Title
Data Exchange Problems: Algorithms and Complexity

Author
Milosavljevic, Nebojsa

Publication Date
2013

Peer reviewed|Thesis/dissertation

Data Exchange Problems: Algorithms and Complexity

by

Nebojsa Milosavljevic

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering-Electrical Engineering & Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Kannan Ramchandran, Co-Chair
Professor Michael Gastpar, Co-Chair
Professor Jim Pitman

Fall 2013

Data Exchange Problems: Algorithms and Complexity

Abstract

Data Exchange Problems: Algorithms and Complexity

by

Nebojsa Milosavljevic

Doctor of Philosophy in Engineering-Electrical Engineering & Computer Sciences

University of California, Berkeley

Professor Kannan Ramchandran, Co-Chair

Professor Michael Gastpar, Co-Chair

In this thesis 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 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 thesis, each user can have any arbitrary convex cost function. We provide a polynomial time deterministic algorithm (in the number of users and packets) that finds 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. In the later chapters we consider a general form of side-information, where each user observes independent realizations of some joint random process. For such scenario, we provide a polynomial-time algorithm (in the number of users and packets) that finds an optimal communication rate allocations for all the users. Next, we study two extensions to the original data exchange problem. First, we consider the problem where not all users in the system are interested in obtaining the file, but they are willing to help users who are. Also, we explore the problem where each user can communicate only to its immediate neighbors through a wireline network. For both the problems, we provide a polynomial time algorithm that is inspired by the original data exchange problem.

To my family

Contents

Chapter 1

Introduction

1.1 Data Exchange Problem

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.

An example of such a setting is shown in Figure 1.1, 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 1.1, 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 1.1), 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, 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.

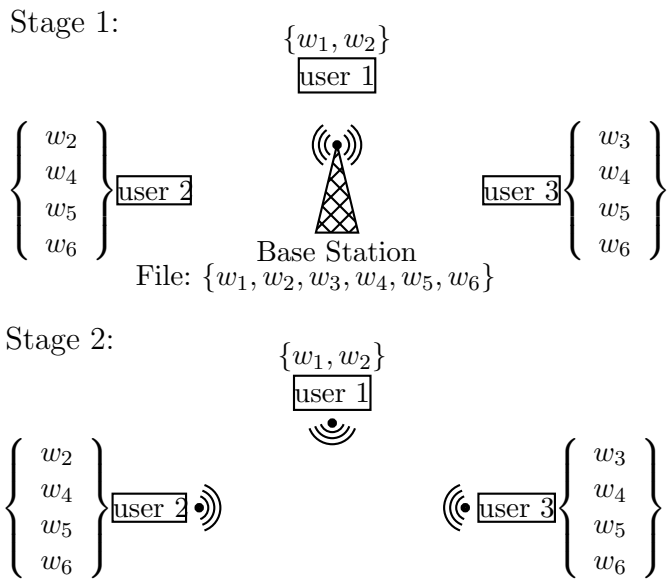


Figure 1.1: An example of the data exchange problem. A base station has a file formed of six packets $w_1, \ldots, w_6 \in \mathbb{F}_q$ and wants to deliver it to three users over an unreliable wireless channel. The base station stops transmitting once all users collectively have all the packets, even 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.

- 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 1.1, 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 fair 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

[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.

minimizes a convex fairness cost.

1.2 Source Model

In the example from Figure 1.1, we considered only a simple form of side-information, where different users observe subset of uncoded "raw" packets of the original file. Content distribution networks [6, 5, 24] are increasingly using codes, such as linear network codes or Fountain codes [25], 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*.

Each packet takes a value from a finite field $\mathbb{F}_q$. In a broader sense, we can think of the case where each packet's value is a realization of uniform distribution over $\{0, 1, \ldots, q-1\}$. This gives rise to considering an "information theoretic" version of this problem, where every user observes independent realizations of some random process. More specifically, say there are m users, and user i observes n independent realizations of the i^{th} component of an arbitrary discrete memoryless process defined by a joint probability mass function (pmf) $P_{X_1,X_2,\ldots,X_m}$. The goal is for each user to reconstruct all n realizations of all m components of the joint process while minimizing the communication cost. In the literature [11], this model is known as the discrete memoryless multiple source (DMMS) model.

1.3 Thesis Overview and Contributions

This thesis is outlined as follows:

- **Chapter 1:** In the remainder of this chapter, we outline the main contributions of this thesis, and we summarize the previous work.
- **Chapter 2:** In this chapter 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. In this chapter, we make the following contributions:

    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 is $\mathcal{O}((m^6 \cdot N^3 + m^7) \cdot \log N)$, where m is the total number of users, and N is the number of packets in the file. The second technique is based

on subgradient methods, in which case the total complexity of the algorithm can be bounded by $\mathcal{O}((N^2 \cdot m^4 \log m + N^5 \cdot m^4) \cdot \log N)$ 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 $\mathcal{O}(m \cdot N^4 \log N)$. 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 [34], where the authors considered linear communication cost.
3. For the data exchange problem with the 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.

- **Chapter 3:** We study the data exchange problem under the DMMS model, and the linear communication cost. In this chapter, we make the following contributions:

1. We propose a combinatorial algorithm of polynomial complexity that finds an optimal rate allocation w.r.t. the communication cost. The complexity of the algorithm is $\mathcal{O}(m^7 \cdot \gamma + m^8)$, where γ is the complexity of computing entropy function.
2. For the linear communication cost, we propose a non-combinatorial algorithm of polynomial complexity that computes an approximately optimal rate allocation. This algorithm recovers a primal optimal solution from an LP dual optimal solution by using dual subgradient methods, and averaging technique for the primal solution recovery. As mentioned above, the algorithm provides a near optimal solution to the problem that is within ε distance from the optimal one, and it is of complexity $\mathcal{O}((m^4 \log m + m^4\gamma) \cdot \lceil \frac{1}{\varepsilon^2} \rceil)$.

- **Chapter 4:** We study two extensions of the original data exchange problem. First, the data exchange problem with helpers, where some users are not interested in gaining access to the file, but they are willing to help other users in doing so. The second problem can be regraded as an extension of the data exchange problem with helpers, where all users can communicate to its immediate neighbors through a wireline network. The communication network is represented by an acyclic directed graph, and users

interested in the file are its sinks. In the literature, this problem is known as a multi-source multicast problem. For both of these problems, we propose a non-combinatorial algorithm of polynomial complexity that is based on the techniques we developed in Chapter 3.

1.4 Previous Work

Data exchange problem was originally introduced by by El Rouayheb *et al.* in [13], for the "raw" packet model. The communication cost considered was the total number of bits transmitted over a noiseless broadcast channel. A randomized algorithm was proposed in [39], while Tajbakhsh *et al.* [41] formulated this problem as a linear program (LP). The solution proposed in [41] is approximate.

The linear cost data exchange problem was studied by Ozgul *et al.* [34], where the authors proposed a randomized algorithm. A deterministic polynomial time algorithm was proposed by Milosavljevic *et al.* in [28], and by Courtade and Wesel in [8]. For the general separable convex communication cost, in [29] we proposed polynomial time deterministic and randomized algorithms. For the data exchange problem with helpers, in [30] we proposed a deterministic polynomial time algorithm based on dual subgradient methods, and averaging technique for the primal solution recovery.

In [9, 16], the authors considered the data exchange problem where users can only broadcast messages to their immediate neighbors. In [9] it was shown that the problem is NP-hard, while an approximate solution is provided in [16]. In [26], Lucani *et al.* considered the problem of data exchange when the channel between different users can have erasures.

For the general DMMS model, in [11], Csiszár and Narayan posed a related security problem referred to as the "multi-terminal key agreement" problem. They showed that obtaining the file among the users in minimum number of bits exchanged over the public broadcast channel is sufficient to maximize the size of the secret key shared between the users. This applies to the both versions of the data exchange problem:

- When all users in the system are interested in agreeing on a key, this problem is equivalent to the data exchange problem.
- When a subset of users is interested in agreeing on a key, and the rest of the users are willing to help, the problem is equivalent to the data exchange problem with helpers.

This result establishes a connection between the Multi-party key agreement and the data exchange problems. In [28], [30] we proposed a deterministic polynomial-time algorithm that computes optimal communication rates of each user w.r.t. the linear cost.

Minimum linear communication cost problem was also studied in the network coding literature. Lun *et al.* [27] proposed a polynomial time algorithm for the single source multicast problem over a directed acyclic graph. Ramamoorthy in [37] proposed an efficient

algorithm to the multi-source multicast problem based on purely convex optimization techniques. In Chapter 4, for the multi-source multicast problem, we propose a polynomial time algorithm that is based on both the convex and combinatorial optimization techniques, and we theoretically show its convergence.

Chapter 2

Data Exchange Problem - Linear Packet Model

2.1 System Model and Preliminaries

In this chapter, 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.

$$\mathbf{x}_i = \mathbf{A}_i \mathbf{w}, \ i \in \mathcal{M}, \tag{2.1}$$

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 (2.1) as a linear packet model.

Let us denote by $\mathbf{v}_i$, a transmission of user $i \in \mathcal{M}$. In [11] it was shown that in order for each user to recover the file, interaction among them is not needed. Hence, without loss of generality, we can assume that $\mathbf{v}_i$ is a function of user i's initial observation. We define

$$R_i \triangleq |\mathbf{v}_i|_q \tag{2.2}$$

to be the size of user's i 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

$$\psi_i : \mathbb{F}_q^{\ell_i} \times \mathbb{F}_q^{R_1} \times \cdots \times \mathbb{F}_q^{R_m} \to \mathbb{F}_q^N, \tag{2.3}$$

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

$$\psi_i(\mathbf{x}_i, \mathbf{v}_1, \ldots, \mathbf{v}_m) = \mathbf{w}. \tag{2.4}$$

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 (2.4) 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

$$\mathcal{R} \triangleq \left\{\mathbf{R} \in \mathbb{R}^m : R(\mathcal{S}) \geq N - \text{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}),\ \forall \mathcal{S} \subset \mathcal{M}\right\}, \tag{2.5}$$

where

$$R(\mathcal{S}) \triangleq \sum_{i\in\mathcal{S}} R_i, \quad \text{and} \quad \mathbf{A}_{\mathcal{M}\setminus\mathcal{S}} \triangleq \bigcup_{i\in\mathcal{M}\setminus\mathcal{S}} \mathbf{A}_i.$$

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.*

Proof. 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

$$R_2 = \text{rank}\left(\mathbf{A}_{\{1,2\}}\right) - \text{rank}\left(\mathbf{A}_{\{1\}}\right) \tag{2.6}$$

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

$$\begin{aligned} R_3 &= \text{rank}\left(\mathbf{A}_{\{1,2,3\}}\right) - \text{rank}\left(\mathbf{A}_{\{1,2\}}\right) && (2.7)\\ &\vdots \\ R_m &= \text{rank}\left(\mathbf{A}_{\mathcal{M}}\right) - \text{rank}\left(\mathbf{A}_{\mathcal{M}\setminus\{m\}}\right) \\ &= N - \text{rank}\left(\mathbf{A}_{\mathcal{M}\setminus\{m\}}\right). && (2.8) \end{aligned}$$

Observe that the number of linear equations each user sends depends upon the ordering of users in equations (2.6) through (2.8). 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.

$$\begin{aligned} R_{j(2)} &= \text{rank}\left(\mathbf{A}_{\{1,j(2)\}}\right) - \text{rank}\left(\mathbf{A}_{\{1\}}\right) && (2.9)\\ R_{j(3)} &= \text{rank}\left(\mathbf{A}_{\{1,j(2),j(3)\}}\right) - \text{rank}\left(\mathbf{A}_{\{1,j(2)\}}\right) && (2.10)\\ &\vdots \\ R_{j(m)} &= N - \text{rank}\left(\mathbf{A}_{\mathcal{M}\setminus\{j(m)\}}\right). && (2.11) \end{aligned}$$

From (2.9)-(2.11), observe that

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

By using this method of ordering, we can reconstruct any vertex of the region

$$\sum_{i=t}^{m} R_{j(i)} \geq N - \text{rank}\left(\mathbf{A}_{\{1,j(2),\ldots,j(t-1)\}}\right), \quad t = 2, \ldots, m,$$
$$\text{for all permutations } j(2), \ldots, j(m) \text{ of the set } \mathcal{M} \setminus \{1\}. \tag{2.12}$$

The region in (2.12) is equivalent to

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

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 (2.5). Then, it is not hard to show that

$$\mathcal{R}_{int} = \mathcal{R}_1 \cap \mathcal{R}_2 \cap \cdots \cap \mathcal{R}_m. \tag{2.13}$$

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 [1]. Hence, every rate tuple that belongs to $\mathcal{R}_{int}$ can be achieved via linear network coding. □

In Section 2.5 we show 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.

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

$$\mathbf{v}_i = \mathbf{B}_i \mathbf{x}_i = \mathbf{B}_i \mathbf{A}_i \mathbf{w} = \mathbf{U}_i \mathbf{w}, \tag{2.14}$$

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,

$$\text{rank}\left(\begin{bmatrix} \mathbf{A}_i \\ \mathbf{U} \end{bmatrix}\right) = N, \quad \forall i \in \mathcal{M}, \tag{2.15}$$

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 (2.15) 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 (2.5) and the above mentioned cost function, the problem considered in this work can be formulated as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{R}\in\mathbb{Z}^m} & \sum_{i=1}^{m} \varphi_i(R_i), \\ \text{s.t.} \quad & R(\mathcal{S}) \geq N - \operatorname{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}), \ \forall \mathcal{S} \subset \mathcal{M}. \end{aligned} \tag{2.16}$$

Optimization problem (2.16) is a convex integer problem with $2^m - 2$ constraints. It was shown in [7] that only n of these constraints are active but the challenge is how to determine which of them are. Solving the optimization problem (2.16) answers the question of how many symbols in $\mathbb{F}_q$ each user has to transmit in an optimal scheme. In this chapter we provide a polynomial time algorithm that solves problem (2.16). 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 [22], [17].

2.2 Deterministic Algorithm

Our goal is to solve problem (2.16) 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 4) since the objective function is w.r.t. β. First, let us identify these two problems by rewriting problem (2.16) as follows

$$\min_{\beta\in\mathbb{Z}_+} h(\beta), \tag{2.17}$$

where

$$\begin{aligned} h(\beta) \triangleq \min_{\mathbf{R}\in\mathbb{Z}^m} & \sum_{i=1}^{m} \varphi_i(R_i), \\ \text{s.t.} \quad & R(\mathcal{M}) = \beta, \quad R(\mathcal{S}) \geq N - \operatorname{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}), \ \forall \mathcal{S} \subset \mathcal{M}. \end{aligned} \tag{2.18}$$

Note that the optimizations (2.17) and (2.18) are associated with Problem 2 and Problem 1 defined above, respectively. Next we will explain our approach to solving these two problems.

2.2.1 Optimization with a Given Sum-Rate Budget β

Now, let us focus on the set of constraints of optimization problem (2.18). First, we introduce some concepts from combinatorial optimization theory.

Definition 2 (Polyhedron)**.** Let y_β be a set function defined over the set $\mathcal{M} = \{1, 2, \ldots, m\}$, *i.e.*, $y_\beta : 2^{\mathcal{M}} \to \mathbb{Z}$ such that $\varphi_\beta(\emptyset) = 0$, where $2^{\mathcal{M}}$ is the power set of $\mathcal{M}$. Then the *integer polyhedron* $P(y_\beta, \geq)$ and the *integer base polyhedron* $B(y_\beta, \geq)$ of y_β are defined as follows

$$P(y_\beta, \geq) \triangleq \{\mathbf{R} \in \mathbb{Z}^m \mid R(\mathcal{S}) \geq y_\beta(\mathcal{S}),\ \forall \mathcal{S} \subseteq \mathcal{M}\}, \tag{2.19}$$

$$B(y_\beta, \geq) \triangleq \{\mathbf{R} \in P(y_\beta, \geq) \mid R(\mathcal{M}) = y_\beta(\mathcal{M})\}. \tag{2.20}$$

Analogously, we can define $P(y_\beta, \leq)$, and $B(y_\beta, \leq)$.

Note that the set of constraints of problem (2.18), for any fixed $\beta \in \mathbb{Z}_+$, constitutes the integer base polyhedron $B(y_\beta, \geq)$ of the set function

$$y_\beta(\mathcal{S}) = \begin{cases} N - \operatorname{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}) & \text{if } \mathcal{S} \subset \mathcal{M}, \\ \beta & \text{if } \mathcal{S} = \mathcal{M}. \end{cases} \tag{2.21}$$

Example 1. Let us consider the source model from Figure 1.1, where the three users observe the following parts of the file $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix}^T$:

$$\begin{aligned} \mathbf{x}_1 &= \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T, \\ \mathbf{x}_2 &= \begin{bmatrix} w_2 & w_4 & w_5 & w_6 \end{bmatrix}^T, \\ \mathbf{x}_3 &= \begin{bmatrix} w_3 & w_4 & w_5 & w_6 \end{bmatrix}^T. \end{aligned} \tag{2.22}$$

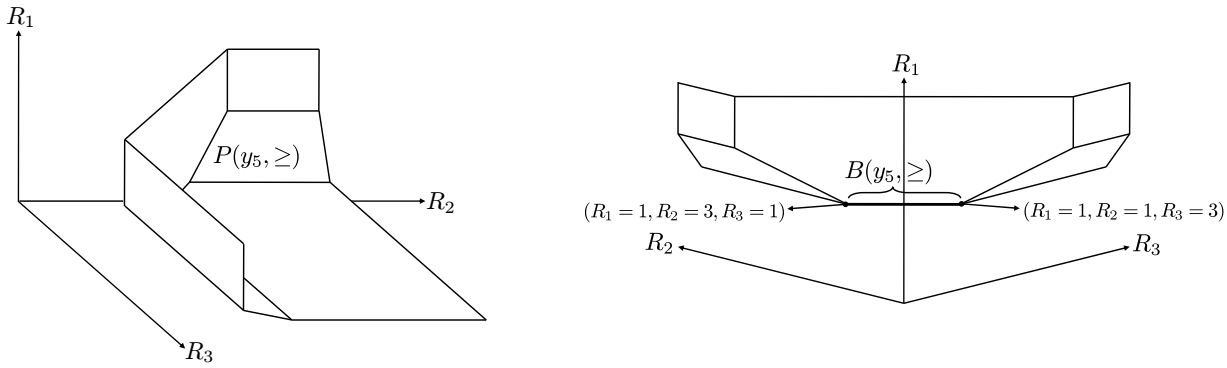
For $\beta = 5$, polyhedron $P(y_5, \geq)$ (see Figure 2.1) is defined by the following set of inequalities

$$\begin{aligned} &R_1 \geq y_5(\{1\}) = 1, \quad R_2 \geq y_5(\{2\}) = 0, \quad R_3 \geq y_5(\{3\}) = 1, \\ &R_1 + R_2 \geq y_5(\{1,2\}) = 2, \quad R_1 + R_3 \geq y_5(\{1,3\}) = 2, \quad R_2 + R_3 \geq y_5(\{2,3\}) = 4, \\ &R_1 + R_2 + R_3 \geq y_5(\{1,2,3\}) = 5. \end{aligned} \tag{2.23}$$

The base polyhedron $B(y_5, \geq)$ (see Figure 2.1(b)) can be interpreted as the intersection of the polyhedron $P(y_5, \geq)$ with a hyperplane $R_1 + R_2 + R_3 = y_5(\{1,2,3\}) = 5$.

Definition 3 (Dual set function [14])**.** For a set function y_β, its *dual set function* $f_\beta : 2^{\mathcal{M}} \to \mathbb{Z}$ is defined as follows

$$f_\beta(\mathcal{S}) \triangleq y_\beta(\mathcal{M}) - y_\beta(\mathcal{M} \setminus \mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}. \tag{2.24}$$



(a) Polyhedron $P(y_5, \geq)$

(b) Base Polyhedron $B(y_5, \geq)$

Figure 2.1: For the source model given by (2.22), and the set function y_5 obtained from (2.21), polyhedron and base polyhedron are shown above.

Thus, the dual of function y_β, as defined in (2.21), is given by

$$f_\beta(\mathcal{S}) = \begin{cases} \beta - N + \text{rank}(\mathbf{A}_\mathcal{S}) & \text{if } \emptyset \neq \mathcal{S} \subseteq \mathcal{M}, \\ 0 & \text{if } \mathcal{S} = \emptyset. \end{cases} \tag{2.25}$$

Lemma 1 (Dual Polyhedron [14])**.** *If $B(y_\beta, \geq) \neq \emptyset$, then $B(y_\beta, \geq) = B(f_\beta, \leq)$.*

Proof. Let $\mathbf{R} \in B(y_\beta, \geq)$. The base polyhedron $B(y_\beta, \geq)$ is defined by the following set of inequalities

$$R(\mathcal{S}) \geq y_\beta(\mathcal{S}), \quad \forall \mathcal{S} \subset \mathcal{M}, \tag{2.26}$$
$$R(\mathcal{M}) = y_\beta(\mathcal{M}). \tag{2.27}$$

Equality 2.27 can be rewritten as

$$R(\mathcal{M}) = R(\mathcal{S}) + R(\mathcal{M} \setminus \mathcal{S}) = y_\beta(\mathcal{M}). \tag{2.28}$$

Since $R(\mathcal{M} \setminus \mathcal{S}) \geq y_\beta(\mathcal{M} \setminus \mathcal{S})$ (by (2.26)), from (2.28) it follows that

$$R(\mathcal{S}) \leq y_\beta(\mathcal{M}) - y_\beta(\mathcal{M} \setminus \mathcal{S}) = f_\beta(\mathcal{S}). \tag{2.29}$$

By Definition 3,

$$f_\beta(\mathcal{M}) = y_\beta(\mathcal{M}). \tag{2.30}$$

From (2.29) and (2.30) it immediately follows that $\mathbf{R} \in B(f_\beta, \leq)$. Similarly, starting from a rate a rate vector that belongs to $B(f_\beta, \leq)$, it is straightforward to show that such rate vector belongs to $B(y_\beta, \geq)$. □

Corollary 1. If $B(y_\beta, \geq) \neq \emptyset$, then $P(y_\beta, \geq) \cap P(f_\beta, \leq) = B(y_\beta, \geq) = B(f_\beta, \leq)$.

Example 2. Let us consider the same source model as in Example 3. For $\beta = 4$, the polyhedron $P(y_4, \geq)$ is defined by

$$\begin{aligned} &R_1 \geq y_4(\{1\}) = 1, \quad R_2 \geq y_4(\{2\}) = 0, \quad R_3 \geq y_4(\{3\}) = 1, \\ &R_1 + R_2 \geq y_4(\{1,2\}) = 2, \quad R_1 + R_3 \geq y_4(\{1,3\}) = 2, \quad R_2 + R_3 \geq y_4(\{2,3\}) = 4, \\ &R_1 + R_2 + R_3 \geq y_4(\{1,2,3\}) = 4. \end{aligned} \tag{2.31}$$

It can be verified that no rate vector (R_1, R_2, R_3) exists such that $R_1+R_2+R_3 = 4$. Therefore, $B(y_4, \geq) = \emptyset$, and $P(y_4, \geq) = P(y_5, \geq)$.

The polyhedron of the dual set function f_4 (see Definition 3) is defined by

$$\begin{aligned} &R_1 \leq f_4(\{1\}) = 0, \quad R_2 \leq f_4(\{2\}) = 2, \quad R_3 \leq f_4(\{3\}) = 2, \\ &R_1 + R_2 \leq f_4(\{1,2\}) = 3, \quad R_1 + R_3 \leq f_4(\{1,3\}) = 4, \quad R_2 + R_3 \leq f_4(\{2,3\}) = 3, \\ &R_1 + R_2 + R_3 \leq f_4(\{1,2,3\}) = 4. \end{aligned} \tag{2.32}$$

From Lemma 1, since $B(y_4, \geq) = \emptyset$, it also holds that $B(f_4, \leq) = \emptyset$, and we observe that $P(y_4 \geq) \cap P(f_4, \leq) = \emptyset$ (see Figure 2.2(a)). The maximum sum-rate over polyhedron $P(f_4, \leq)$ is 3.

On the other hand, for $\beta = 5$, $B(y_5, \geq)$ is not an empty set, and by Lemma 1 $B(y_5, \geq) = B(f_5, \leq)$. Polyhedron $P(f_5, \leq)$ is defined by

$$\begin{aligned} &R_1 \leq f_5(\{1\}) = 1, \quad R_2 \leq f_5(\{2\}) = 3, \quad R_3 \leq f_5(\{3\}) = 3, \\ &R_1 + R_2 \leq f_5(\{1,2\}) = 4, \quad R_1 + R_3 \leq f_5(\{1,3\}) = 5, \quad R_2 + R_3 \leq f_5(\{2,3\}) = 4, \\ &R_1 + R_2 + R_3 \leq f_5(\{1,2,3\}) = 5. \end{aligned} \tag{2.33}$$

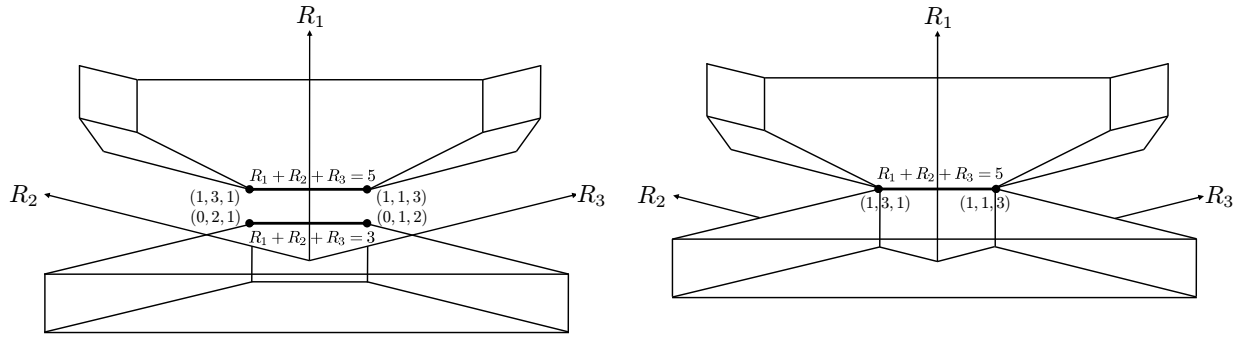
The direct consequence of Lemma 1 is that $P(y_5, \geq) \cap P(f_5, \leq) = B(y_5, \geq) = B(f_5, \leq)$ (see Figure 2.2(b)).

From Lemma 1 it follows that if the optimization problem (2.18) is feasible, *i.e.*, if $B(y_\beta, \geq) \neq \emptyset$, then it is equivalent to

$$\min_{\beta \in \mathbb{Z}_+} \min_{\mathbf{R} \in \mathbb{Z}^m} \sum_{i=1}^{m} \varphi_i(R_i), \quad \text{s.t.} \quad \mathbf{R} \in B(f_\beta, \leq). \tag{2.34}$$

For now, let us assume that parameter β is chosen such that the optimization problem (2.34) is feasible, *i.e.*, $B(f_\beta, \leq) \neq \emptyset$. We will explain later how the condition $B(f_\beta, \leq) \neq \emptyset$ can be efficiently verified.

The main idea behind solving the optimization problem in (2.34) efficiently, is to utilize the combinatorial properties of the set function f_β.



(a) $P(y_4, \geq)$ and $P(f_4, \leq)$ for $\beta = 4$.

(b) $P(y_5, \geq)$ and $P(f_5, \leq)$ for $\beta = 5$.

Figure 2.2: For $\beta = 4$, $B(y_4, \geq) = \emptyset$. Therefore, $P(y_4, \geq) \cap P(f_4, \leq) = \emptyset$ (see Figure 2.2(a)). For $\beta = 5$, $B(y_5 \geq) \neq \emptyset$. Therefore, $P(y_5, \geq) \cap P(f_5, \leq) = B(y_5, \geq) = B(f_5, \leq)$ (see Figure 2.2(b)).

Definition 4. We say that a set function $f : 2^{\mathcal{M}} \to \mathbb{Z}$ is *intersecting submodular* if

$$\begin{aligned} f(\mathcal{S}) + f(\mathcal{T}) \geq f(\mathcal{S} \cup \mathcal{T}) + f(\mathcal{S} \cap \mathcal{T}), \\ \forall \mathcal{S}, \mathcal{T} \subseteq \mathcal{M} \text{ s.t. } \mathcal{S} \cap \mathcal{T} \neq \emptyset. \end{aligned} \tag{2.35}$$

When the inequality conditions in (2.35) are satisfied for all sets $\mathcal{S}, \mathcal{T} \subseteq \mathcal{M}$, the function f is *fully submodular*.

Lemma 2. *The function f_β is* intersecting submodular *for any β. When $\beta \geq N$, f_β is fully submodular.*

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

$$\begin{aligned} & f_\beta(\mathcal{S}) + f_\beta(\mathcal{T}) \\ & = \text{rank}(\mathbf{A}_\mathcal{S}) + \text{rank}(\mathbf{A}_\mathcal{T}) - 2(N - \beta) \\ & \geq \text{rank}(\mathbf{A}_{\mathcal{S}\cup\mathcal{T}}) + \text{rank}(\mathbf{A}_{\mathcal{S}\cap\mathcal{T}}) - 2(N - \beta) \\ & = f_\beta(\mathcal{S} \cup \mathcal{T}) + f_\beta(\mathcal{S} \cap \mathcal{T}). \end{aligned} \tag{2.36}$$

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

$$\begin{aligned} & f_\beta(\mathcal{S}) + f_\beta(\mathcal{T}) \\ & = \text{rank}(\mathbf{A}_\mathcal{S}) + \text{rank}(\mathbf{A}_\mathcal{T}) - 2(N - \beta) \\ & \geq \text{rank}(\mathbf{A}_{\mathcal{S}\cup\mathcal{T}}) - (N - \beta) = f_\beta(\mathcal{S} \cup \mathcal{T}). \end{aligned} \tag{2.37}$$

The inequality in (2.37) directly follows from the submodularity of the rank function

$$\text{rank}(\mathbf{A}_{\mathcal{S}}) + \text{rank}(\mathbf{A}_{\mathcal{T}}) - \text{rank}(\mathbf{A}_{\mathcal{S}\cup\mathcal{T}}) \geq 0 \geq \beta - N.$$

This completes the proof. □

Theorem 2 (Dilworth Truncation [14])**.** *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., $P(g_\beta, \leq) = P(f_\beta, \leq)$, and g_β can be expressed as*

$$g_\beta(\mathcal{S}) = \min_{\mathcal{P}} \left\{ \sum_{\mathcal{V}\in\mathcal{P}} f_\beta(\mathcal{V}) : \mathcal{P} \text{ is a partition of } \mathcal{S} \right\}. \tag{2.38}$$

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, $P(g_\beta, \leq) = P(f_\beta, \leq)$, it follows that $B(g_\beta, \leq) = B(f_\beta, \leq)$ whenever $g_\beta(\mathcal{M}) = f_\beta(\mathcal{M}) = \beta$, *i.e.*, when $B(f_\beta, \leq) \neq \emptyset$ which implies feasibility of the optimization problem (2.34).

Continuing with Example 2, the Dilworth truncation of the set function f_4 is given by

$$\begin{aligned} &g_4(\{1\}) = 0,\ g_4(\{2\}) = 2,\ g_4(\{3\}) = 2,\\ &g_4(\{1,2\}) = 2,\ g_4(\{1,3\}) = 2,\ g_4(\{2,3\}) = 3,\\ &g_4(\{1,2,3\}) = 3. \end{aligned} \tag{2.39}$$

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 problems (2.18) and (2.34). Also, from Figure 2.2(a) we can see that there is a "duality gap" between the polyhedrons $P(f_4, \leq)$ and $P(y_4, \geq)$ indicating infeasibility of $\beta = 4$.

On the other hand, for $\beta = 5$, Dilworth truncation of a set function f_5 is given by

$$\begin{aligned} &g_5(\{1\}) = 1,\ g_5(\{2\}) = 3,\ g_5(\{3\}) = 3,\\ &g_5(\{1,2\}) = 4,\ g_5(\{1,3\}) = 4,\ g_5(\{2,3\}) = 4,\\ &g_5(\{1,2,3\}) = 5. \end{aligned} \tag{2.40}$$

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 problems (2.18) and (2.34). This can be also verified by observing that there is no gap between the polyhedrons $P(f_5, \leq)$ and $P(y_5, \geq)$ in Figure 2.2(b). Hence, the optimization problem (2.18) can be written as

$$\min_{\mathbf{R}\in\mathbb{Z}^m} \sum_{i=1}^{m} \varphi_i(R_i), \quad \text{s.t.,} \quad \mathbf{R} \in B(g_\beta, \leq) \tag{2.41}$$

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

Remark 2. Parameter β is feasible w.r.t. the problem (2.18) if $g_\beta(\mathcal{M}) = \beta$. Otherwise, $g_\beta(\mathcal{M}) < \beta$. This is the direct consequence of the Dilworth truncation (2.38).

Depending upon a cost function, in the sequel, we provide several algorithms that are efficiently solving problem (2.17). First, we analyze a special case when the cost function is linear,

$$\varphi_i(R_i) = \alpha_i R_i, \quad \alpha_i > 0, \quad \forall i \in \mathcal{M}. \tag{2.42}$$

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

2.2.2 Linear Cost - Edmonds' Algorithm

When the cost function is linear, the optimization problem (2.41) has the following form

$$\min_{\mathbf{R}\in\mathbb{Z}^m} \sum_{i=1}^{m} \alpha_i R_i, \quad \text{s.t.,} \quad \mathbf{R} \in B(g_\beta, \leq). \tag{2.43}$$

Due to the submodularity of function g_β, the optimization problem (2.43) can be solved analytically using Edmonds' greedy algorithm [12] (see Algorithm 1).

Algorithm 1 Edmonds' Algorithm

1: Set $j(1), j(2), \ldots, j(m)$ to be an ordering of $1, 2, \ldots, m$, such that

$$\alpha_{j(1)} \leq \alpha_{j(2)} \leq \cdots \leq \alpha_{j(m)}.$$

2: Initialize $\mathbf{R}^* = \mathbf{0}$.
3: **for** $i = 1$ to m **do**
4:

$$R_{j(i)} = g_\beta(\{j(1), j(2), \ldots, j(i)\}) - g_\beta(\{j(1), j(2), \ldots, j(i-1)\}).$$

5: $\mathbf{R}^* = \mathbf{R}$ is an optimal rate vector w.r.t. the problem (2.43).
6: **end for**

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

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

In other words, at each iteration, the individual user's rate update reaches the boundary of polyhedron $P(g_\beta, \leq)$. Optimality of this approach is the direct consequence of submodularity of function g_β [12].

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

$$\sum_{i=1}^{m} R_i^* = g_\beta(\mathcal{M}). \tag{2.44}$$

Remark 4. Step 1 of Algorithm 1 involves sorting a vector of m elements. The best algorithm to our knowledge is merge sort which can execute sorting in $\mathcal{O}(m \log m)$ time. Therefore, the complexity of Edmonds' algorithm is $\mathcal{O}(m \log m + m \cdot \vartheta)$, where ϑ is the complexity of computing function $g_\beta(\mathcal{S})$ for any given set $\mathcal{S} \subseteq \mathcal{M}$.

Example 3. Let us consider the same source model as in Example 3, 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 (2.33) and (2.40), respectively. The rate vector is updated in an increasing order w.r.t. the weight vector. In this case, the order is $1 \to 3 \to 2$.

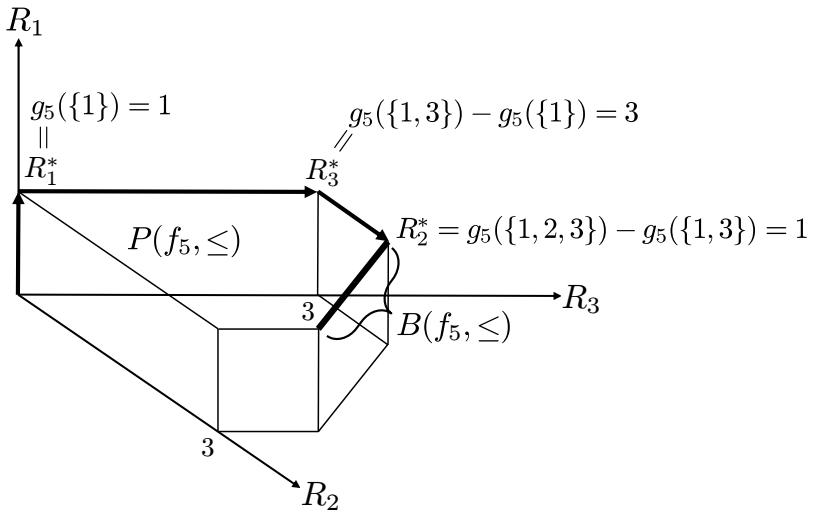


Figure 2.3: Edmonds' algorithm applied to the three-user problem described in Example 3, with the cost function $R_1+3R_2+2R_3$. To minimize the cost, the order in which we greedily update communication rates should be increasing w.r.t. the weight vector, i.e., $1 \to 3 \to 2$. The optimal *DE*-rate vector is $R_1^* = 1$, $R_2^* = 1$, $R_3^* = 3$.

2.2.3 Proof of Correctness of Edmonds' Algorithm

To show why Edmonds' algorithm provides optimal rate vector w.r.t. the problem (2.43), we introduce some additional concepts from combinatorial optimization theory.

Definition 5. For a submodular function g_β, define *saturation function* $\text{sat} : P(g_\beta, \leq) \to 2^{\mathcal{M}}$ by

$$\text{sat}(\mathbf{R}) = \{i \mid i \in \mathcal{M},\ \mathbf{R} + d \cdot \mathbf{e}(i) \notin P(g_\beta, \leq) \text{ for any } d > 0\} \tag{2.45}$$

In other words, $\text{sat}(\mathbf{R})$ is a set of all $j \in \mathcal{M}$ such that an increase in R_j will produce $\mathbf{R} \notin P(g_\beta, \leq)$.

Definition 6. For a submodular function g_β, *dependence function* $\text{dep} : P(g_\beta, \leq) \times \mathcal{M} \to 2^{\mathcal{M}}$ is defined as follows

$$\text{dep}(\mathbf{R}, i) = \begin{cases} \{u \mid u \in \mathcal{M},\ \mathbf{R} + d \cdot (\mathbf{e}(i) - \mathbf{e}(u)) \in P(g_\beta, \leq) \text{ for some } d > 0\} & \text{if } i \in \text{sat}(\mathbf{R}) \\ \emptyset & \text{if } i \notin \text{sat}(\mathbf{R}). \end{cases} \tag{2.46}$$

If $\mathbf{R}$ and $i \in \text{sat}(\mathbf{R})$, then $\text{dep}(\mathbf{R}, i)$ denotes the set of all $u \in \mathcal{M}$ such that increase in R_i by $d > 0$ and decrease in R_u by d will produce a rate vector that still belongs to $P(g_\beta, \leq)$. This is particularly interesting when we focus on the base polyhedron $B(g_\beta, \leq)$. Since such manipulation preserves the sum-rate, *i.e.*, $R(\mathcal{M}) = g_\beta(\mathcal{M})$, the dependance function provides a set of rate vector updates that can be performed over the base polyhedron. For instance, lets consider the optimal rate allocation from Figure 2.3: $R_1^* = 1$, $R_2^* = 1$, $R_3^* = 3$. As mentioned before, such rate vector belongs to the base polyhedron $B(g_5, \leq)$, and thus any positive update of its components will produce a vector that does not belong to $P(g_5, \leq)$. Hence, $\text{sat}(\mathbf{R}^*) = \mathcal{M}$. On the other hand, $\text{dep}(\mathbf{R}^*, 2) = \{3\}$. This means that coordinates 2 and 3 can be updated according to (2.46) in order to produce other optimal solutions: $R_1^* = 1$, $R_2^* = 2$, $R_3^* = 2$, and $R_1^* = 1$, $R_2^* = 3$, $R_3^* = 1$.

Lemma 3. *Let* $\mathbf{R} \in P(g_\beta, \leq)$. *If* $R(\mathcal{U}) = g_\beta(\mathcal{U})$, *and* $R(\mathcal{V}) = g_\beta(\mathcal{V})$, *then* $R(\mathcal{U} \cup \mathcal{V}) = g_\beta(\mathcal{U} \cup \mathcal{V})$, *and* $R(\mathcal{U} \cap \mathcal{V}) = g_\beta(\mathcal{U} \cap \mathcal{V})$.

Proof.

$$\begin{aligned} g_\beta(\mathcal{U}) + g_\beta(\mathcal{V}) &= R(\mathcal{U}) + R(\mathcal{V}) = \\ &= R(\mathcal{U} \cup \mathcal{V}) + R(\mathcal{U} \cap \mathcal{V}) \\ &\overset{(a)}{\leq} g_\beta(\mathcal{U} \cup \mathcal{V}) + g_\beta(\mathcal{U} \cap \mathcal{V}) \\ &\overset{(b)}{\leq} g_\beta(\mathcal{U}) + g_\beta(\mathcal{V}), \end{aligned}$$

where (a) comes from the fact that $\mathbf{R} \in P(g_\beta, \leq)$, and (b) is due to submodularity of g_β. Therefore, (a) and (b) have to hold with equality. From $R(\mathcal{U} \cup \mathcal{V}) \leq g_\beta(\mathcal{U} \cup \mathcal{V})$, and $R(\mathcal{U} \cap \mathcal{V}) \leq g_\beta(\mathcal{U} \cap \mathcal{V})$, the proof of this lemma immediately follows. □

In the next theorem, we show that any type of greedy algorithm, where a positive rate vector updates are confined to be inside polyhedron $P(g_\beta, \leq)$, will eventually result in a rate vector that belongs to the base polyhedron $B(f_\beta, \leq)$ (see Algorithm 2).

Definition 7. Let us define by $\mathbf{e}(i)$ an m dimensional vector where all its coordinates are zero except for the i^{th} coordinate which is equal to 1.

Algorithm 2 Greedy Algorithm

1: Initialize $\mathbf{R} = \mathbf{0}$.
2: Select $i \in \mathcal{M}$ any $v \in \mathbb{Z}_+$ such that $\mathbf{R} + v \cdot \mathbf{e}(i) \in P(g_\beta, \leq)$, and set $\mathbf{R} = \mathbf{R} + v \cdot \mathbf{e}(i)$
3: If $\mathbf{R}$ cannot be updated then stop, otherwise go to Step 2.

Theorem 3. *When Algorithm 2 terminates,* $\mathbf{R} \in B(g_\beta, \leq)$.

Proof. Note that after some finite number of realizations of Step 2 of Algorithm 2, the boundary of polyhedron $P(g_\beta)$ is reached. In other words there exists some non-empty set $\text{sat}(\mathbf{R})$.

For every $i \in \text{sat}(\mathbf{R})$, there exists a set $\mathcal{S} \subseteq \mathcal{M}$ such that $i \in \mathcal{S}$, and $R(\mathcal{S}) = g_\beta(\mathcal{S})$. This can be proved by contradiction. Assume that for some $i \in \text{sat}(\mathbf{R})$ it holds that

$$R(\mathcal{T}) < g_\beta(\mathcal{T}), \quad \forall \mathcal{T} \subseteq \mathcal{M}, \quad \text{s.t.} \quad i \in \mathcal{T}. \tag{2.47}$$

This would mean that it is possible to increase R_i by some positive amount and still have $\mathbf{R} \in P(g_\beta, \leq)$. This implies that $i \notin \text{sat}(\mathbf{R})$ which is a contradiction.

Therefore, we have established that

$$\forall i \in \text{sat}(\mathbf{R}), \quad \exists \mathcal{S}_i \subseteq \text{sat}(\mathbf{R}) \quad \text{s.t.} \quad R(\mathcal{S}_i) = g_\beta(\mathcal{S}_i). \tag{2.48}$$

Note that $\mathcal{S}_i \subseteq \text{sat}(\mathbf{R})$ in (2.48), because otherwise $\text{sat}(\mathbf{R})$ would also include elements from $\mathcal{S}_i \setminus \text{sat}(\mathbf{R})$

Observe that $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_{|\text{sat}(\mathbf{R})|} = \text{sat}(\mathbf{R})$, and by Lemma 3 it follows that

$$R(\text{sat}(\mathbf{R})) = g_\beta(\text{sat}(\mathbf{R})). \tag{2.49}$$

While keep on executing Algorithm 2 we eventually saturate all elements in the set $\mathcal{M}$, *i.e.*, $\text{sat}(\mathbf{R}) = \mathcal{M}$. Hence, by (2.49), it follows that

$$R(\mathcal{M}) = g_\beta(\mathcal{M}), \tag{2.50}$$

or in other words $\mathbf{R} \in B(g_\beta, \leq)$. □

Theorem 3 shows that any type of greedy algorithm leads to a rate vector that belongs to the base polyhedron $B(g_\beta, \leq)$. Now, let us show that the rate allocation provided by Algorithm 1 is feasible, *i.e.*,

$$R^*(\mathcal{S}) \leq g_\beta(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}. \tag{2.51}$$

We show this by induction on the size of $\mathcal{S}$. Without loss of generality, let us assume that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_m$. For each $i \in \mathcal{M}$,

$$\begin{aligned} R_i^* &\overset{(a)}{=} g_\beta(\{1,2,\ldots,i\}) - g_\beta(\{1,2,\ldots,i-1\}) \\ &\overset{(b)}{\leq} g_\beta(\{i\}) - g_\beta(\emptyset) \\ &= g_\beta(\{i\}), \end{aligned} \tag{2.52}$$

where (a) follows from Algorithm 1, and (b) is due to submodularity of function g_β. Therefore, (2.51) is true for all $\mathcal{S} \subseteq \mathcal{M}$ such that $|\mathcal{S}| = 1$. Assume that (2.51) holds for all $\mathcal{S}$ such that $|\mathcal{S}| = p-1$, and let $\mathcal{S} = \{i_1, i_2, \ldots, i_p\}$, where $i_1 < i_2 < \cdots < i_p$. Then,

$$\begin{aligned} R^*(\mathcal{S}) &= R^*(\mathcal{S} \setminus \{i_p\}) + R_{i_p}^* \\ &\overset{(a)}{\leq} g_\beta(\mathcal{S} \setminus \{i_p\}) + R_{i_p}^* \\ &= g_\beta(\mathcal{S} \setminus \{i_p\}) + g_\beta(\{1,2,\ldots,i_p\}) - g_\beta(\{1,2,\ldots,i_p-1\}) \\ &\overset{(b)}{\leq} g_\beta(\mathcal{S}), \end{aligned} \tag{2.53}$$

where (a) is due to induction hypothesis, and (b) is due to submodularity of function g_β.

Therefore, Algorithm 1 would lead to the rate vector that belongs to the base polyhedron $B(g_\beta, \leq)$. It remains to prove that Algorithm 1 minimizes the cost $\sum_{i=1}^m \alpha_i R_i$. We show this by proving (by induction) that each iteration i of Edmonds' algorithm outputs the rate vector that is optimal w.r.t. the cost $\sum_{j=1}^i \alpha_j R_j$. Since we assumed that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_m$, for $i = 2$,

$$\begin{aligned} R_1 &= g_\beta(\{1\}), \\ R_2 &= g_\beta(\{1,2\}) - g_\beta(\{1\}), \end{aligned} \tag{2.54}$$

is an optimal rate allocation. To see this, we first observe that R_1 cannot be increased furthermore. Therefore, the only manipulation we can perform over vector (R_1, R_2) is to decrease R_1 by some amount while increasing R_2 by the same amount. More formally, if $\text{dep}(\mathbf{R}, 2) = \{1\}$, let $d \in \mathbb{Z}_+$ be such that $(R_1 - d, R_2 + d) \in B(g_\beta, \leq)$. Since $\alpha_1 \leq \alpha_2$, it immediately follows that

$$\alpha_1 R_1 + \alpha_2 R_2 \leq \alpha_1 (R_1 - d) + \alpha_2 (R_2 + d). \tag{2.55}$$

Now, let Algorithm 1 be optimal at iteration $i = m - 1$. After one more iteration, we obtain

$$R_m = g_\beta(\mathcal{M}) - g_\beta(\mathcal{M} \setminus \{m\}). \tag{2.56}$$

Since $\sum_{j=1}^{m-1} \alpha_j R_j$ is the minimal cost, any attempt to change vector components

$$(R_1, R_2, \ldots, R_{m-1})$$

would lead to the higher cost. Therefore, we only need to check whether the increase of R_m and decrease of one of the components in $\mathcal{M} \setminus \{m\}$ by the same amount would lead to the lower cost. More formally, for $k \in \mathcal{M} \setminus \{m\}$, if $k \in \text{dep}(\mathbf{R}, m)$, let $d_k \in \mathbb{Z}_+$ be such that

$$\mathbf{R} + d_k(\mathbf{e}(m) - \mathbf{e}(k)) \in B(g_\beta, \leq), \quad \forall k \in \mathcal{M} \setminus \{m\}. \tag{2.57}$$

Since $\alpha_k \leq \alpha_m$, $\forall k \in \mathcal{M} \setminus \{m\}$, it follows that for any d_k that satisfies (2.57) we have

$$\sum_{j=1}^{m} \alpha_j R_j \leq \sum_{j=1, j\neq k}^{m-1} \alpha_j R_j + \alpha_i(R_k - d_k) + \alpha_m(R_m + d_k). \tag{2.58}$$

Therefore, Algorithm 1 computes an optimal rate allocation w.r.t. the linear cost.

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 (2.38). 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 (2.21)), 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 $P(g_\beta)$. Since we don't explicitly have function g_β, this polyhedron boundary can be calculated by applying the Dilworth truncation formula (2.38). For the three-user problem in Example 3 this procedure would go as follows

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

Generalization of this procedure to an arbitrary number of users is straightforward (see Algoruthm 3). We refer the interested reader to the references [14] and [31] where this algorithm is explained in more details for an arbitrary intersecting submodular functions.

In each iteration i, the minimization problem (2.59) is over all subsets of $\{j(1), \ldots, j(i)\}$. Using the fact that all the subsets considered in (2.59) 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 $\{j(1), j(2), \ldots, j(i)\}$. Now the polynomial time solution of Algorithm 3 follows from the fact that minimization of a fully submodular function can be done in polynomial time [33].

Algorithm 3 Modified Edmonds' Algorithm

1: Set $j(1), j(2), \ldots, j(m)$ to be an ordering of $1, 2, \ldots, m$, such that

$$\alpha_{j(1)} \leq \alpha_{j(2)} \leq \cdots \leq \alpha_{j(m)}.$$

2: Initialize $\mathbf{R} = \mathbf{0}$.
3: **for** $i = 1$ to m **do**
4:

$$R_{j(i)} = \min_{\mathcal{S}} \{f_\beta(\mathcal{S} \cup j(i)) - R(\mathcal{S}) : \mathcal{S} \subseteq \{j(1), j(2), \ldots, j(i-1)\}\}. \tag{2.59}$$

5: **end for**
6: $\mathbf{R}^* = \mathbf{R}$ is an optimal rate vector w.r.t. the problem (2.43).

Remark 5. The complexity of Algorithm 3 is $\mathcal{O}(m \cdot SFM(m))$, where $SFM(m)$ is the complexity of minimizing submodular function. The best known algorithm to our knowledge is proposed by Orlin in [33], and has complexity $\mathcal{O}(m^5 \cdot \gamma + m^6)$, where γ is complexity of computing the submodular function. For the submodular function defined in (2.59), 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 $\mathcal{O}(N)$.

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 3 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 (2.34). Hence, for any given β, the feasibility of such sum-rate can be verified in $\mathcal{O}(m \cdot SFM(m))$ time.

2.3 Finding the Optimal Value of β

So far we have shown how to compute function $h(\beta)$ defined in (2.18) for any β when $\varphi_i(R_i) = \alpha_i R_i$. To complete our solution, *i.e.*, to solve the problem defined in (2.17), it remains to show how to minimize function $h(\beta)$ efficiently.

Theorem 4. *Function $h(\beta)$, defined in (2.18), is convex when β is a feasible sum-rate w.r.t. the optimization problem (2.18).*

Proof. Consider two feasible sum-rates β_1 and β_2 w.r.t. the problem (2.18). 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. Note that

$$\begin{aligned}
&\lambda h(\beta_1) + (1-\lambda)h(\beta_2) \\
&= \sum_{i=1}^{m} \left(\lambda \varphi_i(R_i^{(1)}) + (1-\lambda)\varphi_i(R_i^{(2)}) \right) \\
&\overset{(a)}{\geq} \sum_{i=1}^{m} \varphi_i(\lambda R_i^{(1)} + (1-\lambda) R_i^{(2)}) = \sum_{i=1}^{m} \varphi_i(R_i^{(\lambda)}),
\end{aligned} \tag{2.60}$$

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 (2.18) 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

$$\begin{aligned}
R^{(\lambda)}(\mathcal{M}) &= \lambda R^{(1)}(\mathcal{M}) + (1-\lambda) R^{(2)}(\mathcal{M}) \\
&= \lambda\beta_1 + (1-\lambda)\beta_2.
\end{aligned} \tag{2.61}$$

Since

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

we have

$$\begin{aligned}
R^{(\lambda)}(\mathcal{S}) &= \lambda R^{(1)}(\mathcal{S}) + (1-\lambda) R^{(2)}(\mathcal{S}) \\
&\geq N - \operatorname{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}), \quad \forall \mathcal{S} \subset \mathcal{M}.
\end{aligned} \tag{2.62}$$

From (2.61) and (2.62) it follows that $\mathbf{R}^{(\lambda)}$ is a feasible *DE*-rate vector w.r.t. optimization problem (2.18) 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 (2.60), it follows that

$$h(\lambda\beta_1 + (1-\lambda)\beta_2) \leq \lambda h(\beta_1) + (1-\lambda)h(\beta_2), \tag{2.63}$$

which completes the proof. □

In order to minimize function h, first, we identify the set of sum-rates β that are feasible w.r.t. the problem (2.17). 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 3 provides an optimal rate allocation w.r.t. the linear cost. It is only left to find β that minimizes $h(\beta)$ in (2.17). Let us first consider the sum-rate cost, *i.e.*, $\varphi_i(R_i) = R_i$. From the equivalence of the Algorithms 1 and 3, and from Remark 3 it follows that for any given parameter β, the output rate vector $\mathbf{R}^*$ of Algorithm 3 satisfies

$$\sum_{i=1}^{m} R_i^* = g_\beta(\mathcal{M}). \tag{2.64}$$

Thus, for a randomly chosen parameter β we can verify whether it is feasible w.r.t. the problem (2.18) 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 4).

Algorithm 4 Minimum Sum-Rate Algorithm (binary search)

Initialize $\beta_{start} = 0$, $\beta_{end} = N$.
while $\beta_{end} - \beta_{start} > 1$ **do**
  $\beta = \lceil \frac{\beta_{start}+\beta_{end}}{2} \rceil$.
  Execute Algorithm 3 with parameter β.
  if $\sum_{i=1}^{m} R_i^* = \beta$, **then**
    $\beta_{end} = \beta$.
  else $\beta_{start} = \beta$.
end while
β_{end} is the minimum sum-rate.

Remark 7. The complexity of Algorithm 4 is $\mathcal{O}(m \cdot SFM(m) \cdot \log N)$.

For the general linear cost function $\varphi_i(R_i) = \alpha_i R_i$, we showed in Theorem 4 that $h(\beta)$ is convex for β greater than the minimum sum-rate (obtained from Algorithm 4). In Section 2.4.1, Lemma 8, 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 (2.17) 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 5 Minimum Linear Cost Algorithm

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

Remark 8. Since for any fixed β, $h(\beta)$ can be found by using Algorithm 3, and β_{sum}^* can be found by applying Algorithm 4, the complexity of Algorithm 5 is $\mathcal{O}(m \cdot SFM(m) \cdot \log N)$.

2.4 Using Subgradient Methods to Solve Step 4 of Algorithm 3

In this section we propose an alternative solution to the minimization problem (2.59) in Algorithm 3 that does not involve minimization of a submodular function. The underlying linear optimization problem has the following form

$$\min_{\mathbf{R}\in\mathbb{Z}^m} \sum_{i=1}^{m} \alpha_i R_i, \quad \text{s.t. } \mathbf{R} \in B(f_\beta, \leq), \tag{2.65}$$

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 3 of Algorithm 3 can be written as

$$R_i^* = \min_{\mathcal{S}} \{f_\beta(\mathcal{S}) - R(\mathcal{S}) : i \in \mathcal{S}, \quad \mathcal{S} \subseteq \{1, 2, \ldots, i\}\}, \quad i = 1, 2, \ldots, m. \tag{2.66}$$

Minimization (2.66) 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 minimization problem

$$\begin{aligned} R_i^* &= \max_{\mathbf{R}\in\mathbb{R}^i} R_i, \\ \text{s.t. } & R_k \geq R_k^*, \ k = 1, 2, \ldots, i-1, \\ & R(\mathcal{S} \cup \{i\}) \leq f_\beta(\mathcal{S} \cup \{i\}), \quad \forall \mathcal{S} \subseteq \{1, 2, \ldots, i-1\}. \end{aligned} \tag{2.67}$$

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 (2.67) and (2.66) are equivalent.

Let us denote by $\mathcal{R}^{(i)}$ the rate region that corresponds to the optimization problem (2.67)

$$\begin{aligned} \mathcal{R}^{(i)} = \{\mathbf{R} \in \mathbb{R}^i \mid & R(\mathcal{S} \cup \{i\}) \leq f_\beta(\mathcal{S} \cup \{i\}), \\ & \forall \mathcal{S} \subseteq \{1, 2, \ldots, i-1\}\}. \end{aligned} \tag{2.68}$$

To solve optimization problem (2.67), we apply the dual subgradient method. First, the Lagrangian function of the problem (2.67) is

$$\mathcal{L}(\mathbf{R}, \boldsymbol{\lambda}) = R_i + \sum_{k=1}^{i-1} \lambda_k (R_k - R_k^*), \quad \mathbf{R} \in \mathcal{R}^{(i)}, \tag{2.69}$$

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

$$\begin{aligned} \delta(\boldsymbol{\lambda}) &= \max_{\mathbf{R}\in\mathcal{R}^{(i)}} \mathcal{L}(\mathbf{R}, \lambda) \\ &= \max_{\mathbf{R}\in\mathcal{R}^{(i)}} \left\{ R_i + \sum_{k=1}^{i-1} \lambda_k R_k \right\} - \sum_{k=1}^{i-1} \lambda_k R_k^*. \end{aligned} \tag{2.70}$$

Due to the maximization step in (2.70) over multiple hyper-planes, it immediately follows that $\delta(\boldsymbol{\lambda})$ is a convex function. By the weak duality theorem [3],

$$\delta(\lambda) \geq R_i^*, \quad \forall \lambda_k \geq 0, \ k = 1, 2, \ldots, i-1. \tag{2.71}$$

Hence,

$$\min_{\boldsymbol{\lambda}} \{\delta(\boldsymbol{\lambda}) \mid \lambda_k \geq 0, \ k = 1, 2, \ldots, i-1\} \geq R_i^* \tag{2.72}$$

Since optimization problem (2.67) is linear, there is no duality gap, *i.e.*,

$$R_i^* = \min_{\boldsymbol{\lambda}} \{\delta(\boldsymbol{\lambda}) \mid \lambda_k \geq 0, \ k = 1, 2, \ldots, i-1\}. \tag{2.73}$$

To solve optimization problem (2.73), we apply the dual subgradient method [4] as follows. Starting with a feasible iterate $\lambda_k[0]$, $k = 1, 2, \ldots, i-1$, w.r.t. the optimization problem (2.73), 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

$$\lambda_k[j+1] = \left\{\lambda_k[j] - \theta_j(\tilde{R}_k[j] - R_k^*)\right\}_+, \tag{2.74}$$

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

$$\max_{\mathbf{R} \in \mathcal{R}^{(i)}} R_i + \sum_{k=1}^{i-1} \lambda_k[j] R_k. \tag{2.75}$$

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

Lemma 4. *An optimal solution to the problem (2.75) 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,*

$$\begin{aligned} \tilde{R}_i[j] &= \begin{cases} f_\beta(\{i\}), & \text{if } \lambda_{t(1)} \leq 1, \\ 0, & \text{otherwise.} \end{cases} \\ \tilde{R}_{t(k)} &= f_\beta(\mathcal{S}_{t(k)} \cup \{i\}) - \sum_{u=1}^{k-1} \tilde{R}_{t(u)}[j] - \tilde{R}_i[j], \ k = 1, 2, \ldots, i-1, \end{aligned} \tag{2.76}$$

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

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

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

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

$$R(\mathcal{S} \cup \{i\}) \leq g(\mathcal{S} \cup \{i\}) = f_\beta(\mathcal{S} \cup \{i\}). \tag{2.77}$$

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

Now, let $\mathbf{R} \in \mathcal{R}^{(i)}$. From (2.68) we have

$$R(\mathcal{S} \cup \{i\}) \leq f_\beta(\mathcal{S} \cup \{i\}) = g(\mathcal{S} \cup \{i\}), \quad \forall \mathcal{S} \subseteq \{1, 2, \ldots, i-1\} \tag{2.78}$$

Since the rate vector is positive, (2.78) implies that

$$R(\mathcal{S}) \leq f_\beta(\mathcal{S} \cup \{i\}) = g(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \{1, 2, \ldots, i-1\}. \tag{2.79}$$

From (2.78) and (2.79) it follows that $\mathbf{R} \in P(g, \leq)$. Hence, $\mathcal{R}^{(i)} = P(g, \leq)$.

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

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

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

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

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

$$\begin{aligned} g(\mathcal{S}) + g(\mathcal{T}) &= f_\beta(\mathcal{S} \cup \{i\}) + f_\beta(\mathcal{T} \cup \{i\}) \\ &\geq f_\beta(\mathcal{S} \cup \mathcal{T} \cup \{i\}) + f_\beta((\mathcal{S} \cap \mathcal{T}) \cup \{i\}) \\ &= g(\mathcal{S} \cup \mathcal{T}) + g(\mathcal{S} \cap \mathcal{T}). \end{aligned}$$

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

$$\begin{aligned} g(\mathcal{S}) + g(\mathcal{T}) &= f_\beta(\mathcal{S}) + f_\beta(\mathcal{T}) \\ &\geq f_\beta(\mathcal{S} \cup \mathcal{T}) + f_\beta(\mathcal{S} \cap \mathcal{T}) \\ &= g(\mathcal{S} \cup \mathcal{T}) + g(\mathcal{S} \cap \mathcal{T}). \end{aligned}$$

Therefore, function g is indeed fully submodular. Hence, problem (2.75) is a linear optimization problem over a submodular polyhedron, and it can be solved via Edmonds' algorithm.

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

$$\begin{aligned}\tilde{R}_i[j] &= g(\{i\}) = f_\beta(\{i\}),\\ \tilde{R}_{t(k)}[j] &= g(\mathcal{S}_{t(k)} \cup \{i\}) - g(\mathcal{S}_{t(k-1)} \cup \{i\})\\ &= f_\beta(\mathcal{S}_{t(k)} \cup \{i\}) - \sum_{u=1}^{k-1} \tilde{R}_{t(u)}[j] - \tilde{R}_i[j],\ k = 1, 2, \ldots, i-1. \end{aligned} \tag{2.80}$$

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

$$\begin{aligned}\tilde{R}_i[j] &= g(\mathcal{S}_{t(r)} \cup \{i\}) - g(\mathcal{S}_{t(r)}) = 0,\\ \tilde{R}_{t(k)}[j] &= g(\mathcal{S}_{t(k)} \cup \{i\}) - g(\mathcal{S}_{t(k-1)} \cup \{i\})\\ &= f_\beta(\mathcal{S}_{t(k)} \cup \{i\}) - \sum_{u=1}^{k-1} \tilde{R}_{t(u)}[j],\ k = 1, 2, \ldots, i-1. \end{aligned} \tag{2.81}$$

This completes the proof of this lemma. □

Remark 9. The complexity of the algorithm proposed by Lemma 4 is $\mathcal{O}(i \log i + i \cdot N^3)$.

The reason why we apply subgradient methods instead of a gradient descent is because function $\delta(\boldsymbol{\lambda})[j]$ even though convex, is not differentiable. From Lemma 4, it follows that for a given $\boldsymbol{\lambda}[j]$, there may be more than one maximizer of the problem (2.75). Due to possibility of having more than one direction along which we can update vector $\boldsymbol{\lambda}[j]$ according to (2.74), 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

$$\tilde{\boldsymbol{\lambda}}[j] = \operatorname{argmin}\{\delta(\boldsymbol{\lambda}[0]), \delta(\boldsymbol{\lambda}[1]), \ldots, \delta(\boldsymbol{\lambda}[j])\}. \tag{2.82}$$

Before we go any further, note that the primal optimization problem (2.67) is over real vectors. However, the minimization (2.65) is an integer optimization problem. As pointed out above, the optimal solution of the problem (2.67) is equal to the solution of the problem (2.65). 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,

$$\left|\delta(\tilde{\boldsymbol{\lambda}}[l]) - R_i^*\right| \leq \varepsilon, \tag{2.83}$$

where $\varepsilon < 0.5$. Then,

$$R_i^* = \operatorname{round}(\delta(\tilde{\boldsymbol{\lambda}}[l])). \tag{2.84}$$

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 (2.84) provides the optimal solution. Following the notes on subgradient methods provided in [4], let $\boldsymbol{\lambda}^*$ be an arbitrary optimal vector that minimizes the dual function δ. Then,

$$\begin{aligned}
&\sum_{k=1}^{i-1}(\lambda_k[j+1]-\lambda_k^*)^2 = \sum_{k=1}^{i-1}\left(\left\{\lambda_k[j]-\theta_j(\tilde{R}_k[j]-R_k^*)\right\}_+ - \lambda_k^*\right)^2 \\
&\leq \sum_{k=1}^{i-1}\left(\lambda_k[j]-\theta_j(\tilde{R}_k[j]-R_k^*)-\lambda_k^*\right)^2 \\
&= \sum_{k=1}^{i-1}(\lambda_k[j]-\lambda_k^*)^2 - 2\theta_j\sum_{k=1}^{i-1}(\tilde{R}_k[j]-R_k^*)(\lambda_k[j]-\lambda_k^*) + \theta_j^2\sum_{k=1}^{i-1}\left(\tilde{R}_k[j]-R_k^*\right)^2 \\
&\leq \sum_{k=1}^{i-1}(\lambda_k[j]-\lambda_k^*)^2 - 2\theta_j\left(\delta(\boldsymbol{\lambda}[j])-\delta(\boldsymbol{\lambda}^*)\right) + \theta_j^2\sum_{k=1}^{i-1}\left(\tilde{R}_k[j]-R_k^*\right)^2, \qquad (2.85)
\end{aligned}$$

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

$$\delta(\boldsymbol{\lambda}[j])-\delta(\boldsymbol{\lambda}^*) \leq \sum_{k=1}^{i-1}(\tilde{R}_k[j]-R_k^*)(\lambda_k[j]-\lambda_k^*), \qquad (2.86)$$

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 (2.85) over j from 0 to $l-1$, we obtain

$$\begin{aligned}
\sum_{k=1}^{i-1}(\lambda_k[l]-\lambda_k^*)^2 \leq \sum_{k=1}^{i-1}(\lambda_k[0]-\lambda_k^*)^2 - 2\sum_{j=0}^{l-1}\theta_j\left(\delta(\boldsymbol{\lambda}[j])-\delta(\boldsymbol{\lambda}^*)\right) \\
+\sum_{j=0}^{l-1}\theta_j^2\sum_{k=1}^{i-1}\left(\tilde{R}_k[j]-R_k^*\right)^2. \qquad (2.87)
\end{aligned}$$

Therefore,

$$2\sum_{j=0}^{l-1}\theta_j\left(\delta(\boldsymbol{\lambda}[j])-\delta(\boldsymbol{\lambda}^*)\right) \leq \sum_{k=1}^{i-1}(\lambda_k[0]-\lambda_k^*)^2 + \sum_{j=0}^{l-1}\theta_j^2\sum_{k=1}^{i-1}\left(\tilde{R}_k[j]-R_k^*\right)^2. \qquad (2.88)$$

Since,

$$\sum_{j=0}^{l-1}\theta_j\left(\delta(\boldsymbol{\lambda}[j])-\delta(\boldsymbol{\lambda}^*)\right) \geq \sum_{j=0}^{l}\theta_j \min_{j\in\{0,1,\ldots,l-1\}}\left(\delta(\boldsymbol{\lambda}[j])-\delta(\boldsymbol{\lambda}^*)\right), \qquad (2.89)$$

from (2.88) and (2.82) we obtain

$$\begin{aligned}
\delta(\tilde{\boldsymbol{\lambda}}[l-1]) - \delta(\boldsymbol{\lambda}^*) &= \min_{j\in\{0,1,\ldots,l-1\}} \delta(\boldsymbol{\lambda}[j]) - \delta(\boldsymbol{\lambda}^*) \\
&\leq \frac{\sum_{k=1}^{i-1}(\lambda_k[0]-\lambda_k^*)^2 + \sum_{j=0}^{l-1}\theta_j^2 \sum_{k=1}^{i-1}\left(\tilde{R}_k[j]-R_k^*\right)^2}{2\sum_{j=0}^{l-1}\theta_j} \\
&\leq \frac{\sum_{k=1}^{i-1}(\lambda_k[0]-\lambda_k^*)^2 + \sum_{j=0}^{l-1}\theta_j^2 \left(\sum_{k=1}^{i-1}\left(\tilde{R}_k[j]\right)^2 + \sum_{k=1}^{i-1}(R_k^*)^2\right)}{2\sum_{j=0}^{l}\theta_j} \\
&\leq \frac{\sum_{k=1}^{i-1}(\lambda_k[0]-\lambda_k^*)^2 + \sum_{j=0}^{l-1}\theta_j^2 \left(\left(\sum_{k=1}^{i-1}\tilde{R}_k[j]\right)^2 + \left(\sum_{k=1}^{i-1}R_k^*\right)^2\right)}{2\sum_{j=0}^{l-1}\theta_j} \\
&\leq \frac{\sum_{k=1}^{i-1}(\lambda_k[0]-\lambda_k^*)^2 + 2N^2\sum_{j=0}^{l-1}\theta_j^2}{2\sum_{j=0}^{l-1}\theta_j}, \qquad (2.90)
\end{aligned}$$

where the last inequality holds because $R(\mathcal{M}) \leq f_\beta(\mathcal{M}) \leq N$ for any achievable DE-rate vector $\mathbf{R}$. Continuing with (2.90), we have

$$\delta(\tilde{\boldsymbol{\lambda}}[l-1]) - \delta(\boldsymbol{\lambda}^*) \leq \frac{\left(\sum_{k=1}^{i-1}\lambda_k[0]\right)^2 + \left(\sum_{k=1}^{i-1}\lambda_k^*\right)^2 + 2N^2\sum_{j=0}^{l-1}\theta_j^2}{2\sum_{j=0}^{l-1}\theta_j}. \qquad (2.91)$$

Since $\boldsymbol{\lambda}^*$ is an arbitrary minimizer of the dual function δ, let us pick $\boldsymbol{\lambda}^*$ as suggested by the following lemma.

Lemma 5. *There exists an optimal solution to the problem (2.73) that satisfies*

$$\sum_{k=1}^{i-1}\lambda_k^* \leq m. \qquad (2.92)$$

Proof. For any $\boldsymbol{\lambda}^*$, let us denote by $\tilde{\mathbf{R}}$ an optimal solution of the problem (2.70) obtained by applying Lemma 4. 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

$$\sum_{k=1}^{i-1}\tilde{R}_k - R_k^* \geq R_i^* - \tilde{R}_i. \qquad (2.93)$$

The minimum value of the dual function δ is R_i^*. Therefore,

$$\sum_{k=1}^{i-1}\lambda_k^*(\tilde{R}_k - R_k^*) = R_i^* - \tilde{R}_i. \qquad (2.94)$$

From Algorithm 3 and Theorem 2, it follows that

$$\sum_{k=1}^{i} R_i^* = \min_{\mathcal{P}} \left\{ \sum_{\mathcal{S}\in\mathcal{P}} f_\beta(\mathcal{S}) : \mathcal{P} \text{ is a partition of } \{1,2,\ldots,i\} \right\}. \tag{2.95}$$

Let us denote by $\mathcal{S}_i^*$, a set that belongs to an optimal partitioning $\mathcal{P}^*$ w.r.t. problem (2.95) such that $i \in \mathcal{S}_i^*$. In this case, we have

$$\sum_{k\in\mathcal{S}_i^*} R_k^* = f_\beta(\mathcal{S}_i^*). \tag{2.96}$$

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

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

To verify that this choice of $\boldsymbol{\lambda}^*$ is indeed a dual optimal solution, note that from Lemma 4, we have

$$\sum_{k\in\mathcal{S}_i^*} \tilde{R}_k = f_\beta(\mathcal{S}_i^*). \tag{2.97}$$

Therefore,

$$\sum_{k\in\mathcal{S}_i^*} \tilde{R}_k - R_k^* = 0. \tag{2.98}$$

Expression (2.94) can be rewritten as

$$\sum_{k\notin\mathcal{S}_i^*} \lambda_k^*(\tilde{R}_k - R_k^*) = R_i^* - \tilde{R}_i + \sum_{k\in\mathcal{S}_i^*\setminus\{i\}} \lambda_k^*(\tilde{R}_k - R_k^*). \tag{2.99}$$

From (2.97) and (2.98), it follows that both sides of equality are equal to 0, and thus $\boldsymbol{\lambda}^*$ is indeed a dual optimal solution. Hence,

$$\sum_{k=1}^{i-1} \lambda_k^* \le i - 1 \le m. \tag{2.100}$$

□

Initial, feasible $\boldsymbol{\lambda}[0]$ can be chosen as follows

$$\lambda_k[0] = 0, \quad \forall k \in \{1,2,\ldots,i-1\}. \tag{2.101}$$

Combining (2.91), (2.92) and (2.102), we obtain

$$\delta(\tilde{\boldsymbol{\lambda}}[l-1]) - \delta(\boldsymbol{\lambda}^*) \leq \frac{m^2 + 2N^2 \sum_{j=0}^{l-1} \theta_j^2}{2 \sum_{j=0}^{l-1} \theta_j}. \tag{2.102}$$

There are many ways to choose the step size that satisfies (2.102). Here, we briefly examine the constant step size, where $\theta_j = \theta$, $j = 0, 1, 2, \ldots$. For other choices on selecting an appropriate step size θ_j, we refer the interested reader to the notes on subgradient methods. When $\theta_j = \theta$, the inequality (2.102) becomes

$$\delta(\tilde{\boldsymbol{\lambda}}[l-1]) - \delta(\boldsymbol{\lambda}^*) \leq \frac{m^2 + 2N^2 l\theta^2}{2l\theta}. \tag{2.103}$$

Hence, the condition (2.83) is satisfied when

$$\frac{m^2 + 2N^2 l\theta^2}{2l\theta} < \frac{1}{2}. \tag{2.104}$$

It can be easily verified that (2.104) holds when

$$\theta < \frac{1}{2N^2}, \tag{2.105}$$

$$l > \frac{m^2}{\theta(1 - 2N^2\theta)}. \tag{2.106}$$

Putting all these results together, the minimization (2.66) can be obtained by running Algorithm 6.

Remark 10. From Remark 9 it follows that the complexity of Algorithm 6 is $SFM(m) = \mathcal{O}(lm \log m + lm\gamma)$. For a constant step size θ, from (2.105) and (2.106) it follows that the complexity of Algorithm 6 can be bounded by $\mathcal{O}(N^2 m^3 \log m + N^2 m^3 \gamma)$.

Remark 11. Note that Algorithm 6 can be applied to solve problem (2.65) when f_β is an arbitrary intersecting submodular function over integers.

2.4.1 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 $P(g_\beta, \leq)$. This effectively means that Edmonds' algorithm provides rate allocations that are vertices of the base polyhedron $B(g_\beta, \leq)$. 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 $B(g_\beta, \leq)$. We will show this in Example 4.

Algorithm 6 Minimization (2.66) of Algorithm 3

1: Select parameters l, and θ_j, $j = 0, 1, \ldots, l-1$ such that

$$\frac{m^2 + 2N^2 \sum_{j=0}^{l-1} \theta_j^2}{2 \sum_{j=0}^{l-1} \theta_j} < \frac{1}{2}. \tag{2.107}$$

2: Set $\lambda_k[0] = 0$, $k = 1, 2, \ldots, i-1$, and $\tilde{\boldsymbol{\lambda}}[0] = \boldsymbol{\lambda}[0]$.
3: **for** $j = 0$ to $l-1$ **do**
4:

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

where $\tilde{\mathbf{R}}[j]$ is computed according to Lemma 4.

5:

$$\tilde{\boldsymbol{\lambda}}[j+1] = \operatorname{argmin} \left\{ \delta(\boldsymbol{\lambda}[j+1]), \delta(\tilde{\boldsymbol{\lambda}}[j]) \right\}.$$

6: **end for**
7:

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

The general convex cost optimization problem

$$\min_{\mathbf{R}\in\mathbb{Z}^m} \sum_{i=1}^{m} \varphi_i(R_i), \quad \text{s.t.} \quad \mathbf{R} \in B(g_\beta, \leq) \tag{2.108}$$

is known as a *resource allocation problem under submodular constraints* [20], 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 7).

Algorithm 7 Minimizing separable convex cost under submodular constraints

1: Set $R_i = 0$, $\forall i \in \mathcal{M}$
2: **for** $j = 1$ to β **do**
3: Find $i^* \in \mathcal{M}$ such that

$$i^* = \underset{i\in\mathcal{M}}{\operatorname{argmin}} \left\{ d_i(R_i + 1) \mid \mathbf{R} + \mathbf{e}(i) \in P(g_\beta, \leq) \right\},$$

where $d_i(R_i+1) \triangleq \varphi_i(R_i+1) - \varphi_i(R_i)$, and $\mathbf{e}(i)$ is the unit basis m-dimensional vector with i^{th} coordinate equals to 1.
4: Set $R_{i^*} = R_{i^*} + 1$.
5: **end for**
6: $\mathbf{R}^* = \mathbf{R}$ is an optimal rate vector w.r.t. the problem (2.108).

Definition 8. Let us define set $\mathcal{T}_j$ to be the set of all users that are in iteration j of Algorithm 7 allowed to update their transmission rates

$$\mathcal{T}_j \triangleq \left\{ i \mid \mathbf{R} + \mathbf{e}(i) \in P(g_\beta, \leq) \right\}. \tag{2.109}$$

The question is how to efficiently recover set $\mathcal{T}_j$ in each round of Algorithm 7. First, we observe that $P(g_\beta, \leq) = P(f_\beta, \leq)$ according to Theorem 2. Second, note that in Algorithm 3, the minimization (2.59) 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

$$\min_{\mathcal{S}\subseteq\mathcal{M}\setminus\{i\}} \left\{ f_\beta(\mathcal{S}\cup\{i\}) - R(\mathcal{S}\cup\{i\}) \right\} \geq 1. \tag{2.110}$$

Putting these results together, we can obtain a polynomial time solution to problem (2.18) by applying Algorithm 8.

The complexity of (2.110) is $SFM(m)$, since the function $f_\beta(\mathcal{S}) - R^*(\mathcal{S})$ is fully submodular. This check can be done either by minimizing submodular function as suggested

Algorithm 8 Minimizing separable convex cost under intersecting submodular constraints

1: Set $R_i = 0$, $\forall i \in \mathcal{M}$
2: **for** $j = 1$ to β **do**
3: Construct set $\mathcal{T}_j$ as follows

$$\mathcal{T}_j = \left\{ i : \min_{\mathcal{S} \subseteq \mathcal{M} \setminus \{i\}} \{ f_\beta(\mathcal{S} \cup \{i\}) - R(\mathcal{S} \cup \{i\}) \} \geq 1 \right\}. \tag{2.111}$$

4: Find $i^* \in \mathcal{T}_j$ such that

$$i^* = \underset{i \in \mathcal{T}_j}{\operatorname{argmin}} \{ d_i(R_i + 1) \}.$$

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 (2.108).

in (2.110) or by running the dual subgradient algorithm similar to the one proposed in Section 2.4. Here, we briefly explain the differences. First, rate region $\mathcal{R}^{(i)}$ defined in (2.68), now has the following form

$$\begin{aligned} \mathcal{R}^{(i)} = \{ \mathbf{R} \in \mathbb{R}^m \mid & R(\mathcal{S} \cup \{i\}) \leq f_\beta(\mathcal{S} \cup \{i\}), \\ & \forall \mathcal{S} \subseteq \mathcal{M} \setminus \{i\} \}. \end{aligned} \tag{2.112}$$

Let us denote by $\mathbf{R}^* \in \mathbb{R}^m$ the current rate allocation in round j of Algorithm 8. Then, if the maximization

$$\max_{\mathbf{R} \in \mathcal{R}^{(i)}} R_i, \tag{2.113}$$

$$\text{s.t. } R_k \geq R_k^*, \ k = 1, 2, \ldots, m, \tag{2.114}$$

is at least 1, then $i \in \mathcal{T}_j$. Problem (2.113) can be solved by following the same steps in solving the dual problem as in Section 2.4.

At each iteration, Algorithm 8 calls (2.111) m times, and there are total of β iterations, which is of the order of the file size N.

Remark 12. The complexity of Algorithm 8 is $\mathcal{O}(m \cdot N \cdot SFM(m))$.

2.4.2 Proof of Correctness of Algorithm 7

Now we provide a sketch of proof of optimality of Algorithm 7 based on Chapters 4 and 9 in [20]. Notice that from time to time, after increasing R_k by one for some $k \in \mathcal{M}$ in Step 4,

some set $\mathcal{S} \subseteq \mathcal{M}$ becomes newly saturated, *i.e.*,

$$\exists \mathcal{S} \subseteq \mathcal{M}, \text{ s.t. } R(\mathcal{S}) = g_\beta(\mathcal{S}). \tag{2.115}$$

Once (2.115) occurs for some $k \in \mathcal{M}$, it never occurs again for the same k. The sets that were saturated before this update remain saturated. Moreover, observe that using Lemma 3, we can form the maximal size saturated set that includes k as follows

$$\mathcal{S}_k = \bigcup_{R(\mathcal{S})=g_\beta(\mathcal{S})} \mathcal{S}. \tag{2.116}$$

For the current rate vector $\mathbf{R}$ note that by the proof of Theorem 3, it follows that $\text{sat}(\mathbf{R}) = \mathcal{S}_k$. It is important to mention that saturation may not happen in each round of Algorithm 7. Let there be p such instances, where $p = m$ if no vector components were saturated before execution of Algorithm 7, *i.e.*, if $\text{sat}(\mathbf{0}) = \emptyset$. Without loss of generality, we denote by $\{\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_p\}$, the ordered set of the maximal size saturated sets which occur while executing Algorithm 7, where $\mathcal{S}_0 = \text{sat}(\mathbf{R})$, when $\mathbf{R} = \mathbf{0}$. Obviously, $\mathcal{S}_p = \mathcal{M}$. Also, it is easy to show that

$$R^*(\mathcal{S}_i) = g_\beta(\mathcal{S}_i), \quad i = 1, 2, \ldots, p, \tag{2.117}$$

where $\mathbf{R}^*$ is an optimal rate vector obtained by applying Algorithm 7. From (2.115) it follows that

$$\mathcal{S}_0 \subset \mathcal{S}_1 \subset \cdots \subset \mathcal{S}_p. \tag{2.118}$$

Lemma 6. *The optimal rate vector* $\mathbf{R}^*$ *can be obtained by solving the following set of resource allocation problems*

$$\begin{aligned} &\min \sum_{i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}} \varphi_i(R_i) \\ &s.t. \sum_{i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}} R_i = g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1}), \end{aligned} \tag{2.119}$$

where $k = 1, 2, \ldots, p$.

Proof. From the previous discussion, we know that $i \in \mathcal{S}_k$, and $i \notin \mathcal{S}_{k-1}$, where $\mathcal{S}_{k-1} \subset \mathcal{S}_k$. Also, observe that before $\mathcal{S}_k$ gets saturated, $u \notin \text{sat}(\mathbf{R})$ holds for all $u \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}$. Therefore, all the elements $u \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}$ can be updated such that $\mathbf{R} + \mathbf{e}(u) \in P(g_\beta, \leq)$. Hence, in the Step 3 Algorithm 7 selects $i^* \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}$ such that

$$d_{i^*}(R_{i^*} + 1) = \min \{d_i(R_i + 1) \mid i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}\}. \tag{2.120}$$

Since $R^*(\mathcal{S}_k) = g_\beta(\mathcal{S}_k)$, and $R^*(\mathcal{S}_{k-1}) = g_\beta(\mathcal{S}_{k-1})$, we conclude that

$$R^*(\mathcal{S}_k \setminus \mathcal{S}_{k-1}) = g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1}). \tag{2.121}$$

This completes the proof. □

To complete the proof of correctness of Algorithm 7, it is left to show that subsequent application of (2.120) computes an optimal rate allocation for the problem (2.119). To prove this claim, let us consider the Lagrangian relaxation of the problem (2.119):

$$L(\lambda) = \min \sum_{i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}} \varphi_i(R_i) - \lambda \sum_{i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}} R_i, \tag{2.122}$$

where λ is a given real number. Let us denote by $\mathbf{R}^*_k(\lambda)$ an optimal solution of problem (2.122) for a given λ. Then, $\mathbf{R}^*_k(\lambda)$ is also a minimizer of the problem

$$\begin{aligned} \min & \sum_{i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}} \varphi_i(R_i) \\ \text{s.t.} & \sum_{i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}} R_i = \sum_{i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}} R^*_{k,i}(\lambda). \end{aligned} \tag{2.123}$$

In other words, if we find λ for which

$$\sum_{i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}} R^*_{k,i}(\lambda) = g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1}), \tag{2.124}$$

then $\mathbf{R}^*_k(\lambda)$ is an optimal solution of problem (2.119). To find such λ note that by the convexity of φ_i, d_i is a non-decreasing function. Hence,

$$d_i(1) \le d_i(2) \le \cdots \le d_i(g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1})).$$

Next, we observe that the objective function $L(\lambda)$ of (2.122) can be written as follows:

$$\sum_{i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}} (\varphi_i(R_i) - \lambda R_i) = \sum_{i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}} \left[\varphi_i(0) + (d_i(1) - \lambda) + \cdots + (d_i(R_i) - \lambda)\right]. \tag{2.125}$$

Let us denote by $\mathcal{D}_{min}$ the set of $g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1})$ smallest elements (discrete derivatives) in

$$\left\{ d_i(R_i) \mid i = 1, \ldots, m,\ R_i = 1, \ldots, g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1}) \right\}.$$

Then, by setting $\lambda = \lambda^*$ to be the largest $d_i(R_i)$ in $\mathcal{D}_{min}$, we have

$$\begin{aligned} d_i(R_i) - \lambda^* \le 0 \quad & \text{if } d_i(R_i) \in \mathcal{D}_{min}, \\ d_i(R_i) - \lambda^* \ge 0 \quad & \text{if } d_i(R_i) \notin \mathcal{D}_{min}. \end{aligned}$$

Therefore, problem (2.125) is minimized by the following rate vector $\mathbf{R}^*(\lambda)$

$$R^*_{k,i}(\lambda^*) = \begin{cases} 0 & \text{if } d_i(1) \notin \mathcal{D}_{min} \\ g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1}) & \text{if } d_i(g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1})) \in \mathcal{D}_{min} \\ R_i & \text{if } d_i(R_i) \in \mathcal{D}_{min}, \text{ and } d_i(R_i + 1) \notin \mathcal{D}_{min}. \end{cases}$$

Since

$$\sum_{i\in\mathcal{S}_k\setminus\mathcal{S}_{k-1}} R_i^*(\lambda) = |\mathcal{D}_{min}| = g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1}),$$

$\mathbf{R}_k^*(\lambda)$ is an optimal solution of the problem (2.119). Using these results, we can now show that Algorithm 9 computes an optimal rate allocation of problem (2.119).

Algorithm 9 Minimizing separable convex cost for the fixed sum-rate

1: Set $R_i = 0$, $\forall i \in \mathcal{M}$
2: **for** $j = 1$ to $g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1})$ **do**
3: Find $i^* \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}$ such that

$$i^* = \underset{i\in\mathcal{S}_k\setminus\mathcal{S}_{k-1}}{\operatorname{argmin}} \{d_i(R_i+1)\}.$$

4: Set $R_{i^*} = R_{i^*} + 1$.
5: **end for**
6: $\mathbf{R}_k^* = \mathbf{R}$ is an optimal rate vector w.r.t. the problem (2.119).

The initial rate vector in Step 1 of Algorithm 9 is set to zero for each $k = 1, \ldots, p$, by subtracting the optimal rate vector from the previous round $k-1$. The correctness of Algorithm 9 is obvious since Step 3 computes $g_\beta(\mathcal{S}_k) - g_\beta(\mathcal{S}_{k-1})$ smallest $d_i(R_i)$'s in non-decreasing order. To complete proof of correctness of Algorithm 7, it is left to show that Step 3 of Algorithm 7 and Step 3 of Algorithm 9 are equivalent. This is proved in the following lemma.

Lemma 7. *Let* $\mathbf{R}^*$ *be an output rate vector of Algorithm 7. Then,*

$$d_i(R_i^*) \le \min\{d_t(R_t^*+1) \mid t \in \mathcal{M}\setminus\mathcal{S}_{k-1}\}, \tag{2.126}$$

where $i \in \mathcal{S}_k \setminus \mathcal{S}_{k-1}$, *and* $k = 1, \ldots, p$.

Proof. Let $\hat{\mathbf{R}}$ denote the vector obtained just before $R_i = R_i^* - 1$ is increased in Step 3 of Algorithm 7. Since none of the sets $\mathcal{S}_k, \ldots \mathcal{S}_p$ are saturated, we have that $t \in \mathcal{M}\setminus\mathcal{S}_{k-1}$ satisfies $\hat{\mathbf{R}} + \mathbf{e}(t) \in P(g_\beta, \le)$. Therefore,

$$d_i(R_i^*) = d_i(\hat{R}_i + 1) = \min\left\{d_t(\hat{R}_t+1) \mid t \in \mathcal{M}\setminus\mathcal{S}_{k-1}\right\}.$$

This proves the lemma since $d_t(\hat{R}_t+1) \le d_t(R_t^*+1)$ for $t \in \mathcal{M}\setminus\mathcal{S}_{k-1}$ by $\hat{R}_t \le R_t^*$ and the convexity of φ_t. □

From Lemma 7 it follows that no new rate vectors are generated by expanding search space of Step 3 in Algorithm 9 from $\mathcal{S}_k \setminus \mathcal{S}_{k-1}$ to $\mathcal{M} \setminus \mathcal{S}_{k-1}$. Since sets $\mathcal{S}_1, \dots \mathcal{S}_{k-1}$ were already saturated in the previous iterations of Algorithm 7, we conclude that Step 3 of Algorithm 7 and Step 3 of Algorithm 9 are equivalent. This completes the proof of correctness of Algorithm 7.

Lemma 8. *Let us denote by β^* the minimizer of the function h defined in (2.18). Then, $\beta^* \leq N$.*

Proof. By Lemma 2 we know that set functions f_N and f_{N+1}, defined in (2.25), are fully submodular,

$$f_N(\mathcal{S}) = \begin{cases} \operatorname{rank}(\mathbf{A}_\mathcal{S}) & \text{if } \emptyset \neq \mathcal{S} \subseteq \mathcal{M}, \\ 0 & \text{if } \mathcal{S} = \emptyset. \end{cases} \tag{2.127}$$

$$f_{N+1}(\mathcal{S}) = \begin{cases} 1 + \operatorname{rank}(\mathbf{A}_\mathcal{S}) & \text{if } \emptyset \neq \mathcal{S} \subseteq \mathcal{M}, \\ 0 & \text{if } \mathcal{S} = \emptyset. \end{cases} \tag{2.128}$$

Let us denote by $\mathbf{R}^*$ an optimal vector obtained by applying Algorithm 7. In Section 2.2.3 we showed that all faces of a submodular polyhedron $P(g_\beta, \leq)$ are achievable, *i.e.*, for any $\mathcal{S} \subseteq \mathcal{M}$, there exists a rate vector $\mathbf{R}$ such that $R(\mathcal{S}) = g_\beta(\mathcal{S})$. Comparing f_N and f_{N+1}, we see that all "faces" of polyhedron $P(f_{N+1}, \leq)$ expended by 1 compared to polyhedron $P(f_N, \leq)$ (and they are all achievable). Hence, while applying Algorithm 7 for $\beta = N + 1$, we can see that the optimal rate vector $\tilde{\mathbf{R}}$ will differ from $\mathbf{R}^*$ in one coordinate:

$$\tilde{R}_j = \begin{cases} R_j^* + 1 & \text{if } j = \operatorname{argmin} \{d_i(R_i^* + 1) \mid \mathbf{R}^* + \mathbf{e}(i) \in P(f_{N+1}, \leq)\} \\ R_j^* & \text{otherwise.} \end{cases} \tag{2.129}$$

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

$$h(N) = \sum_{i=1}^{m} \varphi_i(R_i^*) = \sum_{i \neq j} \varphi_i(R_i^*) + \varphi_j(R_j^*). \tag{2.130}$$

$$h(N+1) = \sum_{i=1}^{m} \varphi_i(\tilde{R}_i) = \sum_{i \neq j} \varphi_i(R_i^*) + \varphi_j(R_j^* + 1). \tag{2.131}$$

Comparing (2.127) and (2.128), 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 4), it immediately follows that $\beta^* \leq N$. □

For the general non-decreasing set of functions φ_i, $i \in \mathcal{M}$, from Theorem 4 we know that function h is convex. Moreover, by Lemma 8 it follows that the minimizer of h is at most equal to N. Therefore, in order to minimize h, we can apply Algorithm 5 which calls Algorithm 7 approximately $\log N$ times. Thus, the overall complexity of the proposed solution is $\mathcal{O}(m \cdot SFM(m) \cdot N \log N)$.

2.4.3 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.

Example 4. Consider the same three-user problem as in Example 3

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

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

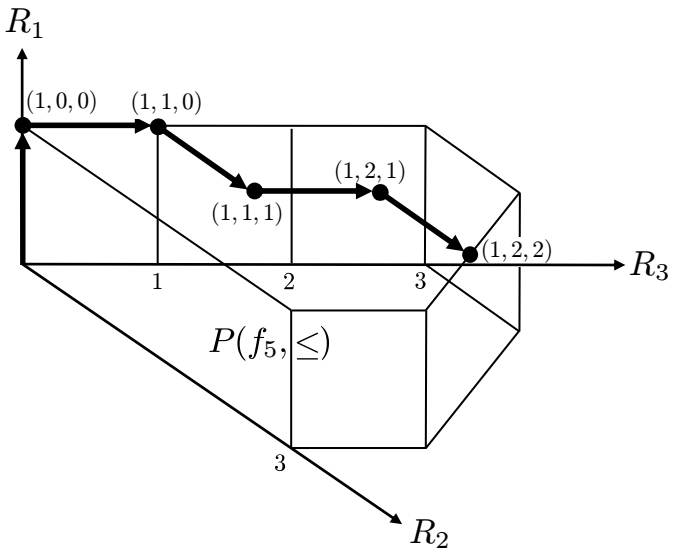


Figure 2.4: Algorithm 7 applied to the three-user problem from Example 4, with the cost function $\sum_{i=1}^{3} R_i \log R_i$ and the fixed sum-rate $R_1 + R_2 + R_3 = 5$. To minimize the cost, in each iteration we update the rate of the user who has transmitted the least amount of symbols in $\mathbb{F}_q$ such that the update still belongs to polyhedron $P(f_\beta, \leq)$.

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

$$R_1^* = 1, \quad R_2^* = 3, \quad R_3^* = 1. \tag{2.132}$$

Let us now analyze the case when the objective is $\varphi_i(R_i) = R_i \log R_i$, $i = 1, 2, 3$. Following the notation of Algorithm 7, we have that

$$d_i(R_i + 1) = (R_i + 1)\log(R_i + 1) - R_i \log R_i. \tag{2.133}$$

It is not hard to show that the above function $d_i(\cdot)$ is increasing. Hence, the minimization step in Algorithm 8 can be written as

$$i^* = \underset{i \in \mathcal{T}_j}{\operatorname{argmin}}\, R_i, \tag{2.134}$$

where $\mathcal{T}_j$ can be computed from (2.110), and $j = 1, \ldots, \beta$ is an iteration of Algorithm 8. The condition (2.134) 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. A dual set function f_5, that corresponds to this source model, has the following evaluations:

$$\begin{aligned} &f_5(\{1\}) = 1, \quad f_5(\{2\}) = 3, \quad f_5(\{3\}) = 3, \\ &f_5(\{1,2\}) = 4, \quad f_5(\{1,3\}) = 5, \quad f_5(\{2,3\}) = 4, \\ &f_5(\{1,2,3\}) = 5. \end{aligned} \tag{2.135}$$

The execution steps of Algorithm 7 are shown below (see also Figure 7 for the reference).

- Set $R_1 = R_2 = R_3 = 0$.
- $j = 1$: Check if user 1 belongs to $\mathcal{T}_1$:

$$\begin{aligned} \min\{&f_5(\{1\}) - R_1, f_5(\{1,2\}) - R_1 - R_2, f_5(\{1,3\}) - R_1 - R_3, \\ &f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 1 \geq 1. \end{aligned}$$

  Hence $1 \in \mathcal{T}_1$. Check if user 2 belongs to $\mathcal{T}_1$:

$$\begin{aligned} \min\{&f_5(\{2\}) - R_2, f_5(\{1,2\}) - R_1 - R_2, f_5(\{2,3\}) - R_2 - R_3, \\ &f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 3 \geq 1. \end{aligned}$$

  Hence $2 \in \mathcal{T}_1$. Check if user 3 belongs to $\mathcal{T}_1$:

$$\begin{aligned} \min\{&f_5(\{3\}) - R_2, f_5(\{1,3\}) - R_1 - R_3, f_5(\{2,3\}) - R_2 - R_3, \\ &f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 3 \geq 1. \end{aligned}$$

  Hence $3 \in \mathcal{T}_1$. Updates of the rate vector $\mathbf{R}$ are selected according to the rule (2.134):

$$\operatorname{argmin}\{R_i \mid i \in \mathcal{T}_1 = \{1,2,3\}\} = \{1,2,3\}.$$

  Set $R_1 = R_1 + 1 = 1$.

- $j = 2$: Check if user 1 belongs to $\mathcal{T}_2$:

$$\begin{aligned}\min\{&f_5(\{1\}) - R_1, f_5(\{1,2\}) - R_1 - R_2, f_5(\{1,3\}) - R_1 - R_3,\\ &f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 0 \geq 1.\end{aligned}$$

  Hence $1 \notin \mathcal{T}_2$. Check if user 2 belongs to $\mathcal{T}_2$:

$$\begin{aligned}\min\{&f_5(\{2\}) - R_2, f_5(\{1,2\}) - R_1 - R_2, f_5(\{2,3\}) - R_2 - R_3,\\ &f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 3 \geq 1.\end{aligned}$$

  Hence $2 \in \mathcal{T}_2$. Check if user 3 belongs to $\mathcal{T}_2$:

$$\begin{aligned}\min\{&f_5(\{3\}) - R_2, f_5(\{1,3\}) - R_1 - R_3, f_5(\{2,3\}) - R_2 - R_3,\\ &f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 3 \geq 1.\end{aligned}$$

  Hence $3 \in \mathcal{T}_2$. Vector $\mathbf{R}$ is updated according to the rule:

$$\operatorname{argmin}\{R_i \mid i \in \mathcal{T}_2 = \{2,3\}\} = \{2,3\}.$$

  Set $R_3 = R_3 + 1 = 1$.

- $j = 3$: Check if user 1 belongs to $\mathcal{T}_3$:

$$\begin{aligned}\min\{&f_5(\{1\}) - R_1, f_5(\{1,2\}) - R_1 - R_2, f_5(\{1,3\}) - R_1 - R_3,\\ &f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 0 \geq 1.\end{aligned}$$

  Hence $1 \notin \mathcal{T}_3$. Check if user 2 belongs to $\mathcal{T}_3$:

$$\begin{aligned}\min\{&f_5(\{2\}) - R_2, f_5(\{1,2\}) - R_1 - R_2, f_5(\{2,3\}) - R_2 - R_3,\\ &f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 3 \geq 1.\end{aligned}$$

  Hence $2 \in \mathcal{T}_3$. Check if user 3 belongs to $\mathcal{T}_3$:

$$\begin{aligned}\min\{&f_5(\{3\}) - R_2, f_5(\{1,3\}) - R_1 - R_3, f_5(\{2,3\}) - R_2 - R_3,\\ &f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 2 \geq 1.\end{aligned}$$

  Hence $3 \in \mathcal{T}_3$. Vector $\mathbf{R}$ is updated according to the rule:

$$\operatorname{argmin}\{R_i \mid i \in \mathcal{T}_3 = \{2,3\}\} = \{2\}.$$

  Set $R_2 = R_2 + 1 = 1$.

- $j = 4$: Check if user 1 belongs to $\mathcal{T}_4$:

$$\min\{f_5(\{1\}) - R_1, f_5(\{1,2\}) - R_1 - R_2, f_5(\{1,3\}) - R_1 - R_3, \\ f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 0 \geq 1.$$

  Hence $1 \notin \mathcal{T}_4$. Check if user 2 belongs to $\mathcal{T}_4$:

$$\min\{f_5(\{2\}) - R_2, f_5(\{1,2\}) - R_1 - R_2, f_5(\{2,3\}) - R_2 - R_3, \\ f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 2 \geq 1.$$

  Hence $2 \in \mathcal{T}_4$. Check if user 3 belongs to $\mathcal{T}_4$:

$$\min\{f_5(\{3\}) - R_2, f_5(\{1,3\}) - R_1 - R_3, f_5(\{2,3\}) - R_2 - R_3, \\ f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 2 \geq 1.$$

  Hence $3 \in \mathcal{T}_4$. Vector $\mathbf{R}$ is updated according to the rule:

$$\operatorname{argmin}\{R_i \mid i \in \mathcal{T}_4 = \{2,3\}\} = \{2,3\}.$$

  Set $R_3 = R_3 + 1 = 2$.

- $j = 5$: Check if user 1 belongs to $\mathcal{T}_5$:

$$\min\{f_5(\{1\}) - R_1, f_5(\{1,2\}) - R_1 - R_2, f_5(\{1,3\}) - R_1 - R_3, \\ f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 0 \geq 1.$$

  Hence $1 \notin \mathcal{T}_5$. Check if user 2 belongs to $\mathcal{T}_5$:

$$\min\{f_5(\{2\}) - R_2, f_5(\{1,2\}) - R_1 - R_2, f_5(\{2,3\}) - R_2 - R_3, \\ f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 1 \geq 1.$$

  Hence $2 \in \mathcal{T}_5$. Check if user 3 belongs to $\mathcal{T}_5$:

$$\min\{f_5(\{3\}) - R_2, f_5(\{1,3\}) - R_1 - R_3, f_5(\{2,3\}) - R_2 - R_3, \\ f_5(\{1,2,3\}) - R_1 - R_2 - R_3\} = 1 \geq 1.$$

  Hence $3 \in \mathcal{T}_5$. Vector $\mathbf{R}$ is updated according to the rule:

$$\operatorname{argmin}\{R_i \mid i \in \mathcal{T}_5 = \{2,3\}\} = \{2\}.$$

  Set $R_2 = R_2 + 1 = 2$.

- An optimal *DE*-rate vector is $R_1^* = 1$, $R_2^* = 2$, $R_3^* = 2$.

2.5 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 3, where

$$\begin{aligned}\mathbf{x}_1 &= \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T, \\ \mathbf{x}_2 &= \begin{bmatrix} w_2 & w_4 & w_5 & w_6 \end{bmatrix}^T, \\ \mathbf{x}_3 &= \begin{bmatrix} w_3 & w_4 & w_5 & w_6 \end{bmatrix}^T. \end{aligned} \tag{2.136}$$

For the objective function $\min R_1 + R_2 + R_3$, we showed that the optimal *DE*-rate vector is $R_1^* = 1$, $R_2^* = 1$, and $R_3^* = 3$. 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 2.5). 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 2.5). 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^*, $\forall i \in \mathcal{M}$.

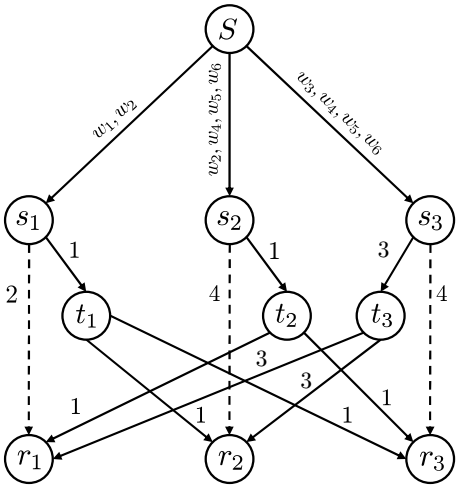


Figure 2.5: Multicast network constructed from the source model and the sum-rate optimal *DE*-rate vector $R_1^* = 1$, $R_2^* = 1$, $R_3^* = 2$. Hence, in an optimal scheme users 1, 2 and 3 are transmitting 1, 1, and 3 linear combinations of their own observations in $\mathbb{F}_q$, respectively. Each 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 $t_i \to r_j$, $i, j \in \{1, 2, 3\}$, $i \neq j$).

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 [21]. 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 [2]. 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), we can directly apply Jaggi's algorithm to this problem by slightly modifying the upper portion of the multicast network from Figure 2.5 (see Figure 2.6). 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 (2.136). In other words, the entire source model and the communication model can be represented by multicast acyclic graph. Therefore, Jaggi's algorithm would find actual transmissions of each user in polynomial time.

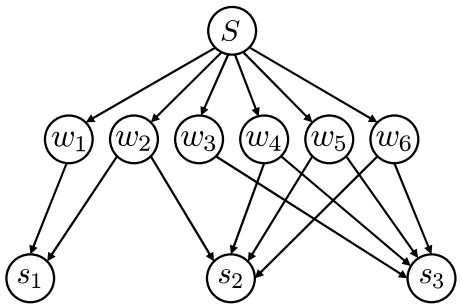


Figure 2.6: When each user observes subset of the file packets, we can model the observations by adding an extra layer of $N = 6$ nodes to the graph in Figure 2.5. Each extra node represents one file packet, and all extra edges are of capacity 1. Then, users' observations can be modeled by connecting nodes from this layer to the users' nodes s_1, s_2 and s_3 according to (2.136).

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 2.6. 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. For that reason, it is more suitable to apply Harvey's algorithm [17] which is based on matrix representation of transmissions in the network [22], and simultaneous matrix completion problem over finite fields.

Before we go any further let us consider the simplest version of a simultaneous matrix completion problem on the following example.

Example 5. Let the three users have access to the following parts of the three packet file

$\mathbf{w} = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix}^T$,

$$\mathbf{x}_1 = \begin{bmatrix} w_1 + w_2 \\ w_2 + w_3 \end{bmatrix},$$
$$\mathbf{x}_2 = \begin{bmatrix} w_2 + w_3 \\ w_1 + w_3 \end{bmatrix},$$
$$\mathbf{x}_3 = \begin{bmatrix} w_1 + w_2 \\ w_1 + w_3 \end{bmatrix}, \tag{2.137}$$

where $w_1, w_2, w_3 \in \mathbb{F}_q$, $q > 2$. It can be shown that the minimum sum-rate is 2, with the rate allocation $R_1^* = 1$, $R_2^* = 1$, $R_3^* = 0$. In general, we can represent transmission of user 1 as

$$v_1 = \alpha_1(w_1 + w_2) + \alpha_2(w_2 + w_3) = \alpha_1 w_1 + (\alpha_1 + \alpha_2)w_2 + \alpha_2 w_3, \tag{2.138}$$

where $\alpha_1, \alpha_2 \in \mathbb{F}_q$. Similarly, transmission of user 2 is

$$v_2 = \beta_1(w_2 + w_3) + \beta_2(w_1 + w_3) = \beta_2 w_1 + \beta_1 w_2 + (\beta_1 + \beta_2)w_3, \tag{2.139}$$

where $\beta_1, \beta_2 \in \mathbb{F}_q$. After all the users transmitted optimal number of packets, each of them updates its observation matrix.

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \beta_2 & \beta_1 & \beta_1 + \beta_2 \end{bmatrix}, \ \mathbf{A}_2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ \alpha_1 & \alpha_1 + \alpha_2 & \alpha_2 \end{bmatrix}, \ \mathbf{A}_3 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ \alpha_1 & \alpha_1 + \alpha_2 & \alpha_2 \\ \beta_2 & \beta_1 & \beta_1 + \beta_2 \end{bmatrix}.$$

In order to construct the coding scheme it is necessary to find coefficients α_1, α_2, β_1, β_2 such that the rank of each updated observation matrix is full, *i.e.*, $N = 3$. This problem is called the simultaneous matrix completion problem.

Polynomial time solution to the simultaneous matrix completion problem, as suggested in [17], exists if all indeterminate elements, in this example α_1, α_2, β_1, β_2, appear no more than once in each observation matrix. However, this is not true in the case of the linear packet model; for instance in Example 5, β_1 appears twice in matrix $\mathbf{A}_1$.

Therefore, in order to have a polynomial time code construction, it is necessary to construct new matrices assigned to each user, where all indeterminate elements appear at most once in each matrix, and possibly more times across different matrices. This can be done by using algebraic approach to the multicast network coding problem as in [22], which we briefly explain in the remainder of this section.

Going back to the source model (2.136), let us annotated all the edges from Figure 2.5 (see Figure 2.7). We denote by $Y(e_i)$ transmission process through link e_i, $i \in \{1, 2, \ldots, 35\}$.

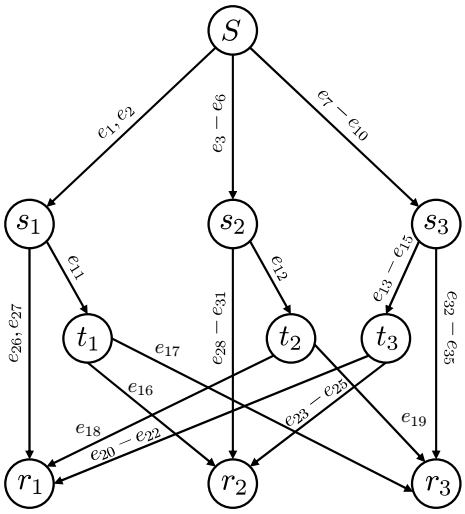


Figure 2.7: Annotated edges for the multicast network in Figure 2.5.

It is obvious that for the first layer of links $e_1 - e_{10}$, $Y(e_i)$ corresponds to the underlying source distribution. More formally,

$$\begin{aligned} &Y(e_1) = w_1, \quad Y(e_2) = w_2, \\ &Y(e_3) = w_2, \quad Y(e_4) = w_4, \quad Y(e_5) = w_5, \quad Y(e_6) = w_6, \\ &Y(e_7) = w_3, \quad Y(e_8) = w_4, \quad Y(e_9) = w_5, \quad Y(e_{10}) = w_6. \end{aligned} \tag{2.140}$$

In the linear multicast network, each node transmits on its outgoing edge a linear combination of the symbols received on the incoming edges. Hence, the transmissions on the remaining edges can be modeled as

$$\begin{aligned} Y(e_{11}) &= \gamma_{e_1,e_{11}} Y(e_1) + \gamma_{e_2,e_{11}} Y(e_2), \\ Y(e_{12}) &= \gamma_{e_3,e_{12}} Y(e_3) + \gamma_{e_4,e_{12}} Y(e_4) + \gamma_{e_5,e_{12}} Y(e_5) + \gamma_{e_6,e_{12}} Y(e_6), \\ Y(e_{13}) &= \gamma_{e_7,e_{13}} Y(e_7) + \gamma_{e_8,e_{13}} Y(e_8) + \gamma_{e_9,e_{13}} Y(e_9) + \gamma_{e_{10},e_{13}} Y(e_{10}), \\ Y(e_{14}) &= \gamma_{e_7,e_{14}} Y(e_7) + \gamma_{e_8,e_{14}} Y(e_8) + \gamma_{e_9,e_{14}} Y(e_9) + \gamma_{e_{10},e_{14}} Y(e_{10}), \\ Y(e_{15}) &= \gamma_{e_7,e_{15}} Y(e_7) + \gamma_{e_8,e_{15}} Y(e_8) + \gamma_{e_9,e_{15}} Y(e_9) + \gamma_{e_{10},e_{15}} Y(e_{10}), \end{aligned} \tag{2.141}$$

where γ_{e_i,e_j} is a coefficient in $\mathbb{F}_q$ that defines a linear operation performed by the node between edges e_i and e_j. Note that edges $e_{26} - e_{35}$ are just routing the side information of nodes s_1, s_2 and s_3 to nodes r_1, r_2 and r_3, respectively. Therefore, we model these transmissions in the following way

$$\begin{aligned} &Y(e_{26}) = Y(e_1), \quad Y(e_{27}) = Y(e_2), \\ &Y(e_{28}) = Y(e_3), \quad Y(e_{29}) = Y(e_4), \quad Y(e_{30}) = Y(e_5), \quad Y(e_{31}) = Y(e_6), \\ &Y(e_{32}) = Y(e_7), \quad Y(e_{33}) = Y(e_8), \quad Y(e_{34}) = Y(e_9), \quad Y(e_{35}) = Y(e_{10}). \end{aligned} \tag{2.142}$$

The remaining edges $e_{16} - e_{25}$ are simply routing information observed by nodes t_1, t_2 and t_3, *i.e.*,

$$\begin{array}{ll} Y(e_{16}) = Y(e_{11}), & Y(e_{17}) = Y(e_{11}), \\ Y(e_{18}) = Y(e_{12}), & Y(e_{19}) = Y(e_{12}), \\ Y(e_{20}) = Y(e_{13}), & Y(e_{23}) = Y(e_{13}), \\ Y(e_{21}) = Y(e_{14}), & Y(e_{24}) = Y(e_{14}), \\ Y(e_{22}) = Y(e_{15}), & Y(e_{25}) = Y(e_{15}). \end{array} \tag{2.143}$$

It is only left to write down linear combinations observed by each receiver r_1, r_2 and r_3. The goal is for each receiver to obtain all $N = 6$ data packets. Using the same notation as in [22] let us denote by $\left[\begin{array}{cccccc} Z_{r_i,w_1} & Z_{r_i,w_2} & Z_{r_i,w_3} & Z_{r_i,w_4} & Z_{r_i,w_5} & Z_{r_i,w_6} \end{array} \right]$, $i = 1, 2, 3$, the output process at sink r_i which correspond to each individual packet that has to be decoded. To that end, for the receivers r_1, r_2 and r_3, the corresponding output process can be written as follows

$$\begin{aligned} Z(r_1, w_i) &= \delta_{e_{18},w_i} Y(e_{18}) + \delta_{e_{20},w_i} Y(e_{20}) + \delta_{e_{21},w_i} Y(e_{21}) \\ &\quad + \delta_{e_{22},w_i} Y(e_{22}) + \delta_{e_{26},w_i} Y(e_{26}) + \delta_{e_{27},w_i} Y(e_{27}), \\ Z(r_2, w_i) &= \delta_{e_{16},w_i} Y(e_{16}) + \delta_{e_{23},w_i} Y(e_{23}) + \delta_{e_{24},w_i} Y(e_{24}) + \delta_{e_{25},w_i} Y(e_{25}) \\ &\quad + \delta_{e_{28},w_i} Y(e_{28}) + \delta_{e_{29},w_i} Y(e_{29}) + \delta_{e_{30},w_i} Y(e_{30}) + \delta_{e_{31},w_i} Y(e_{31}), \\ Z(r_3, w_i) &= \delta_{e_{17},w_i} Y(e_{17}) + \delta_{e_{19},w_i} Y(e_{19}) + \delta_{e_{32},w_i} Y(e_{32}) \\ &\quad + \delta_{e_{33},w_i} Y(e_{33}) + \delta_{e_{34},w_i} Y(e_{34}) + \delta_{e_{35},w_i} Y(e_{35}), \quad i = 1, 2, \ldots, 6. \end{aligned} \tag{2.144}$$

In [22] authors derived the transfer matrix $\mathbf{M}_i$ from the super-node S to any receiver r_i, $i = 1, 2, 3$. It is a 6×6 matrix with the input vector $\mathbf{w}$. In order to construct matrix $\mathbf{M}_i$, $i = 1, 2, 3$, let us consider the following block matrices. Linear combinations described in (2.140) correspond to the underlying source distribution. To that end, let us define a 6×35 matrix $\mathbf{A}$ which rows correspond to $\mathbf{w}$, and columns correspond to the observations on the network edges $Y(e_i)$, $i = 1, 2, \ldots, 35$.

$$\mathbf{A} = \left[\begin{array}{ccccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{array} \right]_{6 \times 35}. \tag{2.145}$$

For a general linear packet model defined in (2.1), matrix $\mathbf{A}$ can be written as follows

$$\mathbf{A} = \left[\begin{array}{ccccccc} \mathbf{A}_1^T & \mathbf{A}_2^T & \cdots & \mathbf{A}_m^T & \mathbf{0} & \cdots & \mathbf{0} \end{array} \right]. \tag{2.146}$$

$\mathbf{A}$ is an $N \times \ell$ matrix, where ℓ can be easily obtained by inspecting Figure 2.5.

$$\ell = 2\sum_{i=1}^{m} \ell_i + m\sum_{i=1}^{m} R_i^*. \tag{2.147}$$

Next, we describe transmissions $Y(e_i)$ for $i = 13, 14, \ldots, 35$. In particular, let as define a 35×35 matrix $\mathbf{\Gamma}$ as follows

$$\mathbf{\Gamma}_{i,j} = \gamma_{e_i,e_j}, \tag{2.148}$$

where γ_{e_i,e_j} are coefficients defined in (2.141), (2.142) and (2.143). All the coefficients γ_{e_i,e_j} that do not appear in (2.141), (2.142) and (2.143) are set to be zero. In general case, matrix $\mathbf{\Gamma}$ is $\ell \times \ell$ and its entries are equal to

$$\mathbf{\Gamma}_{i,j} = \begin{cases} \gamma_{e_i,e_j}, & \text{if } \ head(e_i) = tail(e_j), \\ 0, & \text{otherwise.} \end{cases} \tag{2.149}$$

Notice that some of the γ_{e_i,e_j}'s in (2.149) are set to zero or one according to (2.142) and (2.143). Finally, let us define an output process matrix $\mathbf{D}(k)$ of dimension 33×6, for each user $k = 1, 2, 3$. From linear combinations defined in (2.144) we construct matrix $\mathbf{D}(k)$ as follows

$$\mathbf{D}_{i,j}(k) = \delta_{e_i,w_j}, \quad \forall w_j \in \mathbf{w}, \quad k = 1, 2, 3. \tag{2.150}$$

For all coefficients δ_{e_i,w_j} which do not appear in (2.144), $\mathbf{D}_{i,j}$ equals to zero. In general case, matrix $\mathbf{D}(k)$, $k = 1, 2, \ldots, m$, is $\ell \times N$ and its entries are equal to

$$\mathbf{D}_{i,j}(k) = \begin{cases} \delta_{e_i,w_j}, & \text{if } \ head(e_i) = r_k, \\ 0 & \text{otherwise.} \end{cases} \tag{2.151}$$

From the construction of matrices $\mathbf{A}$, $\mathbf{\Gamma}$, $\mathbf{D}(k)$, authors in [22] derived a transfer matrix $\mathbf{M}(k)$ for each terminal $k = 1, 2, \ldots, m$. This result is stated in next theorem

Theorem 5 (Transfer Matrix [22]). *For a network defined by matrices $\mathbf{A}$, $\mathbf{\Gamma}$, $\mathbf{D}(k)$, $k = 1, 2, \ldots, m$, the transfer matrix $\mathbf{M}(k)$ for each receiver r_k is given as*

$$\mathbf{M}(k) = \mathbf{A}(\mathbf{I} - \mathbf{\Gamma})^{-1}\mathbf{D}(k), \quad k = 1, 2, \ldots, m, \tag{2.152}$$

where $\mathbf{I}$ is the $\ell \times \ell$ identity matrix.

A multicast problem has a network coding solution if and only if each matrix $\mathbf{M}(k)$ is non-singular. It should be noted that all unknown entries in matrices $\mathbf{\Gamma}$, $\mathbf{D}(k)$ are independent

from each other, *i.e.*, all $\gamma_{i,j}$'s and $\delta_{i,j}$'s are independent. In [17] it was shown that for the *expanded transfer matrix* defined as

$$\mathbf{E}(k) = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{I} - \mathbf{\Gamma} & \mathbf{D}(k) \end{bmatrix}, \quad k = 1, 2, \ldots, m, \tag{2.153}$$

it holds that $\det(\mathbf{M}(k)) = \pm \det(\mathbf{E}(k))$. This means that if we pick entries $\gamma_{i,j}$ and $\delta_{i,j}$ such that the matrices $\mathbf{E}(k)$, $k = 1, 2, \ldots, m$, have full rank, then we immediately obtain a network coding solution. Note that matrix $\mathbf{\Gamma}$ appears in all matrices $\mathbf{E}(k)$, and that all indeterminate elements in matrices $\mathbf{\Gamma}$, and $\mathbf{D}(k)$ appear only once in each matrix $\mathbf{E}(k)$, $k = 1, 2, \ldots, m$. This simultaneous matrix completion problem can be solved in polynomial time by using Harvey's algorithm [17].

Lemma 9 (Harvey, [17])**.** *Polynomial time solution for the simultaneous matrix completion problem exists if and only if* $|\mathbb{F}_q| > m$. *Complexity of the proposed algorithm is* $\mathcal{O}(m^4 \cdot N^3 \log(m \cdot N))$, *where* ℓ *is defined in* (2.147).

Hereafter, we assume that the field size $|\mathbb{F}_q|$ is large enough to accommodate for the polynomial time solution. Summarizing this section, in order to obtain polynomial time solution to the data exchange problem, it is necessary to complete the following steps

1. Compute an optimal rate allocation $\mathbf{R}^*$ w.r.t. the communication cost.
2. Based on $\mathbf{R}^*$ construct a multicast acyclic graph as shown in Figures 2.5 and 2.7.
3. Compute matrices $\mathbf{A}$, $\mathbf{\Gamma}$ and $\mathbf{D}(k)$ for $k = 1, 2, \ldots, m$.
4. Apply Harvey's algorithm to the extended matrices $\mathbf{E}(k)$, $k = 1, 2, \ldots, m$, to find indeterminate elements in the matrices $\mathbf{\Gamma}$, $\mathbf{B}(\mathbf{k})$.

2.6 Randomized Algorithm

In this section we combine Algorithm 7 with the linear network coding scheme to produce a randomized solution to the optimization problem (2.18) of linear complexity (in number of users). First, note that Algorithm 7 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 7. Along with the rate update, let user i transmit an appropriately chosen linear combination of its observations; using the notation from Section 2.1, we have

$$v_i^{(j)} = \mathbf{b}_i^{(j)} \cdot \mathbf{A}_i \cdot \mathbf{w}, \tag{2.154}$$

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 2.5 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 (2.154), let us define $\mathbf{u}^{(j)} \in \mathbb{F}_q^N$ as

$$\mathbf{u}^{(j)} \triangleq \mathbf{b}_i^{(j)} \cdot \mathbf{A}_i. \tag{2.155}$$

Then, we can write (2.154) as

$$v_i^{(j)} = \mathbf{u}^{(j)} \cdot \mathbf{w}. \tag{2.156}$$

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 $P(f_\beta, \leq)$. This result is stated in the following theorem.

Theorem 6. *Let the set $\mathcal{T}_j$ be defined as in (2.109). Then,*

$$\mathcal{T}_j = \{i \in \mathcal{M} \mid \mathit{rank}\left(\mathbf{A}_i \cup \mathbf{u}^{(1)} \cup \cdots \cup \mathbf{u}^{(j-1)}\right) > N - (\beta - j + 1)\}. \tag{2.157}$$

Proof. Let us start by considering round $j = 1$ of Algorithm 7. 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 $P(f_\beta, \leq)$

$$R^*(S) + 1 \leq f_\beta(\mathcal{S}), \quad \forall \mathcal{S}, \quad \text{s.t.} \quad i \in \mathcal{S}, \tag{2.158}$$

where f_β is defined in (2.25). Since $\mathbf{R}^*$ is a zero vector, we can write the condition (2.158) as

$$1 \leq \beta - N + \operatorname{rank}(\mathbf{A}_\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \quad \text{s.t.} \quad i \in \mathcal{S}, \tag{2.159}$$

which is equivalent to

$$1 \leq \min_{i \in \mathcal{S} \subseteq \mathcal{M}} \{\beta - N + \operatorname{rank}(\mathbf{A}_\mathcal{S})\}. \tag{2.160}$$

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

$$\operatorname{rank}(\mathbf{A}_i) > N - \beta, \tag{2.161}$$

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

$$v_i^{(1)} = \mathbf{u}^{(1)} \cdot \mathbf{w}, \tag{2.162}$$

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

$$\mathbf{A}_k \cup \mathbf{u}^{(1)}, \quad \forall k \in \mathcal{M} \setminus \{i\}. \tag{2.163}$$

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 (2.161) and (2.163) it is easy to see that in round j, the condition (2.161) becomes

$$\text{rank}\left(\mathbf{A}_i \cup \mathbf{u}^{(1)} \cup \cdots \cup \mathbf{u}^{(j-1)}\right) > N - (\beta - j + 1), \tag{2.164}$$

which completes the proof. □

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 2.5 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 (2.154) 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 10. *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* $1 - \frac{const}{q}$, *provided that the field size is large enough.*

The proof of Lemma 10 directly follows from Lemma 4 in [18]. The idea is to relate this problem to a multicast problem as in Section 2.5, and analyze determinant of the extended matrices $\mathbf{E}(k)$, $k = 1, \ldots, m$ given in (2.153). Then, by choosing the indeterminate elements in matrix $\mathbf{\Gamma}$ given in (2.149), randomly over $\mathbb{F}_q$, we obtain exactly the same formulation as in [18].

Putting all these results together, from Algorithm 7 we can devise its Randomized counterpart as follows (see Algorithm 10).

Remark 13. The complexity of Algorithm 10 is $\mathcal{O}(m \cdot \gamma \cdot N)$, where γ is the complexity of computing rank.

Remark 14. When β is not a feasible sum-rate w.r.t. the optimization problem (2.18), then after β iterations of Algorithm 10 there exists a user that cannot reconstruct all the packets. In other words

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

Algorithm 10 Randomized Algorithm

1: Set $R_i = 0$, $\forall i \in \mathcal{M}$
2: **for** $j = 1$ to β **do**
3: Determine $\mathcal{T}_j$ as defined in (2.157).
4: Find $i^* \in \mathcal{T}_j$ such that

$$i^* = \operatorname{argmin} \{d_i(R_i + 1) \mid i \in \mathcal{T}_j\}.$$

5: Let i^* transmit, and create a transmission $v_{i^*}^{(j)}$ by creating a vector $\mathbf{b}_{i^*}^{(j)}$ uniformly at random over $\mathbb{F}_q^{\ell_{i^*}}$.
6: Set $R_{i^*} = R_{i^*} + 1$.
7: **end for**
8: $\mathbf{R}^* = \mathbf{R}$ is an optimal rate vector w.r.t. the problem (2.108).

In order to solve the optimization problem (2.17), we can apply a binary search algorithm similar to Algorithm 5. Thus, the overall complexity of the proposed algorithm is $\mathcal{O}(m \cdot \gamma \cdot N \log N)$.

Example 6. Let us consider the same problem as in Example 4

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

where $w_i \in \mathbb{F}_q$, $i = 1, \ldots, 6$, 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 10 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 (2.134):

$$\operatorname{argmin} \{R_i \mid i \in \mathcal{T}_1 = \{1, 2, 3\}\} = \{1, 2, 3\},$$

  User 1 transmit some random linear combination of its observation, say $v_1^{(1)} = w_1 + 7w_2$. Set

$$R_1 = R_1 + 1 = 1.$$

- $j = 2$: Vector $\mathbf{R}$ is updated according to the rule:

$$\operatorname{argmin} \{R_i \mid i \in \mathcal{T}_2 = \{2, 3\}\} = \{3\}.$$

User 3 transmit some random linear combination of its observation, say $v_3^{(2)} = w_3 + w_4 + 5w_5 + 11w_6$. Set

$$R_3 = R_3 + 1 = 1.$$

- $j = 3$: Vector $\mathbf{R}$ is updated according to the rule:

$$\text{argmin}\,\{R_i \mid i \in \mathcal{T}_3 = \{2, 3\}\} = \{2\}.$$

User 2 transmit some random linear combination of its observation, say $v_2^{(3)} = 4w_2 + 3w_4 + 13w_5 + 8w_6$. Set

$$R_2 = R_2 + 1 = 1.$$

- $j = 4$: Vector $\mathbf{R}$ is updated according to the rule:

$$\text{argmin}\,\{R_i \mid i \in \mathcal{T}_4 = \{2, 3\}\} = \{3\}.$$

User 3 transmit some random linear combination of its observation, say $v_3^{(4)} = 9w_3 + 5w_4 + 14w_5 + 17w_6$. Set

$$R_3 = R_3 + 1 = 2.$$

- $j = 5$: Vector $\mathbf{R}$ is updated according to the rule:

$$\text{argmin}\,\{R_i \mid i \in \mathcal{T}_5 = \{2, 3\}\} = \{2\}.$$

User 2 transmit some random linear combination of its observation, say $v_2^{(5)} = 11w_2 + 2w_4 + 18w_5 + 6w_6$. Set

$$R_2 = R_2 + 1 = 2.$$

- $\mathbf{R}^* = \mathbf{R}$ 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.

2.7 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 from each user. This is particularly useful in the scenarios where communication consumes a lot of power, and we want to "spare" users with low battery. Say that user i is not allowed to transmit more than c_i packets in $\mathbb{F}_q$. Then, optimization problem (2.17) becomes

$$\min_{\beta \in \mathbb{Z}_+} h(\beta), \tag{2.165}$$

where $h(\beta)$ can be obtained from (2.41) by adding capacity constraints.

$$h(\beta) = \min_{\mathbf{R} \in \mathbb{Z}^m} \sum_{i=1}^{m} \varphi_i(R_i), \quad \text{s.t.,} \quad \mathbf{R} \in B(g_\beta, \leq), \tag{2.166}$$
$$R_i \leq c_i, \quad \forall i \in \mathcal{M},$$

provided that $g_\beta(\mathcal{M}) = \beta$. Otherwise, the sum-rate β is infeasible w.r.t. the problem (2.166).

In Section 2.2 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 (2.18) constitutes a base polyhedron of a submodular function. In this section we show that by adding individual capacity constraints, the constraint set in (2.18) 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 [20] for the reference).

Definition 9. 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

$$g_\beta^{\mathbf{c}}(\mathcal{S}) \triangleq \min \left\{ g_\beta(\mathcal{V}) + c(\mathcal{S} \setminus \mathcal{V}) \mid \mathcal{V} \subseteq \mathcal{S} \right\}, \quad \forall \mathcal{S} \subseteq \mathcal{M}. \tag{2.167}$$

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

Theorem 7 (Theorem 8.2.1 in [20])**.** *Let $g_\beta^{\mathbf{c}}$ be the restriction of a submodular function g_β by vector $\mathbf{c}$. Then, $g_\beta^{\mathbf{c}}$ is submodular.*

Proof. For a set function $g_\beta^{\mathbf{c}}$, let us denote by $\mathcal{V}_{\mathcal{S}}$ the minimizer set of the optimization problem (2.167). Then, for any $\mathcal{S}, \mathcal{T} \subseteq \mathcal{M}$, we have

$$\begin{aligned} g_\beta^{\mathbf{c}}(\mathcal{S}) + g_\beta^{\mathbf{c}}(\mathcal{T}) &= g_\beta(\mathcal{V}_{\mathcal{S}}) + c(\mathcal{S} \setminus \mathcal{V}_{\mathcal{S}}) + g_\beta(\mathcal{V}_{\mathcal{T}}) + c(\mathcal{T} \setminus \mathcal{V}_{\mathcal{T}}) \\ &\overset{(a)}{\geq} g_\beta(\mathcal{V}_{\mathcal{S}} \cup \mathcal{V}_{\mathcal{T}}) + c\left((\mathcal{S} \cup \mathcal{T}) \setminus (\mathcal{V}_{\mathcal{S}} \cup \mathcal{V}_{\mathcal{T}})\right) \\ &\quad + g_\beta(\mathcal{V}_{\mathcal{S}} \cap \mathcal{V}_{\mathcal{T}}) + c\left((\mathcal{S} \cap \mathcal{T}) \setminus (\mathcal{V}_{\mathcal{S}} \cap \mathcal{V}_{\mathcal{T}})\right) \\ &\overset{(b)}{\geq} g_\beta^{\mathbf{c}}(\mathcal{S} \cup \mathcal{T}) + g_\beta^{\mathbf{c}}(\mathcal{S} \cap \mathcal{T}) \end{aligned} \tag{2.168}$$

where (a) is due to the submodularity of g_β, and (b) follows from (2.167). Hence, from (2.168) we can see that $g_\beta^{\mathbf{c}}$ is submodular function. □

Theorem 8. *For a submodular function g_β defined in (2.38) and a capacity vector $\mathbf{c}$, the base polyhedron $B(g_\beta^{\mathbf{c}}, \leq)$ of the restriction of g_β by $\mathbf{c}$, is given by*

$$B(g_\beta^{\mathbf{c}}, \leq) = \{\mathbf{R} \mid \mathbf{R} \in B(g_\beta, \leq), \quad R_i \leq c_i, \ \forall i \in \mathcal{M}\}, \tag{2.169}$$

provided that the sum-rate β and the capacity vector $\mathbf{c}$ are feasible w.r.t. the optimization problem (2.166).

Proof. Let $\mathbf{R}$ be any feasible rate vector w.r.t. the problem (2.166), *i.e.*,

$$R(\mathcal{S}) \leq g_\beta(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \tag{2.170}$$
$$R(\mathcal{S}) \leq c(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \tag{2.171}$$
$$R(\mathcal{M}) = g_\beta(\mathcal{M}) = \beta. \tag{2.172}$$

Let us denote by $g_\beta^\star$ the dual function of g_β (see Definition 3),

$$g_\beta^\star(\mathcal{S}) = g_\beta(\mathcal{M}) - g_\beta(\mathcal{M} \setminus \mathcal{S}) = \beta - g_\beta(\mathcal{M} \setminus \mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}. \tag{2.173}$$

By Lemma 1, vector $\mathbf{R}$ must satisfy

$$\begin{aligned} R(\mathcal{S}) &\geq g_\beta^\star(\mathcal{S}) = \beta - g_\beta(\mathcal{M} \setminus \mathcal{S}), \quad \forall \mathcal{S} \subset \mathcal{M}, \\ R(\mathcal{M}) &= g_\beta^\star(\mathcal{M}) = \beta. \end{aligned} \tag{2.174}$$

From (2.171) and (2.174) it follows that

$$g_\beta(\mathcal{M}) - g_\beta(\mathcal{M} \setminus \mathcal{V}) \leq R(\mathcal{V}) \leq c(\mathcal{V}), \quad \forall \mathcal{V} \subseteq \mathcal{M}. \tag{2.175}$$

From (2.175), we have that

$$g_\beta(\mathcal{M}) \leq g_\beta(\mathcal{M} \setminus \mathcal{V}) + c(\mathcal{V}), \quad \forall \mathcal{V} \subseteq \mathcal{M}. \tag{2.176}$$

Hence, (2.167) implies that

$$g_\beta^{\mathbf{c}}(\mathcal{M}) = g_\beta(\mathcal{M}) = \beta. \tag{2.177}$$

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

$$R(\mathcal{S}) = R(\mathcal{V}) + R(\mathcal{S} \setminus \mathcal{V}) \leq g_\beta(\mathcal{V}) + c(\mathcal{S} \setminus \mathcal{V}), \quad \forall \mathcal{V}, \mathcal{S} \ \text{ s.t. } \ \mathcal{V} \subseteq \mathcal{S} \subseteq \mathcal{M}. \tag{2.178}$$

Finally (2.178) implies that

$$R(\mathcal{S}) \leq \min\{g_\beta(\mathcal{V}) + c(\mathcal{S} \setminus \mathcal{V}) \mid \mathcal{V} \subseteq \mathcal{S}\}, \ \forall \mathcal{S} \subseteq \mathcal{M}. \tag{2.179}$$

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

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

$$R(\mathcal{S}) \leq g^{\mathbf{c}}_{\beta}(\mathcal{S}) \leq g_{\beta}(\mathcal{S}) + c(\emptyset) = g_{\beta}(\mathcal{S}), \ \forall \mathcal{S} \subseteq \mathcal{M}, \tag{2.180}$$

$$R(\mathcal{S}) \leq g^{\mathbf{c}}_{\beta}(\mathcal{S}) \leq g_{\beta}(\emptyset) + c(\mathcal{S}) = c(\mathcal{S}), \ \ \forall \mathcal{S} \subseteq \mathcal{M}, \tag{2.181}$$

$$R(\mathcal{M}) = g^{\mathbf{c}}_{\beta}(\mathcal{M}) = \beta \tag{2.182}$$

where the second inequality in (2.180) and (2.181) directly follows from (2.167). From (2.180), (2.181), and (2.182) it follows that

$$\mathbf{R} \in B(g_{\beta}, \leq), \ \ \text{s.t.} \ \ R_i \leq c_i, \ \ \forall i \in \mathcal{M}. \tag{2.183}$$

This completes the proof. □

Corollary 2. For a submodular function g_{β} defined in (2.38) and a capacity vector $\mathbf{c}$, polyhedron $P(g^{\mathbf{c}}_{\beta}, \leq)$ of the restriction of g_{β} by $\mathbf{c}$, is given by

$$P(g^{\mathbf{c}}_{\beta}, \leq) = \{\mathbf{R} \mid \mathbf{R} \in P(g_{\beta}, \leq), \ \ R_i \leq c_i, \ \forall i \in \mathcal{M}\}, \tag{2.184}$$

provided that the sum-rate β and the capacity vector $\mathbf{c}$ are feasible w.r.t. the optimization problem (2.166).

From Theorem 8 it follows that the constraint set of (2.166) forms a submodular polyhedron $B(g^{\mathbf{c}}_{\beta}, \leq)$, 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

$$\begin{aligned} R^*_{j(i)} = \min\{c_{j(i)}, g_{\beta}(\{j(1), j(2), \ldots, j(i)\}) \\ - g_{\beta}(\{j(1), j(2), \ldots, j(i-1)\})\}. \end{aligned} \tag{2.185}$$

This modification propagates to Algorithm 3 as well. Similarly, at iteration j, Step 3 of Algorithms 7 and 10 is modified as follows

$$i^* = \text{argmin}\{d_i(R^*_i + 1) \mid i \in \mathcal{T}_j, \ \text{s.t.}, \ R^*_i + 1 \leq c_i\}. \tag{2.186}$$

Example 7. Let us consider the same problem as in Example 3

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

where $w_i \in \mathbb{F}_q$. Let the cost function be $R_1+3R_2+2R_3$, the sum-rate $\beta = 5$, and the capacity constraints $c_i \leq 2$, $i = 1, 2, 3$. Then, by applying Algorithm 3 with the modification (2.185), we obtain the following result.

$$\begin{aligned} R_1^* &= \min\{c_1, f_5(\{1\})\} = 1, \\ R_3^* &= \min\{c_3, \min\{f_5(\{1,3\}) - R_1^*, f_5(\{2\})\}\} = 2, \\ R_2^* &= \min\{c_2, \min\{f_5(\{1,2,3\}) - R_1^* - R_3^*, f_5(\{1,3\}) - R_1^*, \\ &\qquad f_5(\{2,3\}) - R_3^*, f_5(\{2\})\}\} = 2. \end{aligned}$$

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

Chapter 3

Data Exchange Problem - General Correlations

In this chapter we study data exchange problem under the general source correlation model. All the techniques we developed in Chapter 2 will be useful in devising polynomial time algorithms in this chapter.

3.1 System Model and Preliminaries

We consider a setup where m users are interested in gaining access to a file or a random process. Let $X_1, X_2, \ldots, X_m$, $m \geq 2$, denote the components of a discrete memoryless multiple source (DMMS) with a given joint probability mass function. Each user $i \in \mathcal{M} = \{1, 2, \ldots, m\}$ observes n i.i.d. realizations of the corresponding random variable X_i, denoted by X_i^n. Let us denote $X_{\mathcal{M}} \triangleq \{X_1, X_2, \ldots, X_m\}$. In [11], Csiszár and Narayan showed that in order for each user in $\mathcal{M}$ to learn $X_{\mathcal{M}}^n$ in a setup with general DMMS *interactive communication is not needed*. As a result, in the sequel without loss of generality we can assume that the transmission of each user is only a function of its own initial observations. In particular, let $F_i \triangleq f_i(X_i^n)$ represent the transmission of the user $i \in \mathcal{M}$, where $f_i(\cdot)$ is a mapping of the observations X_i^n to a message that user i transmits. Each user can recover the realization of $X_{\mathcal{M}}^n$ if and only if the transmission vector $\mathbf{F} \triangleq (F_1, F_2, \ldots, F_m)$ satisfies,

$$\lim_{n\to\infty} \frac{1}{n} H(X_{\mathcal{M}}^n | \mathbf{F}, X_i^n) = 0, \quad \forall i \in \mathcal{M}. \tag{3.1}$$

We define the data exchange rate vector in the similar way as in Section 2.1.

Definition 10. 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 a transmission vector $\mathbf{F}$ that satisfies

(2.4), and is such that

$$R_i = \lim_{n\to\infty} \frac{1}{n} H(F_i), \quad \forall i \in \mathcal{M}. \tag{3.2}$$

In [11], using the cut-set bounds it is shown that all the achievable *DE*-rate vectors necessarily belong to the following rate-region

$$\mathcal{R} \triangleq \{\mathbf{R} : R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}),\ \forall \mathcal{S} \subset \mathcal{M}\}, \tag{3.3}$$

where $X_{\mathcal{S}} \triangleq \{X_i \mid i \in \mathcal{S}\}$ and $R(\mathcal{S}) = \sum_{i\in\mathcal{S}} R_i$. Also, using a random coding argument, it can be shown that any rate vector that belongs to $\mathcal{R}$ can be achieved [11]. In [35] and [36] the authors provide explicit structured codes based on syndrome decoding that achieve the rate region for a Slepian-Wolf distributed source coding problem. This approach was further extended in [40] to a multiterminal setting. Achievable schemes proposed in [35], [36] and [40] require coding over large block sizes, and asymptotically converge to the optimum.

Each user wants to learn the realization of the joint process $X_{\mathcal{M}}^n$ while minimizing the communication cost $\sum_{i=1}^m \varphi_i(R_i)$, where φ_i, $i \in \mathcal{M}$, is a nondecreasing differentiable convex function. In this chapter, we propose a polynomial time algorithm that finds an optimal *DE*-rate vector w.r.t. the cost. This can be formally stated as the following convex optimization problem

$$\begin{aligned} \min_{\mathbf{R}} \ & \sum_{i=1}^m \varphi_i(R_i), \\ \text{s.t.} \ & R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}),\ \forall \mathcal{S} \subset \mathcal{M}. \end{aligned} \tag{3.4}$$

3.2 Combinatorial Algorithm

In this section we propose an efficient combinatorial algorithm that solves optimization problem (3.4). Most of the techniques developed in Chapter 2 will be useful here. Like in Section 2.2, we can rewrite the optimization problem 3.4 as follows

$$\min_{\beta\in\mathbb{R}_+} h(\beta), \tag{3.5}$$

where

$$\begin{aligned} h(\beta) \triangleq \min_{\mathbf{R}\in\mathbb{R}^m} & \sum_{i=1}^m \varphi_i(R_i), \\ \text{s.t.} \ & R(\mathcal{M}) = \beta, \ \ R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}),\ \forall \mathcal{S} \subset \mathcal{M}, \end{aligned} \tag{3.6}$$

where $\mathcal{S}^c \triangleq \mathcal{M} \setminus \mathcal{S}$. From Theorem 4, it follows that h is a convex function. It is clear that the rate region provided in (3.6) constitutes a base polyhedron of a set function

$$y_\beta(\mathcal{S}) = \begin{cases} H(X_\mathcal{S}|X_{\mathcal{S}^c}) & \text{if } \mathcal{S} \subset \mathcal{M}, \\ \beta & \text{if } \mathcal{S} = \mathcal{M}. \end{cases} \tag{3.7}$$

Then, by Definition 3, the dual set function f_β has the following form

$$f_\beta(\mathcal{S}) = \begin{cases} \beta - H(X_{\mathcal{S}^c}|X_\mathcal{S}) & \text{if } \emptyset \neq \mathcal{S} \subseteq \mathcal{M}, \\ 0 & \text{if } \mathcal{S} = \emptyset. \end{cases} \tag{3.8}$$

From Lemma 1, it follows that as long as β is a feasible sum-rate w.r.t. problem (3.6), *i.e.*, $B(y_\beta, \geq) \neq \emptyset$, it holds that $B(y_\beta, \geq) = B(f_\beta, \leq)$. Moreover, using the same proof as in Lemma 2, we can easily show that f_β is an intersecting submodular function. From Theorem 2 it follows that there exists a fully submodular function g_β, such that $P(f_\beta, \leq) = P(g_\beta, \leq)$, and it can be expressed as

$$g_\beta(\mathcal{S}) = \min_{\mathcal{P}} \left\{ \sum_{\mathcal{V} \in \mathcal{P}} f_\beta(\mathcal{V}) : \mathcal{P} \text{ is a partition of } \mathcal{S} \right\}. \tag{3.9}$$

Let us denote by $\mathcal{P}^*_\beta$ to be an optimal partitioning w.r.t. optimization (3.9) when $\mathcal{S} = \mathcal{M}$.

Remark 15. Since $P(f_\beta, \leq) = P(g_\beta, \leq)$, it follows that $B(g_\beta, \leq) = B(f_\beta, \leq)$ whenever $g_\beta(\mathcal{M}) = f_\beta(\mathcal{M}) = \beta$. In other words, as long as $\mathcal{P}^*_\beta = \{\{\mathcal{M}\}\}$, we have that β is feasible sum-rate w.r.t. problem (3.6).

3.2.1 Optimal Partitioning w.r.t. Dilworth Truncation

In this Section we briefly explain how to obtain an optimal partition $\mathcal{P}^*_\beta$ w.r.t. (3.9) by using Algorithm 3. From Remark 3, it follows that an optimal rate vector $\mathbf{R}^*$ obtained by applying Algorithm 3 satisfies:

$$\sum_{i=1}^{m} R^*_i = g_\beta(\mathcal{M}) = \sum_{\mathcal{V} \in \mathcal{P}^*_\beta} f_\beta(\mathcal{V}). \tag{3.10}$$

Let us denote by $\mathcal{S}_i$, $i = 1, 2, \ldots, m$, the minimizer set of the problem (2.59) in iteration i of Algorithm 3. By the "greediness" of Edmonds' algorithm, and the equivalence of Algorithms 1 and 3, it follows that

$$R^*(\mathcal{S}_i) = f_\beta(\mathcal{S}_i). \tag{3.11}$$

Lemma 11. *For any two iterations i, j of Algorithm 3, if $\mathcal{S}_i \cap \mathcal{S}_j \neq \emptyset$, then*

$$R^*(\mathcal{S}_i \cup \mathcal{S}_j) = f_\beta(\mathcal{S}_i \cup \mathcal{S}_j). \tag{3.12}$$

The proof of Lemma 11 directly follows from Lemma 3, since whenever $\mathcal{S}_i \cap \mathcal{S}_j \neq \emptyset$, intersecting submodular function f_β "behaves" as a fully submodular.

By making unions of the overlapping sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$ we end up with the disjoint collection of sets $\mathcal{V}_1, \ldots, \mathcal{V}_l$, *i.e.*, $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset$, $\forall i \neq j \in \{1, \ldots, l\}$. From Lemma 11, it follows that

$$R^*(\mathcal{V}_i) = f_\beta(\mathcal{V}_i), \quad \forall i \in \{1, \ldots, l\}. \tag{3.13}$$

From 3.13 and the fact that Edmonds' algorithm provides a rate vector that belongs to the basis polyhedron $B(g_\beta, \leq)$, it follows that

$$\sum_{i=1}^{l} R^*(\mathcal{V}_i) = R^*(\mathcal{M}) = g_\beta(\mathcal{M}). \tag{3.14}$$

By comparing (3.14) with (3.9), we obtain that $\mathcal{P}^*_\beta = \{\mathcal{V}_1, \ldots, \mathcal{V}_l\}$ is an optimal partitioning of set $\mathcal{M}$ w.r.t. (3.9).

Therefore, in order to construct an optimal partitioning of set $\mathcal{M}$ w.r.t. (3.9), we only need to keep track of the minimizing sets in Algorithm 3, and whenever they intersect to unionize them (see Algorithm 11).

Algorithm 11 Optimal Partitioning Algorithm

1: Let $j(1), j(2), \ldots, j(m)$ be any ordering of $1, 2, \ldots, m$.
2: Initialize $\mathcal{P}^0 = \emptyset$, $\mathbf{R}^* = \mathbf{0}$.
3: **for** $i = 1$ to m **do**
4: Let $\mathcal{S}_i$ be the minimizer of

$$R^*_{j(i)} = \min\{f_\beta(\mathcal{S} \cup \{i\}) - R^*(\mathcal{S}) : \ \mathcal{S} \subseteq \{j(1), \ldots, j(i-1)\}\}.$$

5: $\mathcal{T}_i = \mathcal{S}_i \cup [\cup\{\mathcal{V} : \mathcal{V} \in \mathcal{P}^{i-1}, \ \mathcal{S}_i \cap \mathcal{V} \neq \emptyset\}]$
6: $\mathcal{P}^i = \{\mathcal{T}_i\} \cup \{\mathcal{V} : \mathcal{V} \in \mathcal{P}^{i-1}, \ \mathcal{S}_i \cap \mathcal{V} = \emptyset\}$
7: **end for**
8: $\mathcal{P}^*_\beta = \mathcal{P}^m$.

Remark 16. Algorithm 11 compared to Algorithm 3, has an additional step that in iteration i checks whether set $\mathcal{S}_i$ intersects with any of the current partition sets. The complexity of this step is $\mathcal{O}(m)$, which implies that the complexity of Algorithm 11 is $\mathcal{O}(m^2 \cdot SFM(m))$.

Example 8. Let us consider a DMMS version of Example 3, where each user observes n memoryless observations of the joint process

$$\begin{aligned} X_1 &= \begin{bmatrix} W_1 & W_2 \end{bmatrix}^T, \\ X_2 &= \begin{bmatrix} W_2 & W_4 & W_5 & W_6 \end{bmatrix}^T, \\ X_3 &= \begin{bmatrix} W_3 & W_4 & W_5 & W_6 \end{bmatrix}^T, \end{aligned} \tag{3.15}$$

where $W_i \sim \text{Unif}\{0, 1, \ldots, q-1\}$, $i = 1, 2, \ldots, 6$. Let the sum-rate $\beta = 4 \log q$, where $\log q$ factor is because $H(W_i) = \log q$, $i = 1, 2, \ldots, 6$. Then, the intersecting submodular function f_β (see (3.8)) has the following form:

$$\begin{aligned} &f_\beta(\{1\}) = 0, \quad f_\beta(\{2\}) = 2 \log q, \quad f_\beta(\{3\}) = 2 \log q, \\ &f_\beta(\{1,2\}) = 3 \log q, \quad f_\beta(\{1,3\}) = 4 \log q, \quad f_\beta(\{2,3\}) = 3 \log q, \\ &f_\beta(\{1,2,3\}) = 4 \log q. \end{aligned} \tag{3.16}$$

By Applying Algorithm 11, we obtain the following rate allocation

$$\begin{aligned} R_1^* &= f_\beta(\{1\}) = 0, \\ R_2^* &= \min\{f_\beta(\{1,2\}) - R_1^*, f_\beta(\{2\})\} = f_\beta(\{2\}) = 2 \log q, \\ R_3^* &= \min\{f_\beta(\{1,2,3\}) - R_1^* - R_2^*, f_\beta(\{1,3\}) - R_1^*, f_\beta(\{2,3\}) - R_2^*, f_\beta(\{3\})\} \\ &= f_\beta(\{2,3\}) - R_2^* = \log q. \end{aligned}$$

Therefore, $\mathcal{S}_1 = \{1\}$, $\mathcal{S}_2 = \{2\}$, and $\mathcal{S}_\ni = \{2, 3\}$. Hence, $\mathcal{P}_\beta^* = \{\{1\}, \{2, 3\}\}$, and from (3.10) we obtain

$$\sum_{i=1}^{3} R_i^* = g_\beta(\{1,2,3\}) = f_\beta(\{1\}) + f_\beta(\{2,3\}) = 3 \log q. \tag{3.17}$$

3.2.2 Sum-Rate Cost

Let us first consider a simple sum-rate cost, $\varphi_i(R_i) = R_i$, $i \in \mathcal{M}$. Note that φ_i is a nondecreasing function. Hence, by Lemma 8, it follows that the minimizer β^* of function h, defined in (3.6), is at most $H(X_\mathcal{M})$.

To minimize the sum-rate, we set $\varphi_i(R_i) = R_i$ in (2.34):

$$\min_{0 \le \beta \le H(X_\mathcal{M})} \min_{\mathbf{R}} \sum_{i=1}^{m} R_i, \quad \text{s.t. } \mathbf{R} \in B(f_\beta, \le). \tag{3.18}$$

Using Remark 2, we can rewrite the above problem as follows:

$$\min_{0 \le \beta \le N} \beta, \quad \text{s.t. } \beta = g_\beta(\mathcal{M}). \tag{3.19}$$

Note that $g_\beta(\mathcal{M})$ can be obtained from Algorithm 11 along with the optimal partitioning $\mathcal{P}^*_\beta$ of set $\mathcal{M}$ according to (3.9). Next we show how to solve the optimization problem (3.19) with at most m calls of Algorithm 11. From (3.9) it follows that for every β, the function $g_\beta(\mathcal{M})$ can be represented as

$$g_\beta(\mathcal{M}) = |\mathcal{P}^*_\beta|\beta - \sum_{\mathcal{S}\in\mathcal{P}^*_\beta} (N - \text{rank}(\mathbf{A}_\mathcal{S})). \tag{3.20}$$

Therefore, $g_\beta(\mathcal{M})$ is a piecewise linear function in β. Moreover, function $g_\beta(\mathcal{M})$ is also concave, since it is obtained by minimizing the expression (3.9) over all possible partition sets of $\mathcal{M}$. Finally, since the cardinality of the optimal partitioning $\mathcal{P}^*_\beta$ ranges from 1 to m, the function $g_\beta(\mathcal{M})$ can have at most m linear segments. Hence, the optimization problem (3.19) can be solved in polynomial time by applying an algorithm explained using Example 3. From (3.19) we have that the optimal sum-rate is a breakpoint of $g_\beta(\mathcal{M})$ between

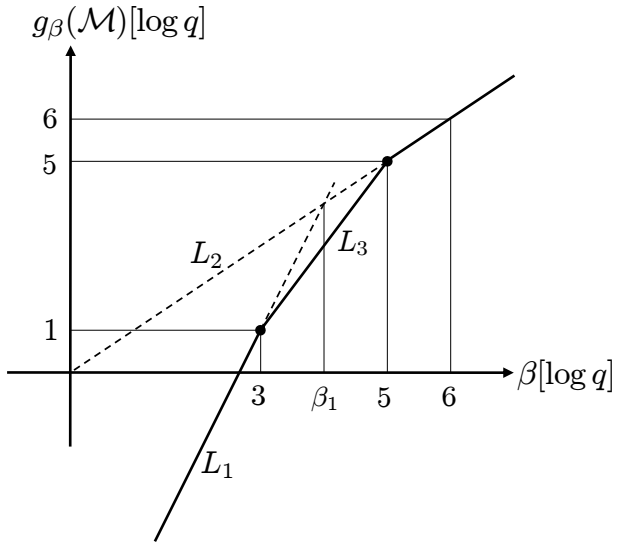


Figure 3.1: Minimal sum-rate can be obtained by intersecting linear segments. First, we intersect the line L_1 which corresponds to $\beta = 0$, with the 45-degree line L_2. The intersecting point β_1 belongs to the linear segment with slope greater than 1. Then, intersecting the segment L_3 to which β_1 belongs to with the 45-degree line L_2, we obtain $\beta_2 = 5$. Since the linear segment at β_2 has slope 1, we conclude that $5\log q$ is the minimum sum-rate.

the linear segment with slope 1 and the consecutive linear segment with the larger slope. For every β one can obtain a value of $g_\beta(\mathcal{M})$ and the corresponding optimal partition $\mathcal{P}^*_\beta$ w.r.t. (3.9) by applying Algorithm 11.

We start our algorithm by intersecting the line L_1 which belongs to the linear segment when $\beta = 0$ and the 45-degree line L_2 which corresponds to the last (rightmost) linear segment. The slope of the line L_1 as well as its value can be obtained in polynomial time by applying Algorithm 11 for $\beta = 0$. Since the function $g_\beta(\mathcal{M})$ is piecewise linear and concave,

the point of intersection β_1 must belong to the linear segment with slope smaller than $|\mathcal{P}_0^*|$, *i.e.*, $|\mathcal{P}_{\beta_1}^*| < |\mathcal{P}_0^*|$. β_1 can be obtained by equating β with $\sum_{\mathcal{S}\in\mathcal{P}_0^*}\beta - H(X_{\mathcal{S}^c}|X_{\mathcal{S}})$. Hence,

$$\beta_1 = \frac{\sum_{\mathcal{S}\in\mathcal{P}_0^*} H(X_{\mathcal{S}^c}|X_{\mathcal{S}})}{|\mathcal{P}_0^*| - 1} = 4\log q. \tag{3.21}$$

Next, by applying Algorithm 11 for $\beta = \beta_1$, we get $(g_{\beta_1}(\mathcal{M}), \mathcal{P}_{\beta_1}^*)$. Since $|\mathcal{P}_{\beta_1}^*| > 1$ (see Figure 3.1), we have not reached the breakpoint of interest yet, because the minimum sum-rate belongs to the linear segment of slope 1. Thus, we proceed by intersecting the line L_3 which belongs to the linear segment when $\beta = \beta_1$ with the the 45-degree line L_2. Like in the previous case, we obtain

$$\beta_2 = \frac{\sum_{\mathcal{S}\in\mathcal{P}_{\beta_1}^*} H(X_{\mathcal{S}^c}|X_{\mathcal{S}})}{|\mathcal{P}_{\beta_1}^*| - 1} = 5\log q. \tag{3.22}$$

Since $|\mathcal{P}_{\beta_2}^*| = 1$, we have reached the breakpoint of interest, and thus $R^*(\mathcal{M}) = 5\log q$. From Algorithm 11 we also obtain an optimal *DE*-rate vector $R_1^* = \log q$, $R_2^* = \log q$, and $R_3^* = 3\log q$. For an arbitrary $g_\beta(\mathcal{M})$, an optimal *DE*-rate vector w.r.t. the optimization (3.18) can be computed as shown in Algorithm 12.

Algorithm 12 Achieving the optimal sum-rate

1: Initialize $\beta = 0$.
2: **while** $|\mathcal{P}_\beta^*| > 1$ **do**
3:

$$\beta = \frac{\sum_{\mathcal{S}\in\mathcal{P}_\beta^*} H(X_{\mathcal{S}^c}|X_{\mathcal{S}})}{|\mathcal{P}_\beta^*| - 1}, \tag{3.23}$$

where $\mathcal{P}_\beta^*$, and thus $\mathbf{R}^*$, are obtained from Algorithm 11.
4: **end while**
5: β is the minimum sum-rate.

It is not hard to see that Algorithm 12 executes at most m iterations, since with each iteration the intersection point moves right to some other linear segment until it hits the optimal β (see Figure 3.1). Therefore, Algorithm 12 calls Algorithm 11 at most m times.

Remark 17. The complexity of Algorithm 12 is $\mathcal{O}(m^3 \cdot SFM(m))$.

It turns out that the sum-rate cost is the only objective for which we can obtain the exact solution to problem (3.5), when the underlying source model is DMMS. For other costs, we can only claim approximate solutions, that are guaranteed to be within some small distance from the actual solution (see Section 3.2.3).

Regarding the linear packet model we explored in Chapter 2, we assumed that the packets are indivisible, *i.e.*, we were not allowed to split packets into smaller chunks. If we consider the sum-rate cost, we can now answer the question: what is the optimal packet split in order to achieve information-theoretic optimal solution.

From (3.23), we have that the minimum sum-rate can be written as

$$\beta^* = \frac{\sum_{\mathcal{S}\in\mathcal{P}^*_{\beta^*}} H(X_{\mathcal{S}^c}|X_{\mathcal{S}})}{|\mathcal{P}^*_{\beta^*}|-1}. \tag{3.24}$$

Therefore, the optimal sum-rate can be achieved by splitting packets into $|\mathcal{P}^*_{\beta^*}|-1$ equally sized chunks, where $\mathcal{P}^*_{\beta^*}$ is a partition of the largest cardinality at break-point β^*. This is illustrated in the following example.

Example 9. Consider the example where three users observe the following parts of the file $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix}$:

$$\mathbf{x}_1 = \begin{bmatrix} w_1 & w_2 \end{bmatrix},$$
$$\mathbf{x}_2 = \begin{bmatrix} w_1 & w_3 \end{bmatrix},$$
$$\mathbf{x}_3 = \begin{bmatrix} w_2 & w_3 \end{bmatrix},$$

where $w_i \in \mathbb{F}_{2^n}$, $i = 1, 2, 3$, and n is an even number. It can be verified that by applying Algorithm 12 we obtain the optimal sum rate

$$R^*_1 + R^*_2 + R^*_3 = \frac{3}{2},$$

where $R^*_1 = R^*_2 = R^*_3 = \frac{1}{2}$. Hence, if we are allowed to split the packets into two equal parts of length $\frac{n}{2}$ bits, then the total communication would require $\frac{3n}{2}$ bits. However, if we were not allowed to that, then by Algorithm 4, the total communication would require $2n$ bits. In such a case, the optimal rate allocation is $R^*_1 = R^*_1 = 1$, $R^*_3 = 0$.

3.2.3 Minimizing Convex Function $h(\beta)$

When $\varphi(R_i) = \alpha_i R_i$, $\alpha_i > 0$, $i \in \mathcal{M}$, Algorithm 3 can be directly applied to the problem (3.6) in order to obtain $h(\beta)$, for any β. Additionally, we can also have capacity constraints on each user. It should be noted that all the results from Section 2.7 are applicable to the DMMS source model as well, and therefore, the capacity constraint problem won't be examined here.

Since $h(\beta)$ can be evaluated for any given β in polynomial time, what is left to do is to minimize convex function h. This can be done y applying a simple gradient descent method [3], which can reach the minimum of h within some precision ε. In other words, if we denote by β^*_{gd} the minimizer of problem (3.5) obtained by applying a gradient descent method, we have

$$|h(\beta^*_{gd}) - h(\beta^*)| \leq \varepsilon, \tag{3.25}$$

where β^* is the minimizer of (3.5).

We can achieve the same performance by discretizing β with some step ϵ, *i.e.*,

$$\hat{\beta}[n] = n\epsilon, \quad n = 0, 1, \ldots, \left\lceil \frac{H(X_{\mathcal{M}})}{\epsilon} \right\rceil. \tag{3.26}$$

Note that by Lemma 8 we only need to consider $\beta \leq H(X_{\mathcal{M}})$. Then, in order to minimize function h we can apply an algorithm similar to Algorithm 5. Before that, we need to figure out what is the minimum sum-rate, in order to obtain the set of feasible sum-rates β w.r.t. the problem (3.5). This can be done by applying Algorithm 11. However, since we are not interested in the exact solution to the problem (3.5), it is not necessary to compute the minimum sum-rate exactly; an approximate solution based on Algorithm 4 is shown in Algorithm 13.

Algorithm 13 An approximate solution to the minimum sum-rate problem

1: Initialize $start = 0$, $end = \left\lceil \frac{H(X_{\mathcal{M}})}{\epsilon} \right\rceil$.
2: **while** $\hat{\beta}[end] - \hat{\beta}[start] > \varepsilon$ **do**
3: $\quad index = \lceil \frac{start+end}{2} \rceil$.
4: $\quad$ Execute Algorithm 3 with parameter $\hat{\beta}[index]$.
5: $\quad$ **if** $\sum_{i=1}^{m} R_i^* = \hat{\beta}[index]$, **then**
6: $\quad\quad end = index$.
7: $\quad$ **else** $start = index$.
8: **end while**
9: $\hat{\beta}[end]$ is the minimum sum-rate.

Remark 18. The complexity of Algorithm 13 is $\mathcal{O}(m \cdot SFM(m) \cdot \log \frac{H(X_{\mathcal{M}})}{\epsilon})$.

Finally, in order to solve problem (3.5) approximately, we can apply Algorithm 14.

Remark 19. The complexity of Algorithm 14 is $\mathcal{O}(m \cdot SFM(m) \cdot \log \frac{H(X_{\mathcal{M}})}{\epsilon})$.

Let us denote by $\hat{\beta}^*$ the sum-rate β that corresponds to the solution obtained by applying Algorithm 14. Then,

$$|\hat{\beta}^* - \beta^*| \leq \epsilon, \tag{3.27}$$

where β^* is the actual minimum of function h. If we want to obtain a solution that is within some ε distance from the optimal solution $h(\beta^*)$, *i.e.*,

$$|h(\hat{\beta}^*) - h(\beta^*)| \leq \varepsilon, \tag{3.28}$$

Algorithm 14 Minimization of function h

1: Initialize $start = index^*$, $end = \left\lceil \frac{H(X_{\mathcal{M}})}{\epsilon} \right\rceil$, where $\hat{\beta}[index^*]$ is an approximate minimum sum-rate obtained from Algorithm 13.
2: $index = \lceil \frac{start+end}{2} \rceil$.
3: Compute $h(\hat{\beta}[index-1])$, $h(\hat{\beta}[index])$, and $h(\hat{\beta}[index+1])$.
4: **if** $h(\hat{\beta}[index]) \leq h(\hat{\beta}[index-1])$ and $h(\hat{\beta}[index]) \leq h(\hat{\beta}[index+1])$, **then**
5: $\quad \mathbf{R}^*$ that corresponds to the sum-rate $\hat{\beta}[index]$ is an optimal rate allocation
6: **else if** $h(\hat{\beta}[index-1]) \geq h(\hat{\beta}[index]) \geq h(\hat{\beta}[index+1])$, **then**
7: $\quad start = index + 1$.
8: **else if** $h(\hat{\beta}[index-1]) \leq h(\hat{\beta}[index]) \leq h(\hat{\beta}[index+1])$, **then**
9: $\quad end = index - 1$.
10: Go to Step 2.

then, we only need to relate parameters ϵ and ε. It is easy to see that the condition (3.28) is satisfied when

$$\epsilon \cdot \max_{0 \leq \beta \leq H(X_{\mathcal{M}})} \frac{dh(\beta)}{d\beta} = \varepsilon. \tag{3.29}$$

3.3 Non-Combinatorial Algorithm

In this section we explore the linear cost data exchange problem under the DMMS source model. The approach we take here is somewhat different from the combinatorial approach studied so far in this thesis. As we have already pointed out in Section 3.2.3, we can only reach an approximate solution to the problem (3.5) that is within some ε distance from the optimal solution. The question we ask here is: if we are not obtaining the exact solution to the problem, are there any other techniques, possibly less complex, that can reach the same "approximate" solution. It turns out that under the linear cost, it is possible to devise an algorithm of polynomial complexity based on convex optimization techniques. Therefore, we consider the following optimization problem:

$$\begin{aligned} \min_{\mathbf{R} \in \mathbb{R}^m} & \sum_{i=1}^{m} \alpha_i R_i, \\ \text{s.t.} \quad & R(\mathcal{S}) \geq H(X_{\mathcal{S}} | X_{\mathcal{S}^c}), \ \forall \mathcal{S} \subset \mathcal{M}, \end{aligned} \tag{3.30}$$

where $\alpha_i \geq 0$, $\forall i \in \mathcal{M}$.

3.3.1 One User Data Exchange Problem

Lets for the moment depart from the original problem, and consider the case where there is only one user interested in gaining access to the joint process, while the other users are helping this user in gaining such knowledge. More precisely, say that only user k is interested in learning the joint process. Then, this problem is known as a multi-terminal Slepian-Wolf problem [10] for which the achievable rate region has the following form:

$$\mathcal{R}_k = \left\{\mathbf{R} \in \mathbb{R}^{m-1} : R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}, X_k),\ \forall \mathcal{S} \subseteq \mathcal{M}_k\right\}, \tag{3.31}$$

where $\mathcal{M}_k \triangleq \mathcal{M} \setminus \{k\}$, and $\mathcal{S}^c \triangleq \mathcal{M}_k \setminus \mathcal{S}$. Hence, the underlying optimization problem has the following form

$$\min_{\mathbf{R}} \sum_{i \in \mathcal{M}_k} \alpha_i R_i, \quad \text{s.t. } \mathbf{R} \in \mathcal{R}_k. \tag{3.32}$$

Let us define a set function $y^{(k)} : 2^{\mathcal{M}_k} \to \mathbb{R}$ as follows

$$y^{(k)}(\mathcal{S}) = H(X_{\mathcal{S}}|X_{\mathcal{S}^c}, X_k), \quad \forall \mathcal{S} \subseteq \mathcal{M}_k. \tag{3.33}$$

Then, the dual set function $f^{(k)}$ can be derived from Definition 3:

$$\begin{aligned} f^{(k)}(\mathcal{S}) &= H(X_{\mathcal{M}_k}|X_k) - H(X_{\mathcal{S}^c}|X_{\mathcal{S}}, X_k) \\ &= H(X_{\mathcal{M}_k}|X_k) - H(X_{\mathcal{M}_k}|X_1) + H(X_{\mathcal{S}}|X_k) = H(X_{\mathcal{S}}|X_k). \end{aligned} \tag{3.34}$$

Due to the submodularity of entropy function, it immediately follows that function $f^{(k)}$ is fully submodular. Since the optimal solution of the problem (3.32) belongs to the base polyhedron $B(y^{(k)}, \geq)$, it immediately follows that optimization problem (3.32) can be equivalently represented as

$$\min_{\mathbf{R}} \sum_{i \in \mathcal{M}_k} \alpha_i R_i, \quad \text{s.t. } \mathbf{R} \in B(f^{(k)}, \leq). \tag{3.35}$$

Therefore, problem (3.32) can be solved analytically by using Edmonds' algorithm (see Algorithm 15)

Remark 20. The complexity of Algorithm 15 is $\mathcal{O}(m \log m + m\gamma)$, where γ is the complexity of computing entropy function.

3.3.2 Multiple User Data Exchange Problem

Going back to the original data exchange problem, in this section we take a convex optimization approach to solve the problem (3.30), where we use the single user solution as a key building block.

Algorithm 15 Edmonds' algorithm applied to problem (3.32)

1: Set $j(1), j(2), \ldots, j(m-1)$ to be an ordering of the elements in $\mathcal{M}_k$ such that

$$\alpha_{j(1)} \leq \alpha_{j(2)} \leq \cdots \leq \alpha_{j(m-1)}.$$

2: **for** $i = 1$ to $m-1$ **do**

3:
$$\begin{aligned} R_{j(i)} &= f^{(k)}(\{j(1), j(2), \ldots, j(i)\}) - f^{(k)}(\{j(1), j(2), \ldots, j(i-1)\}) \\ &= H(X_{j(i)} | X_k, X_{j(1)}, X_{j(2)}, \ldots, X_{j(i-1)}). \end{aligned}$$

4: **end for**

First, we observe that an achievable *DE*-rate vector $\mathbf{R}$ has to simultaneously belong to the rate regions $\mathcal{R}_k$ of each individual user $k \in \mathcal{M}$. Thus, the rate region $\mathcal{R}$ defined in (3.3) can be equivalently represented as

$$\mathcal{R} = \mathcal{R}_1 \cap \mathcal{R}_2 \cap \cdots \cap \mathcal{R}_m. \tag{3.36}$$

It is not hard to see that the following rate region is equivalent to (3.36)

$$\mathcal{R} = \{\mathbf{R} \in \mathbb{R}^m : R_i \geq R_i^{(k)}, \quad \forall i \in \mathcal{M}_k, \quad \text{s.t. } \mathbf{R}^{(k)} \in \mathcal{R}_k, \ \forall k \in \mathcal{M}\}. \tag{3.37}$$

Therefore, the optimal *DE*-rate vector $\mathbf{R}^*$ w.r.t. problem (3.30) can be obtained as follows

$$\begin{aligned} \min_{\mathbf{R}, \mathbf{R}^{(1)}, \ldots \mathbf{R}^{(m)}} & \sum_{i=1}^{m} \alpha_i R_i, \\ \text{s.t. } & R_i \geq R_i^{(k)}, \quad \forall i, k \in \mathcal{M}, \ i \neq k, \\ & \mathbf{R}^{(k)} \in \mathcal{R}_k, \quad \forall k \in \mathcal{M}. \end{aligned} \tag{3.38}$$

Optimization problem (3.38) has an exponential number of constraints, which makes it challenging to solve in polynomial time. To efficiently solve problem (3.38) we consider the Lagrangian dual of this problem. Now, we will go over the most important steps in constructing the dual optimization problem.

The Lagrangian associated with problem (3.38) has the following form.

$$\mathcal{L}(\mathbf{R}, \mathbf{R}^{(1)}, \ldots, \mathbf{R}^{(m)}, \mathbf{\Lambda}) = \sum_{i=1}^{m} \alpha_i R_i + \sum_{k=1}^{m} \sum_{i \in \mathcal{M}_k} \lambda_{i,k} (R_i^{(k)} - R_i), \tag{3.39}$$

where $\mathbf{\Lambda} \triangleq \{\lambda_{i,k} \mid i, k \in \mathcal{M}, \ i \neq k\}$. Then, the Lagrange dual function $\delta(\mathbf{\Lambda})$ equals to

$$\begin{aligned} \delta(\mathbf{\Lambda}) = \min_{\mathbf{R}, \mathbf{R}^{(1)}, \ldots, \mathbf{R}^{(m)}} & \mathcal{L}(\mathbf{R}, \mathbf{R}^{(1)}, \ldots, \mathbf{R}^{(m)}, \mathbf{\Lambda}), \\ \text{s.t. } & \mathbf{R}^{(k)} \in \mathcal{R}_k, \ \forall k \in \mathcal{M}. \end{aligned} \tag{3.40}$$

Let us denote by p^* the optimal value of the primal problem (3.38). Then, it can easily be shown that

$$\delta(\mathbf{\Lambda}) \leq p^*, \quad \forall \lambda_{i,k} \geq 0, \quad i,k \in \mathcal{M},\ i \neq k. \tag{3.41}$$

Hence,

$$\max_{\mathbf{\Lambda}} \{\delta(\mathbf{\Lambda}) \mid \lambda_{i,k} \geq 0, \quad i,k \in \mathcal{M},\ i \neq k\} \leq p^*. \tag{3.42}$$

Since optimization problem (3.38) is linear, it holds that

$$p^* = \max_{\mathbf{\Lambda}} \{\delta(\mathbf{\Lambda}) \mid \lambda_{i,k} \geq 0, \quad i,k \in \mathcal{M},\ i \neq k\}. \tag{3.43}$$

In other words, we have

$$\begin{aligned} p^* &= \max_{\mathbf{\Lambda}} \left\{ \min_{\mathbf{R},\mathbf{R}^{(1)} \in \mathcal{R}_1, \ldots, \mathbf{R}^{(m)} \in \mathcal{R}_m} \left\{ \sum_{i=1}^{m} \alpha_i R_i + \sum_{k=1}^{m} \sum_{i \in \mathcal{M}_k} \lambda_{i,k} (R_i^{(k)} - R_i) \right\} \right\} \\ &= \max_{\mathbf{\Lambda}} \left\{ \min_{\mathbf{R},\mathbf{R}^{(1)} \in \mathcal{R}_1, \ldots, \mathbf{R}^{(m)} \in \mathcal{R}_m} \left\{ \sum_{i=1}^{m} R_i (\alpha_i - \sum_{k=1, k \neq i}^{m} \lambda_{i,k}) + \sum_{k=1}^{m} \sum_{i \in \mathcal{M}_k} \lambda_{i,k} R_i^{(k)} \right\} \right\}. \end{aligned} \tag{3.44}$$

Term

$$\min_{\mathbf{R},\mathbf{R}^{(1)} \in \mathcal{R}_1, \ldots, \mathbf{R}^{(m)} \in \mathcal{R}_m} \sum_{i=1}^{m} R_i (\alpha_i - \sum_{k=1, k \neq i}^{m} \lambda_{i,k})$$

in (3.44) goes to $-\infty$ except when $\sum_{k=1,k\neq i}^{m} \lambda_{i,k} = \alpha_i$, $\forall i \in \mathcal{M}_k$, in which case it equals to 0. Therefore,

$$\begin{aligned} p^* = \max_{\mathbf{\Lambda}} & \sum_{k=1}^{m} \left\{ \min_{\mathbf{R}^{(k)} \in \mathcal{R}_k} \sum_{i \in \mathcal{M}_k} \lambda_{i,k} R_i^{(k)} \right\}, \\ \text{s.t.} & \sum_{k=1, k \neq i}^{m} \lambda_{i,k} = \alpha_i, \quad \lambda_{i,k} \geq 0, \quad \forall i,k \in \mathcal{M},\ i \neq k. \end{aligned} \tag{3.45}$$

Note that the inner minimization problem in (3.45)

$$\min_{\mathbf{R}^{(k)} \in \mathcal{R}_k} \sum_{i \in \mathcal{M}_k} \lambda_{i,k} R_i^{(k)} \tag{3.46}$$

can be solved analytically using Algorithm 15 for any $k \in \mathcal{M}$. Optimization problem (3.45) is a linear program (LP) with $\mathcal{O}(m^2)$ constraints, and it can be solved in polynomial time (w.r.t. the number of users). Here, we apply the dual subgradient method described below.

Starting with a feasible iterate $\lambda_{i,k}[0]$, $k \in \mathcal{M}$, $i \in \mathcal{M}_k$, w.r.t. the optimization problem (3.45), and the step size θ, every subsequent iterate $\lambda_{i,k}[j+1]$ can be recursively represented as an Euclidian projection of the vector

$$\left\{\lambda_{i,k}[j] + \theta \tilde{R}_i^{(k)}[j] \ : \ i,k \in \mathcal{M},\ i \neq k\right\} \tag{3.47}$$

onto the hyperplane

$$P = \left\{\mathbf{\Lambda} \mid \sum_{k=1,k\neq i}^{m} \lambda_{i,k} = \alpha_i, \quad \lambda_{i,k} \geq 0,\ \forall i,k \in \mathcal{M},\ i \neq k\right\}, \tag{3.48}$$

where $\tilde{\mathbf{R}}^{(k)}[j]$ is the optimal rate vector w.r.t the problem

$$\min_{\mathbf{R}^{(k)} \in \mathcal{R}_k} \sum_{i \in \mathcal{M}_k} \lambda_{i,k}[j] R_i^{(k)}. \tag{3.49}$$

Observe that

$$\left\{\tilde{R}_i^{(k)}[j] \ : \ i,k \in \mathcal{M},\ i \neq k\right\}$$

is the derivative w.r.t. $\mathbf{\Lambda}[j]$ of the Lagrange dual function $\delta(\mathbf{\Lambda}[j])$. Therefore with each iteration of the proposed method, we are taking small steps in the direction of the gradient towards the optimal solution. The Euclidian projection ensures that every iterate $\lambda_{i,k}[j]$ is feasible w.r.t. the optimization problem (3.45). It is not hard to verify that the following initial choice of $\lambda_{i,k}[0]$ is feasible.

$$\lambda_{i,k}[0] = \frac{\alpha_i}{m-1}, \quad \forall i,k \in \mathcal{M},\ i \neq k. \tag{3.50}$$

Now we briefly explain how to compute Euclidian projection in an efficient way. The complete analysis of this method can be found in [27]. The Euclidian projection can be formulated as the following optimization problem

$$\min_{\mathbf{\Lambda} \in P} \sum_{i=1}^{m} \sum_{k=1,k\neq i}^{m} \left(\lambda_{i,k} - \left(\lambda_{i,k}[j] + \theta \tilde{R}_i^{(k)}[j]\right)\right)^2. \tag{3.51}$$

Therefore, the dual update defined in (3.47) can be written as follows:

$$\{\lambda_{i,k}[j+1] \ : \ i,k \in \mathcal{M}, i \neq k\} = \operatorname*{argmin}_{\mathbf{\Lambda} \in P} \sum_{i=1}^{m} \sum_{k=1,k\neq i}^{m} \left(\lambda_{i,k} - \left(\lambda_{i,k}[j] + \theta \tilde{R}_i^{(k)}[j]\right)\right)^2. \tag{3.52}$$

Algorithm 16 summarizes the Euclidian projection method proposed in [27].

Algorithm 16 Euclidian projection

1: Let us define by $u_{i,k}$ the elements of the update vector (3.47)

$$u_{i,k} = \lambda_{i,k}[j] + \theta \tilde{R}_i^{(k)}[j], \ \forall i,k \in \mathcal{M}, \ i \neq k. \tag{3.53}$$

2: **for** $i = 1$ to m **do**

3: Sort the elements $u_{i,k}$, $k \in \mathcal{M} \setminus \{i\}$, in the descending order

$$u_{i,k_1} \geq u_{i,k_2} \geq \cdots \geq u_{i,k_{m-1}}.$$

4: Set $\hat{t}$ be the smallest t such that

$$\frac{1}{t}\left(\alpha_i - \sum_{r=1}^{t} u_{i,k_r}\right) \leq -u_{i,k_{t+1}},$$

or set $\hat{t} = m-1$ if no such t exists.

5: Set

$$\lambda_{i,k}[j+1] = \begin{cases} u_{i,k} + \frac{\alpha_i - \sum_{r=1}^{\hat{t}} u_{i,k_r}}{\hat{t}}, & \text{if } \ k \in \{k_1, k_2, \ldots, k_{\hat{t}}\} \\ 0, & \text{otherwise.} \end{cases}$$

6: **end for**

Remark 21. In Step 1 of Algorithm 16 we call Algorithm 15 m times. Thus, the complexity of this step is $\mathcal{O}(m^2 \log m + m^2\gamma)$. In the for loop, the most complex step is sorting the elements of $u_{i,k}$ which can be done in $\mathcal{O}(m \log m)$ time. Therefore, the complexity of Algorithm 16 is $\mathcal{O}(m^2 \log m + m^2\gamma)$.

The dual subgradient method can provide near optimal dual solution for a sufficiently small step size θ, and possibly, large number of iteration. However, such method does not directly provide a primal optimal solution. In [38] and [32], the authors proved that averaging over all iterations of the dual subgradient algorithm can provide near optimal primal solution. For instance,

$$\hat{R}_i^{(k)}[l] = \frac{1}{l}\sum_{j=0}^{l-1} \tilde{R}_i^{(k)}[j], \quad \forall i,k \in \mathcal{M}, \ i \neq k \tag{3.54}$$

is a near optimal collection of vectors $\mathbf{R}^{(k)}$, $k \in \mathcal{M}$, w.r.t. problem (3.38) after l iterations of the dual subgradient method. Then, a near optimal *DE*-rate vector $\hat{\mathbf{R}}$ w.r.t. problem (3.38)

can be obtained as follows

$$\hat{R}_i[l] = \max_{k \in \mathcal{M} \setminus \{i\}} \hat{R}_i^{(k)}[l], \quad \forall i \in \mathcal{M}. \tag{3.55}$$

3.3.3 Convergence Analysis of the Averaging Method

In this section we establish the relationship between the number of iterations l and the step size θ in the dual subgradient method such that the solution obtained by averaging is within ε distance from the primal optimal solution, *i.e.*,

$$\left| \sum_{i=1}^{m} \alpha_i \hat{R}_i[l] - p^* \right| \leq \varepsilon. \tag{3.56}$$

Theorem 9. *For any choice of precision parameter ε, step size θ and number of iterations l that satisfy condition (3.56) can be selected as follows.*

$$\theta = \frac{\varepsilon}{\max_{i \in \mathcal{M}} \alpha_i^2 + m \left(H(X_{\mathcal{M}})\right)^2}, \tag{3.57}$$

$$l = \frac{1}{\theta^2}. \tag{3.58}$$

Proof. First, note that all rate vectors $\tilde{\mathbf{R}}^{(k)}[j]$, $k \in \mathcal{M}$, $j = 0, 1, 2, \ldots$, are feasible w.r.t the primal problem (3.38). Hence, their linear combination (3.54) is also feasible. Finally, according to (3.38), the maximization over k in (3.55) provides a feasible *DE*-rate vector w.r.t. problem (3.38). Therefore, it immediately follows that

$$\sum_{i=1}^{m} \alpha_i \hat{R}_i[l] \geq p^*. \tag{3.59}$$

In order to obtain relationship between l and ε in (3.56), we can apply results from [32], where the main condition that needs to be satisfied is that the domain set of the primal problem (3.38) is compact. We haven't explicitly mentioned that in the formulation of problem (3.38), but all the rate vectors $\mathbf{R}$, $\mathbf{R}^{(k)}$, $\forall k \in \mathcal{M}$ can be trivially bounded. Note that by enforcing $0 \leq R_i \leq H(X_{\mathcal{M}})$, $\forall i \in \mathcal{M}$, and $0 \leq R_i^{(k)} \leq H(X_{\mathcal{M}})$, $\forall i, k \in \mathcal{M}$, $i \neq k$, we do not change the optimal solution of problem (3.38). Now, from [32] to obtain the following bound.

$$\sum_{i=1}^{m} \alpha_i \hat{R}_i[l] \leq p^* + \frac{1}{2l\theta} \sum_{i=1}^{m} \sum_{k=1, k \neq i}^{m} \left(\lambda_{i,k}[0]\right)^2 + \frac{\theta}{2} \sum_{j=0}^{l-1} \sum_{i=1}^{m} \sum_{k=1, k \neq i}^{m} \left(\tilde{R}_i^{(k)}[j]\right)^2. \tag{3.60}$$

To further bound (3.60) we bound each term individually as follows. From (3.50), the following inequality holds.

$$\sum_{i=1}^{m}\sum_{k=1,k\neq i}^{m}\left(\lambda_{i,k}[0]\right)^2=\sum_{i=1}^{m}\sum_{k=1,k\neq i}^{m}\left(\frac{\alpha_i}{m-1}\right)^2\leq 2\max_{i\in\mathcal{M}}\alpha_i. \tag{3.61}$$

Since

$$\sum_{i=1}^{m}\tilde{R}_i^{(k)}[j]\leq H(X_{\mathcal{M}}),\ \forall k\in\mathcal{M},\ k\neq i,\ j=0,1,2,\ldots \tag{3.62}$$

it follows that

$$\sum_{i=1}^{m}\left(\tilde{R}_i^{(k)}[j]\right)^2\leq\left(H(X_{\mathcal{M}})\right)^2,\ \forall k\in\mathcal{M},\ k\neq i,\ j=0,1,2,\ldots \tag{3.63}$$

Therefore,

$$\sum_{j=0}^{l-1}\sum_{i=1}^{m}\sum_{k=1,k\neq i}^{m}\left(\tilde{R}_i^{(k)}[j]\right)^2\leq lm\left(H(X_{\mathcal{M}})\right)^2. \tag{3.64}$$

From (3.59), (3.60), (3.61) and (3.64) it follows that

$$p^*\leq\sum_{i=1}^{m}\alpha_i\hat{R}_i[l]\leq p^*+\frac{\max_{i\in\mathcal{M}}\alpha_i^2}{l\theta}+\frac{m\left(H(X_{\mathcal{M}})\right)^2\theta}{2}. \tag{3.65}$$

Therefore,

$$\left|\sum_{i=1}^{m}\alpha_i\hat{R}_i[l]-p^*\right|\leq\frac{\max_{i\in\mathcal{M}}\alpha_i^2}{l\theta}+m\left(H(X_{\mathcal{M}})\right)^2\theta. \tag{3.66}$$

By comparing (3.56) with (3.66), it follows that if we choose

$$\varepsilon=\frac{\max_{i\in\mathcal{M}}\alpha_i^2}{l\theta}+m\left(H(X_{\mathcal{M}})\right)^2\theta, \tag{3.67}$$

the condition (3.56) is satisfied. Now, let $l=\frac{1}{\theta^2}$. Then, (3.67) becomes

$$\varepsilon=\theta\left(\max_{i\in\mathcal{M}}\alpha_i^2+m\left(H(X_{\mathcal{M}})\right)^2\right). \tag{3.68}$$

□

Remark 22. The choice of parameters θ and l in Theorem 9 guarantees that the condition (3.56) is satisfied. One can easily come up with the different solution with fewer number of iterations.

Putting together all the results so far, minimum linear cost data exchange problem can be solved in polynomial time by applying Algorithm 17.

Remark 23. The complexity of Algorithm 17 is $\mathcal{O}((m^4 \log m + m^4\gamma) \cdot \lceil \frac{1}{\varepsilon^2} \rceil)$.

Example 10. Let us consider a DMMS version of Example 9, where three users observe n memoryless observations of the joint process $W = \begin{bmatrix} W_1 & W_2 & W_3 \end{bmatrix}$:

$$\begin{aligned} X_1 &= \begin{bmatrix} W_1 & W_2 \end{bmatrix}, \\ X_2 &= \begin{bmatrix} W_1 & W_3 \end{bmatrix}, \\ X_3 &= \begin{bmatrix} W_2 & W_3 \end{bmatrix}, \end{aligned}$$

where $W_i \sim \text{Unif}\{0, 1, \ldots, q-1\}$, $i = 1, 2, 3$. The goal is to minimize the sum-rate $R_1 + R_2 + R_3$. For the precision parameter $\varepsilon = 0.01$, from Theorem 9 it follows that $\theta = 0.00036$, and $l = 7.84 \cdot 10^6$. We note that the guaranteed number of iterations required is very large, but nevertheless it shows that the goal can be achieved in finite number of steps. Empirical results on the number of iterations required are much more promising. After 100 iterations of Algorithm 17, we obtain $\hat{R}_1[100] = 0.51$, $\hat{R}_2[100] = 0.5$, $\hat{R}_3[100] = 0.5$, and obviously satisfied condition

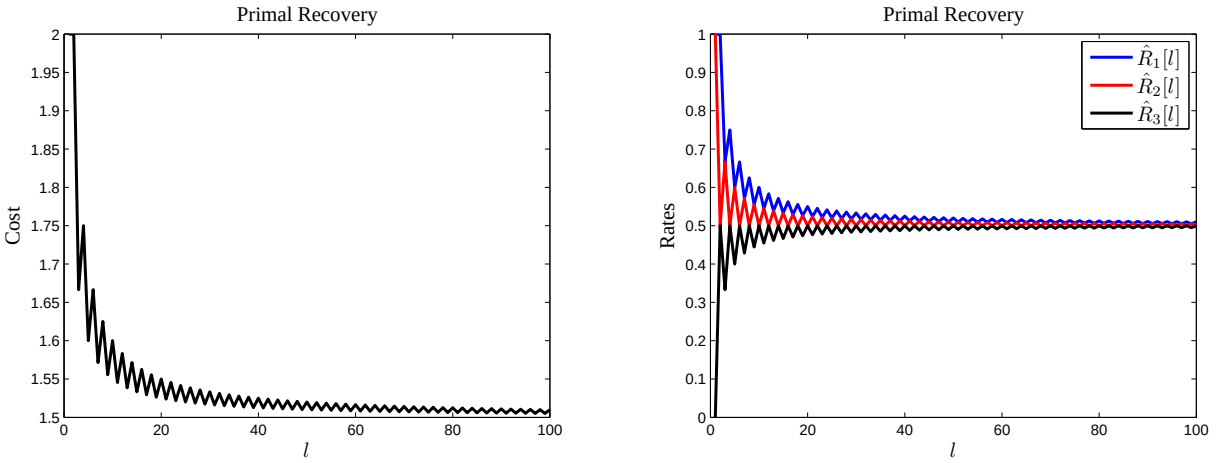
$$\left| \sum_{i=1}^{3} \hat{R}_i[100] - \sum_{i=1}^{3} R_i^* \right| \leq \varepsilon, \tag{3.69}$$

where $R_1^* = R_2^* = R_3^* = 0.5$ according to Example 9.

Example 11. Consider the same DMMS source model as in Example 8,

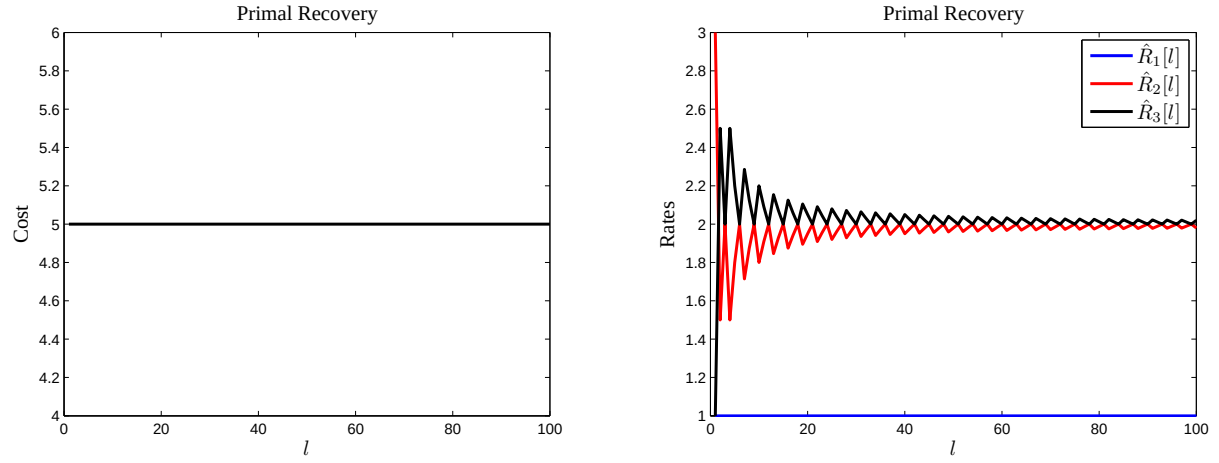
$$\begin{aligned} X_1 &= \begin{bmatrix} W_1 & W_2 \end{bmatrix}^T, \\ X_2 &= \begin{bmatrix} W_2 & W_4 & W_5 & W_6 \end{bmatrix}^T, \\ X_3 &= \begin{bmatrix} W_3 & W_4 & W_5 & W_6 \end{bmatrix}^T, \end{aligned} \tag{3.70}$$

where $W_i \sim \text{Unif}\{0, 1, \ldots, q-1\}$, $i = 1, 2, \ldots, 6$. The goal is to minimize the sum-rate $R_1 + R_2 + R_3$. We use the same precision parameter $\varepsilon = 0.01$ as in Example 10. From Figure 3.3(a) we note that the optimal sum-rate cost is reached after first iteration of Algorithm 17, with optimal *DE*-rate vector $\hat{R}_1[0] = 1$, $\hat{R}_2[0] = 3$, $\hat{R}_3[0] = 1$. After 100 iterations, Algorithm 17 converged to another optimal *DE*-rate vector $\hat{R}_1[100] = 1.01$, $\hat{R}_2[100] = 1.98$, $\hat{R}_3[100] = 2.02$.



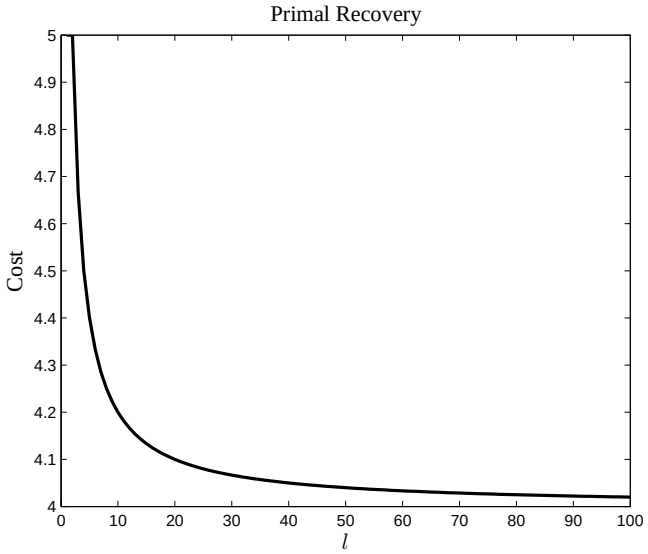
(a) Cost (b) Rates

Figure 3.2: Algorithm 17 applied to Example 10, for the precision parameter $\varepsilon = 0.01$. After 100 iterations of Algorithm 17, the solution obtained is $\hat{R}_1[100] = 0.51$, $\hat{R}_2[100] = 0.5$, $\hat{R}_3[100] = 0.5$, and it satisfies condition (3.69).

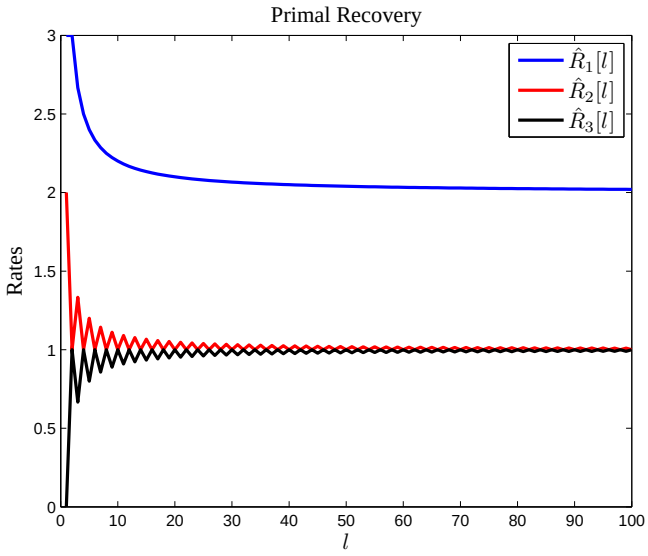


(a) Cost (b) Rates

Figure 3.3: Algorithm 17 applied to Example 11, for the precision parameter $\varepsilon = 0.01$. After 100 iterations of Algorithm 17, the solution obtained is $\hat{R}_1[100] = 1.01$, $\hat{R}_2[100] = 1.98$, $\hat{R}_3[100] = 2.02$.



(a) Cost



(b) Rates

Figure 3.4: Algorithm 17 applied to Example 12, for the precision parameter $\varepsilon = 0.01$. After 100 iterations of Algorithm 17, the solution obtained is $\hat{R}_1[100] = 2.01$, $\hat{R}_2[100] = 1$, $\hat{R}_3[100] = 1$.

Example 12. Consider the following three user example,

$$\begin{aligned} X_1 &= \begin{bmatrix} W_3 & W_4 & W_5 & W_6 \end{bmatrix}^T, \\ X_2 &= \begin{bmatrix} W_1 & W_2 & W_3 \end{bmatrix}^T, \\ X_3 &= \begin{bmatrix} W_1 & W_2 & W_4 \end{bmatrix}^T, \end{aligned} \tag{3.71}$$

where $W_i \sim \text{Unif}\{0, 1, \ldots, q-1\}$, $i = 1, 2, \ldots, 6$. The goal is to minimize the sum-rate $R_1 + R_2 + R_3$. For the parameter $\varepsilon = 0.01$ we again have that 100 iterations of Algorithm 17 are enough to achieve the goal, *i.e.*, $\hat{R}_1[100] = 2.01$, $\hat{R}_2[100] = 1$, $\hat{R}_3[100] = 1$ (see Figure 3.4). It can be verified that the optimal *DE*-rate vector is $R_1^* = 2$, $R_2^* = 1$, $R_3^* = 1$.

Algorithm 17 Minimizing linear cost

1: For a precision ε, set parameters θ and l according to Theorem 9.

2: Set

$$\lambda_{i,k} = \frac{\alpha_i}{m-1}, \quad \forall i,k \in \mathcal{M}, i \neq k.$$

3: **for** $j = 0$ to $l-1$ **do**

4: Using Algorithm 15, compute $\tilde{R}_i^{(k)}[j]$, $\forall i,k \in \mathcal{M}$, $i \neq k$, the minimizer of

$$\min_{\mathbf{R}^{(k)}} \sum_{i \in \mathcal{M}_k} \lambda_{i,k}[j] R_i^{(k)}, \quad \forall k \in \mathcal{M}.$$

5: Using Algorithm 16 compute

$$\{\lambda_{i,k}[j+1] \ : \ i,k \in \mathcal{M}, i \neq k\} = \operatorname*{argmin}_{\mathbf{\Lambda} \in P} \sum_{i=1}^{m} \sum_{k=1, k \neq i}^{m} \left(\lambda_{i,k} - \left(\lambda_{i,k}[j] + \theta \tilde{R}_i^{(k)}[j]\right)\right)^2,$$

where

$$P = \left\{ \mathbf{\Lambda} \mid \sum_{k=1, k \neq i}^{m} \lambda_{i,k} = \alpha_i, \quad \lambda_{i,k} \geq 0, \ \forall i,k \in \mathcal{M}, \ i \neq k \right\}.$$

6: A near optimal *DE*-rate vector $\hat{\mathbf{R}}$ can be obtained via averaging method

$$\hat{R}_i[l] = \max_{k \in \mathcal{M} \setminus \{i\}} \frac{1}{l} \sum_{j=0}^{l-1} \tilde{R}_i^{(k)}[j], \quad \forall i \in \mathcal{M}.$$

7: **end for**

Chapter 4

Data Exchange Problem - Extensions

In this chapter we present two modified versions of the original data exchange problem. First, we consider the problem with helpers, where some group of users is not interested in gaining access to the file or joint process, but they are willing to help other users in achieving their goal. Second, we consider a multisource multicast problem, where nodes in the system are communicating among themselves through wires. The communication network is represented by an acyclic graph, with possibly capacity constrained links.

4.1 Data Exchange Problem with Helpers

We consider a setup with m users out of which some subset of them is interested in gaining access to a file or a random process. Let $X_1, X_2, \ldots, X_m$, $m \geq 2$, denote the components of a discrete memoryless multiple source (DMMS) with a given joint probability mass function. Each user $i \in \mathcal{M} \triangleq \{1, 2, \ldots, m\}$ observes n i.i.d. realizations of the corresponding random variable X_i.

Let $\mathcal{A} = \{1, 2, \ldots, t\} \subseteq \mathcal{M}$ be the subset of users interested in gaining access to the file, *i.e.*, learning the joint process $X_{\mathcal{M}} \triangleq (X_1, \ldots, X_m)$. The remaining users $\{t+1, \ldots, m\}$ serve as helpers, *i.e.*, they are not interested in recovering the file, but they are willing to help users in the set $\mathcal{A}$ to obtain it. In [11], Csiszár and Narayan showed that to deliver the file to all users in a setup with general DMMS, *interactive communication is not needed*. As a result, in the sequel WLOG we can assume that the transmission of each user is only a function of its own initial observations. Let $F_i \triangleq f_i(X_i^n)$ represent the transmission of the user $i \in \mathcal{M}$, where $f_i(\cdot)$ is any desired mapping of the observations X_i^n. For each user in $\mathcal{A}$ in order to recover the entire file, transmissions F_i, $i \in \mathcal{M}$, should satisfy,

$$\lim_{n\to\infty} \frac{1}{n} H(X_{\mathcal{M}}^n | \mathbf{F}, X_l^n) = 0, \quad \forall l \in \mathcal{A}, \tag{4.1}$$

where $\mathbf{F} \triangleq (F_1, F_2, \ldots, F_m)$.

Definition 11. A rate vector $\mathbf{R} = (R_1, R_2, \ldots, R_m)$ is an achievable rate vector if there exists a communication scheme with transmitted messages $\mathbf{F} = (F_1, F_2, \ldots, F_m)$ that satisfies (4.1), and is such that

$$R_i = \lim_{n\to\infty} \frac{1}{n} H(F_i), \quad \forall i \in \mathcal{M}. \tag{4.2}$$

It is easy to show using cut-set bounds that all the achievable rate vectors necessarily belong to the following region

$$\mathcal{R} \triangleq \{\mathbf{R} \in \mathbb{R}^m : R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}),\ \forall \mathcal{S} \subset \mathcal{M},\ \mathcal{A} \not\subseteq \mathcal{S}\}, \tag{4.3}$$

where $R(\mathcal{S}) \triangleq \sum_{i\in\mathcal{S}} R_i$. Also, using a random binning argument, it can be shown that the rate region $\mathcal{R}$ is an achievable rate region [11].

In this section, we aim to design a polynomial complexity algorithm that delivers the file to all users in $\mathcal{A}$ while simultaneously minimizing a linear communication cost function $\sum_{i=1}^m \alpha_i R_i$, where $\alpha_i \geq 0$, $\forall \alpha_i \in \mathcal{M}$. For the general DMMS source model, the question of how many bits each user should transmit in an optimal scheme reduces to the following optimization problem.

$$\begin{aligned} \min_{\mathbf{R}\in\mathbb{R}^m} & \sum_{i=1}^m \alpha_i R_i, \\ \text{s.t.}\ & R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}),\ \forall \mathcal{S} \subset \mathcal{M},\ \mathcal{A} \not\subseteq \mathcal{S}. \end{aligned} \tag{4.4}$$

After solving the above optimization problem, the explicit communication scheme can be designed using methods proposed in [35], [36], [40].

For the linear packet model defined in Section 2.1, the cut-set region $\hat{\mathcal{R}}$ has the following form

$$\hat{\mathcal{R}} \triangleq \left\{\mathbf{R} \in \mathbb{Q}^m : R(\mathcal{S}) \geq N - \text{rank}(\mathbf{A}_{\mathcal{M}\setminus\mathcal{S}}),\ \forall \mathcal{S} \subset \mathcal{M},\ \mathcal{A} \not\subseteq \mathcal{S}\right\}, \tag{4.5}$$

From Theorem 1 it immediately follows that all rational vectors that belong to $\hat{\mathcal{R}}$ can be achieved via linear network coding provided that the underlying field size is large enough, and packet splitting is allowed. The number of symbols in $\mathbb{F}_q$ each user transmits in an optimal scheme can be figured out by solving the following optimization problem

$$\min_{\mathbf{R}} \sum_{i=1}^m \alpha_i R_i, \quad \text{s.t. } \mathbf{R} \in \hat{\mathcal{R}}. \tag{4.6}$$

After solving the above optimization problem, the corresponding communication scheme can be designed using the same methods as in Section 2.5. Optimization problems (4.4) and (4.6) are essentially the same. Methods and algorithms we use to solve these problems will be explained on the DMMS version of this problem.

4.1.1 Deterministic Algorithm

In order to construct an efficient algorithm that solves problem (4.4) we take similar approach as in Section 3.3. First, we observe that an achievable rate vector $\mathbf{R} \in \mathcal{R}$ has to simultaneously belong to the rate regions $\mathcal{R}_k$, $k \in \mathcal{A}$, of each user interested in gaining access to the joint process, where

$$\mathcal{R}_k = \left\{ \mathbf{R} \in \mathbb{R}^{m-1} : R(\mathcal{S}) \geq H(X_{\mathcal{S}} | X_{\mathcal{S}^c}, X_k),\ \forall \mathcal{S} \subseteq \mathcal{M}_k \right\}, \tag{4.7}$$

and $\mathcal{M}_k = \mathcal{M} \setminus \{k\}$. Thus, the rate region $\mathcal{R}$ defined in (4.3) can be equivalently represented as

$$\mathcal{R} = \mathcal{R}_1 \cap \mathcal{R}_2 \cap \cdots \cap \mathcal{R}_t. \tag{4.8}$$

It is not hard to see that the following rate region is equivalent to (4.8)

$$\mathcal{R} = \{\mathbf{R} \in \mathbb{R}^m : R_i \geq R_i^{(k)},\ \ \forall i \in \mathcal{M}_k,\ \ \text{s.t. } \mathbf{R}^{(k)} \in \mathcal{R}_k,\ \forall k \in \mathcal{A}\}. \tag{4.9}$$

Therefore, the optimal rate vector $\mathbf{R}^*$ w.r.t. problem (4.4) can be obtained as follows

$$\begin{aligned} \min_{\mathbf{R}, \mathbf{R}^{(1)}, \ldots \mathbf{R}^{(m)}} & \sum_{i=1}^{m} \alpha_i R_i, \\ \text{s.t. } & R_i \geq R_i^{(k)}, \quad \forall k \in \mathcal{A},\ \forall i \in \mathcal{M}_k, \\ & \mathbf{R}^{(k)} \in \mathcal{R}_k, \quad \forall k \in \mathcal{A}. \end{aligned} \tag{4.10}$$

As in Section 3.3, to efficiently solve problem (4.10) we consider the Lagrangian dual of this problem.

$$\begin{aligned} p^* = \max_{\mathbf{\Lambda}} & \sum_{k=1}^{m} \left\{ \min_{\mathbf{R}^{(k)} \in \mathcal{R}_k} \sum_{i \in \mathcal{M}_k} \lambda_{i,k} R_i^{(k)} \right\}, \\ \text{s.t.} & \sum_{k=1, k \neq i}^{m} \lambda_{i,k} = \alpha_i, \quad \lambda_{i,k} \geq 0, \quad \forall k \in \mathcal{A},\ \forall i \in \mathcal{M}_k. \end{aligned} \tag{4.11}$$

Note that the inner minimization problem in (4.11)

$$\min_{\mathbf{R}^{(k)} \in \mathcal{R}_k} \sum_{i \in \mathcal{M}_k} \lambda_{i,k} R_i^{(k)} \tag{4.12}$$

can be solved analytically using Algorithm 15 for any $k \in \mathcal{A}$. Optimization problem (4.11) is a linear program (LP) with $\mathcal{O}(m^2)$, and it can be solved in polynomial time (w.r.t. the number of users) using the same dual subgradient method as in Section 3.3. Starting with a feasible iterate $\lambda_{i,k}[0]$, $k \in \mathcal{A}$, $i \in \mathcal{M}_k$, w.r.t. the optimization problem (4.11), and the

step size θ, every subsequent iterate $\lambda_{i,k}[j+1]$ can be recursively represented as an Euclidian projection of the vector

$$\left\{\lambda_{i,k}[j] + \theta \tilde{R}_i^{(k)}[j] \ : \ \forall k \in \mathcal{A}, \ \forall i \in \mathcal{M}_k\right\} \tag{4.13}$$

onto the hyperplane

$$P = \left\{\mathbf{\Lambda} \mid \sum_{k=1,k\neq i}^{t} \lambda_{i,k} = \alpha_i, \quad \lambda_{i,k} \geq 0, \ \forall k \in \mathcal{A}, \ \forall i \in \mathcal{M}_k\right\}, \tag{4.14}$$

where $\tilde{\mathbf{R}}^{(k)}[j]$ is the optimal rate vector w.r.t the problem

$$\min_{\mathbf{R}^{(k)} \in \mathcal{R}_k} \sum_{i \in \mathcal{M}_k} \lambda_{i,k}[j] R_i^{(k)}. \tag{4.15}$$

It is not hard to verify that the following initial choice of $\lambda_{i,k}[0]$ is feasible.

$$\lambda_{i,k}[0] = \begin{cases} \frac{\alpha_i}{t} & \text{if } i \notin \mathcal{A} \\ \frac{\alpha_i}{t-1} & \text{if } i \in \mathcal{A} \setminus \{k\} \end{cases}, \quad \forall k \in \mathcal{A}, \ \forall i \in \mathcal{M}_k. \tag{4.16}$$

Finally, in order to obtain a near optimal solution to problem (4.4) we take an average over all iterations of the dual subgradient method

$$\hat{R}_i[l] = \max_{k \in \mathcal{A}\setminus\{i\}} \frac{1}{l} \sum_{j=0}^{l-1} \tilde{R}_i^{(k)}[j], \quad \forall i \in \mathcal{M}. \tag{4.17}$$

Using Theorem 9, we can relate parameters l and θ in order to satisfy the following condition

$$\left| \sum_{i=1}^{m} \alpha_i \hat{R}_i[l] - p^* \right| \leq \varepsilon. \tag{4.18}$$

Putting all these results together, minimum linear cost data exchange problem with helpers can be solved in polynomial time by applying Algorithm 18.

Remark 24. The complexity of Algorithm 18 is $\mathcal{O}((m^4 \log m + m^4\gamma) \cdot \lceil \frac{1}{\varepsilon^2} \rceil)$.

Example 13. Consider the problem with 6 users, where each one of them observes n memoryless observations of the joint process $W = \begin{bmatrix} W_1 & W_2 & W_3 \end{bmatrix}$:

$$X_1 = W_1 + W_2, \quad X2 = W_1 + W_3, \quad X_2 = W_2 + W_3, \quad X_4 = W_1, \quad X_5 = W_2, \quad X_6 = W_3,$$

where $W_i \sim \text{Unif}\{0, 1, \ldots, q-1\}$, $i = 1, 2, 3$. Let $\mathcal{A} = \{1, 2, 3\}$, and $\varepsilon = 0.01$. The goal is to minimize the sum-rate $\sum_{i=1}^{6} R_i$. It can be shown that the optimal rate vector w.r.t the

Algorithm 18 Minimizing linear cost

1: For a precision ε, set parameters θ and l according to Theorem 9.

2: Set

$$\lambda_{i,k}[0] = \begin{cases} \frac{\alpha_i}{t} & \text{if } i \notin \mathcal{A} \\ \frac{\alpha_i}{t-1} & \text{if } i \in \mathcal{A} \setminus \{k\} \end{cases}, \quad \forall k \in \mathcal{A},\ \forall i \in \mathcal{M}_k.$$

3: **for** $j = 0$ to $l-1$ **do**

4: Using Algorithm 15, compute $\tilde{R}_i^{(k)}[j]$, $\forall k \in \mathcal{A}$, $\forall i \in \mathcal{M}_k$, the minimizer of

$$\min_{\mathbf{R}^{(k)}} \sum_{i \in \mathcal{M}_k} \lambda_{i,k}[j] R_i^{(k)}, \quad \forall k \in \mathcal{A}.$$

5: Using Algorithm 16 compute

$$\{\lambda_{i,k}[j+1] \ : \ k \in \mathcal{A},\ i \in \mathcal{M}_k\} = \operatorname*{argmin}_{\mathbf{\Lambda} \in P} \sum_{i=1}^{m} \sum_{k=1, k \neq i}^{m} \left(\lambda_{i,k} - \left(\lambda_{i,k}[j] + \theta \tilde{R}_i^{(k)}[j]\right)\right)^2,$$

where

$$P = \left\{ \mathbf{\Lambda} \mid \sum_{k=1, k \neq i}^{t} \lambda_{i,k} = \alpha_i, \quad \lambda_{i,k} \geq 0,\ \forall k \in \mathcal{M},\ \forall i \in \mathcal{M}_k \right\}.$$

6: A near optimal rate vector $\hat{\mathbf{R}}$ can be obtained via averaging method

$$\hat{R}_i[l] = \max_{k \in \mathcal{A} \setminus \{i\}} \frac{1}{l} \sum_{j=0}^{l-1} \tilde{R}_i^{(k)}[j], \quad \forall i \in \mathcal{M}.$$

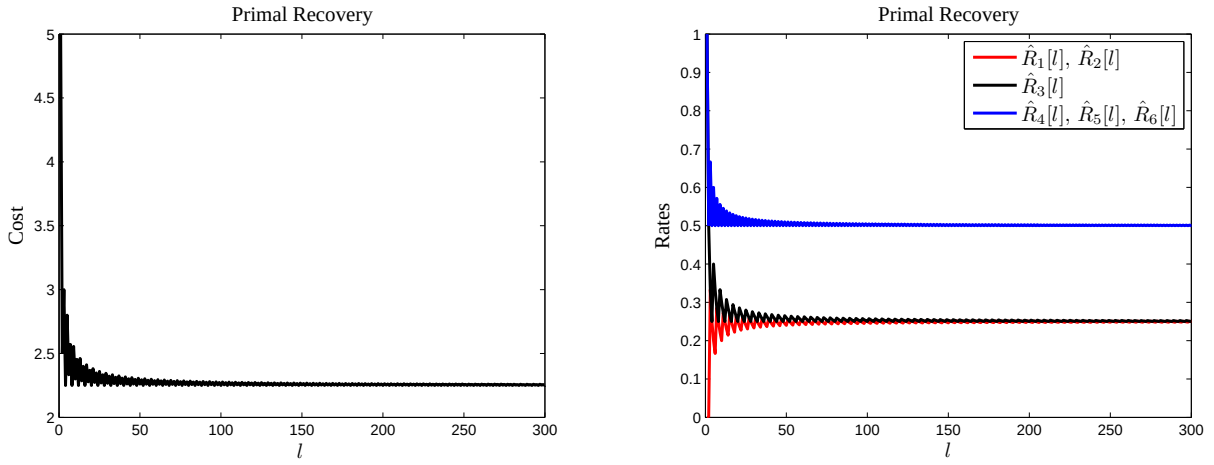
7: **end for**

cost is $R_1^* = R_2^* = R_3^* = \frac{1}{4}$, $R_4^* = R_5^* = R_6^* = \frac{1}{2}$. After 300 iterations of Algorithm 18 (see Figure 4.1), we obtain

$$\hat{R}_1[300] = \hat{R}_3[300] = 0.2525, \quad \hat{R}_2[300] = 0.2492, \quad \hat{R}_4[300] = \hat{R}_5[300] = \hat{R}_6[300] = 0.5017,$$

and obviously satisfied condition

$$\left| \sum_{i=1}^{6} \hat{R}_i[300] - \sum_{i=1}^{6} R_i^* \right| = 0.0091 \leq \varepsilon. \tag{4.19}$$



(a) Cost (b) Rates

Figure 4.1: Algorithm 18 applied to Example 13, for the precision parameter $\varepsilon = 0.01$. After 300 iterations of Algorithm 18, the solution obtained is $\hat{R}_1[300] = \hat{R}_3[300] = 0.2525$, $\hat{R}_2[300] = 0.2492$, $\hat{R}_4[300] = \hat{R}_5[300] = \hat{R}_6[300] = 0.5017$, and it satisfies condition (4.19).

4.2 Multi-source Multicast Problem

In this section we study data exchange problem with helpers under different communication model; instead of noiseless broadcast channel, we consider a model where nodes in the system are communicating among themselves through wires (see Figure 4.2). The communication network is represented by an acyclic graph, with possibly capacity constrained links. This is known as a multi-source multicast problem. There are two types of nodes in the network; *clients* that are interested in recovering the whole content, and *helpers* that may have access to some partial information about the file, and are willing to cooperate in order to distribute the file to the clients. To further illustrate the problem set-up consider the following example.

A file consists of four equally sized packets w_1, w_2, w_3, and w_4 belonging to some finite field $\mathbb{F}_q$. Also, suppose that the data packets are distributed across the helper nodes, that are connected as shown in Figure 4.2. The clients are interested in recovering the entire file. The edges of the graph are denoted by $e_1, \ldots, e_7$ as shown in Figure 4.2. The objective is to minimize a communication cost such that the clients can recover the entire file. For instance, it can be shown that the following coding scheme minimizes the total number of symbols in $\mathbb{F}_q$ communicated: helper 1 transmits w_1 on link e_2, helper 2 transmits w_2, w_3 on link e_3, helper 3 transmits w_3 on link e_5, helper 4 transmits w_1, w_2, w_4 on link e_6 and w_1, w_2, w_3, w_4 on link e_7.

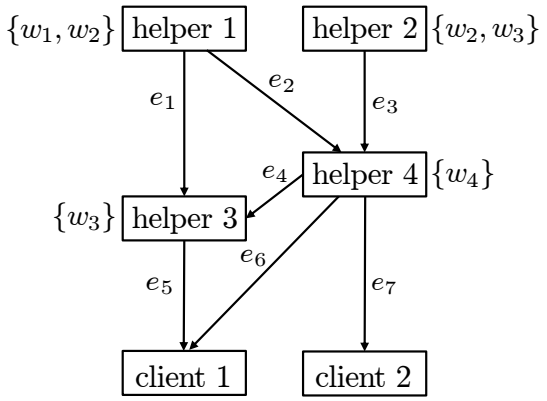


Figure 4.2: An example of the multi-source multicast problem, where helper nodes observe subsets of the file packets $\{w_1, w_2, w_3, w_4\}$ as shown above. Assuming that nodes can communicate reliably over the capacity constrained links, the goal is for the clients 1 and 2 (sinks of the graph) to gain access to the entire file while minimizing the communication cost.

4.2.1 System Model and Preliminaries

In this work we represent the network by a directed acyclic graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, and $\mathcal{E}$ is the set of links that have capacity constraints. We define the capacity function $c : \mathcal{E} \to \mathbb{R}$ to denote the maximum number of bits that can be transmitted over a given link. We distinguish between two types of nodes: 1) helpers $\mathcal{H} = \{1, 2, \ldots, h\}$ that have partial information about the file or the joint process, and 2) clients $\mathcal{T} = \{t_1, t_2, \ldots, t_s\}$ which are interested in recovering the file, and are sinks in the graph G. Let $X_1, X_2, \ldots, X_h$, denote the components of a discrete memoryless multiple source (DMMS) with a given joint probability mass function. Each helper $i \in \mathcal{H}$ observes n i.i.d. realizations of the corresponding random variable X_i, denoted by X_i^n. We note that the results of this section can be applied in a straightforward manner when the clients have side information as well. For the sake brevity, we focus on the case when clients have no side information.

The goal is for each client in $\mathcal{T}$ to gain access to all source nodes' observations. In order to achieve this goal, each helper $i \in \mathcal{H}$ is allowed to send information across the graph G at rate which is limited by the capacity of the outgoing links of that node. Transmission of each source node is a function of its own initial observation and all information it receives from its neighbors. Let us denote transmission on the link $e = (i, j) \in \mathcal{E}$ by

$$F_e = f_e\left(X_i^n, \{F_a : \forall r, \text{ s.t. } a = (r, i) \in \mathcal{E}\}\right), \tag{4.20}$$

where $f_e(\cdot)$ is a mapping of the observations X_i^n and transmissions received from the neighbors of i, $\{r : (r, i) \in \mathcal{E}\}$ to an outgoing message on the link e.

We denote by $\mathcal{M}_{t_i} \subseteq \mathcal{H}$ the set of source nodes which are connected to the client $t_i \in \mathcal{T}$. In other words, there exists a path in graph G from every node in $\mathcal{M}_{t_i}$ to the client $t_i \in \mathcal{T}$. Consequently, we define the graph $G_{t_i} = (\mathcal{V}_{t_i}, \mathcal{E}_{t_i})$ to be a subgraph of G, where $\mathcal{V}_{t_i} = \{\mathcal{M}_{t_i}, t_i\}$, and $\mathcal{E}_{t_i} \subseteq \mathcal{E}$ is a set of links that connects all nodes in $\mathcal{M}_{t_i}$ among themselves

and with client t_i. For the multi-source multicast problem over graph $G = (\mathcal{V}, \mathcal{E})$ shown in Figure 4.3, subgraphs $G_{t_1} = (\mathcal{V}_{t_1}, \mathcal{E}_{t_1})$ and $G_{t_2} = (\mathcal{V}_{t_2}, \mathcal{E}_{t_2})$ are shown in Figure 4.4.

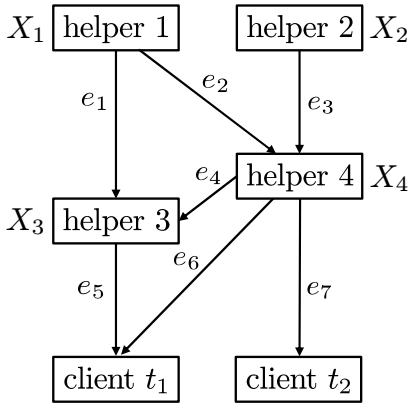
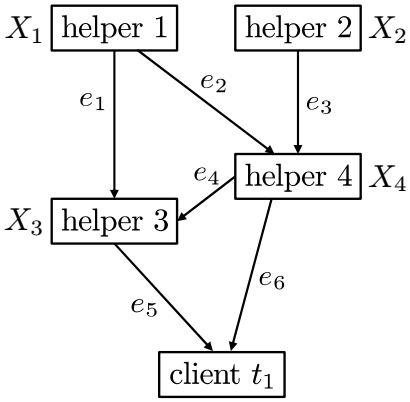
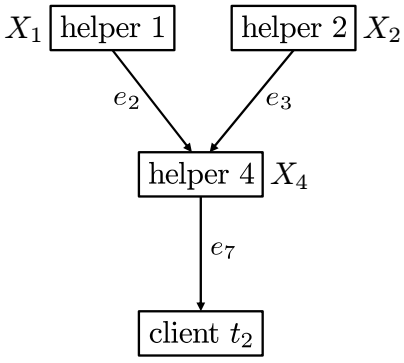


Figure 4.3: Multi-source multicast problem with two clients. The underlying communication model is represented by acyclic graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, 3, 4, t_1, t_2\}$, and $\mathcal{E} = \{e_1, e_2, \ldots, e_7\}$.



(a) Subgraph $G_{t_1} = (\mathcal{V}_{t_1}, \mathcal{E}_{t_1})$



(b) Subgraph $G_{t_2} = (\mathcal{V}_{t_2}, \mathcal{E}_{t_2})$

Figure 4.4: Subgraphs $G_{t_1} = (\mathcal{V}_{t_1}, \mathcal{E}_{t_1})$ and $G_{t_2} = (\mathcal{V}_{t_2}, \mathcal{E}_{t_2})$ derived from the graph $G = (\mathcal{V}, \mathcal{E})$ shown in Figure 4.3. Here, $\mathcal{V}_{t_1} = \{1, 2, 3, 4, t_1\}$, $\mathcal{E}_{t_1} = \{e_1, e_2, \ldots, e_6\}$, $\mathcal{V}_{t_2} = \{1, 2, 4, t_2\}$, $\mathcal{E}_{t_1} = \{e_2, e_3, e_7\}$.

Furthermore, we assume that

$$H\left(X_{\mathcal{M}_{t_1}}\right) = \cdots = H\left(X_{\mathcal{M}_{t_k}}\right) = H\left(X_{\mathcal{H}}\right), \tag{4.21}$$

where $X_{\mathcal{M}_{t_i}} \triangleq \left(X_{m_j} : m_j \in \mathcal{M}_{t_i}\right)$, and $X_{\mathcal{M}} \triangleq (X_{m_1}, \ldots, X_{m_l})$. Equality (4.21) ensures that every client in the network can potentially gain access to the entire process $X_{\mathcal{H}}$.

For each client $t_i \in \mathcal{T}$ to learn the joint process, transmissions Fe, $\forall e \in \mathcal{E}$, must satisfy,

$$\lim_{n\to\infty} \frac{1}{n} H\left(X_{\mathcal{H}}^n | \{F_e \ : \ e = (j, t_i) \in \mathcal{E}, \ \forall j \in \mathcal{H}\}\right) = 0, \quad \forall t_i \in \mathcal{T}. \tag{4.22}$$

Definition 12. A rate tuple $\mathbf{R} = (R_e : e \in \mathcal{E})$ is an *achievable multi-source multicast (MM) rate vector* if there exists a communication scheme with transmitted messages $\mathbf{F} = (F_e : e \in \mathcal{E})$ that satisfies (4.22), and

$$R_e = \lim_{n\to\infty} \frac{1}{n} H(F_e), \quad \forall e \in \mathcal{E}, \tag{4.23}$$

where $R_e \leq c_e$, $\forall e \in \mathcal{E}$.

In this work, we design a polynomial time algorithm for the multi-source multicast problem that minimizes the linear cost function $\sum_{e\in\mathcal{E}} \alpha_e R_e$, where $\alpha_e \geq 0$, $\forall e \in \mathcal{E}$. We allow α_e's to be arbitrary non-negative constants, to account for the case when communication across some group of links in G is more expensive compared to the others. Thus, the problem can be formulated as:

$$\min_{\mathbf{R}} \sum_{e\in\mathcal{E}} \alpha_e R_e, \text{ s.t. } \mathbf{R} \text{ is an achievable } MM\text{-rate vector.} \tag{4.24}$$

4.2.2 Multi-Source Multicast Rate-Flow Region

In order to solve the optimization problem in (4.24) we first establish a region called a "rate-flow region" that contains all possible optimal rate allocations. To identify this rate-flow region for our example of Figure 4.2, in the case of arbitrarily correlated side-information at the source nodes, we start by considering a single client t_1.

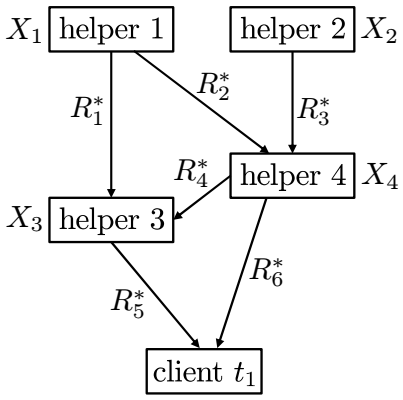


Figure 4.5: Single client multi-source multicast problem over graph $G_{t_1} = (\mathcal{V}_{t_1}, \mathcal{E}_{t_1})$ with link rate allocation $\mathbf{R}^*$.

Suppose the optimal solution w.r.t. problem (4.24) is achieved by $\mathbf{R}^* = (R_1^*, \ldots, R_6^*)$ (see Figure 4.5). Then, it follows that transmissions of helper node 2 have to satisfy

$$\begin{aligned} R_3^* &\geq H(X_2|X_1, X_3, X_4), \\ R_1^* + R_2^* + R_3^* &\geq H(X_1, X_2|X_3, X_4). \end{aligned} \tag{4.25}$$

Let us now consider helper node 4. Its transmission includes information received from helper nodes 1 and 2 combined with its own side information. Since the goal is to minimize the total communication cost, it follows that for the optimal *MM*-rate vector $\mathbf{R}^*$, transmission of helper nodes 1 and 2 cannot be further compressed at helper node 4. Therefore, the transmission of helper node 4 consists of 2 components: 1) routed information from helper nodes 1 and 2, and 2) innovative side-information at helper node 4 w.r.t. all other source nodes in the network. Hence, $\mathbf{R}^*$ must satisfy

$$R_4^* + R_6^* - R_2^* - R_3^* \geq H(X_4|X_1, X_2, X_3). \tag{4.26}$$

In order for client t_1 to learn the joint process, *i.e.*, to gain access to $X_{\mathcal{M}_{t_1}}$, the incoming links to t_1 necessarily have to carry entire information about the process. In other words

$$R_5^* + R_6^* = H(X_{\mathcal{M}_{t_1}}), \tag{4.27}$$

where the equality sign comes from the fact that the goal is to minimize the overall communication cost, and thus, it is wasteful for client t_1 to receive at rate larger than the joint entropy of the process.

Considering all possible subsets of the source node set $\mathcal{M}_{t_1}$, we have that an optimal *MM*-rate vector $\mathbf{R}^*$ must belong to the following rate-flow region

$$\begin{aligned}\partial\mathcal{R}_{t_1} = \{\partial\mathbf{R} : \partial R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{M}_{t_1}\setminus\mathcal{S}}),\ \forall\mathcal{S} \subset \mathcal{M}_{t_1},\\ \partial R(\mathcal{M}_{t_1}) = H(X_{\mathcal{M}_{t_1}})\},\end{aligned} \tag{4.28}$$

where

$$\partial R(\mathcal{S}) \triangleq \sum_{e\in\Delta^+\mathcal{S}} R_e - \sum_{e\in\Delta^-\mathcal{S}} R_e, \tag{4.29}$$

and $\Delta^+\mathcal{S} \subseteq \mathcal{E}_{t_1}$, $(\Delta^-\mathcal{S} \subseteq \mathcal{E}_{t_1})$ denotes the set of links leaving (entering) $\mathcal{S}$. For instance, if $\mathcal{S} = \{m_3, m_4\}$, then the optimal rate vector $\mathbf{R}^*$ satisfies

$$\begin{aligned}\partial R^*(S) &= R_5^* + R_6^* - R_1^* - R_2^* - R_3^*\\ &\geq H(X_3, X_4|X_1, X_2).\end{aligned} \tag{4.30}$$

It can be verified that any rate vector that belongs to the rate-flow region $\partial\mathcal{R}_{t_1}$ can be achieved using multi-terminal Slepian-Wolf random binning scheme [10]. Thus, the rate-flow region $\partial\mathcal{R}_{t_1}$ contains all optimal *MM*-rate vectors w.r.t. the optimization problem (4.24).

Extension of this result to a multiple client case is straightforward: an optimal *MM*-rate vector has to simultaneously belong to all rate-flow regions $\partial\mathcal{R}_{t_i}$ which correspond to the graph G_{t_i}, $\forall t_i \in \mathcal{T}$. The achievability of every rate vector that belongs to $\partial\mathcal{R}_{t_1} \cap \partial\mathcal{R}_{t_2} \cap$

$\cdots \cap \partial\mathcal{R}_{t_s}$ can be easily shown using random binning argument. Hence, the optimization problem (4.24) can be written as

$$\begin{aligned} \min_{\mathbf{R}} \sum_{e\in\mathcal{E}} \alpha_e R_e, \\ \text{s.t. } \partial\mathbf{R} \in \partial\mathcal{R}_{t_1} \cap \partial\mathcal{R}_{t_2} \cap \cdots \cap \partial\mathcal{R}_{t_s}, \\ R_e \leq c_e, \ \forall e \in \mathcal{E}. \end{aligned} \tag{4.31}$$

Before we address the question of efficiently solving the problem (4.31), first we need to answer whether or not the problem is feasible.

4.2.3 Feasibility of the Multi-Source Multicast Problem

As in Section 4.2.2, first, we consider a single client case, *i.e.*, when $\mathcal{T} = \{t_1\}$. Then, the obtained result naturally extends to the setting with arbitrary number of clients. Now, let us consider a set function $y_{t_1} : 2^{\mathcal{M}_{t_1}} \to \mathbb{R}$ given by

$$y_{t_1}(\mathcal{S}) = H(X_{\mathcal{S}}|X_{\mathcal{M}_{t_1}\setminus\mathcal{S}}), \ \forall \mathcal{S} \subseteq \mathcal{M}_{t_1}. \tag{4.32}$$

According to Definition 3, the dual set function $g_{t_1} : 2^{\mathcal{M}_{t_1}} \to \mathbb{R}$ is given by

$$g_{t_1}(\mathcal{S}) = H(X_{\mathcal{S}}), \quad \forall \mathcal{S} \subseteq \mathcal{M}_{t_1}. \tag{4.33}$$

It can be easily shown that g_{t_1} is a fully submodular function. Since $B(y_{t_1}, \geq) = B(g_{t_1}, \leq)$, the rate-flow region $\partial\mathcal{R}_{t_1}$ defined in (4.28) represents the base polyhedron of the function g_{t_1}.

Lemma 12. *For a single client multi-source multicast problem over $G_{t_1} = (\mathcal{V}_{t_1}, \mathcal{E}_{t_1})$, where $\mathcal{V}_{t_1} = \{\mathcal{M}_{t_1}, t_1\}$, there exists an achievable MM-rate vector,* i.e. *$\partial\mathcal{R}_{t_1} \neq \emptyset$, and $R_e \leq c_e$, $\forall e \in \mathcal{E}_{t_1}$, if and only if*

$$c(\Delta^+\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{M}_{t_1}\setminus\mathcal{S}}), \quad \forall \mathcal{S} \subseteq \mathcal{M}_{t_1}, \tag{4.34}$$

where

$$c(\Delta^+\mathcal{S}) = \sum_{e\in\Delta^+\mathcal{S}} c_e, \quad \Delta^+\mathcal{S} \in \mathcal{E}_{t_1}.$$

Proof. As we discussed in Section 4.2.2, the incoming links to t_1 carry entire information about the process. This combined with the fact that the goal is to minimize the communication cost, implies that for any optimal *MM*-rate vector $\mathbf{R}^*$ it holds that

$$\sum_{e=(j,t_1)\in\mathcal{E}_{t_1}} R_e^* = H(X_{\mathcal{M}_{t_1}}). \tag{4.35}$$

Therefore, without loss of generality we can assume that the capacities of the links incoming to t_1 satisfy

$$\sum_{e=(j,t_1)\in\mathcal{E}_{t_1}} c_e = H(X_{\mathcal{M}_{t_1}}), \tag{4.36}$$

provided that the feasible rate-flow region exists. It is not hard to show that the capacity function $c(\Delta^+\mathcal{S})$, $\forall\mathcal{S}\subseteq\mathcal{M}_{t_1}$ is submodular (see Chapter 2 in [14]). Let us denote by $\partial\Psi$, the set of the boundaries $\partial\mathbf{R}$ of a feasible rate-flow region:

$$\partial\Psi \triangleq \{\partial\mathbf{R} : R_e \le c_e,\ \forall e\in\mathcal{E}_{t_1}\} \tag{4.37}$$

In [19] it was shown that

$$\partial\Psi = B(c(\Delta^+)). \tag{4.38}$$

From (4.38) and (4.31) it follows that there exists a feasible *MM*-rate vector iff

$$B(c(\Delta^+))\cap B(g_{t_1}) \neq \emptyset. \tag{4.39}$$

Problem (4.39) is known as a *common base problem* [14] for which the solution exists if and only if

$$c(\Delta^+\mathcal{S}) \ge y_{t_1}(\mathcal{S}),\quad \forall\mathcal{S}\subseteq\mathcal{M}_{t_1}. \tag{4.40}$$

This completes the proof of Lemma 12. □

To verify whether there exists an achievable *MM*-rate vector it is necessary to check whether all $2^{|\mathcal{M}_{t_1}|}$ inequalities in (4.34) are satisfied. Verifying this is, in general, exponentially hard (in number of nodes). However, due to the supermodularity of the function g_{t_1}, the existence of a common base, and thus the feasibility of the multi-source multicast problem, can be verified in polynomial time[1] (see [23] and [14], Chapter 4). This algorithm also provides an achievable *MM*-rate vector (given that it exists) that belongs to the rate-flow region $\partial\mathcal{R}_{t_1}$.

Extensions of the result of Lemma 12 to the case with arbitrary number of clients is straightforward. We just need to check if the inequalities (4.34) are satisfied for all clients in $\mathcal{T}$.

Theorem 10. *For the multi-source multicast problem over $G(\mathcal{V},\mathcal{E})$, with the capacity function c, there exists an achievable* MM*-rate vector if and only if*

$$\begin{aligned} &c(\Delta^+\mathcal{S}) \ge H(X_{\mathcal{S}}|X_{\mathcal{M}_{t_i}\setminus\mathcal{S}}),\\ &\forall\mathcal{S}\subseteq\mathcal{M}_{t_i},\ \partial\Delta^+\mathcal{S}\in\mathcal{E}_{t_i},\ \forall t_i\in\mathcal{T}. \end{aligned} \tag{4.41}$$

From [23], *the common base problem*, and hence the feasibility of the multi-source multicast problem can be verified in $\mathcal{O}(s\cdot|\mathcal{E}|^3)$ time.

[1] Complexity of the *common base* algorithm proposed in [23] is $\mathcal{O}(|\mathcal{E}_{t_1}|^3)$

4.2.4 Deterministic Algorithm for the Single Client Case

When $\mathcal{T} = \{t_1\}$, then, the optimization problem (4.31) can be written as

$$\begin{aligned} &\min_{\mathbf{R}\in\mathbb{R}^{|\mathcal{E}_{t_1}|}} \sum_{e\in\mathcal{E}_{t_1}} R_e, \\ &\text{s.t. } \partial\mathbf{R} \in B(g_{t_1}, \leq), \quad R_e \leq c_e, \quad \forall e \in \mathcal{E}_{t_1}. \end{aligned} \tag{4.42}$$

Optimization problem (4.42) has a form of the *minimum cost submodular flow problem* (see [14] for formal definitions), but with a few differences listed bellow.

1. In the submodular flow problem, function g_{t_1} has to be defined over all vertices $\mathcal{V}_{t_1}$ of graph G_{t_1}. However, in our case g_{t_1} is a set function over the source vertices only.

2. In the submodular flow problem, $g_{t_1}(\mathcal{V}_{t_1})$ must evaluate to 0, whereas in our problem function g_{t_1} is not defined for $\mathcal{V}_{t_1}$.

The first step of solving the problem (4.42) efficiently involves verifying its feasibility. From the common base algorithm we obtain an achievable MM-rate vector that belongs to $B(g_{t_1}, \leq)$ provided that $B(g_{t_1}, \leq) \neq \emptyset$. Given any achievable MM-rate vector that belongs to $B(g_{t_1}, \leq)$, one can construct the auxiliary network over graph G_{t_1}(See Chapter III of [14] for detailed explanation). It can be verified that from this step onwards, we can apply min-cost submodular flow algorithm [14] which involves finding negative cycles of the auxiliary network, and updating the network accordingly along with the achievable MM-rate vector. Comparison between different minimum cost submodular flow algorithms is provided in [15].

4.2.5 Deterministic Algorithm for the Multiple Client Case

In this section we extend the results from the previous section to the case where the set $\mathcal{T}$ contains arbitrary number of clients. Motivated by the results from Section 3.3, the optimization problem (4.31) can be written as follows

$$\begin{aligned} &\min_{\mathbf{R},\mathbf{R}^{(1)},\ldots,\mathbf{R}^{(s)}} \sum_{e\in\mathcal{E}} \alpha_e R_e, \\ &\text{s.t. } R_e \geq R_e^{(t_i)}, \quad \forall t_i \in \mathcal{T}, \ \forall e \in \mathcal{E}_{t_i}, \\ &\quad \partial\mathbf{R}^{(t_i)} \in \partial\mathcal{R}_{t_i}, \ R_e^{(t_i)} \leq c_e, \ \forall e \in \mathcal{E}_{t_i}, \ \forall t_i \in \mathcal{T}, \end{aligned} \tag{4.43}$$

where $\partial\mathcal{R}_{t_i}$ is defined in (4.28) for $i = 1$. Equivalence between the optimization problems (4.31) and (4.43) follows from the fact that transmissions on graph G have to be such that all clients in $\mathcal{T}$ learn the file simultaneously. To obtain a polynomial time solution to

problem (4.43), like in Section 4.1, we consider the Lagrangian dual.

$$\max_{\mathbf{\Lambda}} \sum_{l=1}^{k} z_{t_i}(\mathbf{\Lambda}), \tag{4.44}$$
$$\text{s.t.} \sum_{i=1}^{k} \lambda_{e,t_i} = \alpha_e,\ \lambda_{e,t_i} \geq 0,\ \forall t_i \in \mathcal{T},\ \ \forall e \in \mathcal{E}_{t_i},$$

where

$$z_{t_i}(\mathbf{\Lambda}) = \min_{\mathbf{R}^{(t_i)} \in \mathbb{R}^{|\mathcal{E}_{t_i}|}} \sum_{e \in \mathcal{E}_{t_i}} \lambda_e^{(t_i)} R_e^{(t_i)}, \tag{4.45}$$
$$\text{s.t. } \partial \mathbf{R}^{(t_i)} \in \partial \mathcal{R}_{t_i},\ \ R_e^{(t_i)} \leq c_e,\ \ \forall e \in \mathcal{E}_{t_i}.$$

For any given $t_i \in \mathcal{T}$, the objective function (4.45) of the dual problem (4.44) can be computed in polynomial time as pointed out in Section 4.2.4. Hence, we can apply the dual subgradient method to solve the problem (4.44) in polynomial time. Then, a near optimal solution of the primal problem (4.31) can be recovered by applying averaging method as already discussed in Section 3.3.2.

Starting with a feasible iterate $\lambda_{e,t_i}[0]$, $\forall t_i \in \mathcal{T}$, $\forall e \in \mathcal{E}_{t_i}$, w.r.t. the optimization problem (4.44), and the step size θ, every subsequent iterate $\lambda_{e,t_i}[j+1]$ can be recursively represented as an Euclidian projection of the vector

$$\left\{ \lambda_{e,t_i}[j] + \theta \tilde{R}_e^{(t_i)}[j] \ :\ \forall t_i \in \mathcal{T},\ \forall e \in \mathcal{E}_{t_i} \right\} \tag{4.46}$$

onto the hyperplane

$$P = \left\{ \mathbf{\Lambda} \mid \sum_{t_i : e \in \mathcal{E}_{t_i}}^{t} \lambda_{e,t_i} = \alpha_e, \quad \lambda_{e,t_i} \geq 0,\ \forall t_i \in \mathcal{T},\ \forall e \in \mathcal{E}_{t_i} \right\}, \tag{4.47}$$

where $\tilde{\mathbf{R}}^{(t_i)}[j]$ is an optimal rate vector w.r.t the problem (4.45). In order to obtain a near optimal solution to problem (4.31) we take an average over all iterations of the dual subgradient method

$$\hat{R}_e[l] = \max_{t_i \in \mathcal{T}} \left\{ \frac{1}{l} \sum_{j=0}^{l-1} \tilde{R}_e^{(t_i)}[j] \ :\ e \in \mathcal{E}_{t_i} \right\}, \quad \forall e \in \mathcal{E}. \tag{4.48}$$

The maximization in (4.48) ensures that vector $\hat{\mathbf{R}}[l]$ belongs to all rate-flow regions $\partial \mathcal{R}_{t_i}$, $\forall t_i \in \mathcal{T}$. Convergence analysis of this method can be done in the same way as in Section 3.3.3.

Chapter 5

Conclusion

In this dissertation we investigated the problem of data exchange where multiple users, interested in gaining access to the common file, have only partial information about it stored. In Chapter 2, the information stored at each user was in the for of linearly coded data packets, while in the later chapters, we considered the more general discrete memoryless multiple source (DMMS) model. The main problem studied in this work was to construct a communication scheme that delivers the file to all users, while minimizing some communication cost. In Chapters 2 and 3, the communication model we considered was a noiseless broadcast channel, while in Chapter 4 we investigated a wireline model.

In Chapter 2, we proposed a deterministic polynomial time algorithm for finding an optimal communication scheme w.r.t. the separable convex communication cost. We provided two methods to determine how much should each user transmit in an optimal scheme. The first one is based on minimizing a submodular function, in which case the total complexity of the algorithm is $\mathcal{O}((m^6 \cdot N^3 + m^7) \cdot \log N)$, where m is the total number of users, and N is the number of packets in the file. The second technique is based on subgradient methods, in which case the total complexity of the algorithm can be bounded by $\mathcal{O}((N^2 \cdot m^4 \log m + N^5 \cdot m^4) \cdot \log N)$ given that we use constant step size in the subgradient algorithm. The second method, also provides an alternative solution to the Edmonds' algorithm when the underlying set function is intersecting submodular and over integers. We also devised a randomized algorithm inspired by the deterministic scheme that reduces complexity to $\mathcal{O}(m \cdot N^4 \log N)$.

In Chapter 3, we studied the data exchange problem under the DMMS model, and the linear communication cost. We proposed a combinatorial algorithm of polynomial complexity that finds an optimal rate allocation w.r.t. the communication cost. The complexity of the algorithm is $\mathcal{O}(m^7 \cdot \gamma + m^8)$, where γ is the complexity of computing entropy function. We also proposed a non-combinatorial algorithm of polynomial complexity that computes an approximately optimal rate allocation. The complexity of this algorithm is $\mathcal{O}((m^4 \log m + m^4\gamma) \cdot \lceil \frac{1}{\varepsilon^2} \rceil)$, where ε is the upper bound on the distance from the optimal solution.

In Chapter 4, we studied the data exchange problem with helpers under noiseless broadcast and wireline communication models. For both of these problems we proposed a non-combinatorial algorithm of polynomial complexity that is based on the techniques we developed in Chapter 3.

Bibliography

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network Information Flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[2] D.P. Bertsekas. *Network optimization: Continuous and discrete methods*. Athena Scientific (Belmont, Mass.), 1998.

[3] S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004.

[4] Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004, 2003.

[5] J. Byers, J. Considine, M. Mitzenmacher, and S. Rost. Informed content delivery across adaptive overlay networks. *ACM SIGCOMM Computer Communication Review*, 32(4): 47–60, 2002.

[6] J.W. Byers, M. Luby, and M. Mitzenmacher. Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads. In *Proceedings of INFOCOM*, pages 275–283, 1999.

[7] C. Chan. *Generating Secret in a Network*. PhD thesis, Massachusetts Institute of Technology, 2010.

[8] T.A. Courtade and R.D. Wesel. Weighted universal recovery, practical secrecy, and an efficient algorithm for solving both. In *Proceedings of Allerton Conference on Communication, Control, and Computing*, pages 1349 –1357, Sept. 2011. doi: 10.1109/Allerton.2011.6120324.

[9] Thomas A Courtade, Bike Xie, and Richard D Wesel. Optimal exchange of packets for universal recovery in broadcast networks. In *Military Communications Conference (MILCOM)*, pages 2250–2255. IEEE, 2010.

[10] T.M. Cover and J.A. Thomas. Elements of information theory 2nd edition. 2006.

[11] I. Csiszár and P. Narayan. Secrecy capacities for multiple terminals. *IEEE Transactions on Information Theory*, 50(12):3047–3061, 2004.

[12] J. Edmonds. Submodular functions, matroids, and certain polyhedra. *Combinatorial structures and their applications*, pages 69–87, 1970.

[13] Salim El Rouayheb, Alex Sprintson, and Parastoo Sadeghi. On coding for cooperative data exchange. In *Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2010.

[14] S. Fujishige. *Submodular functions and optimization*. Elsevier Science, 2005. ISBN 0444520864.

[15] S. Fujishige and S. Iwata. Algorithms for submodular flows. *IEICE Transactions on Information and Systems*, 83:322–329, 2000.

[16] M. Gonen and M. Langberg. Coded cooperative data exchange problem for general topologies. *Arxiv preprint arXiv:1202.2088*, 2012.

[17] N.J.A. Harvey, D.R. Karger, and K. Murota. Deterministic network coding by matrix completion. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 489–498, 2005.

[18] T. Ho, M. Médard, R. Koetter, D.R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory*, 52(10):4413–4430, 2006.

[19] A.J. Hoffman. Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. *New York, NY*, pages 113–117, 1958.

[20] Toshihide Ibaraki and Naoki Katoh. *Resource allocation problems: algorithmic approaches*. MIT press, 1988.

[21] S. Jaggi, P. Sanders, P.A. Chou, M. Effros, S. Egner, K. Jain, and L.M.G.M. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6):1973–1982, 2005. ISSN 0018-9448.

[22] R. Koetter and M. Medard. An Algebraic Approach to Network Coding. *IEEE/ACM Transactions on Networking*, 11(5):782 – 795, 2003.

[23] EL Lawler and CU Martel. Computing maximal polymatroidal network flows. *Mathematics of Operations Research*, 7(3):334–347, 1982.

[24] Z. Liu, C. Wu, B. Li, and S. Zhao. Uusee: Large-scale operational on-demand streaming with random network coding. In *Proceedings of INFOCOM*, pages 1–9, 2010.

[25] M. Luby. LT codes. In *Proceedings of Foundations of Computer Science*, pages 271–280. IEEE, 2002.

[26] 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 *7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 1–8, 2009.

[27] D.S. Lun, N. Ratnakar, M. Médard, R. Koetter, D.R. Karger, T. Ho, E. Ahmed, and F. Zhao. Minimum-cost multicast over coded packet networks. *Information Theory, IEEE Transactions on*, 52(6):2608–2623, 2006.

[28] N. Milosavljevic, S. Pawar, S.E. Rouayheb, M. Gastpar, and K. Ramchandran. Optimal deterministic polynomial-time data exchange for omniscience. *Arxiv preprint arXiv:1108.6046*, 2011.

[29] Nebojsa Milosavljevic, Sameer Pawar, Michael Gastpar, and Kannan Ramchandran. Efficient algorithms for the data exchange problem under fairness constraints. In *50th Annual Allerton Conference on Communication, Control, and Computing*, pages 502–508. IEEE, 2012.

[30] Nebojsa Milosavljevic, Sameer Pawar, Salim El Rouayheb, Michael Gastpar, and Kannan Ramchandran. Data exchange problem with helpers. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 2611–2615. IEEE, 2012.

[31] Kiyohito Nagano, Yoshinobu Kawahara, and Satoru Iwata. Minimum average cost clustering. In *Advances in Neural Information Processing Systems*, pages 1759–1767, 2010.

[32] Angelia Nedic and Asuman Ozdaglar. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4):1757–1780, 2009.

[33] J.B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

[34] D. Ozgul and A. Sprintson. An algorithm for cooperative data exchange with cost criterion. In *Proceedings of ITA*, pages 1–4, 2011.

[35] S.S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (discus): Design and construction. *Information Theory, IEEE Transactions on*, 49(3):626–643, 2003.

[36] S.S. Pradhan and K. Ramchandran. Generalized coset codes for distributed binning. *Information Theory, IEEE Transactions on*, 51(10):3457–3474, 2005.

[37] A. Ramamoorthy. Minimum cost distributed source coding over a network. *Information Theory, IEEE Transactions on*, 57(1):461–475, 2011.

[38] H.D. Sherali and G. Choi. Recovery of primal solutions when using subgradient optimization methods to solve lagrangian duals of linear programs. *Operations Research Letters*, 19(3):105–113, 1996.

[39] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb. A randomized algorithm and performance bounds for coded cooperative data exchange. In *Proceedings of ISIT*, pages 1888–1892, 2010.

[40] V. Stankovic, A.D. Liveris, Z. Xiong, and C.N. Georghiades. On code design for the slepian-wolf problem and lossless multiterminal networks. *Information Theory, IEEE Transactions on*, 52(4):1495–1507, 2006.

[41] S.E. Tajbakhsh, P. Sadeghi, and R. Shams. A generalized model for cost and fairness analysis in coded cooperative data exchange. In *Proceedings of NetCod*, pages 1–6, 2011.