Multiple-Source Slepian-Wolf Coding Under a Linear Equation Correlation Model

Shizheng Li and Aditya Ramamoorthy, *Member, IEEE*

***Abstract*—In this work we present practical coding schemes for the problem of lossless distributed source coding for multiple sources. We consider two scenarios - the classical Slepian-Wolf case where there is no feedback from the terminal to the sources and a case where there is feedback from the terminal to the source encoders. The correlation model of interest is given by a system of linear equations, a generalization of the work of Stankovic et al. '06. We propose a transformation of correlation model and a way to determine proper decoding schedules, both of which are required to obtain the optimal sum rate. Our scheme allows us to exploit more correlations than those in the previous work. Simulation results show that the proposed coding scheme has lower sum rate than previous work in both scenarios.**

***Index Terms*—Slepian-Wolf, distributed source coding, rate adaptive codes, LDPC.**

I. Introduction

We consider the design of practical codes for lossless distributed source coding in this paper. Distributed source coding schemes are useful in applications such as sensor networks. In these networks there are typically a large number of resource-limited sensors that observe correlated sources and seek to convey information to a terminal that wants to recover their readings. Distributed source coding schemes allow the sensors to operate without any information exchange amongst themselves, yet allow them to exploit their correlation and transmit at a low overall sum rate to the terminal. The rate region for lossless distributed source coding is given by the Slepian-Wolf theorem [1] when there are two sources. The work of Cover [2] generalized the region to the case of multiple sources as follows. Suppose that the sources $X_1, X_2, \ldots, X_N$ are generating i.i.d. symbols according to the joint distribution $p(x_1, x_2, \ldots, x_N)$. Let R_i denote the rate for source X_i and S denote a nonempty subset of node indices: $S \subseteq \{1, 2, \ldots, N\}$. Let X_S denote the set of random variables $\{X_i : i \in S\}$. The rate region is given by

$$\sum_{i \in S} R_i \geq H(X_S | X_{S^c}) \text{ for all } S \neq \phi.$$

The connection between channel coding and Slepian-Wolf coding for two sources was investigated in [3]. Subsequently, a lot of research work has addressed problems (see [4] and its references) along this line. A majority of the work [4] considers two binary sources with a binary symmetric correlation model, i.e., the sources X_1 and X_2 are related by $X_1 + X_2 = E$, where E is a Bernoulli(p) random variable. Such correlation can be equivalently viewed as a virtual binary symmetric channel by using the idea of [3], when one considers code design where one source is available at the terminal. Let H be the parity check matrix of a channel code. The basic idea of [3] is that the decoder finds the difference (sum in the binary field) of the source values by a slightly modified channel decoding procedure. LDPC and Turbo codes [5], [6] have been designed for this case and they exhibit near-capacity performance under the BSC correlation model. Furthermore, a rate-adaptive LDPC code design for Slepian-Wolf coding was proposed in [7] and demonstrates good performance in the presence of a small amount of feedback from the decoder to the encoders.

The general problem with N (> 2) sources is well known to be challenging. Specifically, under a general correlation model, relating the Slepian-Wolf coding problem to a channel coding problem is no longer straightforward. One could achieve corner points of the Slepian-Wolf region by decoding the sources sequentially, using decoded sources as side information to help decode others. However, this forces the coding scheme to work only at corner points. While focusing on the case of two sources, the work of [8], [9], [10] discussed the applications of their approaches to the N sources case. The work of [8], [9] assumes that the correlations only exist among pairs of the sources, which can be viewed as special cases of the correlation models considered in this paper. In fact, for the correlation models considered in [8] the sum rate of our proposed scheme is as good as those presented in their work. In the work of [10], the authors described a N-machine decoding algorithm for the N-source case operating on a non-corner rate point. However, the complexity is increased and its performance is not studied in the paper. In our work, we use the standard belief propagation algorithm and provide the simulation results for N-source case. In [11], a restricted correlation model for the case of N sources is considered and a channel coding based scheme is proposed. More specifically, assuming that a capacity-achieving channel code is used, the proposed scheme in [11] achieves optimal sum rate when the source correlation is specified only by the modulo-2 sum of all sources. It requires all subsets of size $N-1$ and smaller to be independent. If there are more correlations except the total sum, the scheme ignores them, resulting in a suboptimal rate.

Main Contributions: In this paper, we propose a Slepian-Wolf coding scheme that works for more general correlation models. We consider a model where the correlation between the sources is given by sums of the subsets of sources, i.e., specified by a system of linear equations. Our proposed coding

Paper approved by Z. Xiong, the Editor for Distributed Coding and Processing of the IEEE Communications Society. Manuscript received March 10, 2011; revised August 25, 2011 and February 11, 2012.

This work was supported in part by NSF grant CCF-1018148.

S. Li was with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA. He is now with J. P. Morgan, NY, 10172 USA (e-mail: shizheng.li@jpmorgan.com).

A. Ramamoorthy is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: adityar@iastate.edu).

Digital Object Identifier 10.1109/TCOMM.2012.070912.110062

scheme is able to exploit these correlations in a judicious manner, assuming that a series of rate-adaptive codes [7] are used. Our scheme reduces the Slepian-Wolf coding problem to several channel coding problems in order to capture more correlations. In general, our scheme has a lower sum rate than the scheme in [11]. A key aspect of our work is the design of an appropriate decoding schedule that allows us to be strictly better than straightforward applications of the scheme in [11] in our setting.

In the Slepian-Wolf theorem, the term "rate" refers to the transmission rate per source symbol. Since in this paper we focus on the practical code design, it is occasionally easier to describe the rate as the number of transmitted bits per block of source symbols. For example, when coding over n binary source symbols, the rate will be expressed as $n-\ell$, where ℓ is a positive integer.

This paper is organized as follows. A brief review of the work in [11] is presented in Section II, together with more insights that motivates our solution. Section III presents a motivating example and the our proposed scheme is formally described in Section IV. Some simulation results are given in Section V and Section VI concludes the paper.

II. Background and Related Work

A. Coding scheme for sum correlation

First, we describe the scheme for N sources in [11]. Choose an (n,k) code as the main code with generator matrix G. Choose nonnegative integers $m_1,\ldots,m_N$ such that $\sum_{i=1}^N m_i = k$. Partition G according to $m_1,\ldots,m_N$ to obtain $G_1,\ldots,G_N$, such that $G=[G_1^T\ G_2^T\ \ldots\ G_N^T]^T$. Submatrix G_i corresponds to a parity check matrix H_i, i.e., $G_iH_i^T=0$. The i^{th} source transmits $H_i\mathbf{x}_i=\mathbf{s}_i$, so that the rate $R_i=n-m_i$. The sum rate is $Nn-k$. At the decoder, for each $i=1,\ldots,N$, first find a vector $\mathbf{t}_i$ in the coset with syndrome $\mathbf{s}_i$. Then, $\mathbf{x}_i+\mathbf{t}_i$ is a codeword of the code generated by G_i, i.e., $\mathbf{x}_i+\mathbf{t}_i=\mathbf{a}_iG_i$ for some vector $\mathbf{a}_i$. It is also a codeword of the main code, i.e., $\mathbf{x}_i+\mathbf{t}_i=[\mathbf{0}_{\sum_{j=1}^{i-1}m_j}\ \mathbf{a}_i\ \mathbf{0}_{\sum_{j=i+1}^{N}m_j}]G$, where $\mathbf{0}_\ell$ is a zero vector of length ℓ. Thus, $\sum_{i=1}^N(\mathbf{x}_i+\mathbf{t}_i)=[\mathbf{a}_1\ \mathbf{a}_2\ \ldots\ \mathbf{a}_N]G$. Viewing $\sum_{i=1}^N\mathbf{t}_i$ as the channel output and $\sum_{i=1}^N\mathbf{x}_i$ as the error, we perform standard channel decoding and obtain the channel input $[\mathbf{a}_1,\ldots,\mathbf{a}_N]G$, from which we can recover $\mathbf{a}_1,\ldots,\mathbf{a}_N$. Finally, the i^{th} source $\mathbf{x}_i=\mathbf{t}_i+\mathbf{a}_iG_i$. This scheme works well when the correlation is only given by the sum of the sources and the sum follows a Bernoulli(p) distribution, with p small enough.

B. A rate-equivalent scheme

Next, we show that given any choices of $R_1,\ldots,R_N$ in the rate region of the previously described scheme, we have an equivalent scheme that also works. Let $m_1,\ldots,m_N$ be non-negative integers such that $\sum_{i=1}^N m_i=k$. We explain the scheme from the parity check matrix perspective and this will motivate our proposed scheme. Choose an (n,k) code as the main code with parity check matrix H_{main} of size $(n-k)\times n$. We can simply choose the main code used in [11]. Stack k rows above the matrix H_{main} such that we have a n-by-n full rank matrix H. Partition the newly added k rows according to $m_1,\ldots,m_N$ to obtain $H^1,H^2,\ldots,H^N$, i.e., H^j corresponds to newly added rows numbered from $1+\sum_{k=1}^{j-1}m_k, 2+\sum_{k=1}^{j-1}m_k,\ldots,\sum_{k=1}^{j}m_k$. Let $[N]=\{1,2,\ldots,N\}$ and $[N]\backslash\{i\}=\{1,2,\ldots,i-1,i+1,\ldots,N\}$. To construct the parity check matrix H_i of the subcode for source i, we stack the matrices H_{main} and $H^j: j\in[N]\backslash\{i\}$ together. In other words, H_i is obtained by removing H^i from H. It has $n-k+\sum_{j\in[N]\backslash\{i\}}m_j=n-m_i$ rows. Transmit $H_i\mathbf{x}_i=\mathbf{s}_i$ at each source so that $R_i=n-m_i$. Note that for all i, H_i has H_{main} part in common. Denote the last $(n-k)$ entries of $\mathbf{s}_i$ as $\mathbf{s}_i^{(n-k)}$. Then $\sum_{i=1}^N\mathbf{s}_i^{(n-k)}=H_{main}(\sum_{i=1}^N\mathbf{x}_i)$. By standard channel decoding, we can recover $\sum_{i=1}^N\mathbf{x}_i$ as long as the sum has low enough Hamming weight. Note that H^i appears in every parity check matrix $H_j: j\in[N]\backslash\{i\}$ but not in H_i. From the syndromes $H_j\mathbf{x}_j: j\in[N]\backslash\{i\}$, we know $H^i\mathbf{x}_j$ for all $j\in[N]\backslash\{i\}$ because the latter is a subvector of the former, which allows us to compute $H^i\mathbf{x}_i=H^i(\sum_{j=1}^N\mathbf{x}_i)+\sum_{j\in[N]\backslash\{i\}}H^i\mathbf{x}_j$ since we have already recovered $\sum_{j=1}^N\mathbf{x}_i$. Now, we know both $H^i\mathbf{x}_i$ and $H_i\mathbf{x}_i$, putting them together we know $H\mathbf{x}_i$ and since H is invertible, $\mathbf{x}_i$ can be recovered. This equivalent scheme reveals that in essence, only the correlation given by sum of all sources is exploited in the coding scheme. Other than that, the sources are recovered by matrix inversion, even if there are other form of correlations. Indeed, it can be shown that the scheme in [11] is optimal only when all subsets of sources with size $N-1$ and smaller are independent.

C. Rate adaptive Slepian-Wolf codes

A set of rate adaptive Slepian-Wolf codes is defined to be a set of L linear block codes whose parity check matrices are given by $\{H_1,H_2,\ldots,H_L\}$ with dimensions $n-k_1,n-k_2,\ldots,n-k_L$, where $k_1\geq k_2\ldots\geq k_L$ are such that H_i is a submatrix of H_{i+1} for $i\in[L]$. Using such a set of codes to perform Slepian-Wolf coding, the syndromes $\mathbf{s}_i=H_i\mathbf{e}$ are such that $\mathbf{s}_i$ is a subvector of $\mathbf{s}_{i+1}$. Such codes are very useful in the feedback setting. If the terminal is unable to decode at a certain rate R_i corresponding to H_i, it can indicate decoding failure to the source encoder that can then communicate additional syndrome bits, such that the decoder can obtain the syndrome corresponding to H_{i+1} and attempt decoding again.

Good rate adaptive codes based on LDPC codes for binary sources were presented in [7]. The simulations show that these codes perform very well when there is a feedback channel from the decoder to the encoders that indicates whether the decoding is successful[1]. If the decoding fails, the encoder sends more bits until the decoding is successful. The average minimum required rates are very close to Slepian-Wolf bound [7]. On the other hand, our simulations show that for these codes if there is no feedback and the decoder attempts decoding only once, the performance is not very good.

Our proposed scheme uses rate adaptive codes. In our simulations, we consider two scenarios, one is the classical Slepian-Wolf scenario and in the other scenario there is a

[1]Successful decoding only indicates that the decoder is able to make a decision and does not imply that the decision is correct. For instance, for an LDPC code, successful decoding would imply that the iterative decoding procedure converged to a valid codeword.

feedback from the decoder to the encoder. We shall see our proposed scheme works very close to the Slepian-Wolf bound in the feedback scenario. Even though the code performance is not as good under classical Slepian-Wolf scenario, our scheme still outperforms the work of [11], where capacity-achieving codes are used, because we capture more of the model correlations.

III. A Motivating Example

Consider an example as follows. Suppose that four binary sources X_1, X_2, X_3 and X_4 are given as follows. $X_1 = Y_1, X_2 = Y_1 + E_1, X_3 = Y_1 + E_2, X_4 = Y_1 + E_1 + E_2 + E_3$, where Y_1, E_1, E_2 and E_3 are independent. The source Y_1 has entropy 1, while $H(E_i) < 1$ for $i = 1, \ldots, 3$. Thus, X_2 and X_3 can be viewed as noisy versions of X_1 with different noise levels and their correlation with X_1 can be modeled as a BSC. Source X_4 is an even noisier version of X_1. Equivalently,

$$X_1 + X_2 = E_1, \tag{1}$$

$$X_1 + X_3 = E_2, \tag{2}$$

$$X_1 + X_2 + X_3 + X_4 = E_3. \tag{3}$$

Let $k_i \leq n(1-H(E_i))$ be such that the channel code with rate k_i/n is able to correct the channel error E_i. For a capacity-approaching code, k_i should be close to $n(1 - H(E_i))$. The scheme of [11] captures the last equation and the sum rate is $Nn - k = 4n - k_3$ bits per block.

Suppose that $k_1 \geq k_2 \geq k_3$ and we use a set of rate adaptive codes with rates $k_1/n, k_2/n, k_3/n$ and parity check matrices H_1, H_2, H_3 respectively. According to the definition, H_1 is a submatrix of H_2, and H_2 is a submatrix of H_3. In the first stage, source 1 transmits $H_3\mathbf{x}_1$, which contains $H_1\mathbf{x}_1, H_2\mathbf{x}_1$, and its rate is $n - k_3$. Sources 2, 3 and 4 transmit $H_1\mathbf{x}_2, H_2\mathbf{x}_3, H_3\mathbf{x}_4$ respectively and their rates are $n - k_1, n - k_2, n - k_3$. The decoding of $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ proceeds as follows.

Step 1. From (1), $\mathbf{x}_1 + \mathbf{x}_2 = \mathbf{e}_1$, the terminal knows $H_1\mathbf{x}_1$ and $H_1\mathbf{x}_2$, both of which have length $n - k_1$. It computes $H_1\mathbf{x}_1 + H_1\mathbf{x}_2 = H_1\mathbf{e}_1$ and recovers $\mathbf{e}_1$.

Step 2. From (2), $\mathbf{x}_1 + \mathbf{x}_3 = \mathbf{e}_2$, the terminal knows $H_2\mathbf{x}_1$ and $H_2\mathbf{x}_3$, and recovers $\mathbf{e}_2$.

Step 3. From (1)-(3) it can be observed that

$$\mathbf{x}_1 + \mathbf{x}_4 = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3. \tag{4}$$

The terminal knows the syndromes $H_3\mathbf{x}_1, H_3\mathbf{x}_4$ from the sources, and computes $H_3\mathbf{e}_1, H_3\mathbf{e}_2$ since $\mathbf{e}_1$ and $\mathbf{e}_2$ are both known from the first two steps. Adding these together we get $H_3\mathbf{e}_3$, then we can recover $\mathbf{e}_3$ by syndrome decoding. If we do not add equations (1) and (2) to (3), we need the rate of all the sources to be $n-k_3$ in order to obtain $H_3\mathbf{e}_3$ in the last equation, which is unnecessary for recovering the errors $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$. In general, given a linear equation correlation model, an appropriate transformation needs to be performed in order to obtain a low sum rate. We shall discuss a systematic way to do this in Section IV.

In the second stage, we need to transmit some more encodings such that all the sources can be recovered. Note that if we can recover $\mathbf{x}_1$ we can recover all other sources since we have recovered $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$. We can transmit a linear combination of $\mathbf{x}_1$: $H'\mathbf{x}_1$ (of length k_3) from the source X_1 and such that $[H'^T\ H_3^T]$ is invertible. Alternatively, we can partition the rows of H' into H'_1, H'_2, H'_3, H'_4 and let source X_i transmit $H'_i\mathbf{x}_i$, with rates $a_i, i = 1, \ldots, 4$ such that $\sum_{i=1}^{4} a_i = k_3$. Note that $H'_i\mathbf{x}_1, i = 2, 3, 4$ can be found as follows. $H'_2\mathbf{x}_1 = H'_2\mathbf{x}_2 + H'_2\mathbf{e}_1$, $H'_3\mathbf{x}_1 = H'_3\mathbf{x}_3 + H'_3\mathbf{e}_2$,$H'_4\mathbf{x}_1 = H'_4\mathbf{x}_4 + H'_4\mathbf{e}_1 + H'_4\mathbf{e}_2 + H'_4\mathbf{e}_3$. The last equation is from (4). Thus, $H'\mathbf{x}_1$ can be obtained from the encodings of other sources. This gives us rate flexibility since we do not have to transmit $\mathbf{x}_1$ at rate n. The rates of the sources in this scheme are $R_1 = n-k_3+a_1$, $R_2 = n-k_1+a_2$, $R_3 = n - k_2 + a_3$,$R_4 = n - k_3 + a_4$, $a_1 + a_2 + a_3 + a_4 = k_3$, $a_i \geq 0, i = 1, 2, 3, 4$.

In other words, the rate region of this scheme in terms of bits per block can be expressed by

$$\left\{ \begin{array}{c} R_1, R_4 \geq n - k_3, R_2 \geq n - k_1, R_3 \geq n - k_2, \\ R_1 + R_2 + R_3 + R_4 \geq 4n - k_1 - k_2 - k_3 \end{array} \right\} \tag{5}$$

Thus, the sum rate of the proposed approach is $4n-k_1-k_2-k_3$ bits per block.

Remark: In this example, by applying the scheme in [11] three times to each equation and using previously decoded sources as side information, one can also achieve a sum rate of $4n-k_1-k_2-k_3$. Specifically, applying the scheme in [11] to (1), $\mathbf{x}_1, \mathbf{x}_2$ can be recovered using $2n - k_1$ symbols. Then, $\mathbf{x}_1$ is used as side information and from (2), $\mathbf{x}_3$ can be recovered using $n - k_2$ additional symbols. Then, using $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ as side information, $\mathbf{x}_4$ can be recovered using $n - k_3$ additional symbols from (3).

However, consider the following example: $X_1+X_2+X_4 = E_1$, $X_2+X_3+X_4 = E_2$, $X_1+X_3+X_4 = E_3$. If we apply the scheme in [11] to the first equation, we need $3n-k_1$ symbols to recover $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4$. Then, from the second equation, we need $n-k_2$ additional symbols to recover $\mathbf{x}_3$. The sum rate is $4n-k_1-k_2$. Even if we start with different equations, the best sum rate is $4n - \max\{k_1 + k_2, k_1 + k_3, k_2 + k_3\}$. In contrast, as discussed below our proposed scheme can achieve a sum rate of $4n - k_1 - k_2 - k_3$.

IV. Distributed Source Coding for Linear Correlations

In this section, we propose a practical coding scheme for the linear correlation model considered above. In particular, we design appropriate decoding schedules and a transformation of the system of linear equations such that we can achieve low sum rate assuming the existence of good rate adaptive Slepian Wolf codes. In practice, if we use moderate block length codes, there will be a gap between the joint entropy and the sum transmission rate (see Section V). Denote the index set $\{1, 2, \ldots, L\}$ by $[L]$ for some integer L. Let $S_l, l \in [L]$ be subsets of the sources. The correlation is given by a set of L linear equations $\sum_{i \in S_l} X_i = E_l, l \in [L]$ that are assumed to be linearly independent. The E_is are assumed to be statistically independent. Let k_i/n be the channel code rate that is needed for correcting error E_i.

Our scheme works as follows. We find a set of L linearly independent columns in the coefficient matrix of the system of equations and denote the index set by A. This can always be done because the equations are linearly independent. Denote

the index set $[N]\backslash A$ by B. Note that A is also the index set for the sources that corresponding to the L columns indexed by A. Similarly, B is also an index set for the sources. Without loss of generality, assume that the equations are ordered such that $k_1 \geq k_2 \geq \ldots \geq k_L$. It is important to start with the equations in this order as scheduling the decoding procedure based on such an ordered form gives the best rate performance. As we have seen before in the example in Section III , transforming the system of linear equations properly helps improve the rate performance. In the discussion below we present a decoding scheme that achieves a sum rate of $Nn - \sum_{l=1}^{L} k_l$ bits per block.

In the first stage, we recover the errors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_L$. The rate at this step for the i^{th} source is denoted by P_i. We first discuss the assignments of the rates P_i.

1) Rate allocation: As we will see later, this step also provides a proper decoding procedure (scheduling) for the first stage (recovering the errors).

- For sources in set B, assign $P_i = n - \min_{l\in[L]} k_l = n - k_L, \forall i \in B$.
- The assignment of rates $P_i, i \in A$ is described as follows. Note that the set $A \cap S_l$ indicates the set of sources in A that participate in the l^{th} equation. Let J denote an index set. Let u be the iteration index. At the beginning of each iteration, J is the set of sources in A that has been assigned rate P_i.
  Initialization. $J = \emptyset$; $P_i = 0, \forall i \in A$; $u = 1$.
  (1) Pick a source $j_u \in A \cap S_u$, $J \leftarrow J \cup \{j_u\}$. Assign $P_{j_u} = n - k_u$.
  (2) Add the u^{th} equation to the l^{th} equation for every l such that $l > u$ and $j_u \in A \cap S_l$, i.e., the equations in which the source X_{j_u} appears. Replace the l^{th} equation by this new equation and update S_l accordingly.
  (3) $u \leftarrow u + 1$, if $u < L$, go to (1), otherwise, STOP.

The idea is similar to Gaussian elimination but the main difference is that we do not switch the order of the equations. Gaussian elimination returns a matrix in row echelon form, while this algorithm does not.

Claim: The algorithm assigns rates for each source and the rate allocation is such that $P_i \geq n - k_l, \forall i \in S_l$ for $l = 1, 2, \ldots, L$, where S_l is induced by the linear equations after the transformation performed in the algorithm.

Proof: It is easy to see for $\forall i \in B$, $P_i \geq n - k_l$, $\forall l \in [L]$. For the allocation of $P_i, \forall i \in A$, at each step u, we eliminate the source X_{j_u} in the equations $u+1, \ldots, L$. Thus, for each $1 \leq u \leq L$, at the beginning of step u, $J \cap A \cap S_u = \emptyset$. Therefore, at step u, the sources that are already in J will not be picked again. In addition at each step we can always find $j_u \in A \cap S_u$. This is because the columns indexed by A have full rank, i.e., an all zero row will not appear in the L-by-L submatrix. At the end of the above procedure, $J = A$ and the rate assignment is $P_{j_u} = n - k_u, \forall u \in [L]$. Note that since we start with the equations in an order such that $k_1 \geq k_2 \geq, \ldots, \geq k_L$, we have $P_{j_1} \leq P_{j_2} \leq \ldots \leq P_{j_L}$. In addition, note that for each equation l, $A \cap S_l \cap \{j_1, j_2, \ldots, j_{l-1}\} = \emptyset$, i.e., the sources that have been assigned a rate (lower than $n - k_l$) do not appear in equation l, and the sources in equation l other than j_l will be assign a rate higher than $n - k_l$ in later iterations. Thus, we conclude that for sources in $A \cap S_l$, $P_i \geq n - k_l, \forall i \in A \cap S_l$. ■

$$\begin{bmatrix} 1\,1\,0\,1 \\ 0\,1\,1\,1 \\ 1\,0\,1\,1 \end{bmatrix} \xrightarrow{\text{Iteration 1}} \begin{bmatrix} 1\,1\,0\,1 \\ 1\,0\,1\,0 \\ 1\,0\,1\,1 \end{bmatrix} \xrightarrow{\text{Iteration 2}} \begin{bmatrix} 1\,1\,0\,1 \\ 1\,0\,1\,0 \\ 0\,0\,0\,1 \end{bmatrix}$$

Fig. 1. The evolution of the coefficient matrix for the system of equations $X_1 + X_2 + X_4 = E_1$, $X_2 + X_3 + X_4 = E_2$, $X_1 + X_3 + X_4 = E_3$. In iteration 1, $j_1 = 2, J = \{2\}$ and $P_2 = n - k_1$. In iteration 2, $j_2 = 3, J = \{2, 3\}$ and $P_3 = n - k_2$. In iteration 3, $j_3 = 4, J = \{2, 3, 4\}$ and $P_4 = n - k_3$, the form of the equations does not change at the last iteration.

The sum of P_i values is

$$\begin{aligned} \sum_{i\in B} P_i + \sum_{i\in A} P_i &= (N-L)(n-k_L) + Ln - \textstyle\sum_{l=1}^{L} k_l \\ &= Nn - (N-L)k_L - \textstyle\sum_{l=1}^{L} k_l. \end{aligned} \quad (6)$$

The choice of j_u at each step is not unique so the rate allocation is not unique.

Example. Consider the example in the Remark of Section III. The evolution of the coefficient matrix during the iterations is shown in Fig. 1. The equations after transformation is

$$\begin{bmatrix} 1\,1\,0\,1 \\ 1\,0\,1\,0 \\ 0\,0\,0\,1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 + E_1 \\ E_3 + E_1 + E_2 \end{bmatrix}. \quad (7)$$

2) Code construction and decoding: Choose a set of rate adaptive codes that can adapt the rates among $\{k_1/n, k_2/n, \ldots, k_L/n\}$. The parity check matrices are $H_1, H_2, \ldots, H_L$ and H_i is a submatrix of H_{i+1} for $i \in [L-1]$. For $X_i : i \in B$, transmit $H_L \mathbf{x}_i$. For each $X_i : i \in A$, transmit $H_j \mathbf{x}_i$ if $P_i = n - k_j$. We recover errors according to the ascending order: $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_L$ from equation 1 to L, which are updated during the rate allocation algorithm. Note that $P_i \geq n - k_l$ for all $i \in S_l$. This means that the decoder can obtain $H_l \mathbf{x}_i, \forall i \in S_l$ from the syndromes it receives. For the sources such that $P_i > n - k_l$, $H_l \mathbf{x}_i$ is a portion of the received syndrome $H_{l'} \mathbf{x}_i$ for some $l' > l$. Note that the right hand side of the equation may become $\mathbf{e}_l$ plus some $\mathbf{e}_u$ terms for $u < l$. However, those additional error terms are recovered earlier and we can compute $H_l \mathbf{e}_u$ for those u values. The effective error is still $\mathbf{e}_l$ and we can compute $H_l \mathbf{e}_l$ and recover $\mathbf{e}_l$.

Example (Continued.) In the example above, suppose the parity check matrices are $\{H_1, H_2, H_3\}$. Source X_1 transmits $H_3 \mathbf{x}_1$, X_2 transmits $H_1 \mathbf{x}_2$, X_3 transmits $H_2 \mathbf{x}_3$ and X_4 transmits $H_3 \mathbf{x}_4$ so that their rates are $n - k_3, n - k_1, n - k_2, n - k_3$ respectively.

We look at the equations after the rate allocation, i.e., equation (7). Start with the first equation. Note that $H_1 \mathbf{x}_1$ is a subvector of $H_3 \mathbf{x}_1$ and $H_1 \mathbf{x}_4$ is a subvector of $H_3 \mathbf{x}_4$. So the decoder knows $H_1(\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_4) = H_1 \mathbf{e}_1$ and it is able to recover $\mathbf{e}_1$. In the second equation, note that $H_2 \mathbf{x}_1$ is a subvector of $H_3 \mathbf{x}_1$ and X_3 transmits $H_2 \mathbf{x}_3$, the decoder knows $H_2(\mathbf{x}_1 + \mathbf{x}_3)$ and $H_2 \mathbf{e}_1$ since $\mathbf{e}_1$ was recovered. Thus, it finds $H_2 \mathbf{e}_2$ and recovers $\mathbf{e}_2$. In the third equation, note that X_4 transmits $H_3 \mathbf{x}_4$, and the decoder knows $H_3(\mathbf{e}_1 + \mathbf{e}_2)$ so it knows $H_3 \mathbf{e}_3$ and can recover $\mathbf{e}_3$. Therefore, $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$ can be recovered. ■

In the second stage, we transmit more encodings such that all sources can be recovered. The rate of additional encodings at the second stage for source i is denoted by Q_i. Thus, the transmission rate for source i is $R_i = P_i + Q_i$. If $\mathbf{x}_i, \forall i \in B$ are recovered, $\mathbf{x}_i, \forall i \in A$ can be recovered by matrix inversion. The simplest way is to transmit k_L additional encodings $H'\mathbf{x}_i$ for each $\mathbf{x}_i, \forall i \in B$ and such that $[H'^T H_L^T]$ has full rank. This is equivalent to transmitting $X_i, i \in B$ uncoded. In this case, $Q_i = k_L, \forall i \in B, Q_i = 0, \forall i \in A$. Using the expression of $\sum_{i\in[N]} P_i$, from (6), the sum rate of our scheme in terms of bits per block is

$$\sum_{i\in[N]} R_i = Nn - \sum_{l=1}^{L} k_l. \tag{8}$$

We could also partition the rows of H' and transmit the encodings of other sources such that $H'\mathbf{x}_i, i \in B$ can be recovered based on the errors that we have already found. By doing this, the rates $R_i, i \in B$ do not have to be n. This is similar to the scheme in Section II-B. To obtain a representation of $\mathbf{x}_i$, $H'\mathbf{x}_i$, one can obtain $H'\mathbf{x}_j$ for other sources X_j that participate in the same equation with X_i. Since the right hand side of each equation is recovered in the first stage, $H'\mathbf{x}_i$ can be computed. In Section II-B, only one equation is used, while here we have L equations. The exact rate region depends on the form of the system of equations. Note that the choice of A, B may not be unique and different choices of A, B give different rate assignments.

The optimal sum rate is the joint entropy $H(X_{[N]}) = H(X_B) + H(X_A|X_B) = H(X_B) + H(E_1, E_2, \ldots, E_L|X_B)$. If there exists a choice of A and B such that the columns indexed by A are independent, the sources in the set B are uniformly distributed, and the sources in the set $\{X_B, E_1, \ldots, E_L\}$ are statistically independent, then $H(X_B) = (N - L)$, $H(E_1, E_2, \ldots, E_L|X_B) = \sum_{i=1}^{L} H(E_i) \approx \sum_{i=1}^{L}(1 - k_i/n)$. Thus, if the above assumptions hold and the channel codes are capacity achieving, the sum rate of the proposed scheme achieves the optimum. The practical performance of our scheme is shown in Section V.

If the set of random variables $\{E_i\}_{i=1}^{L}$ are dependent, our scheme will still work. One can use previously decoded $\mathbf{e}_i$ vectors to help decode $\mathbf{e}_j, j > i$. The input to the LDPC decoder will need to be suitably modified. However, the correlations among E_is could be arbitrary and the performance of the LDPC codes cannot be guaranteed to be very good. Note that one special case of our scheme is that when $L = N$, i.e., when the correlation is given by a full rank system of linear equations. In this case if $E_1, E_2, \ldots, E_L$ are independent and the codes are capacity achieving, our scheme achieves optimal sum rate.

V. Simulation Results

We present Monte Carlo simulation results in this section. Note that we only need to determine the rates for the recovery of the error terms. This stage uses error control codes and their performance are evaluated by simulation. The recovery of the actual sources $\mathbf{x}_i$ is done by matrix inversion and vector addition operations and these steps are guaranteed to be correct as long as the error terms are recovered correctly. The rate-adaptive codes designed in [7] are used in our simulations.

TABLE I
The Example in Section III, configuration 1

i	$p(E_i = 1)$	$H(E_i)$	Tx Rate (classical)	Tx Rate (feedback)
1	0.11	0.50	0.77	0.59
2	0.12	0.53	0.82	0.62
3	0.13	0.56	0.89	0.65
Sum rate (classical SWC): $4 - (k_1 + k_2 + k_3)/n = 3.48$				
Average sum rate (feedback): 2.86				
Joint Entropy: 2.59				

The irregular LDPC code has length 6336 and variable node degrees ranging from 2 to 21. We consider two scenarios, the classical Slepian-Wolf coding scenario and the feedback scenario. In [7], the rate-adaptive codes are designed by starting from a parity check matrix with n rows and removing several rows each time to achieve each rate, where n is the code length. We use the same selections of the rows (termed puncturing pattern of syndromes in [7]) as in [7] for our two cases.

In the classical Slepian-Wolf coding scenario, we shall find the lowest transmission rate, i.e., the largest k_i values, that result in near- error-free recovery, measured as a frame error rate $< 10^{-3}$. We say that one frame is in error if any of $\mathbf{e}_i, i = 1, \ldots, L$ is not decoded correctly. In order to obtain FER $< 10^{-3}$ for the whole coding scheme, we roughly need the individual FER for recovering each E_i to be $10^{-3}/L$, where L is the number of equations.

In the feedback scenario, when decoding a error sequence, if a decoding attempt fails, the decoder will request additional syndrome bits from all sources that participate in the equation (after transformation) until the decoding is successful. More specifically, the rate adaptive codes designed in [7] contain codes with rates $\{1/66, 2/66, \ldots, 64/66\}$. Each time when we need to transmit additional syndrome bits, we decrease the code rate by $1/66$. It is possible that for one particular error sequence, some of the sources have transmission rates higher than the required transmission rate for this error sequence. For example, the designed rate of source X_1 is probably high enough for decoding E_1 in our example equation (1). In this case, we do not need to increase the transmission rates for those sources. The simulation results presented below show the average minimum required transmission rate for each source and the average minimum required sum rate for recovering all the sources.

We consider the example in equations (1)-(3) in Section III. Two configurations of probability distribution are used and the results are presented in Table I and Table II. The cases of classical SWC and feedback are both presented. The transmission rate for source i is $1 - k_i/n$. The gaps to joint entropy are compared in Fig. 2. Clearly, the rate-adaptive codes perform better under feedback scenario. In Fig. 2, the theoretical gap means the gap between the transmission rate and the joint entropy when a capacity-achieving code is used, i.e., $k_i/n = 1 - H(E_i)$. For the scheme in [11], when a capacity-achieving code is used, the sum rate will be $4 - (1 - H(E_3))$.

Note that the rate-adaptive codes of [7], do not have very good performance for the classical Slepian-Wolf scenario. Nevertheless, if we use them along with our rate allocation and decoding schedule algorithms and compare them to the scheme of [11] using a capacity-achieving code, our sum

TABLE II
THE EXAMPLE IN SECTION III, CONFIGURATION 2

i	$p(E_i = 1)$	$H(E_i)$	Tx Rate (classical)	Tx Rate (feedback)
1	0.05	0.29	0.56	0.34
2	0.06	0.33	0.61	0.39
3	0.15	0.61	0.89	0.71
Sum rate (classical SWC): $4 - (k_1 + k_2 + k_3)/n = 3.06$				
Average sum rate (feedback): 2.44				
Joint Entropy: 2.22				

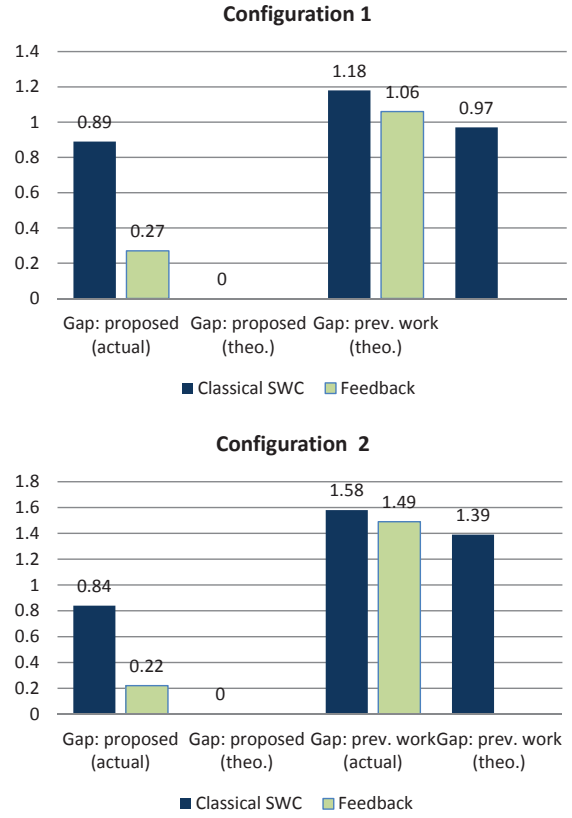

Fig. 2. The comparison of the gaps between sum rate and joint entropy. The actual sum rate is the sum rate observed in the simulations under two different scenarios. The theoretical sum rate is obtained under the assumption that capacity-achieving LDPC codes are used.

rate is still better. This is because our scheme captures more correlations that exist in the joint distribution.

We also considered a full rank system of five equations that contains five sources. This is shown in Fig. 3 and the form after transformation is also given. The corresponding simulation results are presented in Table III.

VI. CONCLUSION AND FUTURE WORK

In this work we considered the design of practical lossless distributed source codes. The correlation model of interest is given by a system of linear equations, a generalization of the models considered in prior work. We propose a transformation of the correlation model and a way to find the proper decoding schedule such that the optimal sum rate can be achieved. More correlations are captured by our scheme as compared to prior work and the simulation results demonstrate the better compression efficiency of our scheme. Our work uses the rate-adaptive LDPC codes designed for asymmetric Slepian-Wolf coding of two sources in [7].

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Fig. 3. The transformation of the coefficient matrix of a full rank system of linear equations with five equations and five sources. Note that $j_1 = 1, j_2 = 2, j_3 = 3, j_4 = 5, j_5 = 4$ and $P_1 = n - k_1, P_2 = n - k_2, P_3 = n - k_3, P_4 = n - k_5, P_5 = n - k_4$.

TABLE III
THE CONFIGURATION AND SIMULATION RESULTS FOR FIVE CORRELATED SOURCES.

i	$p(E_i = 1)$	$H(E_i)$	Tx Rate (classical)	Tx Rate (feedback)
1	0.05	0.29	0.53	0.35
2	0.06	0.33	0.74	0.38
3	0.07	0.37	0.73	0.42
4	0.08	0.40	0.89	0.47
5	0.09	0.44	0.80	0.52
Sum rate (classical SWC): 3.69				
Average sum rate (feedback): 2.15				
Joint Entropy: 1.83				

Simulation results indicate that the performance of these codes is quite good in the presence of feedback. However, for the classical Slepian-Wolf scenario, there is a definite gap with respect to the sum rate bound. Thus, the design of capacity-achieving rate-adaptive Slepian-Wolf codes is an interesting research direction.

REFERENCES

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[2] T. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources (corresp.)," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 226–228, Mar. 1975.

[3] A. Wyner, "Recent results in Shannon theory," *IEEE Trans. Inf. Theory*, vol. 20, no. 1, pp. 2–10, Jan. 1974.

[4] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, Sep. 2004.

[5] A. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 10, pp. 440–442, Oct. 2002.

[6] J. Garcia-Frias, Y. Zhao, and W. Zhong, "Turbo-like codes for transmission of correlated sources over noisy channels," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 58–66, Sep. 2007.

[7] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *Signal Process.*, vol. 86, no. 11, pp. 3123–3130, Nov. 2006.

[8] N. Gehrig and P. Dragotti, "Symmetric and asymmetric Slepian-Wolf codes with systematic and nonsystematic linear codes," *IEEE Commun. Lett.*, vol. 9, no. 1, pp. 61–63, Jan. 2005.

[9] M. Sartipi and F. Fekri, "Distributed source coding using short to moderate length rate-compatible LDPC codes: the entire Slepian-Wolf rate region," *IEEE Trans. Commun.*, vol. 56, no. 3, pp. 400–411, Mar. 2008.

[10] D. Schonberg, K. Ramchandran, and S. Pradhan, "Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources," in *Proc. 2004 Data Compression Conf.*, pp. 292–301.

[11] V. Stankovic, A. Liveris, Z. Xiong, and C. Georghiades, "On code design for the Slepian-Wolf problem and lossless multiterminal networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1495–1507, Apr. 2006.