Efficient Algorithms for the Data Exchange Problem

Nebojsa Milosavljevic, Sameer Pawar, Salim El Rouayheb, *Member, IEEE*,
Michael Gastpar, and Kannan Ramchandran, *Fellow, IEEE*

***Abstract*— In this paper, we study the data exchange problem, where a set of users is interested in gaining access to a common file, but where each has only partial knowledge about it as side-information. Assuming that the file is broken into packets, the side-information considered is in the form of linear combinations of the file packets. Given that the collective information of all the users is sufficient to allow recovery of the entire file, the goal is for each user to gain access to the file, while minimizing some communication cost. We assume that the users can communicate over a noiseless broadcast channel, and that the communication cost is a sum of each user's cost function over the number of bits it transmits. For instance, the communication cost could simply be the total number of bits that needs to be transmitted. In the most general case studied in this paper, each user can have any arbitrary convex cost function. We provide deterministic, polynomial-time algorithms (in the number of users and packets), which find an optimal communication scheme that minimizes the communication cost. To further lower the complexity, we also propose a simple randomized algorithm inspired by our deterministic algorithm, which is based on a random linear network-coding scheme.**

***Index Terms*— Network coding, packet radio networks, combinatorial mathematics, optimization.**

I. Introduction

In recent years cellular systems have witnessed significant improvements in terms of data rates, and are nearly approaching the theoretical limits in terms of the physical layer spectral efficiency. At the same time, the rapid growth in the popularity of data-enabled mobile devices, such as smart phones and tablets, and the resulting explosion in demand for more throughput are challenging our abilities to deliver data, even with the current highly efficient cellular systems. One of the major bottlenecks in scaling the throughput with the increasing number of mobile devices is the "last mile" wireless link between the base station and the mobile devices - a resource that is shared among many users served within the cell. This motivates the study of paradigms where cell phone devices can cooperate among themselves to get the desired data in a peer-to-peer fashion without solely relying on the base station.

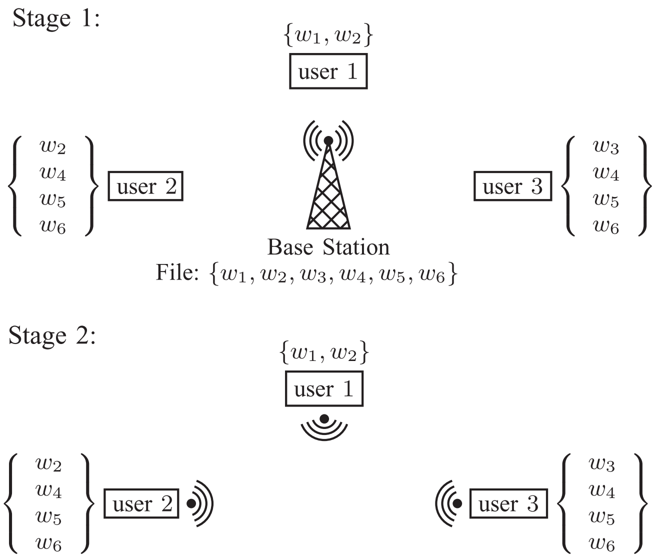

Fig. 1. An example of the data exchange problem. A base station has a file formed of six packets $w_1, \ldots, w_6 \in \mathbb{F}_q$ and wants to deliver it to three users over an unreliable wireless channel. The base station stops transmitting once all users collectively have all the packets, even if individually they have only subsets of the packets (Stage 1). Users can then cooperate among themselves to recover their missing packets by broadcasting over a noiseless public channel (Stage 2). It can be shown that the minimum number of symbols in $\mathbb{F}_q$ needed for the file recovery at all users is 5. A communication scheme that achieves this minimum is: user 1 transmits w_1, user 2 transmits $w_2 + w_4$, while user 3 transmits w_3, w_5, w_6. Now, if the goal is to allocate these 5 transmissions to the users as uniformly as possible, user 1 transmits w_1, user 2 transmits $w_2 + w_4$, w_5, and user 3 transmits w_3, w_6.

An example of such a setting is shown in Figure 1, where a base station wants to deliver the same file to multiple geographically-close users over an unreliable wireless downlink. In the example of Figure 1, we assume that the file consists of six equally sized packets w_1, w_2, w_3, w_4, w_5 and w_6 belonging to some finite field $\mathbb{F}_q$. Suppose that after a few initial transmission attempts by the base station, the three users individually receive only parts of the file (see Figure 1), but collectively have the entire file. Now, if all users are in close vicinity and can communicate with each other, then, it is much more desirable and efficient, in terms of resource usage,

Manuscript received December 1, 2013; revised October 18, 2015; accepted January 6, 2016. Date of publication February 3, 2016; date of current version March 16, 2016. This work was supported in part by the Defense Threat Reduction Agency under Grant HDTRA1-09-1-0032, in part by the National Science Foundation under Grant CCF-0964018 and Grant CCF-0830788, and in part by the Air Force Office of Scientific Research under Grant FA9550-09-1-0120 and Grant FA9550-10-1-0567.

N. Milosavljevic is with the Cloud and Networking Group, Cisco-Meraki, CA 94158 USA (e-mail: nebojsa@eecs.berkeley.edu).

S. Pawar is with the Wireless Division, Intel Corporation, Santa Clara, CA 95054 USA (e-mail: spawar@eecs.berkeley.edu).

S. El Rouayheb is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: salim@iit.edu).

M. Gastpar is with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720 USA, and also with the School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (e-mail: michael.gastpar@epfl.ch).

K. Ramchandran is with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: kannanr@eecs.berkeley.edu).

Communicated by S. Jaggi, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2016.2523980

to reconcile the file among users by letting all of them "talk" to each other without involving the base station. The cooperation among the users has the following advantages:

- Local communication among users has a smaller footprint in terms of interference, thus allowing one to use the shared resources (code, time or frequency) freely without penalizing the base station's resources, *i.e.*, higher resource reuse factor.
- Transmissions within the close group of users is much more reliable than from the base station to any terminal due to geographical proximity of terminals.
- This cooperation allows file recovery even when the connection to the base station is either unavailable after the initial phase of transmission, or it is too weak to meet the delay requirement.

Let us consider the example in Figure 1, and let user 1, user 2 and user 3 transmit R_1, R_2 and R_3 symbols in $\mathbb{F}_q$, respectively. It can be shown that the minimum total number of symbols in $\mathbb{F}_q$ needed to recover the file is 5. One possible communication scheme that achieves it is: user 1 transmits w_1, user 2 transmits $w_2 + w_4$, while user 3 transmits w_3, w_5, w_6. Note that the load of the communication of the system is unevenly distributed among the users, *i.e.*, user 3 transmits 3 out of 5 symbols in $\mathbb{F}_q$. The next question we ask here is out of all communication schemes that deliver the entire file to the users in the minimum number of transmissions, which one distributes the load of communication to the users as fairly as possible. For instance, for the same minimum number of transmissions, we can have the following scheme: user 1 transmits w_1, user 2 transmits $w_2 + w_4$, w_5, and user 3 transmits w_3, w_6. Intuitively, this scheme is more fair[1] than the previous one since it spreads the transmissions more uniformly among the users. And, it can be shown that such scheme minimizes a convex fairness cost.

In the example from Figure 1, we considered only a simple form of side-information, where different users observe uncoded "raw" packets of the original file. Content distribution networks [17]–[19] are increasingly using codes, such as linear network codes or Fountain codes [20], to improve the system efficiency. In such scenarios, the side-information representing the partial knowledge gained by the users would be coded and in the form of linear combinations of the original file packets, rather than the raw packets themselves. We refer to this model of side-information as a *linear packet model*.

Contributions

In this paper, we study the data exchange problem under the linear packet model and the separable convex communication cost. Such cost captures all the communication objectives discussed earlier: 1. Minimization of the (weighted) sum of bits users need to exchange, 2. Fairness. Our contributions can be summarized as follows:

1) We propose a deterministic polynomial time algorithm for finding an optimal communication scheme w.r.t. the communication cost. An important step of this algorithm is to iteratively determine how much should each user transmit in an optimal scheme. We provide two methods to solve this problem. The first one is based on minimizing a submodular function, in which case the total complexity of the algorithm, in case of linear objective, is $\mathcal{O}((m^6N^3 + m^7)\log N)$, where m is the total number of users, and N is the number of packets in the file. In case of the fairness objective, the complexity is $\mathcal{O}((m^6N^3 + m^7)N\log N)$. The second technique is based on subgradient methods, in which case the total complexity of the algorithm can be bounded by $\mathcal{O}((m^4 \log m + N^3m^4)N^2 \log N)$ for linear cost, and $\mathcal{O}((m^4 \log m + N^3m^4)N^3 \log N)$ for the fairness cost, given that we use constant step size in the subgradient algorithm.
2) We devise a randomized algorithm inspired by the deterministic scheme that reduces complexity to $\mathcal{O}(mN^4 \log N)$. The randomized algorithm is based on a random linear network coding scheme, and it achieves the optimal number of transmissions with high probability. To be more precise, the probability of not achieving the optimum is inversely proportional to the underlying field size $|\mathbb{F}_q|$. Our randomized algorithm can be regarded as a generalization of the algorithm proposed in [6], where the authors considered linear communication cost.
3) For the data exchange problem with additional capacity constraints on each user, we provide both deterministic and randomized algorithm of the same complexity as in 1. and 2.

The challenging part of the deterministic algorithm is that the underlying optimization problem has exponential number of constraints coming from the cut-set bound region. By using combinatorial optimization techniques such as *Dilworth truncation* and *Edmonds' algorithm*, we devise an efficient, polynomial time solution.

Literature Overview

The problem of reconciling a file among multiple wireless users having parts of it while minimizing the cost in terms of the total number of bits exchanged is known in the literature as the *data exchange problem* and was introduced by El Rouayheb *et al.* in [4]. A closely related problem was also studied by Csiszár and Narayan in [12] and Chan in [13] and [14], where all users want to agree on a secret key in the presence of an eavesdropper who observes the entire communication. In [12] each user observes realizations of a discrete memoryless multiple source process with an arbitrary joint distribution, whereas in [13] and [14], the author considered a finite linear source model which is essentially an asymptotic version of the file packet model considered in this paper. It was shown that the maximum secret key can be achieved by applying a communication scheme that renders the file to all users in minimum number of bits transmitted over the broadcast network.

A randomized algorithm for the data exchange problem was proposed by Sprintson *et al.* in [5]. Tajbakhsh *et al.* [11]

[1] To be precise, the fairness cost that we consider belongs to the broader class of separable convex costs that is studied in this work.

formulated this problem as a linear program (LP) and gave an approximate solution.

The linear cost data exchange problem was studied by Ozgul and Sprintson [6], where the authors proposed a randomized algorithm. A deterministic polynomial time algorithm was proposed by Courtade and Wesel in [9] concurrently to the authors' work [2]. The minimum linear communication cost problem was also studied in the network coding literature. Lun *et al.* [21] proposed a polynomial time algorithm for the single source multicast problem over a directed acyclic graph.

The general broadcast version of the data exchange problem where users can broadcast messages to their immediate neighbors was studied by Courtade *et al.* in [7], [8], and [10] and by Gonen and Langberg in [22]. In [7] and [8], it is shown that the problem can be solved in the general case by re-casting it as a single-source network coding problem. For the special case of broadcast networks considered in this paper, it is shown that a polynomial solution can be constructed if the file packets are allowed to be split into smaller chunks. For the most general case of broadcast networks, an approximate solution was provided in [22]. In [23], Lucani *et al.* considered the problem of data exchange when the channel between different users can have erasures.

The rest of the paper is organized as follows. In Section II, we describe the model and formulate the optimization problem. In Section III, we provide a polynomial time algorithm that solves for how many symbols in $\mathbb{F}_q$ should each user transmit. We start Section III by analyzing a linear cost function, and then we extend our solution to any separable convex cost. In Section IV, we propose a polynomial time code construction. In Section V, we describe an algorithm based on random linear network coding approach, that achieves the optimal communication cost. In Section VI, we present a polynomial time solution to the problem where each user additionally has capacity constraints, *i.e.*, user i is not allowed to transmit more than c_i symbols in $\mathbb{F}_q$. We conclude our work in Section VII.

II. System Model and Preliminaries

In this paper, we consider a setup with m users that are interested in gaining access to a file. The file is broken into N linearly independent packets $w_1, \ldots, w_N$ each belonging to a field $\mathbb{F}_q$, where q is a power of some prime number. Each user $i \in \mathcal{M} \triangleq \{1, 2, \ldots, m\}$ observes some collection of the linear combinations of the file packets as shown below.

$$\mathbf{x}_i = \mathbf{A}_i \mathbf{w}, \quad i \in \mathcal{M}, \tag{1}$$

where $\mathbf{A}_i \in \mathbb{F}_q^{\ell_i \times N}$ is a given matrix, and $\mathbf{w} = \begin{bmatrix} w_1 \; w_2 \ldots w_N \end{bmatrix}^T$ is a vector of the file packets. In the further text, we refer to (1) as a linear packet model. We assume that matrices $\mathbf{A}_i$, $\forall i \in \mathcal{M}$, are known to each user. That way, they can locally compute the optimal communication strategy and apply it. Alternatively, there can be a central authority that the users report to their side information and can compute the optimal communication strategy and distributes it to everybody.

Let us denote by $\mathbf{v}_i$, a transmission of user $i \in \mathcal{M}$. In order for each user to recover the file, interaction among them is not needed. This follows from the fact that any interactive linear solution can be transformed into an non-interactive one due to linearity of users' observations and noiseless broadcast nature of communication as noticed in [14]. Hence, without loss of generality, we can assume that $\mathbf{v}_i$ is a function of user i's initial observation. We define

$$R_i \triangleq |\mathbf{v}_i|_q \tag{2}$$

to be the size of user i's transmission represented in number of symbols in $\mathbb{F}_q$. To decode the file, user i collects transmissions of all the users and creates a decoding function

$$\psi_i : \mathbb{F}_q^{\ell_i} \times \mathbb{F}_q^{R_1} \times \cdots \times \mathbb{F}_q^{R_m} \rightarrow \mathbb{F}_q^N, \tag{3}$$

that reconstructs the file, *i.e.*,

$$\psi_i(\mathbf{x}_i, \mathbf{v}_1, \ldots, \mathbf{v}_m) = \mathbf{w}. \tag{4}$$

Definition 1: A rate vector $\mathbf{R} = (R_1, R_2, \ldots, R_m)$ is an *achievable data exchange (DE) rate vector* if there exists a communication scheme with transmitted messages $(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m)$ that satisfies (4) for all $i = 1, \ldots, m$.

Remark 1: Using cut-set bounds, it follows that all the achievable *DE*-rate vectors necessarily belong to the following region

$$\begin{aligned}\mathcal{R} \triangleq \{&\mathbf{R} \in \mathbb{R}^m : R(\mathcal{S}) \\ &\geq N - \operatorname{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}),\ \forall \mathcal{S} \subset \mathcal{M}\},\end{aligned} \tag{5}$$

where

$$R(\mathcal{S}) \triangleq \sum_{i \in \mathcal{S}} R_i,$$

and where for any $\mathcal{S} = \{i_1, i_2, ..., i_{|\mathcal{S}|}\} \subseteq \mathcal{M}$ we define $\mathbf{A}_\mathcal{S}$ as follows:

$$\mathbf{A}_\mathcal{S} \triangleq \begin{bmatrix} \mathbf{A}_{i_1} \\ \mathbf{A}_{i_2} \\ \vdots \\ \mathbf{A}_{i_{|\mathcal{S}|}} \end{bmatrix}. \tag{6}$$

Theorem 1: For a sufficiently large field size $|\mathbb{F}_q|$, any integer DE*-rate vector $\mathbf{R} \in \mathbb{Z}^m$ that belongs to the cut-set region $\mathcal{R}$, can be achieved via linear network coding,* i.e.*, it is sufficient for each user $i \in \mathcal{M}$ to transmit R_i properly chosen linear combinations of the data packets it observes.*

The proof of Theorem 1 can be found in Appendix A. In [14] and the followup work [15], the author considered an asymptotic linear packet model that allows packet splitting. Theorem 1 generalizes [14, Th. 1] to the non-asymptotic case. In [7], the authors considered the data exchange problem with an arbitrary broadcast communication scheme that can be represented by an undirected network graph, where each user observes some set of raw file packets. For the fully connected network graph, Theorem 1 generalizes [7, Th. 1] to the linear packet model. In [7], the authors converted the problem to a multicast network, which has a linear network solution. This method cannot be directly applied to the general linear packet model (1), which we briefly argue in Section IV.

From the network code construction argument [16], it follows that any field size $|\mathbb{F}_q|$ larger than the number of users is sufficient to guarantee the existence of such solution. In general, finding the minimum field size can be a hard problem. More details on code construction for the data exchange problem can be found in Section IV.

In order for each user to recover the entire file, it is necessary to receive a sufficient number of linear combinations of the other users' observations. Hence, $\mathbf{v}_i$, $i \in \mathcal{M}$, defined above is a vector of R_i symbols in $\mathbb{F}_q$. Therefore, $\mathbf{v}_i$ can be written as follows

$$\mathbf{v}_i = \mathbf{B}_i \mathbf{x}_i = \mathbf{B}_i \mathbf{A}_i \mathbf{w} = \mathbf{U}_i \mathbf{w}, \tag{7}$$

where $\mathbf{B}_i$ is an $R_i \times \ell_i$ transmission matrix with elements belonging to $\mathbb{F}_q$. In order for each user to recover the file, the transmission matrices $\mathbf{B}_i$, $i \in \mathcal{M}$ should satisfy,

$$\text{rank}\left(\begin{bmatrix} \mathbf{A}_i \\ \mathbf{U} \end{bmatrix}\right) = N, \quad \forall i \in \mathcal{M}, \tag{8}$$

where $\mathbf{U} \triangleq \bigcup_{i=1}^{m} \mathbf{U}_i$. Hence, the decoding function ψ_i of user $i \in \mathcal{M}$ involves inverting the matrix given in (8) in order to obtain $\mathbf{w}$.

In this work, we design a polynomial complexity scheme that achieves the file exchange among all the users while simultaneously minimizing a separable convex cost function $\sum_{i=1}^{m} \varphi_i(R_i)$, where φ_i, $i \in \mathcal{M}$ is a non-decreasing convex function. Such assumption on monotonicity of function φ_i is consistent with the nature of the problem at hand; sending more bits is always more expensive than sending fewer. From (5) and the above mentioned cost function, the problem considered in this work can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{R} \in \mathbb{Z}^m} \sum_{i=1}^{m} \varphi_i(R_i), \qquad\qquad (9)\\ &\text{s.t.} \quad R(\mathcal{S}) \geq N - \text{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}), \quad \forall \mathcal{S} \subset \mathcal{M}. \end{aligned}$$

Optimization problem (9) is a convex integer problem with $2^m - 2$ constraints. It was shown in [13] that only n of these constraints are active but the challenge is how to determine which of them are. Solving the optimization problem (9) answers the question of how many symbols in $\mathbb{F}_q$ each user has to transmit in an optimal scheme. In this paper we provide a polynomial time algorithm that solves problem (9). Once we obtain an optimal rate allocation, the actual transmissions of each user can be solved in polynomial time by using the algebraic network coding framework [24], [25]. This is explained in Section IV.

III. Deterministic Algorithm

Our goal is to solve problem (9) efficiently. To do so, we will split it into two subproblems:

1) Given a total budget constraint β, *i.e.*, $R(\mathcal{M}) = R_1 + R_2 + \cdots R_m = \beta$, determine whether β is feasible or not. If β is feasible, find the feasible rate split among the users that will achieve the total budget β and minimize the cost $\sum_{i=1}^{m} \varphi_i(R_i)$.
2) Find β that minimizes the objective function.

The bottleneck here is how to solve Problem 1 efficiently. The optimal value of β can then be found using binary search (see Algorithm 3) since the objective function is w.r.t. β. First, let us identify these two problems by rewriting problem (9) as follows

$$\min_{\beta \in \mathbb{Z}_+} h(\beta), \tag{10}$$

where

$$\begin{aligned} h(\beta) \triangleq \min_{\mathbf{R} \in \mathbb{Z}^m} &\sum_{i=1}^{m} \varphi_i(R_i),\\ \text{s.t.} \quad &R(\mathcal{M}) = \beta, \quad R(\mathcal{S}) \geq N - \text{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}),\\ &\forall \mathcal{S} \subset \mathcal{M}. \qquad (11) \end{aligned}$$

Note that the optimizations (10) and (11) are associated with Problem 2 and Problem 1 defined above, respectively. Next we will explain our approach to solving these two problems.

A. Optimization With a Given Sum-Rate Budget β

Now, let us focus on the set of constraints of optimization problem (11). By substituting $\mathcal{S}$ with $\mathcal{M} \setminus \mathcal{S}$, we obtain

$$\begin{aligned} R(\mathcal{M}) &= \beta,\\ R(\mathcal{M} \setminus \mathcal{S}) &= R(\mathcal{M}) - R(\mathcal{S}) = \beta - R(\mathcal{S})\\ &\geq N - \text{rank}(\mathbf{A}_{\mathcal{S}}), \quad \forall \mathcal{S} \subset \mathcal{M},\ \mathcal{S} \neq \emptyset. \qquad (12) \end{aligned}$$

Therefore, optimization problem (11) can be equivalently represented as follows

$$\begin{aligned} h(\beta) = \min_{\mathbf{R} \in \mathbb{Z}^m} &\sum_{i=1}^{m} \varphi_i(R_i),\\ \text{s.t.} \quad &R(\mathcal{M}) = \beta,\\ R(\mathcal{S}) \leq \beta - N + \text{rank}(\mathbf{A}_{\mathcal{S}}), &\quad \forall \mathcal{S} \subset \mathcal{M},\ \mathcal{S} \neq \emptyset. \qquad (13) \end{aligned}$$

Before we go any further, let us introduce some concepts from combinatorial optimization theory.

Definition 2 (Polyhedron): Let f_β be a set function defined over set $\mathcal{M} = \{1, 2, \ldots, m\}$, *i.e.*, $f_\beta : 2^{\mathcal{M}} \to \mathbb{Z}$, where $2^{\mathcal{M}}$ is the power set of $\mathcal{M}$. Then the *polyhedron* $P(f_\beta)$ and the *base polyhedron* $B(f_\beta)$ of f_β are defined as follows

$$P(f_\beta) \triangleq \{\mathbf{R} \in \mathbb{Z}^m | R(\mathcal{S}) \leq f_\beta(\mathcal{S}),\ \forall \mathcal{S} \subseteq \mathcal{M}\}, \tag{14}$$

$$B(f_\beta) \triangleq \{\mathbf{R} \in P(f_\beta) | R(\mathcal{M}) = f_\beta(\mathcal{M})\}. \tag{15}$$

Note that the set of constraints of problem (13), for any fixed $\beta \in \mathbb{Z}_+$, constitutes the base polyhedron $B(f_\beta)$ of the set function

$$f_\beta(\mathcal{S}) = \begin{cases} \beta - N + \text{rank}(\mathbf{A}_{\mathcal{S}}) & \text{if } \mathcal{S} \subset \mathcal{M},\ \mathcal{S} \neq \emptyset\\ \beta & \text{if } \mathcal{S} = \mathcal{M},\\ 0 & \text{if } \mathcal{S} = \emptyset. \end{cases} \tag{16}$$

Example 1: Let us consider the source model from Figure 1, where the three users observe the following parts of the file $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix}^T$:

$$\begin{aligned} \mathbf{x}_1 &= \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T,\\ \mathbf{x}_2 &= \begin{bmatrix} w_2 & w_4 & w_5 & w_6 \end{bmatrix}^T,\\ \mathbf{x}_3 &= \begin{bmatrix} w_3 & w_4 & w_5 & w_6 \end{bmatrix}^T. \end{aligned} \tag{17}$$

For $\beta = 4$, the base polyhedron $P(f_4)$ is defined by the following set of inequalities:

$$\begin{aligned} &R_1 \le f_4(\{1\}) = 0,\ R_2 \le f_4(\{2\}) = 2,\\ &R_3 \le f_4(\{3\}) = 2,\\ &R_1 + R_2 \le f_4(\{1,2\}) = 3,\ R_1 + R_3 \le f_4(\{1,3\}) = 4,\\ &R_2 + R_3 \le f_4(\{2,3\}) = 3,\\ &R_1 + R_2 + R_3 \le f_4(\{1,2,3\}) = 4. \end{aligned} \tag{18}$$

It can be verified that no rate vector $(R_1, R_2, R_3) \in P(f_4)$ exists such that $R_1 + R_2 + R_3 = 4$. Therefore, $B(f_4) = \emptyset$. On the other hand, for $\beta = 5$, the polyhedron $P(f_5)$ is defined as follows

$$\begin{aligned} &R_1 \le f_5(\{1\}) = 1,\ R_2 \le f_5(\{2\}) = 3,\\ &R_3 \le f_5(\{3\}) = 3,\\ &R_1 + R_2 \le f_5(\{1,2\}) = 4,\ R_1 + R_3 \le f_5(\{1,3\}) = 5,\\ &R_2 + R_3 \le f_5(\{2,3\}) = 4,\\ &R_1 + R_2 + R_3 \le f_5(\{1,2,3\}) = 5. \end{aligned} \tag{19}$$

It can be easily verified that the rate vector $R_1 = 1$, $R_2 = 3$, $R_3 = 1$ belongs to the polyhedron $P(f_5)$. Therefore, $B(f_5) \neq \emptyset$.

Summarizing the discussion so far, the optimization problem (13) is equivalent to

$$\min_{\mathbf{R} \in \mathbb{Z}^m} \sum_{i=1}^{m} \varphi_i(R_i), \quad \text{s.t. } \mathbf{R} \in B(f_\beta), \tag{20}$$

where f_β is defined in (16). For now, let us assume that parameter β is chosen such that the optimization problem (20) is feasible, *i.e.*, $B(f_\beta) \neq \emptyset$. We will explain later how the condition $B(f_\beta) \neq \emptyset$ can be efficiently verified.

The main idea behind solving the optimization problem in (20) efficiently, is to utilize the combinatorial properties of the set function f_β.

Definition 3: We say that a set function $f : 2^{\mathcal{M}} \to \mathbb{Z}$ is *intersecting submodular* if

$$\begin{aligned} f(\mathcal{S}) + f(\mathcal{T}) \ge f(\mathcal{S} \cup \mathcal{T}) + f(\mathcal{S} \cap \mathcal{T}),\\ \forall \mathcal{S}, \mathcal{T} \subseteq \mathcal{M} \text{ s.t. } \mathcal{S} \cap \mathcal{T} \neq \emptyset. \end{aligned} \tag{21}$$

When the inequality conditions in (21) are satisfied for all sets $\mathcal{S}, \mathcal{T} \subseteq \mathcal{M}$, the function f is *fully submodular*.

Lemma 1: The function f_β is intersecting submodular *for any β. When $\beta \ge N$, f_β is fully submodular.*

The proof of Lemma 1 is provided in Appendix C.

Theorem 2 (Dilworth Truncation [27]): For every intersecting submodular function f_β there exists a fully submodular function g_β such that both functions have the same polyhedron, i.e., $P(g_\beta) = P(f_\beta)$, and g_β can be expressed as

$$g_\beta(\mathcal{S}) = \min_{\mathcal{P} \in \boldsymbol{\mathcal{P}}(\mathcal{S})} \sum_{\mathcal{V} \in \mathcal{P}} f_\beta(\mathcal{V}), \tag{22}$$

where $\boldsymbol{\mathcal{P}}(\mathcal{S})$ is the set of all partitions of the set $\mathcal{S}$. The function g_β is called the Dilworth truncation *of f_β.*

The base polyhedron of any fully submodular function always exists, *i.e.*, there exists a rate vector $\mathbf{R}$ such that $R(\mathcal{M}) = g_\beta(\mathcal{M})$. Since, $P(g_\beta) = P(f_\beta)$, it follows that $B(g_\beta) = B(f_\beta)$ whenever $g_\beta(\mathcal{M}) = f_\beta(\mathcal{M}) = \beta$, *i.e.*, when $B(f_\beta) \neq \emptyset$ which implies feasibility of the optimization problem (20).

Continuing with Example 1, the Dilworth truncation of the set function f_4 is given by

$$\begin{aligned} &g_4(\{1\}) = 0,\ g_4(\{2\}) = 2,\ g_4(\{3\}) = 2,\\ &g_4(\{1,2\}) = 2,\ g_4(\{1,3\}) = 2,\ g_4(\{2,3\}) = 3,\\ &g_4(\{1,2,3\}) = 3. \end{aligned} \tag{23}$$

Note that $f_4(\{1,2,3\}) \neq g_4(\{1,2,3\})$, and hence, $\beta = 4$ is not a feasible sum-rate for the problem (20). On the other hand, for $\beta = 5$, Dilworth truncation of a set function f_5 is given by

$$\begin{aligned} &g_5(\{1\}) = 1,\ g_5(\{2\}) = 3,\ g_5(\{3\}) = 3,\\ &g_5(\{1,2\}) = 4,\ g_5(\{1,3\}) = 4,\ g_5(\{2,3\}) = 4,\\ &g_5(\{1,2,3\}) = 5. \end{aligned} \tag{24}$$

Now, $f_5(\{1,2,3\} = g_5(\{1,2,3\}) = \beta = 5$ which indicates that $\beta = 5$ is a feasible sum-rate for the problem (20). Hence, the optimization problem (20) can be written as

$$\min_{\mathbf{R} \in \mathbb{Z}^m} \sum_{i=1}^{m} \varphi_i(R_i), \quad \text{s.t.,} \quad \mathbf{R} \in B(g_\beta) \tag{25}$$

provided that $g_\beta(\mathcal{M}) = \beta$.

Remark 2: Parameter β is feasible w.r.t. the problem (20) if $g_\beta(\mathcal{M}) = \beta$. Otherwise, $g_\beta(\mathcal{M}) < \beta$. This is the direct consequence of the Dilworth truncation (22).

Depending upon the cost function $\sum_{i=1}^{m} \varphi(R_i)$, in the sequel, we provide several algorithms that can efficiently solve problem (10). First, we analyze a special case when the cost function is linear,

$$\varphi_i(R_i) = \alpha_i R_i, \quad \alpha_i > 0, \quad \forall i \in \mathcal{M}. \tag{26}$$

The condition $\alpha_i > 0$, $i \in \mathcal{M}$ ensures that φ_i is a non-decreasing function.

B. Linear Cost

In this section, we study a linear cost data exchange problem. This problem was independently solved by the authors of this paper in [1] and [2], and Courtade *et al.* in [9]. When the cost function is linear, the optimization problem (25) has the following form

$$\min_{\mathbf{R} \in \mathbb{Z}^m} \sum_{i=1}^{m} \alpha_i R_i, \quad \text{s.t.,} \quad \mathbf{R} \in B(g_\beta). \tag{27}$$

Due to the submodularity of function g_β, the optimization problem (27) can be solved analytically using Edmonds' greedy algorithm [26] (see Algorithm 1).

The greediness of this algorithm is reflected in the fact that each update of the rate vector is sum-rate optimal:

$$\begin{aligned} R^*_{j(1)} &= g_\beta(\{j(1)\})\\ R^*_{j(1)} + R^*_{j(2)} &= g_\beta(\{j(1), j(2)\})\\ &\vdots\\ \sum_{i=1}^{m} R^*_{j(i)} &= g_\beta(\{j(1), \ldots, j(m)\}). \end{aligned} \tag{29}$$

Algorithm 1 Edmonds' Algorithm

```
1: Set $j(1), j(2), \dots, j(m)$ to be an ordering of $\{1, 2, \dots, m\}$
   such that $\alpha_{j(1)} \le \alpha_{j(2)} \le \cdots \le \alpha_{j(m)}$.
2: Initialize $\mathbf{R}^* = \mathbf{0}$.
3: for $i = 1$ to $m$ do
4:     $R^*_{j(i)} = g_\beta(\{j(1), j(2), \dots, j(i)\}) - g_\beta(\{j(1), j(2), \dots, j(i-1)\}). \quad (28)$
5: end for
```

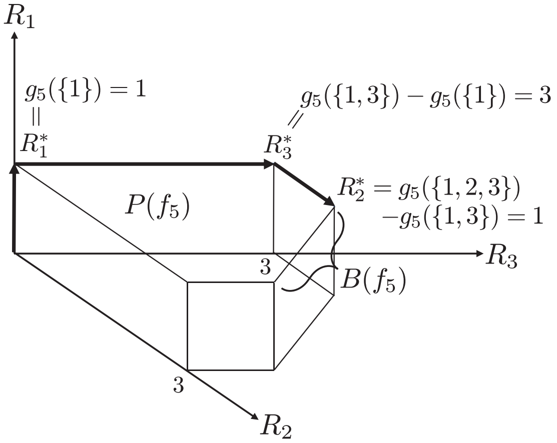

Fig. 2. Edmonds' algorithm applied to the three-user problem described in Example 2, with the cost function $R_1 + 3R_2 + 2R_3$. To minimize the cost, the order in which we greedily update communication rates should be increasing w.r.t. the weight vector, i.e., 1 → 3 → 2. The optimal *DE*-rate vector is $R_1^* = 1$, $R_2^* = 1$, $R_3^* = 3$.

In other words, at each iteration, the individual user's rate update reaches the boundary of polyhedron $P(g_\beta)$. Optimality of this approach is the direct consequence of submodularity of function g_β [26].

Remark 3: The optimal rate vector $\mathbf{R}^*$ belongs to the base polyhedron $B(g_\beta)$. In other words,

$$\sum_{i=1}^{m} R_i^* = g_\beta(\mathcal{M}). \tag{30}$$

Remark 4: The complexity of Edmonds' algorithm is $\mathcal{O}(m\vartheta)$, where ϑ is the complexity of computing function $g_\beta(\mathcal{S})$ for any given set $\mathcal{S} \subseteq \mathcal{M}$.

Example 2: Let us consider the same source model as in Example 1, and let the cost function be $R_1 + 3R_2 + 2R_3$, and $\beta = 5$. The intersecting submodular function f_β, and its Dilworth truncation g_β are given in (19) and (24), respectively. The rate vector is updated in an increasing order w.r.t. the weight vector. In this case, the order is 1 → 3 → 2 (see Figure 2).

The main problem in executing Edmonds' algorithm efficiently is that the function g_β is not available analytically. To compute this function for any given set $\mathcal{S} \subseteq \mathcal{M}$ we need to solve minimization problem (22). Such minimization has to be performed over all partitions of the set $\mathcal{S}$, which annuls the efficiency of the proposed method.

To overcome this problem note that we have access to the function f_β (see (16)), and by Theorem 2, we know that $P(g_\beta) = P(f_\beta)$. As pointed out before, each rate update reaches the boundary of polyhedron $P(g_\beta)$ (see (29)). Since we don't explicitly have function g_β, this polyhedron boundary can be calculated by applying the Dilworth truncation formula (22). For the three-user problem in Example 2 this procedure would go as follows:

$$R_1^* = f_5(\{1\}) = 1, \tag{31}$$

$$R_3^* = \min\{f_5(\{1,3\}) - R_1^*, f_5(\{3\})\} = 3, \tag{32}$$

$$R_2^* = \min\{f_5(\{1,2,3\}) - R_1^* - R_3^*, f_5(\{1,2\}) - R_1^*, f_5(\{2,3\}) - R_3^*, f_5(\{2\})\} = 1. \tag{33}$$

Algorithm 2 Minimizing Linear Cost Under Intersecting Submodular Constraints

```
1: Set $j(1), j(2), \dots, j(m)$ to be an ordering of $\{1, 2, \dots, m\}$
   such that $\alpha_{j(1)} \le \alpha_{j(2)} \le \cdots \le \alpha_{j(m)}$.
2: Initialize $\mathbf{R}^* = \mathbf{0}$.
3: for $i = 1$ to $m$ do
4:     $R^*_{j(i)} = \min_{\mathcal{S}}\{f_\beta(\mathcal{S} \cup \{j(i)\}) - R^*(\mathcal{S}) : \mathcal{S} \subseteq \{j(1), j(2), \dots, j(i-1)\}\}. \quad (34)$
5: end for
```

Generalization of this procedure to an arbitrary number of users is shown in Algorithm 2. We refer the interested reader to references [27]–[29] where this algorithm is explained in more detail for an arbitrary intersecting submodular functions.

In each iteration i, the minimization problem (34) is over all subsets of $\{j(1), \dots, j(i)\}$. Using the fact that all the subsets considered in (34) contain a common element $j(i)$ it is easy to see that $f_\beta(\mathcal{S}) - R^*(\mathcal{S})$ is fully submodular over the domain set $\{j(1), j(2), \dots, j(i-1)\}$. Now the polynomial time solution of Algorithm 2 follows from the fact that minimization of a fully submodular function can be done in polynomial time [30].

Remark 5: The complexity of Algorithm 2 is $\mathcal{O}(mSFM(m))$, where $SFM(m)$ is the complexity of minimizing submodular function. The best known algorithm to our knowledge is proposed by Orlin in [30], and has complexity $\mathcal{O}(m^5\gamma + m^6)$, where γ is complexity of computing the submodular function. For the submodular function defined in (34), γ equals to the complexity of computing rank, and it is a function of the file size N. When users observe linear combinations of the file packets, the rank over $\mathbb{F}_q$ can be computed by Gaussian elimination in $\mathcal{O}(N^3)$ time. For the "raw" packet model, rank computation reduces to counting distinct packets, and therefore its complexity is $\mathcal{O}(N)$. The complexity of sorting in Step 1 is $\mathcal{O}(m \log m)$ which is much lower than the complexity of the for loop in Algorithm 2, and thus it does not contribute to the overall order of complexity of the algorithm.

Remark 6: From Remark 2 and the fact that Edmonds' algorithm provides a rate vector with sum-rate $g_\beta(\mathcal{M})$, it immediately follows that if Algorithm 2 outputs a rate vector $\mathbf{R}^*$ such that $R^*(\mathcal{M}) < \beta$, then $B(f_\beta) = \emptyset$, and such β is not a feasible sum-rate w.r.t. the problem (20). Hence, for

Algorithm 3 Minimum Sum-Rate Algorithm (Binary Search)

```
1: Initialize $\beta_{start} = 0$, $\beta_{end} = N$.
2: while $\beta_{end} - \beta_{start} > 1$ do
3:     $\beta = \lceil \frac{\beta_{start}+\beta_{end}}{2} \rceil$.
4:     Execute Algorithm 2 with parameter $\beta$.
5:     if $\sum_{i=1}^{m} R_i^* = \beta$ then
6:         $\beta_{end} = \beta$.
7:     else
8:         $\beta_{start} = \beta$.
9:     end if
10: end while
11: $\beta_{end}$ is the minimum sum-rate.
```

any given β, the feasibility of such sum-rate can be verified in $\mathcal{O}(mSFM(m))$ time.

C. Finding the Optimal Value of β

So far we have shown how to compute function $h(\beta)$ defined in (13) for any β when $\varphi_i(R_i) = \alpha_i R_i$. To complete our solution, *i.e.*, to solve the problem defined in (10), it remains to show how to minimize function $h(\beta)$ efficiently.

Theorem 3: Function $h(\beta)$, defined in (11)*, is convex when β is a feasible sum-rate w.r.t. the optimization problem* (11).

The proof of Theorem 3 is provided in Appendix B.

In order to minimize function h, first, we identify the set of sum-rates β that are feasible w.r.t. the problem (10). More precisely, we need to find the minimum sum-rate, since every β that is larger than or equal to such value is feasible as well. Hence, we proceed by analyzing the sum-rate objective, *i.e.*, when $\varphi_i(R_i) = R_i$.

For any fixed parameter $\beta \in \mathbb{Z}_+$, Algorithm 2 provides an optimal rate allocation w.r.t. the linear cost. It is only left to find β that minimizes $h(\beta)$ in (10). Let us first consider the sum-rate cost, *i.e.*, $\varphi_i(R_i) = R_i$. From the equivalence of the Algorithms 1 and 2, and from Remark 3 it follows that for any given parameter β, the output rate vector $\mathbf{R}^*$ of Algorithm 2 satisfies

$$\sum_{i=1}^{m} R_i^* = g_\beta(\mathcal{M}). \tag{35}$$

Thus, for a randomly chosen parameter β we can verify whether it is feasible w.r.t. the problem (13) by applying Remark 2, *i.e.*, if $\sum_{i=1}^{m} R_i^* = \beta$, then such sum-rate can be achieved. Therefore, we can apply a simple binary search algorithm to find the minimum sum-rate. Note that the minimum sum-rate is always less than or equal to the file size N. Hence, we can confine our search accordingly (see Algorithm 3).

Remark 7: The complexity of Algorithm 3 is $\mathcal{O}(mSFM(m)\log N)$.

For the general linear cost function $\varphi_i(R_i) = \alpha_i R_i$, by Theorem 3, $h(\beta)$ is convex for β greater than the minimum sum-rate (obtained from Algorithm 3). In Section III-F, Lemma 5, we show that the search space for β that minimizes function h can be limited to the file size N. Hence, in order to solve the minimization problem (10) we can apply a simple binary search algorithm that finds the minimum of $h(\beta)$ by looking for a slope change in function h.

Algorithm 4 Minimum Linear Cost Algorithm

```
1: Initialize $\beta_{start} = \beta_{sum}^*$, $\beta_{end} = N$, where $\beta_{sum}^*$ is the
   minimum sum-rate obtained from Algorithm 3.
2: $\beta = \lceil \frac{\beta_{start}+\beta_{end}}{2} \rceil$.
3: Execute Algorithm 2 for $\beta - 1$, $\beta$, and $\beta + 1$.
4: if $h(\beta) \leq h(\beta-1)$ and $h(\beta) \leq h(\beta+1)$ then
5:     $\mathbf{R}^*$ that corresponds to the sum-rate $\beta$ is an optimal rate
       allocation.
6: else if $h(\beta-1) \geq h(\beta) \geq h(\beta+1)$ then
7:     $\beta_{start} = \beta + 1$.
8: else
9:     $\beta_{end} = \beta - 1$.
10: end if
11: Go to Step 2.
```

Remark 8: Since for any fixed β, $h(\beta)$ can be found by using Algorithm 2, and β_{sum}^* can be found by applying Algorithm 3, the complexity of Algorithm 4 is $\mathcal{O}(mSFM(m)\log N)$.

D. Using Subgradient Methods to Solve Step 4 of Algorithm 2

In this section we propose an alternative solution to the minimization problem (34) in Algorithm 2 that does not involve minimization of a submodular function. The underlying linear optimization problem has the following form

$$\min_{\mathbf{R}\in\mathbb{Z}^m} \sum_{i=1}^{m} \alpha_i R_i, \quad \text{s.t. } \mathbf{R} \in B(f_\beta), \tag{36}$$

given that β is a feasible sum-rate. Without loss of generality, let us assume that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_m$. In this case, the minimization in Step 4 of Algorithm 2 can be written as

$$\begin{aligned} R_i^* = \min_{\mathcal{S}} \{ & f_\beta(\mathcal{S}) - R^*(\mathcal{S}) : i \in \mathcal{S}, \\ & \mathcal{S} \subseteq \{1,2,\ldots,i\}\}, \quad i = 1,2,\ldots,m. \end{aligned} \tag{37}$$

Minimization (37) can be interpreted as a maximal update along the i^{th} coordinate such that R_i^* still belongs to polyhedron $P(f_\beta)$. This problem can be separately formulated as the following maximization problem

$$\begin{aligned} R_i^* = \max_{\mathbf{R}\in\mathbb{R}^i} & \ R_i, \\ \text{s.t. } & R_k \geq R_k^*, \ k = 1,2,\ldots,i-1, \\ & R(\mathcal{S}\cup\{i\}) \leq f_\beta(\mathcal{S}\cup\{i\}), \quad \forall \mathcal{S} \subseteq \{1,2,\ldots,i-1\}. \end{aligned} \tag{38}$$

Note that in an optimal solution, the condition $R_k \geq R_k^*$, $k = 1,\ldots,i-1$, holds with equality because any possible increase of R_k can lead to the smaller value of R_i. Moreover, since the above minimization is over an integer submodular polyhedron, the optimal solution is also an integer number. Therefore, minimization problems (38) and (37) are equivalent.

Let us denote by $\mathcal{R}^{(i)}$ the rate region that corresponds to the optimization problem (38)

$$\mathcal{R}^{(i)} = \{\mathbf{R} \in \mathbb{R}^i | R(\mathcal{S} \cup \{i\}) \leq f_\beta(\mathcal{S} \cup \{i\}), \\ \forall \mathcal{S} \subseteq \{1, 2, \ldots, i-1\}\}. \tag{39}$$

To solve optimization problem (38), we apply the dual subgradient method. First, the Lagrangian function of the problem (38) is

$$\mathcal{L}(\mathbf{R}, \boldsymbol{\lambda}) = R_i + \sum_{k=1}^{i-1} \lambda_k (R_k - R_k^*), \quad \text{s.t. } \mathbf{R} \in \mathcal{R}^{(i)}, \tag{40}$$

where $\lambda_k \geq 0$, $k = 1, 2, \ldots, i-1$. Then, the dual function $\delta(\boldsymbol{\lambda})$ equals to

$$\begin{aligned} \delta(\boldsymbol{\lambda}) &= \max_{\mathbf{R} \in \mathcal{R}^{(i)}} \mathcal{L}(\mathbf{R}, \lambda) \\ &= \max_{\mathbf{R} \in \mathcal{R}^{(i)}} \left\{ R_i + \sum_{k=1}^{i-1} \lambda_k R_k \right\} - \sum_{k=1}^{i-1} \lambda_k R_k^*. \end{aligned} \tag{41}$$

Due to the maximization step in (41) over multiple hyperplanes, it immediately follows that $\delta(\boldsymbol{\lambda})$ is a convex function. By the weak duality theorem [31],

$$\delta(\lambda) \geq R_i^*, \quad \forall \lambda_k \geq 0, \ k = 1, 2, \ldots, i-1. \tag{42}$$

Hence,

$$\min_{\boldsymbol{\lambda}} \{\delta(\boldsymbol{\lambda}) | \lambda_k \geq 0, \ k = 1, 2, \ldots, i-1\} \geq R_i^* \tag{43}$$

Since optimization problem (38) is linear, there is no duality gap, *i.e.*,

$$R_i^* = \min_{\boldsymbol{\lambda}} \{\delta(\boldsymbol{\lambda}) | \lambda_k \geq 0, \ k = 1, 2, \ldots, i-1\}. \tag{44}$$

To solve optimization problem (44), we apply the dual subgradient method [32] as follows. Starting with a feasible iterate $\lambda_k[0]$, $k = 1, 2, \ldots, i-1$, w.r.t. the optimization problem (44), and the step size θ_j, every subsequent iterate $\lambda_k[j+1]$ for all $k = 1, 2, \ldots, i-1$, can be recursively computed as follows

$$\lambda_k[j+1] = \left\{ \lambda_k[j] - \theta_j (\tilde{R}_k[j] - R_k^*) \right\}_+, \tag{45}$$

where $\tilde{R}_k[j]$ is an optimal solution to the problem

$$\max_{\mathbf{R} \in \mathcal{R}^{(i)}} R_i + \sum_{k=1}^{i-1} \lambda_k[j] R_k. \tag{46}$$

Note that $\tilde{R}_k[j] - R_k^*$, $k = 1, 2, \ldots, i-1$, is a derivative of the dual function $\delta(\boldsymbol{\lambda}[j])$.

Lemma 2: An optimal solution to the problem (46) *can be obtained as follows. Let* $t(1), t(2), \ldots, t(i-1)$ *be an ordering of* $1, 2, \ldots, i-1$ *such that* $\lambda_{t(1)} \geq \lambda_{t(2)} \geq \cdots \geq \lambda_{t(i-1)}$. *Then,*

$$\tilde{R}_i[j] = \begin{cases} f_\beta(\{i\}), & \text{if } \lambda_{t(1)} \leq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{47}$$

$$\tilde{R}_{t(k)} = f_\beta(\mathcal{S}_{t(k)} \cup \{i\}) - \sum_{u=1}^{k-1} \tilde{R}_{t(u)}[j] - \tilde{R}_i[j], \tag{48}$$

for $k = 1, 2, \ldots, i-1$, where $\mathcal{S}_{t(k)} \triangleq \{t(1), t(2), \ldots, t(k)\}$.

The proof of this Lemma is provided in Appendix D.

Remark 9: The complexity of the algorithm proposed by Lemma 2 is $\mathcal{O}(i \log i + i\gamma)$, where γ is the complexity of computing rank.

We apply subgradient methods instead of gradient because the function $\delta(\boldsymbol{\lambda})[j]$ even though convex, is not differentiable. From Lemma 2, it follows that for a given $\boldsymbol{\lambda}[j]$, there may be more than one maximizer of the problem (46). Due to possibility of having more than one direction along which we can update vector $\boldsymbol{\lambda}[j]$ according to (45), subgradient method is not technically a descent method; the function value $\delta(\boldsymbol{\lambda}[j])$ may often increase in the consecutive steps. For that reason, at each step we keep track of the smallest solution up to that point in time

$$\tilde{\boldsymbol{\lambda}}[j] = \operatorname{argmin} \{\delta(\boldsymbol{\lambda}[0]), \delta(\boldsymbol{\lambda}[1]), \ldots, \delta(\boldsymbol{\lambda}[j])\}. \tag{49}$$

Before we go any further, note that the primal optimization problem (38) is over real vectors. However, the minimization (36) is an integer optimization problem. As pointed out above, the optimal solution of the problem (38) is equal to the solution of the problem (36). Therefore, we can choose the number of iterations l of the dual subgradient method such that we get "close enough" to an integer solution. In other words,

$$\left| \delta(\tilde{\boldsymbol{\lambda}}[l]) - R_i^* \right| \leq \varepsilon, \tag{50}$$

where $\varepsilon < 0.5$. Then,

$$R_i^* = \operatorname{round}(\delta(\tilde{\boldsymbol{\lambda}}[l])). \tag{51}$$

E. Convergence Analysis

In this section we explore the relationship between the number of iterations of the dual subgradient method l, and the step size θ_j, such that it is guaranteed that (51) provides the optimal solution.

Lemma 3: Let $\boldsymbol{\lambda}^*$ *be an optimal vector that minimizes the dual function* δ. *Then,*

$$\delta(\tilde{\boldsymbol{\lambda}}[l-1]) - \delta(\boldsymbol{\lambda}^*) \\ \leq \frac{\left(\sum_{k=1}^{i-1} \lambda_k[0]\right)^2 + \left(\sum_{k=1}^{i-1} \lambda_k^*\right)^2 + 2N^2 \sum_{j=0}^{l-1} \theta_j^2}{2 \sum_{j=0}^{l-1} \theta_j}. \tag{52}$$

The proof of Lemma 3 can be derived from the notes on subgradient methods presented in [32]. For the sake of completeness, we provide its entire proof in Appendix E.

Since by Lemma 3, $\boldsymbol{\lambda}^*$ can be an arbitrary minimizer of the dual function δ, let us choose $\boldsymbol{\lambda}^*$ that can be bounded as suggested by the following lemma.

Lemma 4: There exists an optimal solution to the problem (44) *that satisfies*

$$\sum_{k=1}^{i-1} \lambda_k^* \leq m. \tag{53}$$

The proof of this Lemma is provided in Appendix F.

An initial feasible $\boldsymbol{\lambda}[0]$ can be chosen as follows

$$\lambda_k[0] = 0, \quad \forall k \in \{1, 2, \ldots, i-1\}. \tag{54}$$

Algorithm 5 Minimization (34) of Algorithm 2

```
Select parameters $l$, and $\theta_j,\ j = 0, 1, \dots, l-1$ such that
    $$\frac{m^2 + 2N^2 \sum_{j=0}^{l-1} \theta_j^2}{2\sum_{j=0}^{l-1} \theta_j} < \frac{1}{2}. \qquad (60)$$
Set $\lambda_k[0] = 0,\ k = 1, 2, \dots, i-1$, and $\tilde{\boldsymbol{\lambda}}[0] = \boldsymbol{\lambda}[0]$.
for $j = 0$ to $l-1$ do
    $$\lambda_k[j+1] = \left\{\lambda_k[j] - \theta_j(\tilde{R}_k[j] - R_k^*)\right\}_+, \qquad (61)$$
    for $k = 1, 2, \dots, i-1$, where $\tilde{\mathbf{R}}[j]$ is computed according to Lemma 2.
    $$\tilde{\boldsymbol{\lambda}}[j+1] = \operatorname{argmin} \left\{\delta(\boldsymbol{\lambda}[j+1]), \delta(\tilde{\boldsymbol{\lambda}}[j])\right\}. \qquad (62)$$
end for
$$R_i^* = \operatorname{round}\left(\delta(\tilde{\boldsymbol{\lambda}}[l])\right). \qquad (63)$$
```

Combining (52), (53) and (54), we obtain

$$\delta(\tilde{\boldsymbol{\lambda}}[l-1]) - \delta(\boldsymbol{\lambda}^*) \leq \frac{m^2 + 2N^2 \sum_{j=0}^{l-1} \theta_j^2}{2\sum_{j=0}^{l-1} \theta_j}. \qquad (55)$$

There are many ways to choose the step size that satisfies the condition (55) (see [32]). Here, we briefly examine the constant step size, where $\theta_j = \theta,\ j = 0, 1, 2, \dots$. In this case, the inequality (55) becomes

$$\delta(\tilde{\boldsymbol{\lambda}}[l-1]) - \delta(\boldsymbol{\lambda}^*) \leq \frac{m^2 + 2N^2 l \theta^2}{2l\theta}. \qquad (56)$$

Hence, the condition (50) is satisfied when

$$\frac{m^2 + 2N^2 l \theta^2}{2l\theta} < \frac{1}{2}. \qquad (57)$$

It can be easily verified that (57) holds when

$$\theta < \frac{1}{2N^2}, \qquad (58)$$

$$l > \frac{m^2}{\theta(1 - 2N^2\theta)}. \qquad (59)$$

Putting all these results together, the minimization (37) can be obtained by running Algorithm 5.

Remark 10: From Remark 9 it follows that the complexity of Algorithm 5 is $SFM(m) = \mathcal{O}(lm \log m + lm\gamma)$. For a constant step size θ, from (58) and (59) it follows that the complexity of Algorithm 5 can be bounded by $\mathcal{O}(N^2 m^3 \log m + N^2 m^3 \gamma)$.

Remark 11: Note that Algorithm 5 can be applied to solve problem (36) when f_β is an arbitrary intersecting submodular function over integers.

Algorithm 6 Minimizing Separable Convex Cost Under Submodular Constraints

```
Set $R_i = 0,\ \forall i \in \mathcal{M}$.
for $j = 1$ to $\beta$ do
    Find $i^* \in \mathcal{M}$ such that
        $$i^* = \operatorname*{argmin}_{i \in \mathcal{M}} \{d_i(R_i + 1) | \mathbf{R} + \mathbf{e}(i) \in P(g_\beta)\}, \qquad (65)$$
    where
        $$d_i(R_i + 1) \triangleq \varphi_i(R_i + 1) - \varphi_i(R_i), \qquad (66)$$
    and $\mathbf{e}(i)$ is the unit basis $m$-dimensional vector with $i^{th}$ coordinate equals to 1.
    Set $R_{i^*} = R_{i^*} + 1$.
end for
$\mathbf{R}^* = \mathbf{R}$ is an optimal rate vector w.r.t. the problem (64).
```

F. General Separable Convex Cost

In the previous section, for the linear cost function, we applied Edmonds' algorithm in order to obtain the optimal rate allocation. Edmonds' algorithm is greedy by its nature since all rate updates are reaching the boundary of polyhedron $P(g_\beta)$. This effectively means that Edmonds' algorithm provides rate allocations that are vertices of the base polyhedron $B(g_\beta)$. While this was an optimal approach in the case of linear objectives, for the general separable convex cost function the optimal rate vector may not belong to a vertex of $B(g_\beta)$. We will show this in Example 3.

The general convex cost optimization problem

$$\min_{\mathbf{R} \in \mathbb{Z}^m} \sum_{i=1}^{m} \varphi_i(R_i), \quad \text{s.t.} \quad \mathbf{R} \in B(g_\beta) \qquad (64)$$

is known as a *resource allocation problem under submodular constraints* [33], and it can be solved by applying the following intuitive approach: instead of applying greedy scheme, we will incrementally update by one symbol in $\mathbb{F}_q$ a communication rate of a user that has the minimal discrete derivative (see Algorithm 6).

Definition 4: Let us define set $\mathcal{T}_j$ to be the set of all users that are in iteration j of Algorithm 6 allowed to update their transmission rates

$$\mathcal{T}_j \triangleq \left\{i | \mathbf{R} + \mathbf{e}(i) \in P(g_\beta)\right\}. \qquad (67)$$

The question is how to efficiently recover set $\mathcal{T}_j$ in each round of Algorithm 6. First, we observe that $P(g_\beta) = P(f_\beta)$ according to Theorem 2. Second, note that in Algorithm 2, the minimization (34) outputs the maximum rate vector update along one coordinate. Therefore, we only need to verify whether such update is at least equal to one symbol in $\mathbb{F}_q$. In other words, $i \in \mathcal{T}_j$ if

$$\min_{\mathcal{S} \subseteq \mathcal{M} \setminus \{i\}} \{f_\beta(\mathcal{S} \cup \{i\}) - R(\mathcal{S} \cup \{i\})\} \geq 1. \qquad (68)$$

Now we can obtain a polynomial time solution to problem (13) by applying Algorithm 7.

The complexity of (68) is $SFM(m)$, since the function $f_\beta(\mathcal{S}) - R^*(\mathcal{S})$ is fully submodular. This check can be done

Algorithm 7 Minimizing separable Convex Cost Under Intersecting Submodular Constraints

```
1: Set $R_i = 0, \forall i \in \mathcal{M}$.
2: for $j = 1$ to $\beta$ do
3:     Construct set $\mathcal{T}_j$ as follows
           $\mathcal{T}_j = \{i : \min_{\mathcal{S} \subseteq \mathcal{M} \setminus \{i\}} \{f_\beta(\mathcal{S} \cup \{i\}) - R(\mathcal{S} \cup \{i\})\} \geq 1\}.$   (69)
4:     Find $i^* \in \mathcal{T}_j$ such that
           $i^* = \operatorname{argmin}_{i \in \mathcal{T}_j} \{d_i(R_i + 1)\}.$   (70)
5:     Set $R_{i^*} = R_{i^*} + 1$.
6: end for
7: $\mathbf{R}^* = \mathbf{R}$ is an optimal rate vector w.r.t. the problem (64).
```

either by minimizing submodular function as suggested in (68) or by running the dual subgradient algorithm similar to the one proposed in Section III-D. Here, we briefly explain the differences. First, rate region $\mathcal{R}^{(i)}$ defined in (39), now has the following form

$$\mathcal{R}^{(i)} = \{\mathbf{R} \in \mathbb{R}^m | R(\mathcal{S} \cup \{i\}) \leq f_\beta(\mathcal{S} \cup \{i\}),\ \forall \mathcal{S} \subseteq \mathcal{M} \setminus \{i\}\}. \tag{71}$$

Let us denote by $\mathbf{R}^* \in \mathbb{R}^m$ the current rate allocation in round j of Algorithm 7. Then, if the maximization

$$\max_{\mathbf{R} \in \mathcal{R}^{(i)}} R_i, \quad \text{s.t. } R_k \geq R_k^*, \quad k = 1, 2, \ldots, m, \tag{72}$$

is at least 1, then $i \in \mathcal{T}_j$. Problem (72) can be solved by following the same steps in solving the dual problem as in Section III-D.

Remark 12: At each iteration, Algorithm 7 calls (69) m times, and there are total of β iterations. Therefore, the complexity of Algorithm 7 is $\mathcal{O}(m\beta SFM(m))$.

Lemma 5: Let us denote by β^* the minimizer of the function h defined in (13). *Then, $\beta^* \leq N$.*

The proof of Lemma 5 is provided in Appendix G.

For the general non-decreasing set of convex functions φ_i, $i \in \mathcal{M}$, from Theorem 3 we know that function h is convex. Moreover, by Lemma 5 it follows that the minimizer of h is at most equal to N. Therefore, in order to minimize h, we can apply Algorithm 4 which computes $h(\beta)$ for any β by applying Algorithm 7. Thus, the overall complexity of the proposed solution is $\mathcal{O}(mSFM(m)N \log N)$.

G. Fairness Under the Fixed Sum-Rate Budget

In this section we study the problem where for the fixed feasible sum-rate budget β, the goal is to distribute communication load to users as evenly as possible. Linear cost function is by its nature "unfair," since it can potentially result in a communication scheme where only a small group of users transmit packets. For the fixed sum-rate budget, the "fairness" can be achieved by introducing an uniform, non-decreasing (in the integer domain) objective $\varphi_i(R_i) = R_i \log R_i$, $i = 1, \ldots, m$, and it is illustrated in the example below.

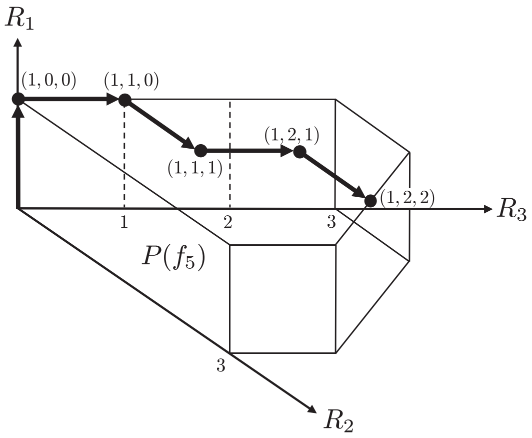

Fig. 3. Algorithm 7 applied to the three-user problem from Example 3, with the cost function $\sum_{i=1}^{3} R_i \log R_i$ and the fixed sum-rate $R_1 + R_2 + R_3 = 5$. To minimize the cost, in each iteration we update the rate of the user who has transmitted the least amount of symbols in $\mathbb{F}_q$ such that the update still belongs to polyhedron $P(f_\beta)$.

Example 3: Consider the same three-user problem as in Example 1

$$\mathbf{x}_1 = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T, \quad \mathbf{x}_2 = \begin{bmatrix} w_2 & w_4 & w_5 & w_6 \end{bmatrix}^T, \quad \mathbf{x}_3 = \begin{bmatrix} w_3 & w_4 & w_5 & w_6 \end{bmatrix}^T, \tag{73}$$

where $w_i \in \mathbb{F}_q$, $i = 1, \ldots, 6$.

In case of the linear objective $R_1 + 3R_2 + 2R_3$, and the given sum-rate $\beta = 5$, we showed in Example 2 that the optimal *DE*-rate vector obtained by using Algorithm 2, belongs to a vertex of the base polyhedron $B(f_\beta)$:

$$R_1^* = 1, \quad R_2^* = 1, \quad R_3^* = 3. \tag{74}$$

Let us now analyze the case when the objective is $\varphi_i(R_i) = R_i \log R_i$, $i = 1, 2, 3$. Then, from (66), it follows that

$$d_i(R_i + 1) = (R_i + 1)\log(R_i + 1) - R_i \log R_i. \tag{75}$$

It is not hard to show that the above function $d_i()$ is increasing. Hence, step 4 of Algorithm 7 can be written as

$$i^* = \operatorname{argmin}_{i \in \mathcal{T}_j} R_i, \tag{76}$$

where $\mathcal{T}_j$ can be computed from (68), and $j = 1, \ldots, \beta$ is an iteration of Algorithm 7. The condition (76) proves that $\varphi_i(R_i) = R_i \log R_i$ is a good measure for fairness, since it is enforcing the transmission vector $\mathbf{R}$ to be as uniform as possible. The execution steps of Algorithm 7 are shown in Figure 3. It can be verified that

$$\mathcal{T}_1 = \{1, 2, 3\}, \ \mathcal{T}_2 = \mathcal{T}_3 = \mathcal{T}_4 = \mathcal{T}_5 = \{2, 3\}. \tag{77}$$

Therefore, the optimal *DE*-rate vector for this example is $R_1^* = 1$, $R_2^* = 2$, $R_3^* = 2$.

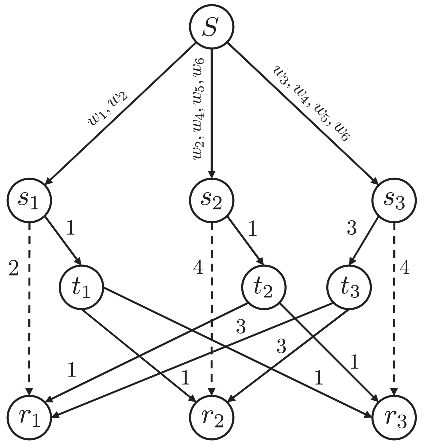

Fig. 4. Multicast network constructed from the source model and *DE*-rate vector $R_1^* = 1$, $R_2^* = 1$, $R_3^* = 3$. Hence, in an optimal scheme users 1, 2 and 3 are transmitting 1, 1, and 3 linear combinations of their own observations in $\mathbb{F}_q$, respectively. Each user receives side-information from "itself" (through the links $s_i \to r_i$, $i = 1, 2, 3$) and from the other users (through the links $t_i \to r_j$, $i, j \in \{1, 2, 3\}$, $i \neq j$).

IV. Code Construction

In Theorem 1, we showed that in order to achieve optimal communication rates, it is sufficient for each user to transmit the optimal number of linear combinations of its observations. In this section, we show how to efficiently design the transmission scheme. We explain the code construction on the three user problem from Example 2, where

$$\begin{aligned}\mathbf{x}_1 &= \begin{bmatrix} w_1 & w_2 \end{bmatrix},\\ \mathbf{x}_2 &= \begin{bmatrix} w_2 & w_4 & w_5 & w_6 \end{bmatrix},\\ \mathbf{x}_3 &= \begin{bmatrix} w_3 & w_4 & w_5 & w_6 \end{bmatrix},\end{aligned} \tag{78}$$

and $R_1^* = 1, R_2^* = 1, R_3^* = 3$. This means that in an optimal scheme users 1, 2 and 3 transmit 1, 1, and 3 linear combinations of their own observations in $\mathbb{F}_q$, respectively. We design the coding scheme by first constructing the corresponding multicast network (see Figure 4). In this construction, notice that there are several types of nodes. First, there is a super node S that has all the packets. Each user in the system is a transmitter, while in addition, each user is also a receiver. To model this, we denote s_1, s_2 and s_3 to be the "transmitting" nodes, and r_1, r_2 and r_3 to be the "receiving" nodes. The side-information observed by users 1, 2 and 3 gets directly routed from s_1, s_2 and s_3 to the receivers r_1, r_2 and r_3 through direct edges (dashed edges in Figure 4). To model the broadcast nature of each transmission, we introduce the "dummy" nodes t_1, t_2 and t_3, such that the capacity of the links (s_i, t_i) is the same as link capacity (t_i, r_j), $j \neq i$, and is equal to R_i^*, $\forall i \in \mathcal{M}$.

Now, when we have a well-defined network it is only left to figure out transmissions on all the edges. For instance, this can be achieved using Jaggi et al. algorithm [16]. The first step of this algorithm is to determine $N = 6$ disjoint paths from the super-node S to each receiver r_1, r_2 and r_3 by using the Ford-Fulkerson algorithm [34]. Such paths are designed to carry linearly independent messages from the super node to the receivers. When each user observes some subset of the file packets (as it is the case in this example), this problem can be solved as a special case of [7]. Namely, we can directly apply Jaggi *et al.* algorithm to this problem by slightly modifying the upper portion of the multicast network from Figure 4 (see Figure 5). Note that in this case, we were able to model observations of each user simply by adding one more layer of nodes which represent individual file packets, and then connecting these packet nodes with each user according to (78). In other words, the entire source model and the communication model can be represented by multicast acyclic graph. Therefore, Jaggi *et al.* algorithm would find actual transmissions of each user in polynomial time.

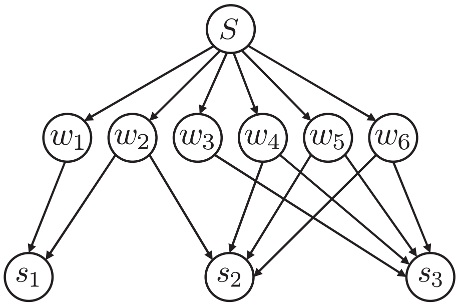

Fig. 5. When each user observes subset of the file packets, we can model the observations by adding an extra layer of $N = 6$ nodes to the graph in Figure 4. Each extra node represents one file packet, and all extra edges are of capacity 1. Then, users' observations can be modeled by connecting nodes from this layer to the users' nodes s_1, s_2 and s_3 according to (78).

In the case of general linear packet model, it is not possible to represent users' observations just by adding one extra layer of nodes to the multicast graph as in Figure 5. This is because there is an underlying correlation between all the linear combinations that appear in the users' observation vectors, and it would be suboptimal to treat all these combinations independently. Thus, the only way to solve this problem is to directly give the linear combinations to the users' nodes s_1, s_2 and s_3. This can be achieved by applying Harvey's algorithm [25] which is based on matrix representation of transmissions in the network [24], [35], and simultaneous matrix completion problem over finite fields. In the remainder of this section, we briefly examine building blocks of this code construction algorithm.

First, we choose source matrix $\mathbf{A}$ to be the side-information matrix of all users as,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1^T & \dots & \mathbf{A}_m^T & \mathbf{0} \dots \mathbf{0} \end{bmatrix}, \tag{79}$$

where $\mathbf{A}_i$ corresponds to the observation matrix defined in (1). Matrix $\mathbf{A}$ is an $N \times \ell$ matrix, where ℓ is the total number of edges in the network.

The transfer matrix $\mathbf{M}(r_i)$ from the super-node S to any receiver r_i, $i \in \mathcal{A}$ can be obtained as shown in [24]. It is a $N \times N$ matrix with the input vector $\mathbf{w}$, and the output vector corresponding to the observations at the receiver r_i

$$\mathbf{M}(r_i) = \mathbf{A}(\mathbf{I} - \Gamma)^{-1}\mathbf{D}(r_i), \quad i = 1, 2, \dots, m, \tag{80}$$

where Γ is adjacency matrix of the multicast network, and $\mathbf{D}(r_i)$ is an output matrix. For more details on how these matrices are constructed, we refer the interested reader to the reference [24].

A multicast problem has a network coding solution if and only if each matrix $\mathbf{M}(r_i)$ is non-singular. In [25], the author showed that for the *expanded transfer matrix* defined as

$$\mathbf{E}(r_i) = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{I} - \Gamma & \mathbf{D}(r_i) \end{bmatrix}, \quad i = 1, 2, \ldots, m, \tag{81}$$

it holds that $\det(\mathbf{M}(r_i)) = \pm \det(\mathbf{E}(r_i))$.

Some entries of the matrices Γ and $\mathbf{D}(r_i)$, $i = 1, 2, \ldots, m$, are unknowns. To obtain the actual transmissions on all the edges, it is necessary to replace those unknown entries with elements from $\mathbb{F}_q$ such that all matrices $\mathbf{E}(r_i)$, $i = 1, 2, \ldots, m$, have full rank. This is known as a simultaneous matrix completion problem and it is solved in [25] in polynomial time provided that $|\mathbb{F}_q| > m$.

Remark 13: The complexity of the algorithm proposed in [25] applied to our problem is $\mathcal{O}(m^4 \gamma \log(mN))$, where γ is the complexity of computing rank.

V. Randomized Algorithm

In this section we combine Algorithm 6 with the linear network coding scheme to produce a randomized solution to the optimization problem (13) of linear complexity (in number of users). First, note that Algorithm 6 is incremental by its nature, *i.e.*, in each iteration we update the rate vector by one symbol in $\mathbb{F}_q$. Say that user i updates its rate at round j of Algorithm 6. Along with the rate update, let user i transmit an appropriately chosen linear combination of its observations; using the notation from Section II, we have

$$v_i^{(j)} = \mathbf{b}_i^{(j)} \mathbf{A}_i \mathbf{w}, \tag{82}$$

where $\mathbf{b}_i^{(j)} \in \mathbb{F}_q^{\ell_i}$, is the vector of coefficients that lead to the optimal communication scheme. We note that those coefficients are not known a priori; they can be figured out by applying the algorithm proposed in Section IV only after the entire optimal *DE*-rate vector is recovered. For now, let us just assume that we have access to the vectors $\mathbf{b}_i^{(j)}$ for all iterations $j = 1, \ldots, \beta$, and for all users $i \in \mathcal{M}$ that are scheduled to update their communication rates. Later, we will use random linear network coding argument to relax these assumptions.

In the expression (82), let us define $\mathbf{u}^{(j)} \in \mathbb{F}_q^N$ as

$$\mathbf{u}^{(j)} \triangleq \mathbf{b}_i^{(j)} \mathbf{A}_i. \tag{83}$$

Then, we can write (82) as

$$v_i^{(j)} = \mathbf{u}^{(j)} \mathbf{w}. \tag{84}$$

By generating transmissions along with the rate updates, we can actually reduce the complexity of verifying whether the rate vector update still belongs to the polyhedron $P(f_\beta)$. This result is stated in the following theorem.

Theorem 4: Let the set $\mathcal{T}_j$ be defined as in (67). *Then,*

$$\begin{aligned} \mathcal{T}_j = \{ & i \in \mathcal{M} | \operatorname{rank}(\mathbf{A}_i \cup \mathbf{u}^{(1)} \cup \cdots \cup \mathbf{u}^{(j-1)}) \\ & > N - (\beta - j + 1)\}. \end{aligned} \tag{85}$$

The proof of Theorem 4 is provided in Appendix H.

Algorithm 8 Randomized Algorithm

```
Set $R_i = 0$, $\forall i \in \mathcal{M}$.
for $j = 1$ to $\beta$ do
    Find $\mathcal{T}_j$ as defined in (85).
    Find $i^* \in \mathcal{T}_j$ such that
        $i^* = \operatorname{argmin} \{ d_i(R_i + 1) | i \in \mathcal{T}_j \}$,   (86)
    where $d_i()$ is defined in (66).
    Let $i^*$ transmit, and create a transmission $v_{i^*}^{(j)}$ by creating a vector $\mathbf{b}_{i^*}^{(j)}$ uniformly at random over $\mathbb{F}_q^{\ell_{i^*}}$.
    Set $R_{i^*} = R_{i^*} + 1$.
end for
$\mathbf{R}^* = \mathbf{R}$ is an optimal rate vector w.r.t. the problem (64).
```

So far we have assumed that the vectors $\mathbf{u}^{(j)}$ are provided to us deterministically, and that they render optimal communication scheme. However, this assumption is unjustifiable since we saw in Section IV that in order to construct a deterministic communication scheme we need to know optimal *DE*-rate vector beforehand. To go around this problem we invoke a random linear network coding scheme. The basic idea behind the random linear network coding argument is that if user i is scheduled to transmit in round j, then we can choose vectors $\mathbf{b}_i^{(j)}$ in (82) uniformly at random over $\mathbb{F}_q^{\ell_i}$. The following lemma provides a relationship between probability of generating optimal transmissions and the field size q.

Lemma 6: For the random linear network coding scheme, the probability of choosing an optimal sequence of vectors $\mathbf{u}^{(j)}$, $j = 1, 2, \ldots, \beta$, is at least $(1 - \frac{m}{q})^\beta$.

The proof of Lemma 6 directly follows from [35]. The idea is to relate this problem to a multicast problem as in Section IV, while assuming that the optimal rates are given. Then, by randomly generating transmissions on each link, we obtain the exactly same formulation as in [35].

Putting all these results together, from Algorithm 6 we can devise its Randomized counterpart as follows (see Algorithm 8).

Remark 14: The complexity of Algorithm 8 is $\mathcal{O}(m\gamma N)$, where γ is the complexity of computing rank.

Remark 15: When β is not a feasible sum-rate w.r.t. the optimization problem (13), then after β iterations of Algorithm 8 there exists a user that cannot reconstruct all the packets. In other words

$$\exists i \in \mathcal{M}, \quad \text{s.t. } \operatorname{rank}\left(\mathbf{A}_i \cup \mathbf{u}^{(1)} \cup \cdots \cup \mathbf{u}^{(\beta)}\right) < N. \tag{87}$$

In order to solve the optimization problem (10), we can apply a binary search algorithm similar to Algorithm 4. Thus, the overall complexity of the proposed algorithm is $\mathcal{O}(m\gamma N \log N)$.

Example 4: Let us consider the same problem as in Example 3

$$\begin{aligned} \mathbf{x}_1 &= \begin{bmatrix} w_1 & w_2 \end{bmatrix}, \\ \mathbf{x}_2 &= \begin{bmatrix} w_2 & w_4 & w_5 & w_6 \end{bmatrix}, \\ \mathbf{x}_3 &= \begin{bmatrix} w_3 & w_4 & w_5 & w_6 \end{bmatrix}, \end{aligned} \tag{88}$$

where $w_i \in \mathbb{F}_q$, $i = 1, \dots, 6$, and q is some large prime number. For the uniform objective $\sum_{i=1}^{3} R_i \log R_i$ with a fixed sum-rate $\sum_{i=1}^{3} R_i = 5$, Algorithm 8 executes the following steps:

- Set $R_1 = R_2 = R_3 = 0$.
- $j = 1$: Updates of the rate vector $\mathbf{R}^*$ are selected according to the rule (76):
$$\text{argmin}\ \{R_i | i \in \mathcal{T}_1 = \{1,2,3\}\} = \{1,2,3\}, \tag{89}$$
User 1 transmit some random linear combination of its observation, say $v_1^{(1)} = w_1 + 7w_2$. Set
$$R_1 = R_1 + 1 = 1. \tag{90}$$
- $j = 2$: Vector $\mathbf{R}$ is updated according to the rule:
$$\text{argmin}\ \{R_i | i \in \mathcal{T}_2 = \{2,3\}\} = \{3\}. \tag{91}$$
User 3 transmit some random linear combination of its observation, say $v_3^{(2)} = w_3 + w_4 + 5w_5 + 11w_6$. Set
$$R_3 = R_3 + 1 = 1. \tag{92}$$
- $j = 3$: Vector $\mathbf{R}$ is updated according to the rule:
$$\text{argmin}\ \{R_i | i \in \mathcal{T}_3 = \{2,3\}\} = \{2\}. \tag{93}$$
User 2 transmit some random linear combination of its observation, say $v_2^{(3)} = 4w_2 + 3w_4 + 13w_5 + 8w_6$. Set
$$R_2 = R_2 + 1 = 1. \tag{94}$$
- $j = 4$: Vector $\mathbf{R}$ is updated according to the rule:
$$\text{argmin}\ \{R_i | i \in \mathcal{T}_4 = \{2,3\}\} = \{3\}. \tag{95}$$
User 3 transmit some random linear combination of its observation, say $v_3^{(4)} = 9w_3 + 5w_4 + 14w_5 + 17w_6$. Set
$$R_3 = R_3 + 1 = 2. \tag{96}$$
- $j = 5$: Vector $\mathbf{R}$ is updated according to the rule:
$$\text{argmin}\ \{R_i | i \in \mathcal{T}_5 = \{2,3\}\} = \{2\}. \tag{97}$$
User 2 transmit some random linear combination of its observation, say $v_2^{(5)} = 11w_2 + 2w_4 + 18w_5 + 6w_6$. Set
$$R_2 = R_2 + 1 = 2. \tag{98}$$
- $\mathbf{R}^* = \mathbf{R}$ is an optimal *DE*-rate vector w.r.t. the uniform objective and the condition $R(\mathcal{M}) = 5$.

It can be verified that after this round of communication all the users are able to recover the file.

VI. Introducing Capacity Constraints

In this section we explore a data exchange problem where the transmissions of each user can be further restricted. For instance, we can limit the total number of packets sent by each user. Say that user i is not allowed to transmit more than c_i packets in $\mathbb{F}_q$. Then, optimization problem (10) becomes

$$\min_{\beta \in \mathbb{Z}_+} h(\beta), \tag{99}$$

where $h(\beta)$ can be obtained from (25) by adding capacity constraints

$$h(\beta) = \min_{\mathbf{R} \in \mathbb{Z}^m} \sum_{i=1}^{m} \varphi_i(R_i),$$
$$\text{s.t.,} \quad \mathbf{R} \in B(g_\beta), \quad R_i \le c_i, \quad \forall i \in \mathcal{M}, \tag{100}$$

provided that $g_\beta(\mathcal{M}) = \beta$. We also assume that the capacity vector $\mathbf{c}$ is feasible, *i.e.*, there exists a rate vector $\mathbf{R} \in B(g_\beta)$ such that the capacity constraints in (100) are satisfied.

In Section III we pointed out that the optimality of all the algorithms we studied is guaranteed due to the fact that the constraint set of the problem (13) constitutes a base polyhedron of a submodular function. In this section we show that by adding individual capacity constraints, the constraint set in (13) also forms a base polyhedron of a submodular function. This implies that in such a case we can still apply every algorithm developed so far in order to obtain an optimal *DE*-rate vector.

We begin our analysis by defining the restriction of a submodular function (see [33] for the reference).

Definition 5: For a submodular function $g_\beta : 2^{\mathcal{M}} \to \mathbb{Z}$, and a vector $\mathbf{c} \in \mathbb{Z}^m$, define a function $g_\beta^{\mathbf{c}} : 2^{\mathcal{M}} \to \mathbb{Z}$ by

$$g_\beta^{\mathbf{c}}(\mathcal{S}) \triangleq \min_{\mathcal{V} \subseteq \mathcal{S}} \left\{ g_\beta(\mathcal{V}) + c(\mathcal{S} \setminus \mathcal{V}) \right\}, \quad \forall \mathcal{S} \subseteq \mathcal{M}. \tag{101}$$

The submodular function $g_\beta^{\mathbf{c}}$ is called the *restriction of g_β by vector* $\mathbf{c}$.

Theorem 5 [33, Th. 8.2.1]: Let $g_\beta^{\mathbf{c}}$ be restriction of a submodular function g_β by vector $\mathbf{c}$. *Then, $g_\beta^{\mathbf{c}}$ is submodular.*

Theorem 6: For a submodular function g_β defined in (22) *and a feasible capacity vector* $\mathbf{c}$ *w.r.t. problem* (100)*, the base polyhedron $B(g_\beta^{\mathbf{c}})$ of the restriction of g_β by* $\mathbf{c}$*, is given by*

$$B(g_\beta^{\mathbf{c}}) = \left\{ \mathbf{R} | \mathbf{R} \in B(g_\beta), \ R_i \le c_i, \ \forall i \in \mathcal{M} \right\}, \tag{102}$$

provided that the sum-rate β and the capacity vector $\mathbf{c}$ *are feasible w.r.t. the optimization problem* (13)*.*

The proof of Theorem 6 is provided in Appendix I.

From Theorem 6 it follows that the constraint set of (100) forms a submodular polyhedron $B(g_\beta^{\mathbf{c}})$, which further implies that all the algorithms developed so far can be applied to obtain an optimal *DE*-rate vector. For instance, with capacity constraints, Step 4 of Algorithm 1 becomes

$$R^*_{j(i)} = \min\{c_{j(i)}, g_\beta(\{j(1), j(2), \dots, j(i)\}) - g_\beta(\{j(1), j(2), \dots, j(i-1)\})\}. \tag{103}$$

This modification propagates to Algorithm 2 as well. Similarly, at iteration j, Step 4 of Algorithms 7 and 8 is modified as follows

$$i^* = \text{argmin}\{d_i(R_i^* + 1) | i \in \mathcal{T}_j, \ \text{s.t.,} \ R_i^* + 1 \le c_i\}. \tag{104}$$

Remark 16: If the capacity vector $\mathbf{c}$ was not feasible w.r.t. problem (100), then all algorithms considered so far would terminate before reaching a sum-rate equal to β.

Example 5: Let us consider the same problem as in Example 2

$$\begin{aligned}\mathbf{x}_1 &= \begin{bmatrix} w_1 & w_2 \end{bmatrix},\\ \mathbf{x}_2 &= \begin{bmatrix} w_2 & w_4 & w_5 & w_6 \end{bmatrix},\\ \mathbf{x}_3 &= \begin{bmatrix} w_3 & w_4 & w_5 & w_6 \end{bmatrix},\end{aligned} \tag{105}$$

where $w_i \in \mathbb{F}_q$. For the cost function $R_1 + 3R_2 + 2R_3$, and the sum-rate $\beta = 5$, let the capacity constraints be $c_i \leq 2$, $i = 1, 2, 3$. Then, by applying Algorithm 2 with the modification (103), we obtain the following result

$$R_1^* = \min\{f_5(\{1\}), c_1\} = 1, \tag{106}$$

$$R_3^* = \min\{\min\{f_5(\{1,3\}) - R_1^*, f_5(\{2\})\}, c_3\} = 2, \tag{107}$$

$$\begin{aligned}R_2^* = \min\{\min\{&f_5(\{1,2,3\}) - R_1^* - R_3^*, f_5(\{1,3\}) - R_1^*,\\ &f_5(\{2,3\}) - R_3^*, f_5(\{2\})\}, c_2\} = 2.\end{aligned} \tag{108}$$

Without capacity constraints, as it was the case in Example 2, user 3 would transmit 3 packets in $\mathbb{F}_q$.

VII. Conclusion

In this work we addressed the problem of the data exchange, where each user has some side-information about the file, and is interested in recovering it. We assumed that the users are allowed to "talk" to each other over a noiseless broadcast channel. For the case when the side information is in the form of the linearly coded packets, we provided deterministic and randomized polynomial time algorithms for finding an optimal communication scheme, w.r.t. a separable convex communication cost. For the deterministic algorithm, we proposed two methods to determine how much should each user transmit in an optimal scheme. The first one was based on minimizing a submodular function, while the second technique was based on subgradient methods. The latter technique also provides an alternative solution to the Edmonds' algorithm when the underlying set function is intersecting submodular and over integers.

Appendix A
Proof of Theorem 1

In order for each user in $\mathcal{M}$ to reconstruct the file, it is necessary for all of them to receive a sufficient number of linear combinations over $\mathbb{F}_q$ so that the observation rank of each user is full. For instance, in order for user 1 to recover all N packets of the file, it is sufficient for him to select $N - \ell_1$ linear equations from the remaining $m - 1$ users. In this case, user 2 can send to user 1

$$R_2 = \text{rank}\left(\mathbf{A}_{\{1,2\}}\right) - \text{rank}\left(\mathbf{A}_{\{1\}}\right) \tag{109}$$

of its linear equations, after which user 1's observation rank will be $\text{rank}\left(\mathbf{A}_{\{1,2\}}\right)$. Following this procedure, we have that the number of linear equations sent by the remaining users is

$$R_3 = \text{rank}\left(\mathbf{A}_{\{1,2,3\}}\right) - \text{rank}\left(\mathbf{A}_{\{1,2\}}\right) \tag{110}$$

$$\vdots$$

$$\begin{aligned}R_m &= \text{rank}\left(\mathbf{A}_{\mathcal{M}}\right) - \text{rank}\left(\mathbf{A}_{\mathcal{M}\setminus\{m\}}\right)\\ &= N - \text{rank}\left(\mathbf{A}_{\mathcal{M}\setminus\{m\}}\right).\end{aligned} \tag{111}$$

Observe that the number of linear equations each user sends depends upon the ordering of users in equations (109) through (111). Let $j(2), \ldots, j(m)$ be any ordering of $2, \ldots, m$. Then, by applying the same approach as above, we obtain other feasible rate tuples

$$R_{j(2)} = \text{rank}\left(\mathbf{A}_{\{1,j(2)\}}\right) - \text{rank}\left(\mathbf{A}_{\{1\}}\right) \tag{112}$$

$$R_{j(3)} = \text{rank}\left(\mathbf{A}_{\{1,j(2),j(3)\}}\right) - \text{rank}\left(\mathbf{A}_{\{1,j(2)\}}\right) \tag{113}$$

$$\vdots$$

$$R_{j(m)} = N - \text{rank}\left(\mathbf{A}_{\mathcal{M}\setminus\{j(m)\}}\right). \tag{114}$$

From (112)-(114), observe that

$$\sum_{i=t}^{m} R_{j(i)} = N - \text{rank}\left(\mathbf{A}_{\{1,j(2),\ldots,j(t-1)\}}\right), \tag{115}$$

for $t = 2, \ldots, m$. Note that (112) through (114) constitute a feasible solution provided by Edmonds' algorithm (see Algorithm 1) for a fully submodular function $g_\beta(\mathcal{S}) = \beta - N + \text{rank}(\mathbf{A}_{\mathcal{S}\cup\{1\}})$, $\mathcal{S} \subseteq \{2, \ldots, m\}$, for $\beta = N - \text{rank}(\mathbf{A}_{\{1\}})$. Hence, it immediately follows that any rate vector $\mathbf{R}$ obtained by applying (112) through (114) for any ordering of $2, \ldots, m$, should satisfy

$$\sum_{i\in\mathcal{S}} R_i \leq \text{rank}\left(\mathbf{A}_{\mathcal{S}\cup\{1\}}\right) - \text{rank}\left(\mathbf{A}_{\{1\}}\right),\ \forall \mathcal{S} \subseteq \mathcal{M} \text{ s.t. } \{1\} \notin \mathcal{S}, \tag{116}$$

which is equivalent to

$$\sum_{i\in\mathcal{S}} R_i \geq N - \text{rank}\left(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}\right), \forall \mathcal{S} \subseteq \mathcal{M} \quad \text{s.t.} \quad \{1\} \notin \mathcal{S}. \tag{117}$$

Let us denote the above region by $\mathcal{R}_1$. Similarly, for users 2 through m, we can define regions $\mathcal{R}_2, \ldots, \mathcal{R}_m$. Let us denote by $\mathcal{R}_{int}$ the set of all integer vectors $\mathbb{Z}^m$ that belong to the cut-set region $\mathcal{R}$ defined in (5). Then, it is not hard to show that

$$\mathcal{R}_{int} = \mathcal{R}_1 \cap \mathcal{R}_2 \cap \cdots \cap \mathcal{R}_m. \tag{118}$$

From the discussion above, we know that if $\mathbf{R} \in \mathcal{R}_{int}$, then it is sufficient for user i to send R_i linear equations separately to all the users, which makes the total of $(m-1)R_i$ equations over $\mathbb{F}_q$ sent by user i. The key property of the linear network codes is that there exists one set of R_i linear equations that user i can broadcast and simultaneously satisfy demands of all the remaining users in $\mathcal{M}$, provided that the field size $|\mathbb{F}_q|$ is large enough [36]. Hence, every rate tuple that belongs to $\mathcal{R}_{int}$ can be achieved via linear network coding.

Appendix B
Proof of Theorem 3

Consider two feasible sum-rates β_1 and β_2 w.r.t. the problem (13). We show that for any $\lambda \in [0, 1]$ such that $\lambda\beta_1 + (1-\lambda)\beta_2 \in \mathbb{Z}_+$ it holds that $h(\lambda\beta_1 + (1-\lambda)\beta_2) \leq \lambda h(\beta_1) + (1-\lambda)h(\beta_2)$. Let $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(2)}$ be the optimal rate tuples w.r.t. $h(\beta_1)$ and $h(\beta_2)$, respectively. Then,

$$\begin{aligned}\lambda h(\beta_1) &+ (1-\lambda)h(\beta_2)\\ &= \sum_{i=1}^{m}\Big(\lambda\varphi_i(R_i^{(1)}) + (1-\lambda)\varphi_i(R_i^{(2)})\Big) \quad (119)\\ &\overset{(a)}{\geq} \sum_{i=1}^{m}\varphi_i(\lambda R_i^{(1)} + (1-\lambda)R_i^{(2)}) = \sum_{i=1}^{m}\varphi_i(R_i^{(\lambda)}), \quad (120)\end{aligned}$$

where (a) follows from the convexity of φ_i, $\forall i \in \mathcal{M}$, and $\mathbf{R}^{(\lambda)} \triangleq \lambda\mathbf{R}^{(1)} + (1-\lambda)\mathbf{R}^{(2)}$. Now, we show that $\mathbf{R}^{(\lambda)}$ is a feasible *DE*-rate vector for the problem (13) when $\beta = \lambda\beta_1 + (1-\lambda)\beta_2$.

Since $R^{(1)}(\mathcal{M}) = \beta_1$ and $R^{(2)}(\mathcal{M}) = \beta_2$, it follows that

$$\begin{aligned}R^{(\lambda)}(\mathcal{M}) &= \lambda R^{(1)}(\mathcal{M}) + (1-\lambda)R^{(2)}(\mathcal{M})\\ &= \lambda\beta_1 + (1-\lambda)\beta_2. \quad (121)\end{aligned}$$

From

$$R^{(i)}(\mathcal{S}) \geq N - \text{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}),\ \forall\mathcal{S}\subset\mathcal{M},\quad i = 1,2, \quad (122)$$

we have

$$\begin{aligned}R^{(\lambda)}(\mathcal{S}) &= \lambda R^{(1)}(\mathcal{S}) + (1-\lambda)R^{(2)}(\mathcal{S})\\ &\geq N - \text{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}),\quad \forall\mathcal{S}\subset\mathcal{M}. \quad (123)\end{aligned}$$

From (121) and (123) it follows that $\mathbf{R}^{(\lambda)}$ is a feasible *DE*-rate vector w.r.t. optimization problem (13) when $\beta = \lambda\beta_1 + (1-\lambda)\beta_2$. Therefore, $\sum_{i=1}^{m}\varphi_i(R_i^{(\lambda)}) \geq h(\lambda\beta_1 + (1-\lambda)\beta_2)$. Hence, from (120), it follows that

$$h(\lambda\beta_1 + (1-\lambda)\beta_2) \leq \lambda h(\beta_1) + (1-\lambda)h(\beta_2), \quad (124)$$

which completes the proof of Theorem 3.

Appendix C
Proof of Lemma 1

When $\mathcal{S}\cap\mathcal{T}\neq\emptyset$, the following inequality holds due to the submodularity of the rank function

$$\begin{aligned}f_\beta(\mathcal{S}) &+ f_\beta(\mathcal{T})\\ &= \text{rank}(\mathbf{A}_\mathcal{S}) + \text{rank}(\mathbf{A}_\mathcal{T}) - 2(N-\beta) \quad (125)\\ &\geq \text{rank}(\mathbf{A}_{\mathcal{S}\cup\mathcal{T}}) + \text{rank}(\mathbf{A}_{\mathcal{S}\cap\mathcal{T}}) - 2(N-\beta) \quad (126)\\ &= f_\beta(\mathcal{S}\cup\mathcal{T}) + f_\beta(\mathcal{S}\cap\mathcal{T}). \quad (127)\end{aligned}$$

To show that the function f_β is submodular when $\beta \geq N$, it is only left to consider the case $\mathcal{S}\cap\mathcal{T} = \emptyset$. Since $f_\beta(\emptyset) = 0$, we have

$$\begin{aligned}f_\beta(\mathcal{S}) &+ f_\beta(\mathcal{T})\\ &= \text{rank}(\mathbf{A}_\mathcal{S}) + \text{rank}(\mathbf{A}_\mathcal{T}) - 2(N-\beta) \quad (128)\\ &\geq \text{rank}(\mathbf{A}_{\mathcal{S}\cup\mathcal{T}}) - (N-\beta) = f_\beta(\mathcal{S}\cup\mathcal{T}). \quad (129)\end{aligned}$$

The inequality in (129) directly follows from the submodularity of the rank function

$$\begin{aligned}\text{rank}(\mathbf{A}_\mathcal{S}) + \text{rank}(\mathbf{A}_\mathcal{T}) - \text{rank}(\mathbf{A}_{\mathcal{S}\cup\mathcal{T}}) &\geq 0 \quad (130)\\ &\geq N-\beta. \quad (131)\end{aligned}$$

This completes the proof of Lemma 1.

Appendix D
Proof of Lemma 2

Let us construct the set function $y : 2^{\{1,2,\ldots,i\}} \to \mathbb{Z}$ as follows

$$y(\mathcal{S}) = \begin{cases}0 & \text{if } \mathcal{S} = \emptyset,\\ f_\beta(\mathcal{S}) & \text{if } i\in\mathcal{S},\\ f_\beta(\mathcal{S}\cup\{i\}) & \text{if } i\notin\mathcal{S}.\end{cases} \quad (132)$$

First, we show that $\mathcal{R}^{(i)} = P(y)$. Let $\mathbf{R}\in P(y)$. Then, for any $\mathcal{S}\subseteq\{1,2,\ldots,i-1\}$, it follows that

$$R(\mathcal{S}\cup\{i\}) \leq y(\mathcal{S}\cup\{i\}) = f_\beta(\mathcal{S}\cup\{i\}). \quad (133)$$

Therefore, $\mathbf{R}\in\mathcal{R}^{(i)}$.

Now, let $\mathbf{R}\in\mathcal{R}^{(i)}$. From (39) we have

$$R(\mathcal{S}\cup\{i\}) \leq f_\beta(\mathcal{S}\cup\{i\}) = y(\mathcal{S}\cup\{i\}), \forall\mathcal{S}\subseteq\{1,2,\ldots,i-1\}. \quad (134)$$

Since the rate vector is non-negative, (134) implies that

$$R(\mathcal{S}) \leq f_\beta(\mathcal{S}\cup\{i\}) = y(\mathcal{S}),\quad \forall\mathcal{S}\subseteq\{1,2,\ldots,i-1\}. \quad (135)$$

From (134) and (135) it follows that $\mathbf{R}\in P(y)$. Hence, $\mathcal{R}^{(i)} = P(y)$.

Next, we show that function y is fully submodular. For any $\mathcal{S},\mathcal{T}\subseteq\{1,2,\ldots,i\}$, let us consider the following 3 cases

Case 1: $i\in\mathcal{S}$, $i\notin\mathcal{T}$,

$$\begin{aligned}y(\mathcal{S}) &+ y(\mathcal{T})\\ &= f_\beta(\mathcal{S}) + f_\beta(\mathcal{T}\cup\{i\}) \quad (136)\\ &\overset{(a)}{\geq} f_\beta(\mathcal{S}\cup\mathcal{T}) + f_\beta((\mathcal{S}\cap\mathcal{T})\cup\{i\}) \quad (137)\\ &= y(\mathcal{S}\cup\mathcal{T}) + y(\mathcal{S}\cap\mathcal{T}), \quad (138)\end{aligned}$$

where (a) is due to intersecting submodularity of function f_β.

Case 2: $i\notin\mathcal{S}$, $i\notin\mathcal{T}$,

$$\begin{aligned}y(\mathcal{S}) &+ y(\mathcal{T})\\ &= f_\beta(\mathcal{S}\cup\{i\}) + f_\beta(\mathcal{T}\cup\{i\}) \quad (139)\\ &\geq f_\beta(\mathcal{S}\cup\mathcal{T}\cup\{i\}) + f_\beta((\mathcal{S}\cap\mathcal{T})\cup\{i\}) \quad (140)\\ &= y(\mathcal{S}\cup\mathcal{T}) + y(\mathcal{S}\cap\mathcal{T}). \quad (141)\end{aligned}$$

Case 3: $i\in\mathcal{S}$, $i\in\mathcal{T}$,

$$\begin{aligned}y(\mathcal{S}) + y(\mathcal{T}) &= f_\beta(\mathcal{S}) + f_\beta(\mathcal{T}) \quad (142)\\ &\geq f_\beta(\mathcal{S}\cup\mathcal{T}) + f_\beta(\mathcal{S}\cap\mathcal{T}) \quad (143)\\ &= y(\mathcal{S}\cup\mathcal{T}) + y(\mathcal{S}\cap\mathcal{T}). \quad (144)\end{aligned}$$

Therefore, function y is indeed fully submodular. Hence, problem (46) is a linear optimization problem over a submodular polyhedron, and it can be solved by applying algorithm similar to Algorithm 1 (see reference [26]). The only difference is that in this case, the weights in Step 1 of Algorithm 1 should be ordered in a non-increasing order.

If $\lambda_{t(1)} \leq 1$, then

$$\begin{aligned}\tilde{R}_i[j] &= y(\{i\}) = f_\beta(\{i\}), \quad (145)\\ \tilde{R}_{t(k)}[j] &= y(\mathcal{S}_{t(k)}\cup\{i\}) - y(\mathcal{S}_{t(k-1)}\cup\{i\}) \quad (146)\\ &= f_\beta(\mathcal{S}_{t(k)}\cup\{i\}) - \sum_{u=1}^{k-1}\tilde{R}_{t(u)}[j] - \tilde{R}_i[j], \quad (147)\end{aligned}$$

for $k = 1,2,\ldots,i-1$.

If for some $r \in \{1, 2, \ldots, i-2\}$, $\lambda_{t(r)} \geq 1 \geq \lambda_{t(r+1)}$, then

$$\tilde{R}_{t(k)}[j] = y(\mathcal{S}_{t(k)}) - y(\mathcal{S}_{t(k-1)}) \tag{148}$$

$$= f_\beta(\mathcal{S}_{t(k)} \cup \{i\}) - \sum_{u=1}^{k-1} \tilde{R}_{t(u)}[j], \; t(k) \in \{t(1), \ldots, t(r)\}, \tag{149}$$

$$\tilde{R}_i[j] = y(\mathcal{S}_{t(r)} \cup \{i\}) - y(\mathcal{S}_{t(r)}) = 0, \tag{150}$$

$$\tilde{R}_{t(k)}[j] = y(\mathcal{S}_{t(k)} \cup \{i\}) - y(\mathcal{S}_{t(k-1)} \cup \{i\}) \tag{151}$$

$$= f_\beta(\mathcal{S}_{t(k)} \cup \{i\}) - \sum_{u=1}^{k-1} \tilde{R}_{t(u)}[j], \quad t(k) \in \{t(r+1), \ldots, t(i-1)\}, \tag{152}$$

for $k = 1, 2, \ldots, i-1$.

If $\lambda_{t(i-1)} > 1$, then

$$\tilde{R}_{t(k)}[j] = y(\mathcal{S}_{t(k)}) - y(\mathcal{S}_{t(k-1)})$$

$$= f_\beta(\mathcal{S}_{t(k)} \cup \{i\}) - \sum_{u=1}^{k-1} \tilde{R}_{t(u)}[j], \tag{153}$$

$$\tilde{R}_i[j] = y(\mathcal{S}_{t(i-1)} \cup \{i\}) - y(\mathcal{S}_{t(i-1)}) = 0, \tag{154}$$

for $k = 1, 2, \ldots, i-1$. This completes the proof of Lemma 2.

Appendix E
Proof of Lemma 3

After $j+1$ iterations of the subgradient algorithm, the Euclidian distance between $\boldsymbol{\lambda}[j+1]$ and a minimizer $\boldsymbol{\lambda}^*$ of the dual function δ, can be bounded as follows

$$\sum_{k=1}^{i-1} (\lambda_k[j+1] - \lambda_k^*)^2$$

$$= \sum_{k=1}^{i-1} \left(\left\{ \lambda_k[j] - \theta_j (\tilde{R}_k[j] - R_k^*) \right\}_+ - \lambda_k^* \right)^2 \tag{155}$$

$$\leq \sum_{k=1}^{i-1} \left(\lambda_k[j] - \theta_j (\tilde{R}_k[j] - R_k^*) - \lambda_k^* \right)^2 \tag{156}$$

$$= \sum_{k=1}^{i-1} \left(\lambda_k[j] - \lambda_k^* \right)^2 - 2\theta_j \sum_{k=1}^{i-1} (\tilde{R}_k[j] - R_k^*)(\lambda_k[j] - \lambda_k^*) + \theta_j^2 \sum_{k=1}^{i-1} \left(\tilde{R}_k[j] - R_k^* \right)^2 \tag{157}$$

$$\leq \sum_{k=1}^{i-1} \left(\lambda_k[j] - \lambda_k^* \right)^2 - 2\theta_j \left(\delta(\boldsymbol{\lambda}[j]) - \delta(\boldsymbol{\lambda}^*) \right) + \theta_j^2 \sum_{k=1}^{i-1} \left(\tilde{R}_k[j] - R_k^* \right)^2, \tag{158}$$

where the last inequality is due to the convexity of function $\delta(\boldsymbol{\lambda})$, *i.e.*,

$$\delta(\boldsymbol{\lambda}[j]) - \delta(\boldsymbol{\lambda}^*) \leq \sum_{k=1}^{i-1} (\tilde{R}_k[j] - R_k^*)(\lambda_k[j] - \lambda_k^*), \tag{159}$$

since $\tilde{R}_k[j] - R_k^*$ is a partial derivative of $\delta(\boldsymbol{\lambda}[j])$ at coordinate $\lambda_k[j]$, $k = 1, 2, \ldots, i-1$. Summing both sides of inequality (158) over j from 0 to $l-1$, we obtain

$$\sum_{k=1}^{i-1} (\lambda_k[l] - \lambda_k^*)^2 \leq \sum_{k=1}^{i-1} (\lambda_k[0] - \lambda_k^*)^2 - 2 \sum_{j=0}^{l-1} \theta_j \left(\delta(\boldsymbol{\lambda}[j]) - \delta(\boldsymbol{\lambda}^*) \right) + \sum_{j=0}^{l-1} \theta_j^2 \sum_{k=1}^{i-1} \left(\tilde{R}_k[j] - R_k^* \right)^2. \tag{160}$$

Therefore,

$$2 \sum_{j=0}^{l-1} \theta_j \left(\delta(\boldsymbol{\lambda}[j]) - \delta(\boldsymbol{\lambda}^*) \right) \leq \sum_{k=1}^{i-1} (\lambda_k[0] - \lambda_k^*)^2 + \sum_{j=0}^{l-1} \theta_j^2 \sum_{k=1}^{i-1} \left(\tilde{R}_k[j] - R_k^* \right)^2. \tag{161}$$

Since,

$$\sum_{j=0}^{l-1} \theta_j \left(\delta(\boldsymbol{\lambda}[j]) - \delta(\boldsymbol{\lambda}^*) \right) \geq \sum_{j=0}^{l-1} \theta_j \min_{j \in \{0,1,\ldots,l-1\}} \left(\delta(\boldsymbol{\lambda}[j]) - \delta(\boldsymbol{\lambda}^*) \right), \tag{162}$$

from (161) and (49) we obtain

$$\delta(\tilde{\boldsymbol{\lambda}}[l-1]) - \delta(\boldsymbol{\lambda}^*) = \min_{j \in \{0,1,\ldots,l-1\}} \delta(\boldsymbol{\lambda}[j]) - \delta(\boldsymbol{\lambda}^*)$$

$$\leq \frac{\sum_{k=1}^{i-1} (\lambda_k[0] - \lambda_k^*)^2}{2 \sum_{j=0}^{l-1} \theta_j} + \frac{\sum_{j=0}^{l-1} \theta_j^2 \sum_{k=1}^{i-1} \left(\tilde{R}_k[j] - R_k^* \right)^2}{2 \sum_{j=0}^{l-1} \theta_j} \tag{163}$$

$$\leq \frac{\sum_{k=1}^{i-1} (\lambda_k[0] - \lambda_k^*)^2}{2 \sum_{j=0}^{l-1} \theta_j} + \frac{\sum_{j=0}^{l-1} \theta_j^2 \left(\sum_{k=1}^{i-1} \left(\tilde{R}_k[j] \right)^2 + \sum_{k=1}^{i-1} \left(R_k^* \right)^2 \right)}{2 \sum_{j=0}^{l-1} \theta_j} \tag{164}$$

$$\leq \frac{\sum_{k=1}^{i-1} (\lambda_k[0] - \lambda_k^*)^2}{2 \sum_{j=0}^{l-1} \theta_j} + \frac{\sum_{j=0}^{l-1} \theta_j^2 \left(\left(\sum_{k=1}^{i-1} \tilde{R}_k[j] \right)^2 + \left(\sum_{k=1}^{i-1} R_k^* \right)^2 \right)}{2 \sum_{j=0}^{l-1} \theta_j} \tag{165}$$

$$\leq \frac{\sum_{k=1}^{i-1} (\lambda_k[0] - \lambda_k^*)^2 + 2N^2 \sum_{j=0}^{l-1} \theta_j^2}{2 \sum_{j=0}^{l-1} \theta_j}, \tag{166}$$

where the last inequality holds because $R(\mathcal{M}) \leq f_\beta(\mathcal{M}) \leq N$ for any achievable *DE*-rate vector $\mathbf{R}$. From (166), it immediately follows that

$$\delta(\tilde{\boldsymbol{\lambda}}[l-1]) - \delta(\boldsymbol{\lambda}^*) \leq \frac{\left(\sum_{k=1}^{i-1} \lambda_k[0]\right)^2 + \left(\sum_{k=1}^{i-1} \lambda_k^*\right)^2 + 2N^2 \sum_{j=0}^{l-1} \theta_j^2}{2\sum_{j=0}^{l-1} \theta_j}, \quad (167)$$

which completes the proof of Lemma 3.

Appendix F Proof of Lemma 4

For a minimizer $\boldsymbol{\lambda}^*$ of a dual function δ, let us denote by $\tilde{\mathbf{R}}$ an optimal solution of the problem (41) obtained by applying Lemma 2. Since $\sum_{k=1}^{i} \tilde{R}_k = f_\beta(\{1, 2, \ldots, i\})$, and $\sum_{k=1}^{i} R_k^* \leq f_\beta(\{1, 2, \ldots, i\})$, it follows that

$$\sum_{k=1}^{i-1} \tilde{R}_k - R_k^* \geq R_i^* - \tilde{R}_i. \quad (168)$$

By the formulation of the optimization problem (38), the minimum value of the dual function δ is R_i^*. Therefore,

$$\sum_{k=1}^{i-1} \lambda_k^* (\tilde{R}_k - R_k^*) = R_i^* - \tilde{R}_i. \quad (169)$$

From Theorem 2, it follows that

$$\sum_{k=1}^{i} R_i^* = \min_{\mathcal{P} \in \boldsymbol{\mathcal{P}}(\{1,\ldots,i\})} \sum_{\mathcal{S} \in \mathcal{P}} f_\beta(\mathcal{S}), \quad (170)$$

where $\boldsymbol{\mathcal{P}}(\{1, \ldots, i\})$ is the set of all partitions of the set $1, \ldots, i$. Let us denote by $\mathcal{S}_i^*$, a set that belongs to an optimal partitioning $\mathcal{P}^*$ w.r.t. problem (170) such that $i \in \mathcal{S}_i^*$. Then,

$$\sum_{k \in \mathcal{S}_i^*} R_k^* = f_\beta(\mathcal{S}_i^*). \quad (171)$$

Now, let us select $\boldsymbol{\lambda}^*$ as follows

$$\lambda_k^* = \begin{cases} 1 & \text{if } k \in \mathcal{S}_i^*, \\ 0 & \text{otherwise.} \end{cases} \quad (172)$$

To verify that this choice of $\boldsymbol{\lambda}^*$ is indeed a dual optimal solution, note that from Lemma 2, we have

$$\sum_{k \in \mathcal{S}_i^*} \tilde{R}_k = f_\beta(\mathcal{S}_i^*). \quad (173)$$

Therefore,

$$\sum_{k \in \mathcal{S}_i^*} \tilde{R}_k - R_k^* = 0. \quad (174)$$

From (172) and (174), it follows that

$$\sum_{k \in \{1, \ldots i-1\} \setminus \mathcal{S}_i^*} \lambda_k^* (\tilde{R}_k - R_k^*) = R_i^* - \tilde{R}_i + \sum_{k \in \mathcal{S}_i^* \setminus \{i\}} \lambda_k^* (R_k^* - \tilde{R}_k). \quad (175)$$

This is consistent with (169), and hence, $\boldsymbol{\lambda}^*$ is indeed a dual optimal solution. Therefore,

$$\sum_{k=1}^{i-1} \lambda_k^* \leq i - 1 \leq m. \quad (176)$$

This completes the proof of Lemma 4

Appendix G Proof of Lemma 5

By Lemma 1 we know that set functions f_N and f_{N+1}, defined in (16), are fully submodular,

$$f_N(\mathcal{S}) = \begin{cases} \text{rank}(\mathbf{A}_{\mathcal{S}}) & \text{if } \emptyset \neq \mathcal{S} \subseteq \mathcal{M}, \\ 0 & \text{if } \mathcal{S} = \emptyset. \end{cases} \quad (177)$$

$$f_{N+1}(\mathcal{S}) = \begin{cases} 1 + \text{rank}(\mathbf{A}_{\mathcal{S}}) & \text{if } \emptyset \neq \mathcal{S} \subseteq \mathcal{M}, \\ 0 & \text{if } \mathcal{S} = \emptyset. \end{cases} \quad (178)$$

Let us denote by $\mathbf{R}^*$ an optimal vector obtained by applying Algorithm 6 for $\beta = N$. Edmonds' algorithm implies that all faces of the submodular polyhedron $P(f_N)$ are achievable, *i.e.*, for any $\mathcal{S} \subseteq \mathcal{M}$, there exists a rate vector $\mathbf{R}$ such that $R(\mathcal{S}) = g_N(\mathcal{S}) = f_N(\mathcal{S})$, where the second equality is due to submodularity of f_N. Comparing f_N and f_{N+1}, we see that all "faces" of polyhedron $P(f_{N+1})$ expended by 1 compared to polyhedron $P(f_N)$ (and they are all achievable). Hence, while applying Algorithm 6 for $\beta = N + 1$, we can see that the optimal rate vector $\tilde{\mathbf{R}}$ will differ from $\mathbf{R}^*$ in one coordinate. Let

$$j^* = \text{argmin}\, \{d_i(R_i^* + 1) | \mathbf{R}^* + \mathbf{e}(i) \in P(f_{N+1})\}. \quad (179)$$

Then,

$$\tilde{R}_i = \begin{cases} R_i^* + 1 & \text{if } i = j^* \\ R_i^* & \text{otherwise.} \end{cases} \quad (180)$$

Evaluating costs for $\beta = N$ and $\beta = N + 1$, we obtain

$$h(N) = \sum_{i=1}^{m} \varphi_i(R_i^*) = \sum_{i \neq j^*} \varphi_i(R_i^*) + \varphi_{j^*}(R_{j^*}^*). \quad (181)$$

$$h(N+1) = \sum_{i=1}^{m} \varphi_i(\tilde{R}_i) = \sum_{i \neq j^*} \varphi_i(R_i^*) + \varphi_{j^*}(R_{j^*}^* + 1). \quad (182)$$

Comparing (181) and (182), we conclude that $h(N) \leq h(N + 1)$ since φ_{j^*} is a non-decreasing function. Since h is a convex function (see Theorem 3), it immediately follows that $\beta^* \leq N$. This completes the proof of Lemma 5.

Appendix H Proof of Theorem 4

Let us start by considering round $j = 1$ of Algorithm 6. All rates are set to zero, *i.e.*, $R_i^* = 0$, $i = 1, \ldots, m$. To check whether user i belongs to set $\mathcal{T}_1$, we need to verify whether its update belongs to polyhedron $P(f_\beta)$

$$R^*(S) + 1 \leq f_\beta(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \quad \text{s.t.,} \quad i \in \mathcal{S}, \quad (183)$$

where f_β is defined in (16). Since $\mathbf{R}^*$ is a zero vector, we can write the condition (183) as

$$1 \leq \beta - N + \text{rank}(\mathbf{A}_\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \quad \text{s.t.} \quad i \in \mathcal{S}, \tag{184}$$

which is equivalent to

$$1 \leq \min_{i \in \mathcal{S} \subseteq \mathcal{M}} \{\beta - N + \text{rank}(\mathbf{A}_\mathcal{S})\}. \tag{185}$$

It is easy to see that $\mathcal{S} = \{i\}$ is the minimizer of the above problem. Hence, $i \in \mathcal{T}_1$ if

$$\text{rank}(\mathbf{A}_i) > N - \beta, \tag{186}$$

which matches the theorem statement for $j = 1$.

Say that user i belongs to $\mathcal{T}_1$ and that he is scheduled to transmit in the first round according to the cost function. Thus, user i transmits

$$v_i^{(1)} = \mathbf{u}^{(1)}\mathbf{w}, \tag{187}$$

where $\mathbf{u}^{(1)}$ is appropriately chosen vector. All the remaining users update their observation matrix by appending vector $\mathbf{u}^{(1)}$ to it

$$\mathbf{A}_k \cup \mathbf{u}^{(1)}, \quad \forall k \in \mathcal{M} \setminus \{i\}. \tag{188}$$

In the next round we reduce parameter β by 1, and again ask the same question whether user i belongs to $\mathcal{T}_2$ for the updated set of observations. Combining (186) and (188) it is easy to see that in round j, the condition (186) becomes

$$\text{rank}\left(\mathbf{A}_i \cup \mathbf{u}^{(1)} \cup \cdots \cup \mathbf{u}^{(j-1)}\right) > N - (\beta - j + 1), \tag{189}$$

which completes the proof of Theorem 4.

Appendix I
Proof of Theorem 6

Let $\mathbf{R}$ be any feasible rate vector w.r.t. the problem (100), *i.e.*,

$$R(\mathcal{S}) \leq g_\beta(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \tag{190}$$

$$R(\mathcal{S}) \leq c(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \tag{191}$$

$$R(\mathcal{M}) = g_\beta(\mathcal{M}) = \beta. \tag{192}$$

By substituting $\mathcal{S}$ with $\mathcal{M} \setminus \mathcal{S}$ in (190), we obtain

$$R(\mathcal{M} \setminus \mathcal{S}) \leq g_\beta(\mathcal{M} \setminus \mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \tag{193}$$

This can be rewritten as

$$\begin{aligned} R(\mathcal{S}) &\geq R(\mathcal{M}) - g_\beta(\mathcal{M} \setminus \mathcal{S}) \\ &= \beta - g_\beta(\mathcal{M} \setminus \mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \end{aligned} \tag{194}$$

where the last equality comes from (192). From (191), (192), and (194) it follows that

$$g_\beta(\mathcal{M}) - g_\beta(\mathcal{M} \setminus \mathcal{V}) \leq R(\mathcal{V}) \leq c(\mathcal{V}), \quad \forall \mathcal{V} \subseteq \mathcal{M}. \tag{195}$$

From (195), we have that

$$g_\beta(\mathcal{M}) \leq g_\beta(\mathcal{M} \setminus \mathcal{V}) + c(\mathcal{V}), \quad \forall \mathcal{V} \subseteq \mathcal{M}, \tag{196}$$

which implies that

$$g_\beta(\mathcal{M}) \leq \min_{\mathcal{V} \subseteq \mathcal{M}} \left\{g_\beta(\mathcal{M} \setminus \mathcal{V}) + c(\mathcal{V})\right\} = g_\beta^{\mathbf{c}}(\mathcal{M}). \tag{197}$$

From (101), (192) and (197), we conclude that

$$g_\beta^{\mathbf{c}}(\mathcal{M}) = g_\beta(\mathcal{M}) = \beta. \tag{198}$$

Hence, $R(\mathcal{M}) = g_\beta^{\mathbf{c}}(\mathcal{M})$. Since $R_i \leq c_i$, it follows that

$$\begin{aligned} R(\mathcal{S}) &= R(\mathcal{V}) + R(\mathcal{S} \setminus \mathcal{V}) \\ &\leq g_\beta(\mathcal{V}) + c(\mathcal{S} \setminus \mathcal{V}), \quad \forall \mathcal{V}, \mathcal{S} \text{ s.t. } \mathcal{V} \subseteq \mathcal{S} \subseteq \mathcal{M}. \end{aligned} \tag{199}$$

Finally (199) implies that

$$R(\mathcal{S}) \leq \min_{\mathcal{V} \subseteq \mathcal{S}} \left\{g_\beta(\mathcal{V}) + c(\mathcal{S} \setminus \mathcal{V})\right\}, \quad \forall \mathcal{S} \subseteq \mathcal{M}. \tag{200}$$

Hence, $\mathbf{R} \in B(g_\beta^{\mathbf{c}})$.

Conversely, let $\mathbf{R}$ be such that $\mathbf{R} \in B(g_\beta^{\mathbf{c}})$. Then,

$$\begin{aligned} R(\mathcal{S}) &\leq g_\beta^{\mathbf{c}}(\mathcal{S}) \leq g_\beta(\mathcal{S}) + c(\emptyset) \\ &= g_\beta(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \end{aligned} \tag{201}$$

$$\begin{aligned} R(\mathcal{S}) &\leq g_\beta^{\mathbf{c}}(\mathcal{S}) \leq g_\beta(\emptyset) + c(\mathcal{S}) \\ &= c(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \end{aligned} \tag{202}$$

$$R(\mathcal{M}) = g_\beta^{\mathbf{c}}(\mathcal{M}) = \beta \tag{203}$$

where the second inequality in (201) and (202) directly follows from (101). From (201), (202), and (203) it follows that

$$\mathbf{R} \in B(g_\beta), \quad \text{s.t.} \quad R_i \leq c_i, \quad \forall i \in \mathcal{M}. \tag{204}$$

This completes the proof of Theorem 6.

References

[1] N. Milosavljevic, S. Pawar, S. El Rouayheb, M. Gastpar, and K. Ramchandran, "Deterministic algorithm for the cooperative data exchange problem," in *Proc. ISIT*, 2011, pp. 410–414.

[2] N. Milosavljevic, S. Pawar, S. El Rouayheb, M. Gastpar, and K. Ramchandran. (2011). "Optimal deterministic polynomial-time data exchange for omniscience." [Online]. Available: http://arxiv.org/abs/1108.6046

[3] N. Milosavljevic, S. Pawar, M. Gastpar, and K. Ramchandran, "Efficient algorithms for the data exchange problem under fairness constraints," in *Proc. 50th Annu. Allerton Conf. Commun., Control, Comput.*, 2012, pp. 502–508.

[4] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Jan. 2010, pp. 1–5.

[5] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb, "A randomized algorithm and performance bounds for coded cooperative data exchange," in *Proc. ISIT*, 2010, pp. 1888–1892.

[6] D. Ozgul and A. Sprintson, "An algorithm for cooperative data exchange with cost criterion," in *Proc. ITA*, 2011, pp. 1–4.

[7] T. A. Courtade, B. Xie, and R. D. Wesel, "Optimal exchange of packets for universal recovery in broadcast networks," in *Proc. Military Commun. Conf. (MILCOM)*, 2010, pp. 2250–2255.

[8] T. A. Courtade and R. D. Wesel, "Efficient universal recovery in broadcast networks," in *Proc. 48th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep./Oct. 2010, pp. 1542–1549.

[9] T. A. Courtade and R. D. Wesel, "Weighted universal recovery, practical secrecy, and an efficient algorithm for solving both," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2011, pp. 1349–1357.

[10] T. A. Courtade and R. D. Wesel, "Coded cooperative data exchange in multihop networks," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1136–1158, Feb. 2014.

[11] S. E. Tajbakhsh, P. Sadeghi, and R. Shams, "A generalized model for cost and fairness analysis in coded cooperative data exchange," in *Proc. NetCod*, 2011, pp. 1–6.

[12] I. Csiszár and P. Narayan, "Secrecy capacities for multiple terminals," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3047–3061, Dec. 2004.

[13] C. Chan, "Generating secret in a network," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Institute Technology, Cambridge, MA, USA, 2010.

[14] C. Chan, "Linear perfect secret key agreement," in *Proc. IEEE Inf. Theory Workshop*, Oct. 2011, pp. 723–726.

[15] C. Chan, "Delay of linear perfect secret key agreement," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput.*, 2011, pp. 1128–1135.

[16] S. Jaggi *et al.*, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973–1982, Jun. 2005.

[17] J. W. Byers, M. Luby, and M. Mitzenmacher, "Accessing multiple mirror sites in parallel: Using Tornado codes to speed up downloads," in *Proc. 18th INFOCOM*, 1999, pp. 275–283.

[18] J. Byers, J. Considine, M. Mitzenmacher, and S. Rost, "Informed content delivery across adaptive overlay networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 47–60, 2002.

[19] Z. Liu, C. Wu, B. Li, and S. Zhao, "UUSee: Large-scale operational on-demand streaming with random network coding," in *Proc. INFOCOM*, 2010, pp. 1–9.

[20] M. Luby, "LT codes," in *Proc. 43rd Annu. Found. Comput. Sci.*, 2002, pp. 271–280.

[21] D. S. Lun *et al.*, "Minimum-cost multicast over coded packet networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2608–2623, Jun. 2006.

[22] M. Gonen and M. Langberg, "Coded cooperative data exchange problem for general topologies," *IEEE Trans. Inf. Theory*, vol. 61, no. 10, pp. 5656–5669, Oct. 2015.

[23] D. E. Lucani, F. H. P. Fitzek, M. Médard, and M. Stojanovic, "Network coding for data dissemination: It is not what you know, but what your neighbors don't know," in *Proc. 7th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw.*, 2009, pp. 1–8.

[24] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[25] N. J. A. Harvey, D. R. Karger, and K. Murota, "Deterministic network coding by matrix completion," in *Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2005, pp. 489–498.

[26] J. Edmonds, "Submodular functions, matroids, and certain polyhedra," in *Proceedings Calgary International Conference on Combinatorial Structures and Their Applications*, R. Guy, H. Hanani, N. Sauer, and J. Schönheim, Eds. New York, NY, USA: Gordon and Breach, 1970, pp. 69-87.

[27] S. Fujishige, *Submodular Functions and Optimization*. New York, NY, USA: Elsevier, 2005.

[28] K. Nagano, Y. Kawahara, and S. Iwata, "Minimum average cost clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1759–1767.

[29] S. Fujishige, "Optimization over the polyhedron determined by a submodular function on a co-intersecting family," *Math. Program.*, vol. 42, nos. 1–3, pp. 565–577, 1988.

[30] J. B. Orlin, "A faster strongly polynomial time algorithm for submodular function minimization," *Math. Program.*, vol. 118, no. 2, pp. 237–251, 2009.

[31] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[32] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," in *Lecture Notes of EE392o*. Stanford, CA, USA: Stanford Univ. Press, 2004.

[33] T. Ibaraki and N. Katoh, *Resource Allocation Problems: Algorithmic Approaches*. Cambridge, MA, USA: MIT Press, 1988.

[34] D. P. Bertsekas, *Network Optimization: Continuous And Discrete Models*. Belmont, MA, USA: Athena Scientific, 1998.

[35] T. Ho *et al.*, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[36] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

Nebojsa Milosavljevic received the B.S. degree (2007) in electrical engineering from University of Belgrade, Serbia. He received the M.S. (2010) and Ph.D. (2013) in electrical engineering and computer science (EECS) from University of California at Berkeley. His main areas of research interest are multi-terminal information theory, network coding and combinatorial optimization. Since 2013, he has been with the Cloud Networking Group at Cisco-Meraki, San Francisco, CA.

Sameer Pawar received the M.S. degree in electrical engineering from Indian Institute of Science (IISc), Bangalore, India, in 2005. He received the Ph. D. degree in electrical engineering and computer science (EECS) from University of California at Berkeley, in 2013. Since 2014, he has been with the wireless division at Intel corporation, Santa Clara, CA. Prior to that, he had been with the Communications Department, Infineon Technologies India. His research interests include information theory and Coding theory. He is a recipient of Gold Medal for the Best Masters thesis in Electrical Division at IISc.

Salim El Rouayheb (S'07–M'09) received the Diploma degree in electrical engineering from the Lebanese University, Faculty of Engineering, Roumieh, Lebanon, in 2002, and the M.S. degree in computer and communications engineering from the American University of Beirut, Lebanon, in 2004. He received the Ph.D. degree in electrical engineering from Texas A&M University, College Station, in 2009. He was a postdoctoral research fellow at UC Berkeley (2010-2011) and a research scholar at Princeton University (2012-2013).

He is currently an assistant professor in the ECE department at the Illinois Institute of Technology, Chicago. His research interests are in the broad area of information theory and coding theory with a focus on network coding, coding for distributed storage and information theoretic security.

Michael Gastpar received the Dipl. El.-Ing. degree from the Eidgenössishe Technische Hochschule (ETH), Zürich, Switzerland, in 1997, the M.S. degree in electrical engineering from the University of Illinois at Urbana- Champaign, Urbana, IL, USA, in 1999, and the Doctoratès Science degree from the Ecole Polytechnique Fédérale (EPFL), Lausanne, Switzerland, in 2002. He was also a student in engineering and philosophy at the Universities of Edinburgh and Lausanne.

During the years 2003-2011, he was an Assistant and tenured Associate Professor at the Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley. Since 2011, he has been a Professor in the School of Computer and Communication Sciences, Ecole Polytechnique Fédérale (EPFL), Lausanne, Switzerland. He is also a professor at Delft University of Technology, The Netherlands. He was a Researcher with the Mathematics of Communications Department, Bell Labs, Lucent Technologies, Murray Hill, NJ. His research interests are in network information theory and related coding and signal processing techniques, with applications to sensor networks and neuroscience.

Dr. Gastpar received the IEEE Communications Society and Information Theory Society Joint Paper Award in 2013 and the EPFL Best Thesis Award in 2002. He was an Information Theory Society Distinguished Lecturer (2009-2011), an Associate Editor for Shannon Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY (2008-2011), and he has served as Technical Program Committee Co-Chair for the 2010 International Symposium on Information Theory, Austin, TX.

Kannan Ramchandran (F'93) is a Professor of Electrical Engineering and Computer Science at the University of California at Berkeley, where he has been since 1999. Prior to that, he was with the University of Illinois at Urbana-Champaign from 1993 to 1999, and was at AT&T Bell Laboratories from 1984 to 1990. His current research interests include distributed signal processing algorithms for wireless sensor and ad hoc networks, multimedia and peer-to-peer networking, multi-user information and communication theory, and wavelets and multi-resolution signal and image processing. Prof. Ramchandran is a Fellow of the IEEE. His research awards include the Elaihu Jury award for the best doctoral thesis in the systems area at Columbia University, the NSF CAREER award, the ONR and ARO Young Investigator Awards, two Best Paper awards from the IEEE Signal Processing Society, a Hank Magnuski Scholar award for excellence in junior faculty at the University of Illinois, and an Okawa Foundation Prize for excellence in research at Berkeley. He is a Fellow of the IEEE. He has published extensively in his field, holds 8 patents, serves as an active consultant to industry, and has held various editorial and Technical Program Committee positions.