and the Best Student Paper Award from WiOpt 2017. He has also received the C.W. Gear Outstanding Graduate Student Award from the University of Illinois at Urbana–Champaign and the Silver Prize in the Asian Pacific Mathematics Olympiad.

Alex Sprintson (Senior Member, IEEE) is currently a Faculty Member with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA, where he also conducts research on wireless networks, distributed storage, and software-defined networking. He received the Wolf Award for Distinguished Ph.D. students, the Viterbi Post-Doctoral Fellowship, the TAMU College of Engineering Outstanding Contribution Award, and the NSF CAREER Award. From 2013 and 2019, he has served as an Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He has been a member of the Technical Program Committee of the IEEE Infocom since 2006. He is also serving as a Rotating Program Director for the United States National Science foundation networking technology and systems program.

Authorized licensed use limited to: Texas A M University. Downloaded on September 23,2021 at 19:38:22 UTC from IEEE Xplore. Restrictions apply.