A Rate-Distortion Approach to Caching

Roy Timo†, Shirin Saeedi Bidokhti*, Michèle Wigger† and Bernhard C. Geiger‡
†Institute for Communications Engineering, Technische Universität München
‡Department of Electrical Engineering, Stanford University
‡Communications and Electronics Department, Telecom ParisTech
{roy.timo, bernhard.geiger}@tum.de, saeedi@stanford.edu, michele.wigger@telecom-paristech.fr

Abstract—This paper takes a rate-distortion approach to the caching problem of Maddah-Ali and Niesen. We characterise the optimal tradeoffs between compression rate, reconstruction distortion and cache capacity for a single-user problem and special cases of a two-user problem. These tradeoffs illustrate some interesting connections between optimal caching strategies, Gács-Körner common information, and Wyner’s common information.

I. INTRODUCTION AND SETUP

We address a communication scenario where users request files from a server during peak-traffic periods. The server reduces the peak-traffic by pre-placing information in cache memories close to the users during prior periods of low traffic. In these low-traffic periods, communication rate is not a limiting resource and the amount of pre-placed information is mainly restricted by the cache memory sizes.

More specifically, in this paper we consider the scenario in Figure 1. The server has access to a library with $L$ files:

$$\text{Library} \quad X := (X^1_n, X^2_n, \ldots, X^L_n),$$

where each file is a sequence of $n$ symbols

$$X^\ell_n := (X_{\ell,1}, X_{\ell,2}, \ldots, X_{\ell,n})$$

taking value in a finite alphabet $\mathcal{X}_\ell$. For simplicity, we assume that each file is a sequence of independent and identically distributed (i.i.d.) symbols, where symbols pertaining to different files can be correlated:

$$(X_{1,1}, \ldots, X_{L,1}), \ldots, (X_{1,n}, \ldots, X_{L,n}) \sim \text{i.i.d. } P_X, \quad (1)$$

for some given joint law $P_X$ over $\mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_L$.

Assume that there is a single user, which selects an index

$$\ell \in \mathcal{L} := \{1, 2, \ldots, L\}$$

arbitrarily and requests the corresponding file $X^\ell_n$ from the server. The user has a local cache memory of size $nC$ bits where the server can pre-place information $M_c$, and which the user can access to reconstruct its requested file $X^\ell_n$. A central assumption in our work is that the server has to place the information in the cache before it learns the user’s request. The information $M_c$ stored in the cache should thus be chosen such that it is useful for (or common to) as many files as possible.

Once the server learns the user’s request $\ell \in \mathcal{L}$, it sends an $nR$-bit delivery message $M$ to the user. Based on this message $M$ and the cache content $M_c$, the user attempts to reconstruct its requested file $X^\ell_n$. Hence, the delivery message $M$ should contain all the information about $X^\ell_n$ that is relevant to the user and that is not yet stored in the cache memory.

Such a cache-aided setup was first considered by Maddah-Ali and Niesen in [1] and triggered a series of fruitful results [3]–[9]. The works in [1]–[7] studied the problem where independent files $X^1_n, \ldots, X^L_n$ had to be reconstructed losslessly by multiple users. More specifically, these works presented various upper and lower bounds on the minimum required delivery-rate $R$ for given cache capacity $C$. While we limit ourselves to a single user with cache memory, we extend the analysis to lossy reconstruction of potentially correlated files, cf. [1]. We furthermore analyse the problem when a second user without cache memory is present, see the setup in Figure 4.

The main problem of interest in this paper is thus the optimal tradeoff between the delivery rate $R$, the cache capacity $C$, and the user’s reconstruction distortion. Notice that the delivery rate $R$ is a worst-case rate (or a compound rate) in the sense that it has to be sufficiently large so that the user can reconstruct every file $X^\ell_n$, $\ell \in \mathcal{L}$, with desired accuracy. The problem setup by Wang, Lim and Gastpar [9], can be considered as an ergodic average-case equivalent of our worst-case (or compound) setup.

II. SINGLE USER

A. Formal Problem Definition

Let $X^1_n, \ldots, X^L_n$ be given reconstruction sets. A joint rate-distortion-cache (RDC) code for a given blocklength $n$ consists of $(2L + 1)$-mappings:

(i) A cache encoder $f_c : \mathcal{X}^n \to M_c$, where $M_c$ is finite.
(ii) A file encoder $f_\ell : \mathcal{X}^n \to M$ for each $\ell \in \mathcal{L}$, where $M$ is finite.
(iii) A file decoder $g_\ell : \mathcal{X} \times M_c \to \hat{X}^\ell_n$ for each $\ell \in \mathcal{L}$.

For brevity, we will call the above collection of encoders and decoders an $(n, M, M_c)$-code. Given demand $\ell \in \mathcal{L}$, the cache content and the delivery message are

$$M_c := f_c(X^n) \quad \text{and} \quad M := f_\ell(X^n);$$

Fig. 1. Single-user RD cache problem.
and the user’s reconstruction is
\[ \hat{X}_n^\ell := g_\ell(M, M_c) \in \hat{X}_n^\ell. \]

As per the usual rate-distortion (RD) paradigm, let us assume that the quality of \( \hat{X}_n^\ell \) can be meaningfully measured using average per-letter distortions. Specifically, for each \( \ell \in \mathcal{L} \), let
\[ \delta_\ell : \hat{X}_n^\ell \times X_n^\ell \to [0, \infty) \]
be a bounded distortion function. For simplicity, we assume that for each symbol \( x_\ell \in X_n^\ell \) there always exists an \( \hat{x}_\ell \in \hat{X}_n^\ell \) such that \( \delta_\ell(\hat{x}_\ell, x_\ell) = 0 \).

**Definition 1:** Let \( D := (D_1, D_2, \ldots, D_L) \) and \( C \) be arbitrary nonnegative reals. We say that a delivery rate \( R \geq 0 \) is \((D, C)\)-admissible if for every \( \epsilon > 0 \) there exists a sufficiently large blocklength \( n \) and an \((n, M, M_c)\)-code such that
\[
\forall \ell \in \mathcal{L}: \quad \frac{1}{n} \sum_{i=1}^n \delta_\ell(\hat{x}_\ell,i, x_\ell,i) \leq D_\ell + \epsilon,
\]
and
\[
|M_c| \leq 2^{n(C+\epsilon)} \quad \text{and} \quad |M| \leq 2^{n(R+\epsilon)}.
\]

We call \( C \) the cache capacity and \( D \) the distortion constraints. The optimal RDC tradeoff for blocklengths \( n \to \infty \) is characterised by the following function.

**Definition 2:** The RDC function is
\[
R(D, C) := \inf \{ R \geq 0 : R \text{ is } (D, C)\text{-admissible} \}.
\]

**B. Main Results**

The RDC function has the following properties:

**Proposition 1:**
1. \( R(D, C) \) is jointly convex and non-increasing in \( D \) and \( C \).
2. If \( C \geq H(X) \), then \( R(D, C) = 0 \) for all \( D \).
3. If \( C = 0 \), then
\[
R(D, 0) = \max_{\ell \in \mathcal{L}} R_{X_\ell}(D_\ell),
\]
where \( R_{X_\ell}(D_\ell) \) is the usual RD function for \( X_\ell \).

**Proposition 2:**
\[
R((D_1, D_2), C) = \min_{(C_1, C_2) \in \mathcal{C}(D_1, D_2)} \max \{R_1, R_2\}.
\]

**Corollary 1:**
\[
R_0(C) = R^*(0, C) = \min_{U} \max_{\ell} H(X_\ell|U),
\]
where the minimum is taken over all auxiliary random variables \( U \), jointly distributed with \( X \), satisfying \( H(X_\ell|U) \leq C \) for all \( \ell \in \mathcal{L} \).

**C. Connections to the Gray-Wyner Network**

For the case of \( L = 2 \) files, \( X^n = (X_1^n, X_2^n) \), there is a close connection between the RDC function and Gray and Wyner’s classic “source coding for a simple network” problem [10]. The Gray-Wyner network is illustrated in Figure 2. A transmitter is connected to two different receivers via a common link of rate \( R_c \) and two private links of rates \( R_1 \) and \( R_2 \). The set of all achievable rate tuples \((R_c, R_1, R_2)\) for which receivers 1 and 2 can respectively reconstruct \( X_1^n \) and \( X_2^n \) to within distortions \( D_1 \) and \( D_2 \) is given by [10] Thm. 8
\[
\mathcal{R}_{GW}(D_1, D_2) := \bigcup \left\{ (R_c, R_1, R_2) : \begin{array}{l}
R_c \geq I(X_1; X_2; U) \\
R_1 \geq I(X_1; X_2; U) \\
R_2 \geq I(X_2; X_1; U) 
\end{array} \right\},
\]
where the union is over all tuples \((X_1, X_2, U, \hat{X}_1, \hat{X}_2)\) satisfying \( \mathbb{E}[\delta_\ell(\hat{x}_\ell, x_\ell)] \leq D_\ell \) for \( \ell \in \{1, 2\} \). The next proposition can be proved by associating the common rate \( R_c \) of the Gray-Wyner problem with the rate of the caching message \( M_c \), and the two private rates \( R_1 \) and \( R_2 \) of the Gray-Wyner problem with the rates of our delivery message \( M \) when the user demands \( X_1^n \) and \( X_2^n \), respectively.

**Proposition 2:**
\[
R((D_1, D_2), C) = \min_{(C_1, C_2) \in \mathcal{R}_{GW}(D_1, D_2)} \max \{R_1, R_2\}.
\]

**D. Almost Lossless Compression**

Let us now restrict attention to the case where the user wants to reconstruct \( X_1^n \) (almost) losslessly. Specifically, suppose that \( \hat{X}_\ell = X_\ell \) and \( \delta_\ell(\hat{x}_\ell, x_\ell) = 1\{\hat{x}_\ell \neq x_\ell\} \) for all \( \ell \in \mathcal{L} \) are Hamming distortion functions; and \( 0 := (0, \ldots, 0) \) is a tuple of \( L \) zeros. Given these assumptions, define the rate-cache (RC) function
\[
R_0(C) := R(0, C).
\]
From Theorem 1 we have the next corollary.

**Corollary 1.1:**
\[
R_0(C) = R^*(0, C) = \min_{U} \max_{\ell} H(X_\ell|U),
\]
where the minimum is taken over all auxiliary random variables \( U \), jointly distributed with \( X \), satisfying \( H(X_\ell|U) \leq C \).

Figure 3 shows the typical behaviour of \( R_0(C) \). To obtain better understanding, we propose two lower bounds and study conditions when they are tight.
Further, let
\[ C_{\text{Genie}} := \max_{\ell \in \mathcal{L}} I(X; U), \]
where the maximum is taken over all auxiliary random variables \( U \) jointly distributed with \( X \) for which the following statements hold:

1) For every \( \ell^* \in \mathcal{L}^* \), we have \( U \leftrightarrow X_{\ell^*} \leftrightarrow X_{\mathcal{L}\setminus\ell^*} \), where \( X_{\mathcal{L}\setminus\ell^*} := (X_1, X_2, \ldots, X_{\ell^*-1}, X_{\ell^*+1}, \ldots, X_L) \).

2) For every \( \ell^* \in \mathcal{L}^* \),
\[ H(X_{\ell^*}; U) = \max_{\ell \in \mathcal{L}} H(X_{\ell} | U), \]

3) \( U \) is defined on an alphabet \( \mathcal{U} \) with \( |\mathcal{U}| = |\mathcal{X}| + |\mathcal{L}^*| + L. \)

Proposition 4:
\[ C_{\text{Genie}} = C_{\text{Genie}}^*. \]

The critical cache capacity \( C_{\text{Genie}}^* \) is related to the natural \( L \)-variable generalisation \(^{12}\) of Gács and Körner’s common information:
\[ K_{\text{GK}}^* := \max_{U: H(U) = 0, \ell \in \mathcal{L}} H(U). \]

Proposition 5:
\[ C_{\text{Genie}}^* \geq K_{\text{GK}}^*. \quad (4) \]

If \( H(X_1) = \cdots = H(X_L) \), then (4) holds with equality.

3) Lower Bound \( R_{0, \text{Super}}^*(C) \) on \( R_0(C) \): Now imagine a situation where we have a superuser that requests all the \( L \) sources \( X_1^n, \ldots, X_L^n \) and that obtains \( L \) delivery messages of rate \( R \) each. Moreover, suppose that as before this superuser has a local cache memory of size \( nC \) bits that can be filled by the server. The optimal strategy for this superuser problem is again obvious, since it is equivalent to a standard RD problem with a single compression message of rate \( LR + C \). The server takes an optimal code to compress the entire library \( X^n \) and distributes the produced bits in the cache memory and over the \( L \) delivery messages. The RC function of this superuser system, \( R_{0, \text{Super}}(C) \), hence is:
\[ R_{0, \text{Super}}(C) = R_{0, \text{Super}}^*(C) := \max \left\{ 0, \frac{1}{L} \left( H(X) - C \right) \right\}. \]

If one limits the superuser to reconstruct each source \( X_p^n \), \( \ell \in \mathcal{L} \), solely based on the content in the cache memory and the \( \ell \)-th delivery message, one obtains our original setup. The RC function of the superuser system thus can not exceed the RC function of the original setup:

Proposition 6:
\[ R_0(C) \geq R_{0, \text{Super}}(C). \]

For independent and identically distributed files, above lower bound is tight:

Example 2: Let the DMS \( X \) follow the product distribution \( P_X = \prod_{\ell=1}^L P_X. \) In this case,
\[ R_0(C) = R_{0, \text{Super}}^*(C) = \max \left\{ 0, \frac{1}{L} \left( H(X) - C \right) \right\}. \]

\(^1\)The maximum over \( \ell \in \mathcal{L} \) is needed because we again consider a worst-case (compound) setup over all possible demands \( \ell \in \mathcal{L} \).
4) Connection to Wyner’s Common Information: The superuser lower bound is trivially tight when \( C \geq H(X) \). So it is natural to consider the smallest cache capacity for which there is no rate loss with respect to the optimal superuser system,

\[
C_{\text{Super}} := \inf \left\{ C \geq 0 : R_0(C) = R_{0,\text{Super}}(C) \right\}.
\]

Let

\[
C^*_{\text{Super}} := \min_U I(X; U),
\]

where the minimum is taken over all auxiliary random variables \( U \) jointly distributed with \( X \) such that

(i) \( X_\ell \leftrightarrow U \leftrightarrow X_{\ell\backslash \ell} \) for all \( \ell \in L \);
(ii) \( H(X_1|U) = \cdots = H(X_L|U) \); and
(iii) \( U \) is defined on \( \mathcal{U} \) with \( |\mathcal{U}| \leq |\mathcal{X}| + 2L \).

Proposition 7:

\[
C_{\text{Super}} = C^*_{\text{Super}}.
\]

The critical cache capacity \( C^*_{\text{Super}} \) is related to the natural L-variable generalisation of Wyner’s common information:

\[
K^*_W(X) := \inf_U I(X; U),
\]

where the minimum is taken over all \( U \) jointly distributed with \( X \) for which

(i) \( X_\ell \leftrightarrow U \leftrightarrow X_{\ell\backslash \ell} \) for all \( \ell \in L \); and
(ii) \( U \) is defined on an alphabet \( \mathcal{U} \) with \( |\mathcal{U}| \leq |\mathcal{X}| + L \).

Proposition 8:

\[
C^*_{\text{Super}} \geq K^*_W.
\]

If the source \( X \) is sufficiently symmetric, above inequality holds with equality.

III. TWO-USERS WITH ONE CACHE

A. Setup

We now consider a two-user extension of the problem in Section II. Let us assume that user 1 has a cache with capacity \( C \), while user 2 does not have a cache; see Figure 4. The library consists of the same \( L \) files \( X^n := (X_1^n, \ldots, X_L^n) \) used in Section II and communication again takes place in two phases — a caching phase and a delivery phase. Let \( L_1, L_2 \subseteq L \) denote those indices that can be potentially selected by users 1 and 2, respectively. That is, user \( k \) (for \( k = 1, 2 \)) will request a file from \( \{ X_\ell^n : \ell \in L_k \} \). Let \( L_1 := |L_1| \) and \( L_2 := |L_2| \).

A two-user joint RDC code with blocklength \( n \) consists of

(i) A cache encoder

\[
f_k : \mathcal{X}^n \rightarrow \mathcal{M}_k.
\]

(ii) A file encoder

\[
f_{(\ell_1, \ell_2)} : \mathcal{X}^n \rightarrow \mathcal{M}, \quad (\ell_1, \ell_2) \in L_1 \times L_2.
\]

(iii) A user-1 file decoder

\[
g^{(1)}_{\ell_1, \ell_2} : \mathcal{M} \times \mathcal{M}_k \rightarrow \hat{X}^{(1),n}_{\ell_1}, \quad (\ell_1, \ell_2) \in L_1 \times L_2.
\]

(iv) A user-2 file decoder

\[
g^{(2)}_{\ell_1, \ell_2} : \mathcal{M} \rightarrow \hat{X}^{(2),n}_{\ell_2}, \quad (\ell_1, \ell_2) \in L_1 \times L_2.
\]

Notice that we allow the decoders to depend on the demands of both users. We call the above collection of encoders and decoders an \((n, \mathcal{M}, \mathcal{M}_k)\)-two-user-code.

During the caching phase, the server pre-places the message \( \mathcal{M}_k := f_k(X^n) \) in the cache of user 1. After the demands \( (\ell_1, \ell_2) \in L_1 \times L_2 \) are revealed to the server and both users, the server sends the message \( \mathcal{M} := f_{(\ell_1, \ell_2)}(X^n) \) to both users. Users 1 and 2 respectively output

\[
\hat{X}^{(1),n}_{\ell_1} := g_{(1)}^{(1)}(\mathcal{M}, \mathcal{M}_k), \quad \hat{X}^{(2),n}_{\ell_2} := g_{(2)}^{(2)}(\mathcal{M}).
\]

For convenience, we index user 1’s reconstruction sequence only with its own demand \( \ell_1 \); it can however also depend on user 2’s demand \( \ell_2 \). Similarly, for user 2’s reconstruction.

The users might have differing exigencies regarding the files in the library. To account for this, we admit both users to measure reconstruction accuracy with different bounded per-letter distortion functions \( \delta_{(1)}^{(1)} : \hat{X}^{(1),n}_{\ell_1} \times X_{\ell_1} \rightarrow [0, \infty) \) and \( \delta_{(2)}^{(2)} : \hat{X}^{(2),n}_{\ell_2} \times X_{\ell_2} \rightarrow [0, \infty) \) (for indices \( \ell_1 \in L_1 \) and \( \ell_2 \in L_2 \).

Definition 3: Let \( C \) be a nonnegative real number, and let \( \mathbf{D}^{(1)} := \{ D^{(1)}_{\ell_1} \}_{\ell_1 \in L_1} \) and \( \mathbf{D}^{(2)} := \{ D^{(2)}_{\ell_2} \}_{\ell_2 \in L_2} \) be \( L_1 \)- and \( L_2 \)-tuples of nonnegative real numbers.

We say that a compression rate \( R \geq 0 \) is \((\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, C)\)-admissible if for any \( \epsilon > 0 \) there exists a sufficiently large blocklength \( n \) and an \((n, \mathcal{M}, \mathcal{M}_k)\)-code satisfying

\[
\forall \ k \in \{1, 2\} : \forall \ \ell \in L_k : \quad E \left[ \frac{1}{n} \sum_{i=1}^{n} \delta_{(k)}^{(i)} \left( \hat{X}^{(k),n}_{\ell_i}, X_{\ell_i} \right) \right] \leq D_{\ell_k}^{(k)} + \epsilon.
\]

(5)

Definition 4: The two-user RDC function is

\[
R_{\text{user}}(\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, C) := \inf \{ R \geq 0 : R \text{ is } (\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, C)\text{-admissible} \}.
\]

B. Genie-Aided Lower Bound on the RDC Function

If both users’ demands were revealed by a genie to the server even before the caching phase, our setup would coincide with a “worst-case” (or compound) successive-refinement setup. The rate-distortions function of this worst-demands successive refinement problem thus forms a lower bound on \( R_{\text{user}}(\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, C) \).

Definition 5: Let \( R_{\text{succRef}}^{(1)}(\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, C) \) be the RDC function defined in (6) on top of the next page, where the minimum is taken over all tuples \( (\mathbf{X}, \hat{\mathbf{X}}^{(1)}, \hat{\mathbf{X}}^{(2)}) \) such that for \( k \in \{1, 2\} \):

\[
\forall \ \ell \in L_k : \quad E \left[ \delta_{(k)}^{(i)}(\hat{X}^{(k)}_{\ell_i}, X_{\ell_i}) \right] \leq D_{\ell_k}^{(k)}.
\]

(7)
\[
R_{\text{succRef}}(D^{(1)}, D^{(2)}, C) := \max_{(\ell, \ell_2) \in \mathcal{L}_1 \times \mathcal{L}_2} \min_{P_{X^{(1)}, X^{(2)}}} \max \left\{ I(X; \hat{X}^{(2)}_{\ell_2}), I(X; \hat{X}^{(1)}_{\ell_1}, \hat{X}^{(2)}_{\ell_2}) - C \right\}
\]

\[
R_{\text{2user, Ach}}(D^{(1)}, D^{(2)}, C) := \min_{(\ell, \ell_2) \in \mathcal{L}_1 \times \mathcal{L}_2} \max \left\{ I(X; \hat{X}^{(2)}_{\ell_2}) + I(X; \hat{X}^{(1)}_{\ell_1} | U, \hat{X}^{(2)}_{\ell_2}), I(X; U, \hat{X}^{(1)}_{\ell_1}, \hat{X}^{(2)}_{\ell_2}) - C \right\}
\]

\[
R_{\text{2user}}(0, 0, C) \leq \min_{P_{U|X}} \max_{(\ell, \ell_2) \in \mathcal{L}_1 \times \mathcal{L}_2} \max \left\{ H(\hat{X}^{(2)}_{\ell_2}) + H(\hat{X}^{(1)}_{\ell_1} | U, \hat{X}^{(2)}_{\ell_2}), H(U, \hat{X}^{(1)}_{\ell_1}, \hat{X}^{(2)}_{\ell_2}) - C \right\}
\]

**Theorem 2:**

\[R_{\text{2user}}(D^{(1)}, D^{(2)}, C) \geq R_{\text{succRef}}(D^{(1)}, D^{(2)}, C).\]

**C. Upper Bound on the RDC Function**

We have the following upper bound on the RDC function. **Definition 6:** Let \( R_{\text{2user, Ach}}(D^{(1)}, D^{(2)}, C) \) be defined as in [3] on top of the next page, where the minimum is taken over all tuples \((U, X^{(1)}, X^{(2)})\) and a collection of auxiliary rates \( \{R_{\ell_2}\}_{\ell_2} \in \mathcal{L}_2 \) such that for every pair \((\ell_1, \ell_2) \in \mathcal{L}_1 \times \mathcal{L}_2:\)

\[C + \hat{R}_{\ell_2} \geq I(U; X, \hat{X}^{(2)}_{\ell_2}) - I(U; \hat{X}^{(2)}_{\ell_2}) = I(U; X; \hat{X}^{(2)}_{\ell_2})\]

\[R - \hat{R}_{\ell_2} \geq I(X; \hat{X}^{(2)}_{\ell_2}) + I(X; \hat{X}^{(1)}_{\ell_1} | U, \hat{X}^{(2)}_{\ell_2}).\]

These rates are achieved by the following scheme. The server compresses the entire library \( X^n \) into \( U^n \) using the adaptive conditional RD code for side-information \( \hat{X}^n_{\ell_2} \) that we describe in the next paragraph. Our adaptive RD code produces a first message of \( nC \) bits which the server stores in user 1’s cache, and a second message of \( nR_{\ell_2} \) bits which the server sends as part of the delivery message. In the delivery message it also sends a standard RD message that allows both users to reconstruct \( \hat{X}^{(2)}_{\ell_2} \), and a standard conditional RD message that allows user 1 to reconstruct \( \hat{X}^{(1)}_{\ell_1} \), given that it already knows \((U^n, \hat{X}^{(2)}_{\ell_2})\). Both users first reconstruct \( \hat{X}^{(2)}_{\ell_2} \). User 1 subsequently reconstructs \( U^n \) and \( \hat{X}^{(1)}_{\ell_1}, n \), always using previously reconstructed sequences as side-information.

Our adaptive conditional RD code uses a codebook \( \mathcal{C} := \{U^n(m_n)\} \) with a nested binning structure: it contains \( \approx 2^{nC} \) outer bins that each consist of \( \approx 2^{nR_{\ell_2}} \) inner bins. The outer binning rate \( C \) is fixed in advanced; the inner binning rate however adapts to the quality of the side-information \( \hat{X}^{(2)}_{\ell_2} \) and is fixed only after the demand \( \ell_2 \) is revealed. Encoding is in two steps. In a first step the server picks the unique codeword \( U^n(m_n) \) that for every \( \ell_2 \in \mathcal{L}_2 \) is jointly typical with the pair \((X^n, \hat{X}^{(2)}_{\ell_2})\). The outer bin index of \( U^n(m_n) \) is immediately available and the server stores the \( nC \) bits representing this index in user 1’s cache. Once the demand \( \ell_2 \) is fixed, also the inner bin index is available and the server sends it as part of the delivery message. Decoding is standard using both bin indices and the side-information \( \hat{X}^{(2)}_{\ell_2} \).

**D. Almost Lossless Reconstructions**

Let now both users reconstruct their demanded files \( X^n_{\ell_1} \) and \( X^n_{\ell_2} \) (almost) losslessly. From **Theorem 3**.

**Corollary 3.1:** The RC-function for the lossless setup satisfies the upper bound in (1) on top of this page.

**Corollary 3.2:** Bound (9) holds with equality when

1) \( \mathcal{L}_1 = \mathcal{L}_2 = \{\ell, \ell'\} \) for \( \ell, \ell' \in \mathcal{L}; \)
2) \( \mathcal{L}_1 = \{\ell\} \) for some \( \ell \in \mathcal{L}; \) or
3) \( \mathcal{L}_2 = \{\ell\} \) for some \( \ell \in \mathcal{L}.\)

**Proof:** To prove cases 1) and 2), specialise the lower bound in **Theorem 2** to the lossless case and to \( U = (X_\ell, X_{\ell'}) \) and \( U = X_\ell, \) respectively. For case 3) a new converse is required.

Interestingly, in the first two cases there is no penalty for not knowing the demands during the caching phase.

**ACKNOWLEDGEMENTS**

The work of R. Timo was supported by the Alexander von Humboldt Foundation. The work of S. Saeedi Bidokhti was supported by the Swiss National Science Foundation fellowship no. 158487.

**REFERENCES**