Compress-and-Forward on a Multiaccess Relay Channel With Computation at the Receiver

Mohieddine El Soussi
ICTEAM, UCL
Place du Levant.2,
B-1348 Louvain-la-Neuve, Belgium
Email: mohieddine.elsoussi@uclouvain.be

Abdellatif Zaidi
Universite Paris-Est Marne-La-Vallee
Boulevard Descartes.5,
77454 Marne la Vallee Cedex 2, France
Email: abdellatif.zaidi@univ-mlv.fr

Luc Vandendorpe
ICTEAM, UCL
Place du Levant.2,
B-1348 Louvain-la-Neuve, Belgium
Email: luc.vandendorpe@uclouvain.be

***Abstract*—We study a system in which two sources communicate with a destination with the help of a half-duplex relay. We consider a decoding strategy, based on the compute-and-forward strategy, in which the destination decodes two integer-valued linear combinations that relate the transmitted codewords. In this strategy, the relay compresses its observation using Wyner-Ziv compression and then forwards it to the destination. The destination appropriately combines what it gets from the direct transmission and the relay. Then, using this combination, it computes two integer-valued linear combinations. We discuss the encoding/decoding strategy, and evaluate the achievable sum-rate. Next, we consider the problem of allocating the powers and selecting the integer-valued coefficients of the recovered linear combinations in order to maximize the sum-rate. For the model under consideration, the optimization problem is NP hard. We propose an iterative algorithm to solve this problem using coordinate descent method. The results are illustrated through some numerical examples.**

I. Introduction

The recently proposed Compute-and-forward (CoF) strategy [1] attracts great attention. The idea is that the receiving node decodes a linear combination of the sources' codewords instead of the individual codewords. A receiver that is given a sufficient number of linear combinations decodes the transmitted messages by solving a system of independent linear equations. This strategy has been studied for different communication systems, such as the two-way relay channel [2], the multiple access channel and multiple-input multiple-output channels [1], [3]. A critical requirement in this strategy is that the coefficients of the linear combinations must be integer-valued. This is essential as the combination of codewords should itself be a codeword so that it is decodable. Lattice codes have this property, and are thus good codes for compute-and-forward strategy [4].

We consider two independent sources that communicate with a destination with the help of a relay node, as shown in Figure 1. In previous works [5] [6], we establish a coding strategy based on the compute-and-forward strategy [1]. In this coding strategy, the relay uses what it receives from the sources during the multiaccess transmission to decode an appropriate integer-valued linear combination of the sources' codewords; and then sends this combination to the destination. In addition to the linear combination that it receives from the relay, the destination decodes another integer-valued linear combination from what it gets directly from the sources. In this strategy, the two linear combinations are recovered in a distributed way. By opposition to [5], in this work, we develop a coding strategy in which both linear combinations of the sources' codewords are recovered locally at the destination. More specifically, the relay compresses its observation from the sources' transmission during the first transmission period using Wyner-Ziv compression [7] taking into account the information available at the destination through the direct transmission. It then transmits the compressed version to the destination during the second transmission period. The destination determines the two required linear combinations, as follows. It utilizes an appropriate combination of the output from the sources' transmission during the first period and of the output from the relay' transmission during the second period. From this combination, two independent linear combinations relating the sources' codewords are computed.

For this coding strategy, we target the optimization of the sources and the relay powers, and of the integer coefficients of the linear combinations to maximize the achievable sum-rate. The optimization problem is NP-hard. We develop an iterative approach that finds the appropriate power and integer coefficients alternatively. More specifically, we show that the problem of finding appropriate integer coefficients for a given set of powers has the same solution as an approximated mixed integer quadratic programming (MIQP) problem with quadratic constraints. Also, we show that the problem of finding the appropriate power policy at the sources and the relay for a given set of integer coefficients is a non-linear non-convex optimization problem. We formulate and solve this problem through geometric programming and successive convex approximation approach [8].

We compare our strategy with standard compress-and-forward (CF), standard decode-and-forward (DF) and compute-and-forward strategies [6] for the case of *symmetric* rate. We show that for some channel gain values it is better to recover the linear combinations locally at the destination instead of in a distributed way. Also, we show that our coding strategy achieves as good as regular compress-and-forward; and has the advantage of utilizing feasible linear codes instead of random codes which are infeasible in practice. More specifically, standard CF usually uses the maximum likelihood (ML) receiver which has high computational complexity. However, linear receivers such as the decorrelator and minimum-mean-squared error (MMSE) receiver are often used as low-complexity alternatives.

We use the following notations throughout the paper. Lowercase boldface letters are used to denote column vectors, e.g., $\mathbf{x}$. Upper case boldface letters are used to denote matrices, e.g., $\mathbf{X}$. Also, we use $\mathbf{X}^T$ to designate matrix transpose of $\mathbf{X}$; and $\det(\mathbf{X})$ to designate the determinant of $\mathbf{X}$. We use $\mathbf{I}_n$ to denote the n-by-n identity matrix; and $\mathbf{0}$ to denote a matrix whose elements are all zeros (its size will be evident from the context). For two vectors $\mathbf{x}$ and $\mathbf{y} \in \mathbb{R}^n$, the vector $\mathbf{z} = \mathbf{x} \circ \mathbf{y} \in \mathbb{R}^n$ denotes the Hadamard product of $\mathbf{x}$ and $\mathbf{y}$, i.e., the vector whose ith element is the product of the ith elements of $\mathbf{x}$ and $\mathbf{y}$, i.e., $z_i = (\mathbf{x} \circ \mathbf{y})_i = x_i y_i$. We use $\mathrm{Var}(\mathbf{x})$ to denote the power of $\mathbf{x}$ i.e. $\mathbb{E}[\|\mathbf{x}\|^2]$. Finally, logarithms are taken to base 2; and, for $x \in \mathbb{R}$, $\log^+(x) := \max\{\log(x), 0\}$.

II. System Model

Two sources A and B communicate with a destination D with the help of a relay R. The sources would like to transmit their messages $W_a \in \mathcal{W}_a$, $W_b \in \mathcal{W}_b$ to the destination reliably, in $2n$ uses of the channel. Let R_a and R_b be the transmission rate of message W_a and W_b respectively. In this work, we focus on the *symmetric* rate case, i.e., $R_a = R_b = R$, or equivalently, $|\mathcal{W}_a| = |\mathcal{W}_b| = 2^{2nR}$. We measure the system performance in terms of the allowed achievable sum rate $R_{\text{sum}} = R_a + R_b = 2R$. Also, we divide the transmission time into two transmission periods having each of length n channel uses and we assume that the relay operates in a half-duplex mode.

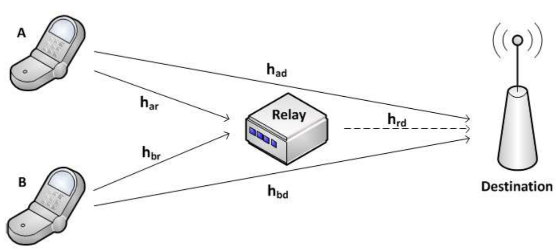

Figure 1. Multiple-access channel with a half-duplex relay

During the first transmission period, the sources encode their messages $W_i \in [1, 2^{2nR}]$ $i \in \{a, b\}$ into codeword $\mathbf{x}_i$ and send them over the channel. Let $\mathbf{y}_r$ and $\mathbf{y}_d$ be the signals received respectively at the relay and at the destination during this period. These signals are given by

$$\begin{aligned} \mathbf{y}_r &= h_{ar}\mathbf{x}_a + h_{br}\mathbf{x}_b + \mathbf{z}_r \\ \mathbf{y}_d &= h_{ad}\mathbf{x}_a + h_{bd}\mathbf{x}_b + \mathbf{z}_d, \end{aligned} \quad (1)$$

where h_{ad} and h_{bd} are the channel gains on the links transmitters-to-destination, h_{ar} and h_{br} are the channel gains on the links transmitters-to-relay, and $\mathbf{z}_r$ and $\mathbf{z}_d$ are additive background noises at the relay and the destination.

During the second transmission period, the relay sends a codeword $\tilde{\mathbf{x}}_r$ to help both sources. During this period, the destination receives

$$\tilde{\mathbf{y}}_d = h_{rd}\tilde{\mathbf{x}}_r + \tilde{\mathbf{z}}_d, \quad (2)$$

where h_{rd} is the channel gain on the link relay-to-destination, and $\tilde{\mathbf{z}}_d$ is additive background noise.

Throughout, we assume that all channel gains are real-valued, fixed and known to all the nodes in the network; and the noises at the relay and the destination are independent of each other, and independently and identically distributed (i.i.d) Gaussian, with zero mean and variance N. Furthermore, we consider the following individual constraints on the transmitted power (per codeword),

$$\begin{aligned} &\mathbb{E}[\|\mathbf{x}_a\|^2] = n\beta_a^2 P \le nP_a, \qquad \mathbb{E}[\|\mathbf{x}_b\|^2] = n\beta_b^2 P \le nP_b, \\ &\mathbb{E}[\|\tilde{\mathbf{x}}_r\|^2] = n\beta_r^2 P \le nP_r, \end{aligned} \quad (3)$$

where $P_a \ge 0$, $P_b \ge 0$ and $P_r \ge 0$ are some constraints imposed by the system; $P \ge 0$ is given, and β_a, β_b and β_r are some scalars that can be chosen to adjust the actual transmitted powers, and are such that $0 \le |\beta_a| \le \sqrt{P_a/P}$, $0 \le |\beta_b| \le \sqrt{P_b/P}$ and $0 \le |\beta_r| \le \sqrt{P_r/P}$. For convenience, we will sometimes use the shorthand vector notation $\mathbf{h}_d = [h_{ad},\, h_{bd}]^T$, $\mathbf{h}_r = [h_{ar},\, h_{br}]^T \in \mathbb{R}^2$. Also, we use $\boldsymbol{\beta} = [\beta_a, \beta_b, \beta_r]^T \in \mathbb{R}^3$, $\boldsymbol{\beta}_s = [\beta_a, \beta_b]^T \in \mathbb{R}^2$, and the shorthand matrix notation $\mathbf{H} = [\mathbf{h}_d^T;\, \mathbf{h}_r^T] \in \mathbb{R}^{2\times 2}$. Finally, the signal-to-noise ratio will be denoted as $\mathrm{snr} = P/N$ in the linear scale, and by $\mathrm{SNR} = 10\log_{10}(\mathrm{snr})$ in decibels in the logarithmic scale.

III. Computation at the Destination

In this section, we develop the coding strategy that is based on the compute-and-forward strategy of [1]. In this strategy, the relay compresses what it receives from the sources during the first transmission period taking into account the information available at the destination through the direct transmission. It then transmits the compressed version to the destination during the second transmission period. The destination computes two linearly independent combinations of the sources' codewords using its outputs from both transmission periods locally. The following proposition provides an achievable sum-rate for the model that we study.

***Proposition** 1:* For any set of channel vector $\mathbf{h} = [h_{ar},\, h_{br},\, h_{ad},\, h_{bd},\, h_{rd}]^T \in \mathbb{R}^5$, the following sum rate is achievable [6]:

$$\begin{aligned} \text{(A): } R_{\text{sum}}^{\text{CoD}} = \max \frac{1}{2} \min \Bigg\{ & \\ & \log^+\left(\frac{\mathrm{snr}}{\mathrm{snr}\|\boldsymbol{\beta}_s \circ \mathbf{H}^T\boldsymbol{\alpha}_t - \mathbf{t}\|^2 + (\boldsymbol{\alpha}_t \circ \boldsymbol{\alpha}_t)^T\mathbf{n}_d}\right), \\ & \log^+\left(\frac{\mathrm{snr}}{\mathrm{snr}\|\boldsymbol{\beta}_s \circ \mathbf{H}^T\boldsymbol{\alpha}_k - \mathbf{k}\|^2 + (\boldsymbol{\alpha}_k \circ \boldsymbol{\alpha}_k)^T\mathbf{n}_d}\right)\Bigg\}, \end{aligned} \quad (4)$$

where $\boldsymbol{\alpha}_t = [\alpha_{1t},\, \alpha_{2t}]^T$ and $\boldsymbol{\alpha}_k = [\alpha_{1k},\, \alpha_{2k}]^T \in \mathbb{R}^2$ are some inflation factors, $\mathbf{n}_d = [1,\, 1 + D/N]^T \in \mathbb{R}^2$, and D is given by

$$D = \frac{N^2\left(1+\text{snr}\|\boldsymbol{\beta_s}\circ\mathbf{h}_r\|^2\right)}{|h_{rd}|^2\beta_r^2 P} - \frac{N^2\left(\text{snr}(\boldsymbol{\beta_s}\circ\mathbf{h}_r)^T(\boldsymbol{\beta_s}\circ\mathbf{h}_d)\right)^2}{|h_{rd}|^2\beta_r^2 P\left(1+\text{snr}\|\boldsymbol{\beta_s}\circ\mathbf{h}_d\|^2\right)}, \tag{5}$$

and the maximization is over $\boldsymbol{\alpha}_t$, $\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}$ such that $0 \le |\beta_a| \le \sqrt{P_a/P}$, $0 \le |\beta_b| \le \sqrt{P_b/P}$, and $0 \le |\beta_r| \le \sqrt{P_r/P}$ and over the integer coefficients $\mathbf{k}$ and $\mathbf{t}$ such that $|\det(\mathbf{k},\mathbf{t})| \ge 1$.

Proof: Let Λ be an n-dimensional lattice of second moment $\sigma_\Lambda^2 = P$ and normalized second moment $G(\Lambda)$, and $\mathcal{V}$ be its fundamental Voronoi region. Also similarly as in [4] [6], let $\Lambda_{\text{FINE}} \supseteq \Lambda$ be chosen such that the codebook $\mathcal{C} = \Lambda_{\text{FINE}} \cap \mathcal{V}$ be of cardinality 2^{2nR}. Let $\mathbf{k} = [k_a, k_b] \in \mathbb{Z}^2$ and $\mathbf{t} = [t_a, t_b] \in \mathbb{Z}^2$ be given such that $|\det(\mathbf{k},\mathbf{t})| = |k_a t_b - k_b t_a| \ge 1$.

Encoding: Let (W_a, W_b) be the pair of messages to be transmitted. Let $\mathbf{u}_a$, $\mathbf{u}_b$ and $\mathbf{u}_r$ be some dither vectors that are drawn independently and uniformly over $\mathcal{V}$ and known by all nodes in the network. Also, let $\phi_a(\cdot)$ and $\phi_b(\cdot)$ be one-to-one mapping functions between the set $\{W_a\}$ and the codebook $\mathcal{C}$ and between the set $\{W_b\}$ and the codebook $\mathcal{C}$, respectively. Let $\mathbf{v}_a = \phi_a(W_a)$ and $\mathbf{v}_b = \phi_b(W_b)$, where $\mathbf{v}_a \in \mathcal{C}$ and $\mathbf{v}_b \in \mathcal{C}$.

During the first transmission period, to transmit message W_i, $i \in \{a, b\}$, the sources send

$$\mathbf{x}_i = \beta_i\left([\mathbf{v}_i - \mathbf{u}_i] \bmod \Lambda\right). \tag{6}$$

During this period, the relay quantizes what it receives using Wyner-Ziv compression [7], taking into account the available side information $\mathbf{y}_d$ at the destination. Let $\hat{\mathbf{y}}_r$ be the compressed version of $\mathbf{y}_r$ given by

$$\hat{\mathbf{y}}_r = \mathbf{y}_r + \mathbf{d} \tag{7}$$

where $\mathbf{d}$ is a Gaussian random vector whose elements are i.i.d with zero mean and variance D; and is independent of all other signals.

During the second transmission period, the relay conveys the description $\hat{\mathbf{y}}_r$ of $\mathbf{y}_r$ to the destination. To this end, it sends an independent Gaussian input $\tilde{\mathbf{x}}_r$ with power $\beta_r^2 P$ and carries the Wyner-Ziv compression index of $\hat{\mathbf{y}}_r$.

Decoding at the destination: During the two transmission periods, the destination receives $\mathbf{y}_d$ and $\tilde{\mathbf{y}}_d$ as given in (1) and (2). The destination first recovers the compressed version of the relay's output sent by the relay during the second transmission period ($\hat{\mathbf{y}}_r$), by utilizing its output $\tilde{\mathbf{y}}_d$ as well as the available side information $\mathbf{y}_d$ [6]. The destination computes two linearly independent combinations with integer coefficients, as follows. It combines $\mathbf{y}_d$ and $\hat{\mathbf{y}}_r$ and uses the obtained signal to compute the two integer-valued linear combinations. More precisely, let

$$\begin{aligned}\mathbf{y}_j &= \alpha_{1j}\mathbf{y}_d + \alpha_{2j}\hat{\mathbf{y}}_r\\ &= \left(\alpha_{1j}h_{ad} + \alpha_{2j}h_{ar}\right)\mathbf{x}_a + \left(\alpha_{1j}h_{bd} + \alpha_{2j}h_{br}\right)\mathbf{x}_b +\\ &\quad \alpha_{1j}\mathbf{z}_d + \alpha_{2j}\mathbf{z}_r + \alpha_{2j}\mathbf{d},\end{aligned} \tag{8}$$

for some $\boldsymbol{\alpha}_j = [\alpha_{1j},\, \alpha_{2j}]^T$ and for $j \in \{t, k\}$. The destination, using the obtained signal $\mathbf{y}_j$, decodes two linear combinations with integer coefficients by performing the modulo-reduction operation [4] [6],

$$\begin{aligned}\mathbf{y}'_j &= [\mathbf{y}_j + j_a\mathbf{u}_a + j_b\mathbf{u}_b] \bmod \Lambda\\ &= [j_a\mathbf{v}_a + j_b\mathbf{v}_b + \mathbf{z}'_j] \bmod \Lambda\end{aligned} \tag{9}$$

where $\mathbf{z}'_j$ is the effective noise given by

$$\begin{aligned}\mathbf{z}'_j \triangleq \Big[&\alpha_{1j}\mathbf{z}_d + \alpha_{2j}\mathbf{z}_r + \alpha_{2j}\mathbf{d} + (\alpha_{1j}h_{ad} + \alpha_{2j}h_{ar} - \frac{j_a}{\beta_a})\mathbf{x}_a +\\ &(\alpha_{1j}h_{bd} + \alpha_{2j}h_{br} - \frac{j_b}{\beta_b})\mathbf{x}_b\Big] \bmod \Lambda.\end{aligned} \tag{10}$$

Finally, by decoding the lattice points $\mathbf{e}_t = [t_a\mathbf{v}_a + t_b\mathbf{v}_b] \in \Lambda$ and $\mathbf{e}_k = [k_a\mathbf{v}_a + k_b\mathbf{v}_b] \in \Lambda$ using $\mathbf{y}'_j$, the destination obtains two linear combinations with integer coefficients. Hence, it obtains the transmitted codewords by solving a system of two equations with two variables.

Rate Analysis:

Using $\mathbf{y}'_j$ given by (9), we get that the destination obtains the desired combination $\mathbf{j}$, $\mathbf{j} \in \{\mathbf{t},\, \mathbf{k}\}$, at mutual information satisfying [4] [6]

$$\begin{aligned}\frac{1}{2n}I(\mathbf{e}_j;\mathbf{y}'_j) &\ge \frac{1}{4}\log^+\left(\frac{\sigma_\Lambda^2}{\text{Var}(\mathbf{z}'_j)}\right) - \frac{1}{4}\log\left(2\pi e G(\Lambda)\right)\\ &\ge \frac{1}{4}\log^+\left(\frac{\text{snr}}{\text{snr}\|\boldsymbol{\beta}_s\circ\mathbf{H}^T\boldsymbol{\alpha}_j - \mathbf{j}\|^2 + (\boldsymbol{\alpha}_j\circ\boldsymbol{\alpha}_j)^T\mathbf{n}_d}\right) -\\ &\quad \frac{1}{4}\log\left(2\pi e G(\Lambda)\right),\end{aligned} \tag{11}$$

where $\boldsymbol{\alpha}_j$ should be chosen to minimize the effective noise $\mathbf{z}'_j$ in (10), i.e., such that

$$\boldsymbol{\alpha}_j^\star = \left(\mathbf{G}\mathbf{G}^T + \mathbf{N}_d\right)^{-1}\mathbf{G}\mathbf{j}, \tag{12}$$

where $\mathbf{G} = [(\boldsymbol{\beta}_s\circ\mathbf{h}_d)^T;\, (\boldsymbol{\beta}_s\circ\mathbf{h}_r)^T] \in \mathbb{R}^{2\times 2}$ and $\mathbf{N}_d = [1/\text{snr},\, 0;\, 0,\, 1/\text{snr} + D/P] \in \mathbb{R}^{2\times 2}$.

Let $R_t(\Lambda)$ be the RHS of (11) for $j = t$ and $R_k(\Lambda)$ the RHS of (11) for $j = k$. The above means that the destination can decode the sources' codewords correctly at the transmission sum rate $R_{\text{sum}}^{\text{CoD}}(\Lambda) = 2\,\min\{R_t(\Lambda), R_k(\Lambda)\}$. Furthermore, investigating the expression of $R_j(\Lambda)$, it can easily be seen that it decreases with increasing D. Also, observing that the RHS of (5) decreases if β_r increases, the largest rate $R_j(\Lambda)$ is then obtained by taking the equality in the distortion constraint (5) with $\beta_r^2 = P_r/P$. Finally, observing that $2\pi e G(\Lambda) \to 1$ when $n \to \infty$ [4], the desired sum rate (4) is obtained by taking the limit of $R_{\text{sum}}^{\text{CoD}}(\Lambda)$ as n goes to infinity; and this completes the proof of Proposition 1. □

IV. Sum Rates Optimization

This section is dedicated to evaluate the optimal powers and integer-coefficients that maximize the sum-rate of Proposition 1. In order to compute $R_{\text{sum}}^{\text{CoD}}$ as given by (4), we develop the following iterative algorithm which optimizes the integer coefficients and the powers alternatively, and to which we refer to as "Algorithm A" in reference to the optimization problem (A) in (4).

Algorithm A Iterative algorithm for computing $R_{\text{sum}}^{\text{CoD}}$ as given by (4) in Proposition 1

1: Choose an initial feasible vector $\boldsymbol{\beta}_{\text{s}}{}^{(0)}$ and set $\iota = 1$.
2: Solve (4) with $\boldsymbol{\beta}_{\text{s}} = \boldsymbol{\beta}_{\text{s}}{}^{(\iota-1)}$ for the optimal $\mathbf{k}$ and $\mathbf{t}$ using Algorithm A-1 and assign it to $\mathbf{k}^{(\iota)}$ and $\mathbf{t}^{(\iota)}$.
3: Solve (4) with $\mathbf{k} = \mathbf{k}^{(\iota)}$ and $\mathbf{t} = \mathbf{t}^{(\iota)}$ for the optimal $\boldsymbol{\beta}_{\text{s}}$ using Algorithm A-2 and assign it to $\boldsymbol{\beta}_{\text{s}}{}^{(\iota)}$.
4: Increment the iteration index as $\iota = \iota + 1$ and go back to Step 2.
5: Terminate if $\|\boldsymbol{\beta}_{\text{s}}{}^{(\iota)} - \boldsymbol{\beta}_{\text{s}}{}^{(\iota-1)}\| \le \epsilon$, $|R_{\text{sum}}^{\text{CoD}}[\iota] - R_{\text{sum}}^{\text{CoD}}[\iota-1]| \le \epsilon$ or if $R_{\text{sum}}^{\text{CoD}}[\iota] \le R_{\text{sum}}^{\text{CoD}}[\iota-1]$.

1) Integer Coefficients Optimization: In this section, we focus on the problem of evaluating the integer vectors $\mathbf{k} \in \mathbb{Z}^2$ and $\mathbf{t} \in \mathbb{Z}^2$ for a given value of $\boldsymbol{\beta}_s$. Examining the objective function in (4), it can easily be seen that this problem can be equivalently stated as

$$\min_{\mathbf{k},\,\mathbf{t},\,\Theta_1} \quad \Theta_1 \tag{13a}$$
$$\text{s. t.} \quad \Theta_1 \ge \mathbf{t}^T \boldsymbol{\Omega} \mathbf{t} \tag{13b}$$
$$\Theta_1 \ge \mathbf{k}^T \boldsymbol{\Omega} \mathbf{k} \tag{13c}$$
$$\det(\mathbf{k},\mathbf{t}) = |k_a t_b - k_b t_a| \ge 1 \tag{13d}$$
$$\mathbf{k},\ \mathbf{t} \in \mathbb{Z}^2,\ \ \Theta_1 \in \mathbb{R}, \tag{13e}$$

where $\boldsymbol{\Omega} = (\mathbf{G}^T(\mathbf{G}\mathbf{G}^T + \mathbf{N}_d)^{-1}\mathbf{G} - \mathbf{I}_2)^T(\mathbf{G}^T(\mathbf{G}\mathbf{G}^T + \mathbf{N}_d)^{-1}\mathbf{G} - \mathbf{I}_2) + ((\mathbf{G}\mathbf{G}^T + \mathbf{N}_d)^{-1}\mathbf{G})^T\mathbf{N}_d((\mathbf{G}\mathbf{G}^T + \mathbf{N}_d)^{-1}\mathbf{G})$. Note that Θ_1 is simultaneously an extra optimization variable and the objective function in (13). Also, we should note that the integer coefficients $\mathbf{k}$ and $\mathbf{t}$ are independent of $\boldsymbol{\alpha}_t$ and $\boldsymbol{\alpha}_k$.

In order to solve the problem (13), we need to reformulate it in a convenient manner. We introduce the following quantities. Let $\mathbf{a}_0 = [0,0,0,0,1]^T$; $\mathbf{a}_1 = \mathbf{a}_2 = [0,0,0,0,-1]^T$ and $\mathbf{a}_3 = [0,0,0,0,0]^T$. Also, let $\mathbf{b} = [t_a, t_b, k_a, k_b, \Theta_1]^T$; and the scalars $c_1 = c_2 = 0$, and $c_3 = -1$. We also introduce the following five-by-five matrices $\mathbf{F}_1$, $\mathbf{F}_2$, and $\mathbf{F}_3$, where

$$\mathbf{F}_1 = \begin{bmatrix} 2\boldsymbol{\Omega} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{F}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 2\boldsymbol{\Omega} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$
$$\text{and} \qquad \mathbf{F}_3 = \begin{bmatrix} \mathbf{0} & 0 & -2 & 0 \\ \mathbf{0} & 2 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{14}$$

The optimization problem (13) can now be reformulated equivalently as

$$\min_{\mathbf{b}} \quad \mathbf{a}_0^T \mathbf{b}$$
$$\text{s. t.} \quad \frac{1}{2}\mathbf{b}^T \mathbf{F}_i \mathbf{b} + \mathbf{a}_i^T \mathbf{b} \le c_i \quad i = 1,2,3$$
$$\mathbf{k} \in \mathbb{Z}^2,\ \mathbf{t} \in \mathbb{Z}^2,\ \ \Theta_1 \in \mathbb{R} \tag{15}$$

The equivalent optimization problem (15) is a MIQP problem with quadratic constraints [9]. There are known algorithms for solving MIQP optimization problems, such as branch and bound algorithm [9]. A critical condition for solving MIQP optimization problems, is that the involved matrices associated with the quadratic constraints must all be semi-definite. In our case, it is easy to see that the matrices $\mathbf{F}_1$ and $\mathbf{F}_2$ are positive semi-definite. However, the matrix $\mathbf{F}_3$ is indefinite, irrespective to the values of $\mathbf{k}$ and $\mathbf{t}$.

In order to transform the optimization problem (15) into one that is MIQP-compatible, we replace the quadratic constraint (13d) with one that is linear. We introduce the following two real-valued vectors $\tilde{\mathbf{k}} = [\tilde{k}_a, \tilde{k}_b]^T \in \mathbb{R}^2$ and $\tilde{\mathbf{t}} = [\tilde{t}_a, \tilde{t}_b]^T \in \mathbb{R}^2$ defined such that they satisfy

$$\mathbf{k} = \boldsymbol{\kappa} \circ \exp(\tilde{\mathbf{k}}), \quad \mathbf{t} = \boldsymbol{\tau} \circ \exp(\tilde{\mathbf{t}}), \tag{16}$$

where $\boldsymbol{\kappa} = [\kappa_a, \kappa_b]^T \in \mathbb{R}^2$ and $\boldsymbol{\tau} = [\tau_a, \tau_b]^T \in \mathbb{R}^2$ are constant vectors to be chosen appropriately. Thus, the constraint (13d) can now be rewritten equivalently as

$$|\kappa_a \tau_b \exp(\tilde{k}_a + \tilde{t}_b) - \kappa_b \tau_a \exp(\tilde{k}_b + \tilde{t}_a)| \ge 1. \tag{17}$$

Now, we linearize the constraint (17) by selecting the constant vectors $\boldsymbol{\kappa}$ and $\boldsymbol{\tau}$ such that the first order Taylor series approximations $\exp(\tilde{\mathbf{k}}) \approx \mathbf{1} + \tilde{\mathbf{k}}$ and $\exp(\tilde{\mathbf{t}}) \approx \mathbf{1} + \tilde{\mathbf{t}}$ hold. Hence, the constraint (13d) can be rewritten as

$$|\kappa_a \tau_b (1 + \tilde{k}_a + \tilde{t}_b) - \kappa_b \tau_a (1 + \tilde{k}_b + \tilde{t}_a)| \gtrsim 1. \tag{18}$$

Note that the constraint (18) is now linear, and the optimization problem (13) has the same solution as the following problem which is MIQP-compatible,

$$\min_{\mathbf{k},\,\mathbf{t},\,\Theta_1} \ \Theta_1 \tag{19a}$$
$$\text{s.t.} \ \Theta_1 \ge \mathbf{t}^T \boldsymbol{\Omega} \mathbf{t} \tag{19b}$$
$$\Theta_1 \ge \mathbf{k}^T \boldsymbol{\Omega} \mathbf{k} \tag{19c}$$
$$-|\kappa_a \tau_b (1 + \tilde{k}_a + \tilde{t}_b) - \kappa_b \tau_a (1 + \tilde{k}_b + \tilde{t}_a)| \lesssim -1 \tag{19d}$$
$$\frac{k_i}{\kappa_i} - 1 - \tilde{k}_i \le 0, \quad -\frac{k_i}{\kappa_i} + 1 + \tilde{k}_i \le 0, \quad i = a,b \tag{19e}$$
$$\frac{t_i}{\tau_i} - 1 - \tilde{t}_i \le 0, \quad -\frac{t_i}{\tau_i} + 1 + \tilde{t}_i \le 0, \quad i = a,b \tag{19f}$$
$$\mathbf{k},\ \mathbf{t} \in \mathbb{Z}^2,\ \ \tilde{\mathbf{k}},\ \tilde{\mathbf{t}},\ \boldsymbol{\kappa},\ \boldsymbol{\tau} \in \mathbb{R}^2, \Theta_1 \in \mathbb{R}. \tag{19g}$$

The optimization problem (19) can be solved iteratively using Algorithm A-1 hereinafter.

Algorithm A-1 Integer coefficients selection for $R_{\text{sum}}^{\text{CoD}}$ as given by (4) in Proposition 1

1: Initialization: set $\iota_1 = 1$.
2: Use the branch-and-bound algorithm of [10] to solve for $\Theta_1^{(\iota_1)}$, $\mathbf{k}^{(\iota_1)}$ and $\mathbf{t}^{(\iota_1)}$ with the constraint (19d) substituted with $-\kappa_a \tau_b (1 + \tilde{k}_a + \tilde{t}_b) + \kappa_b \tau_a (1 + \tilde{k}_b + \tilde{t}_a) \le -1$.
3: Update the values of $\boldsymbol{\kappa}$ and $\boldsymbol{\tau}$ in a way to satisfy (16); and increment the iteration index as $\iota_1 = \iota_1 + 1$.
4: Terminate if $\exp(\tilde{\mathbf{k}}^{(\iota_1)}) \approx \mathbf{1} + \tilde{\mathbf{k}}^{(\iota_1)}$ and $\exp(\tilde{\mathbf{t}}^{(\iota_1)}) \approx \mathbf{1} + \tilde{\mathbf{t}}^{(\iota_1)}$. Denote the found solution as $\Theta_1^{\min,2}$.
5: Redo steps 1 to 4 with in Step 2 the constraint (19d) substituted with $\kappa_a \tau_b (1 + \tilde{k}_a + \tilde{t}_b) - \kappa_b \tau_a (1 + \tilde{k}_b + \tilde{t}_a) \le -1$. In Step 4, denote the found solution as $\Theta_1^{\min,2}$.
6: Select the integer coefficients corresponding to the minimum among $\Theta_1^{\min,1}$ and $\Theta_1^{\min,2}$.

2) Power Allocation Policy: The problem of optimizing the power value $\boldsymbol{\beta}_\mathrm{s}$ for fixed integer coefficients $\mathbf{k}$ and $\mathbf{t}$, can be written as,

$$\min_{\boldsymbol{\beta}_\mathrm{s},\, \boldsymbol{\alpha}_t,\, \boldsymbol{\alpha}_k,\, \Theta_2} \Theta_2 \tag{20a}$$

$$\text{s. t.}\Theta_2 \geq \frac{\mathrm{snr}\|\boldsymbol{\beta}_s \circ \mathbf{H}^T\boldsymbol{\alpha}_t - \mathbf{t}\|^2 + (\boldsymbol{\alpha}_t \circ \boldsymbol{\alpha}_t)^T\mathbf{n}_d}{\mathrm{snr}}, \tag{20b}$$

$$\Theta_2 \geq \frac{\mathrm{snr}\|\boldsymbol{\beta}_s \circ \mathbf{H}^T\boldsymbol{\alpha}_k - \mathbf{k}\|^2 + (\boldsymbol{\alpha}_k \circ \boldsymbol{\alpha}_k)^T\mathbf{n}_d}{\mathrm{snr}}, \tag{20c}$$

$$D \geq \frac{N^2\left(1+\mathrm{snr}\|\boldsymbol{\beta}_s \circ \mathbf{h}_r\|^2\right)}{|h_{rd}|^2P_r} - \frac{N^2\left(\mathrm{snr}(\boldsymbol{\beta}_s \circ \mathbf{h}_r)^T(\boldsymbol{\beta}_s \circ \mathbf{h}_d)\right)^2}{|h_{rd}|^2P_r\left(1+\mathrm{snr}\|\boldsymbol{\beta}_s \circ \mathbf{h}_d\|^2\right)} \tag{20d}$$

$$-\sqrt{\frac{P_i}{P}} \leq \beta_i \leq \sqrt{\frac{P_i}{P}},\ \ i = a, b \tag{20e}$$

$$\boldsymbol{\beta}_s \in \mathbb{R}^2,\ \Theta_2 \in \mathbb{R}. \tag{20f}$$

Note that $\boldsymbol{\beta}_s$ depends on $\boldsymbol{\alpha}_t$ and $\boldsymbol{\alpha}_k$. The complexity of optimizing these variables simultaneously is very high and it decreases if the optimization is carried out in two steps. First, we optimize the power value $\boldsymbol{\beta}_\mathrm{s}$ using geometric programming with successive convex approximation as described below, for a fixed value of $\boldsymbol{\alpha}_t$ and $\boldsymbol{\alpha}_k$. Next, we optimize the value of $\boldsymbol{\alpha}_t$ and $\boldsymbol{\alpha}_k$ for a fixed value of $\boldsymbol{\beta}_\mathrm{s}$ using (12). This process is repeated until convergence.

The optimization problem in (20) is non-linear and non-convex for a fixed value of $\boldsymbol{\alpha}_t$ and $\boldsymbol{\alpha}_k$. We use geometric programming (GP) [8] to solve it. The GP algorithm requires that the constraints are posynomial and the variables are strictly positive [8]. In our case, in the problem (20), the constraints (20b), (20c) and (20d) contain functions that are non posynomial. Also, the variables in (20) are not all positive. In what follows, we first transform the problem (20) into one equivalent in which the constraints involve functions that are all posynomial and the variables are all positive; and then, we develop an algorithm for solving the equivalent problem.

Let $\mathbf{c} = [c_a, c_b]^T \in \mathbb{R}^2$ and $\boldsymbol{\delta}_s = [\delta_a, \delta_b]^T \in \mathbb{R}^2$, such that $c_i > \sqrt{P_i/P}$ and $\delta_i = \beta_i + c_i$ for $i \in \{a, b\}$. Note that the elements of $\boldsymbol{\delta}_s$ are all strictly positive. Also, we define the posynomial functions $f_1(\boldsymbol{\delta}_\mathrm{s}, \Theta_2, \boldsymbol{\alpha}_t, \boldsymbol{\alpha}_k)$ and $g_1(\boldsymbol{\delta}_\mathrm{s}, \Theta_2, \boldsymbol{\alpha}_t, \boldsymbol{\alpha}_k)$ which correspond to the constraint (20b), the posynomial functions $f_2(\boldsymbol{\delta}_\mathrm{s}, \Theta_2, \boldsymbol{\alpha}_t, \boldsymbol{\alpha}_k)$ and $g_2(\boldsymbol{\delta}_\mathrm{s}, \Theta_2, \boldsymbol{\alpha}_t, \boldsymbol{\alpha}_k)$ which correspond to the constraint (20c), and the posynomial functions $f_3(\boldsymbol{\delta}_\mathrm{s})$ and $g_3(\boldsymbol{\delta}_\mathrm{s})$ which correspond to the constraint (20d). These functions are not given here due to space limitation. It is now easy to see that the optimization problem can be stated in the following form,

$$\min_{\boldsymbol{\delta}_\mathrm{s},\, \boldsymbol{\alpha}_t,\, \boldsymbol{\alpha}_k,\, \Theta_2} \Theta_2 \tag{21a}$$

$$\text{s. t.}\ \frac{f_1(\boldsymbol{\delta}_\mathrm{s}, \Theta_2, \boldsymbol{\alpha}_t, \boldsymbol{\alpha}_k)}{g_1(\boldsymbol{\delta}_\mathrm{s}, \Theta_2, \boldsymbol{\alpha}_t, \boldsymbol{\alpha}_k)} \leq 1, \quad \frac{f_2(\boldsymbol{\delta}_\mathrm{s}, \Theta_2, \boldsymbol{\alpha}_t, \boldsymbol{\alpha}_k)}{g_2(\boldsymbol{\delta}_\mathrm{s}, \Theta_2, \boldsymbol{\alpha}_t, \boldsymbol{\alpha}_k)} \leq 1, \tag{21b}$$

$$\frac{f_3(\boldsymbol{\delta}_\mathrm{s})}{g_3(\boldsymbol{\delta}_\mathrm{s})} \leq 1 \tag{21c}$$

$$-\sqrt{\frac{P_i}{P}} + c_i \leq \delta_i \leq \sqrt{\frac{P_i}{P}} + c_i\ ,\ i = a, b \tag{21d}$$

$$\boldsymbol{\delta}_s \in \mathbb{R}^2,\ \mathbf{c} \in \mathbb{R}^2,\ \Theta_2 \in \mathbb{R}. \tag{21e}$$

The functions in (21b) and (21c) consist of ratios of posynomials, i.e., are not posynomial. This is a non-convex class of GP problems known as Complementary GP [8]. We can transform the Complementary GPs into GPs by a series of approximations. The ratio between two posynomials can be turned into GPs by approximating the denominator of the ratio of posynomials, $g()$, with a monomial $\tilde{g}()$ [8].

The optimization problem (21) is now turned to a GP problem, and can be solved using an interior point method. The problem of finding the appropriate value of $\boldsymbol{\delta}_s$ can be solved using Algorithm A-2.

Algorithm A-2 Power allocation policy for $R_{\mathrm{sum}}^{\mathrm{CoD}}$ as given by (4) in Proposition 1

1: Set $\boldsymbol{\delta}_\mathrm{s}{}^{(0,0)}$ to some initial value and set $\iota_2 = 1$ and $\iota_3 = 0$.
2: Compute $\Theta_2^{(\iota_2-1,\iota_3)}$, $\boldsymbol{\alpha}_t^{(\iota_2-1,\iota_3)}$ and $\boldsymbol{\alpha}_k^{(\iota_2-1,\iota_3)}$ using $\boldsymbol{\delta}_s^{(\iota_2-1,\iota_3)}$.
3: Approximate $g(\boldsymbol{\delta}_s^{(\iota_2,\iota_3)}, \Theta_2^{(\iota_2,\iota_3)})$ with $\tilde{g}(\boldsymbol{\delta}_s^{(\iota_2,\iota_3)}, \Theta_2^{(\iota_2,\iota_3)})$ around $\boldsymbol{\delta}_s^{(\iota_2-1,\iota_3)}$ and $\Theta_2^{(\iota_2-1,\iota_3)}$.
4: Solve the resulting approximated GP problem using an interior point approach. Denote the found solutions as $\boldsymbol{\delta}_\mathrm{s}{}^{(\iota_2,\iota_3)}$ and $\Theta_2^{(\iota_2,\iota_3)}$.
5: Increment the iteration index as $\iota_2 = \iota_2 + 1$ and go back to Step 3 using $\boldsymbol{\delta}_\mathrm{s}$ and Θ_2 of step 4.
6: Terminate if $\|\boldsymbol{\delta}_\mathrm{s}{}^{(\iota_2,\iota_3)} - \boldsymbol{\delta}_\mathrm{s}{}^{(\iota_2-1,\iota_3)}\| \leq \epsilon$ or if $\Theta_2^{(\iota_2,\iota_3)} \geq \Theta_2^{(\iota_2-1,\iota_3)}$ and denote by $\boldsymbol{\delta}$ the final value.
7: Increment the iteration index as $\iota_3 = \iota_3 + 1$, set $\iota_2 = 1$, and $\boldsymbol{\delta}_s^{(\iota_2-1,\iota_3)} = \boldsymbol{\delta}$ and then go back to Step 2.
8: Terminate if $|R_{\mathrm{sum}}^{\mathrm{CoD}}[\iota_3] - R_{\mathrm{sum}}^{\mathrm{CoD}}[\iota_3 - 1]| \leq \epsilon$ or if $R_{\mathrm{sum}}^{\mathrm{CoD}}[\iota_3] \leq R_{\mathrm{sum}}^{\mathrm{CoD}}[\iota_3 - 1]$.

V. Numerical Examples

Throughout this section, we assume that the channel coefficients are modeled with independent and randomly generated variables, each generated according to a zero-mean Gaussian distribution whose variance is chosen according to the strength of the corresponding link. More specifically, the channel coefficient associated with the link from source A to the relay is modeled with a zero-mean Gaussian distribution with variance σ_{ar}^2. Similar assumptions and notations are used for the other links. Furthermore, we assume that, at every time instant, all the nodes know, or can estimate with high accuracy, the values taken by the channel coefficients at that time, i.e., full channel state information (CSI). Also, we set $P_a = 20$ dBW, $P_b = 20$ dBW, $P_r = 20$ dBW and $P = 20$ dBW.

We compare the sum-rate of the strategy in proposition 1 with the coding strategy of [5], and with standard CF and DF protocols. The sum-rate is evaluated under the constraint of *symmetric* rate by the sources. Let $R^{\text{CF}}_{\text{sum}}$, $R^{\text{DF}}_{\text{sum}}$, $R^{\text{CoF}}_{\text{sum}}$ denote the sum-rate obtained by using standard CF, standard DF and the coding strategy of [5] respectively.

The coding strategy of [5] and the coding strategy of Proposition 1 both decode two linear combinations of the transmitted codewords. However, these linear combinations are recovered differently in the two strategies. In [5], they are computed in a distributed manner while in Proposition 1, they are both computed locally at the destination.

A direct significance of recovering the linear combinations locally at the destination is that both recovered linear combinations utilize *all* the output available at the destination in a joint manner. The coding strategy of [5] is implemented such that the first linear combination is recovered at the destination using only the output received directly from the sources, and the second one is recovered at the relay and forwarded to the destination. The computation of the second one is limited by the weaker output between the output at the relay and the output at the destination (since this linear combination is decoded at the relay and has to be recovered at the destination).

For the example shown in Figure 2, we notice that; although the strategy of [5] ($R^{\text{CoF}}_{\text{sum}}$) performs better than the DF strategy, it performs worse than the CF strategy for given channel gains. Also, we notice that the strategy in Proposition 1 ($R^{\text{CoD}}_{\text{sum}}$) almost gives the same performance as the CF strategy. However, there are advantages for using the coding strategy of Proposition 1 instead of standard CF. In particular, while standard CF utilizes the rather complex joint typicality decoding, or maximum-likelihood decoding, the coding strategy of Proposition 1 utilizes linear decoding, and thus is more easily feasible.

We also should note that with a good initial point the iterative Algorithm A converges to an optimum value as given by standard CF with almost three iterations.

VI. Conclusion

In this paper, we study a two-source half-duplex multiaccess relay channel. Based on the compute-and-forward strategy, we develop and evaluate the performance of a coding strategy in which the destination does not decode the information messages directly from its output. Instead, it uses its output to first recover two linearly independent integer-valued combinations that relate the sources' codewords. It then decodes the messages using the two linear combinations. We discuss the design criteria and establish the allowed sum rate. Next, we investigate the problem of allocating the powers and the integer-valued coefficients of the recovered equations in a way to maximize the offered sum rate. We show that our coding strategy achieves as good as regular compress-and-forward; and has the advantage of utilizing feasible linear codes instead of random codes which are infeasible in practice.

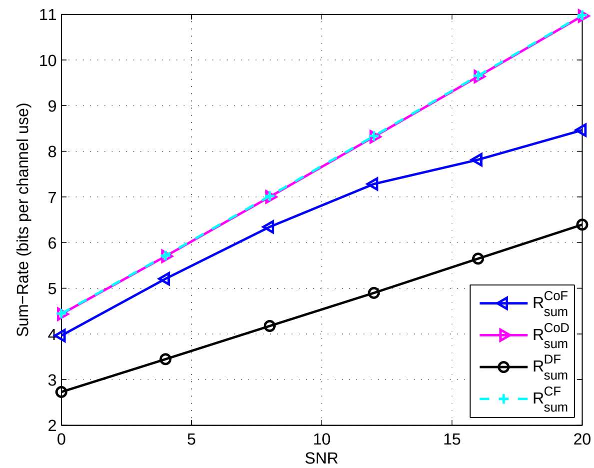

Figure 2. Achievable sum rates under symmetric rate constraint. Numerical values are $P = 20$ dBW, $\sigma^2_{\text{ar}} = \sigma^2_{\text{br}} = 14$ dBW, $\sigma^2_{\text{rd}} = 26$ dBW, and $\sigma^2_{\text{ad}} = \sigma^2_{\text{bd}} = 0$ dBW.

Acknowledgment

The authors would like to thank the European projects NEWCOM#, and SCOOP for the financial support and the IAP BESTCOM project funded by BELSPO. Also, the authors would like to thank Prof. Stephen Boyd and his team, Information Systems Laboratory, Stanford University, USA, and MOSEK Aps for partly helping in the software and programs used in the optimization part in this paper.

References

[1] B. Nazer and M. Gastpar. Compute-and-forward: Harnessing interference through structured codes. *IEEE Trans. Inf. Theory*, 57:6463–6486, 2011.

[2] M. P. Wilson, K. Narayanan, H.D. Pfister, and A. Sprintson. Joint physical layer coding and network coding for bidirectional relaying. *IEEE Trans. Inf. Theory*, 56:5641–5654, Nov. 2010.

[3] J. Zhan, B. Nazer, M. Gastpar, and U. Erez. Mimo compute-and-forward. In *IEEE Int. Symposium on Inf. Theory*, pages 2848–2852, June 2009.

[4] U. Erez and R. Zamir. Achieving $\frac{1}{2}\log(1+snr)$ on the awgn channel with lattice encoding and decoding. *IEEE Trans. Inf. Theory*, 50:2293–2314, Oct. 2004.

[5] M. El Soussi, A. Zaidi, and L. Vandendorpe. Iterative sum-rate optimization for multiple access relay channels with a compute-and-forward relay. In *IEEE International Conference on Communications*, June 2012.

[6] M. El Soussi, A. Zaidi, and L. Vandendorpe. Compute-and-forward on a multiaccess relay channel: Coding and sum-rate optimization. *IEEE Trans. on Wireless Communications*, submitted for publication, Sep. 2012. (Available on http://arxiv.org/abs/1209.4483).

[7] A. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Inf. Theory*, 22:1–10, 1976.

[8] M. Chiang, C. W. Tan, D. P. Palomar, D. O'Neill, and D. Julian. Power control by geometric programming. *IEEE Trans. Wireless Communications*, 6:2640–2651, July 2007.

[9] C. A. Floudas. Nonlinear and mixed integer optimization. In *Oxford University Press*, 1995.

[10] E. L. Lawler and D.E. Wood. Branch-and -bound methods: A survey. *Operations Research*, pages 699–719, 1966.