A Practical and Optimal Symmetric Slepian-Wolf Compression Strategy Using Syndrome Formers and Inverse Syndrome Formers

Peiyu Tan and Jing Li (Tiffany)
Electrical and Computer Engineering Dept, Lehigh University, Bethlehem, 18015
{pet3,jingli}@ece.lehigh.edu

Abstract

This paper considers symmetric Slepian-Wolf coding of two binary memoryless sources. A constructive coding approach, termed the *symmetric SF-ISF framework* (SSIF), is proposed. The idea is to first recover the difference pattern between the sources using the syndrome former (SF) and the inverse syndrome former (ISF) of the linear channel code, and to subsequently recover individual source sequences through syndrome former partitioning. The proposed framework can be efficiently applied to a general linear channel code, incurs no rate loss when converting the channel code to the Slepian-Wolf code, and can achieve an arbitrary point in the Slepian-Wolf rate region. The feasibility and optimality of the framework is rigorously proven using properties of linear codes and cosets, and further demonstrated using low-density parity-check (LDPC) codes.

I. Introduction

The problem of lossless distributed source coding (DSC) rooted back to the seminal work of Slepian and Wolf in 1973. The famous Slepian-Wolf (SW) theorem [1] states that theoretically there is no loss in rate to compress two correlated sources using separate encoding (as compared to joint encoding), provided that the decoding is done jointly and that the source correlation is available to both the encoder(s) and the decoder. The original proof of the achievability of the SW bound is asymptotic and non-constructive. Considerable research effort has since been attempted in terms of practical coding solutions, but it was not until years later that a major breakthrough was made by exploiting the technology of *algebraic code binning* [2]. Since then, various sophisticated formulations using powerful channel codes have been explored for lossless DSC with binary memoryless sources, including low density parity check (LDPC) codes (e.g. [3][4]) and convolutional/turbo codes (e.g [5]-[9]).

The majority of the work in literature targets achieving the corner points of the SW bound, formally known as the asymmetric DSC problem. Practical applications, however, may require the separate encoders to retain comparable compression/transmission rates and to share an equal computational complexity. Although time sharing can be applied on corner points to achieve an arbitrary point in the SW bound (i.e. an arbitrary rate/load allocation), the practice may be inconvenient, expensive, or unfeasible in certain scenarios. For example, inter-source synchronization, as required to perform time sharing, is hard to implement between two non-communicating encoders. For this reason, symmetric compression that can directly realize a rate and load balance is desirable.

The practicality and efficiency of achieving the entire SW rate region using linear channel codes [9]-[13] is a worthy research area. Early work on symmetric SW coding focuses on source splitting. For example, [9] proposes to encode the two (highly) correlated sources X and Y using separate turbo encoders, each followed by a different and complimentary puncturing pattern. The joint decoder iterates between the two turbo decoders in a way similar

This work has been supported by the National Science Foundation under grant CCF-0430634 and by a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development, through the Pennsylvania Infrastructure Technology Alliance (PITA).

to decoding a four-branch parallelly concatenated code. In additional to source splitting, code partitioning is also being actively exploited. The authors of [10] propose to horizontally split the generator matrix of the channel code into two sub generator matrices, each assuming a different compression rate in accordance to the number of rows retained in the respective sub matrix. This approach is shown to be applicable to a general linear block code, but does not warrant lossless conversion; that is, the resulting SW code will likely perform worse as a source code than its original capability as a channel code. When the channel code is a systematic code, [11] provides a different and more efficient code partitioning strategy that exploits the unique properties of a systematic code. Although no proof on the achievability of the theoretical limit is provided, simulations using (systematic) irregular repeat accumulate (IRA) codes demonstrate compression rates that are very close to the SW bound [11]. Besides these excellent results on the general *probabilistic* source correlation (where sources X and Y differ with a probability of $p = \Pr(X \neq Y)$ at any instantaneous output), *constrained* source correlation (where sources X and Y differ by no more than t bits within a block of size n) has also been the subject of interest. t-error correcting linear channel codes and particularly Hamming codes (for $t = 1$) have been exploited for symmetric (and asymmetric) compression in this latter scenario [12].

The main contribution of this paper is the development of a constructive framework for symmetric (and asymmetric) SW coding of binary memoryless sources. The proposed framework is *general*, since it can be readily applied on any linear channel code *and*, for a given channel code, can achieve any SW rate pair within the code capacity. Although not shown here, the framework is also general in the sense that it can be used for both the probabilistic source correlation and the constraint source correlation [13]. Additionally, the framework is *optimal*, which incurs no rate loss during the code conversion. Put another way, how close the resulting SW code gets to the theoretic limit solely depends on how well the channel code performs on the equivalent virtual channel.

In [8], a simple and optimal mechanism that exploits the use of syndrome formers (SF) and inverse syndrome formers (ISF), thereafter referred to as the *Asymmetric SF-ISF Framework* (ASIF), is proposed for efficient asymmetric compression. The asymmetric SF-ISF framework is generally applicable to any linear channel code, but achieves the corner points of the SW boundary only. The new framework discussed here, thereafter referred to as the *Symmetric SF-ISF Framework* (SSIF), is an extension and generalization of the ASIF, and it now achieves an arbitrary point in the SW region [13]!

The key idea of SSIF is to first recover the difference pattern between the sources using the syndrome former and the inverse syndrome former of the linear channel code, and to subsequently recover individual source sequences through SF partitioning. After reviewing the SW theorem and the binning concept in Section II, we propose in Section III the constructive approach. The asymmetric SF-ISF framework is first discussed in brevity, followed by a detailed discussion of the more general symmetric SF-ISF framework. Simulations using LDPC codes are provided in Section IV. More examples on the application of the proposed SSIF, including Hamming codes, turbo product codes (TPC) and convolutional/turbo codes, are available in [13]. Finally, Section V concludes the paper.

II. Background

A. Slepian-Wolf Rate Region

Consider a set of m memoryless discrete sources $\mathcal{U} = \{U_1, U_2, \cdots, U_m\}$ with joint probability distribution $\Pr(U_1, \cdots, S_m)$. The achievable rate region for separate compression and joint decompression, known as the Slepian-Wolf region, is bounded by a convex space

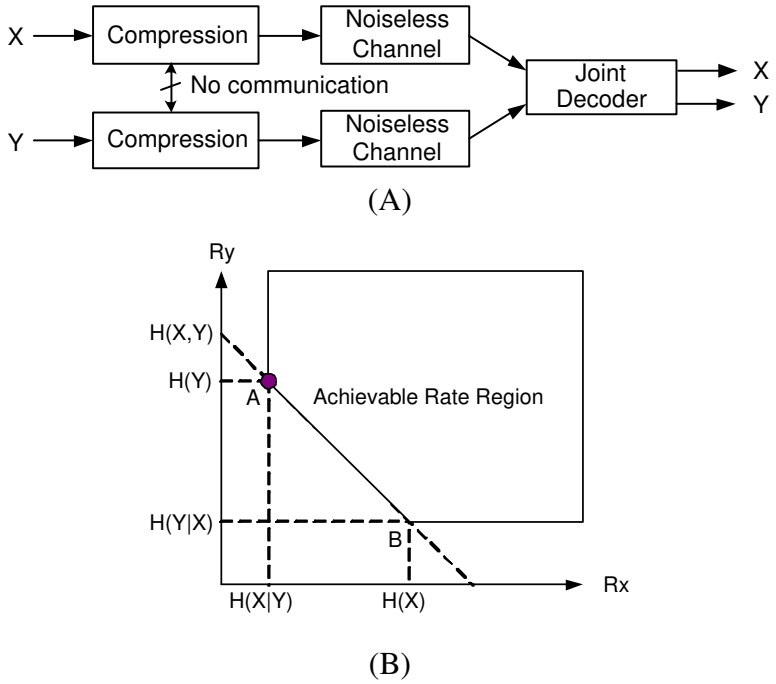

Fig. 1. (A) System model for the (symmetric) SW coding problem. (B) The SW rate region for two discrete memoryless sources.

with boundaries [1][4]:

$$\sum_{U_i \in \mathcal{V}} R_i \geq H(\mathcal{V} \,|\, \mathcal{V}^c), \quad \forall \mathcal{V} \subset \mathcal{U}, \tag{1}$$

where $\mathcal{V}$ is a subset of $\mathcal{U}$, $\mathcal{V}^c$ is the compliment of $\mathcal{V}$, and $H(\cdot)$ is the Shannon entropy function. Specifically, for two binary symmetric sources (X and Y) correlated by BSC(p), the SW region consists of all rate pairs, (R_x, R_y), that satisfy (see Figure 1(B)):

$$R_x \geq H(X|Y) = H(p), \quad R_y \geq H(Y|X) = H(p), \quad R_x + R_y \geq H(X,Y) = 1 + H(p). \tag{2}$$

Each vertex in the SW boundary corresponds to a rate-tuple that is single-user decodable given the side information of the previously decoded user(s), and the process of achieving them is known as asymmetric compression.

B. Code Binning

Central to the practice of Slepian-Wolf coding is the concept of code binning. From the information theory, one realizes that $2^{nH(X,Y)}$ (jointly typical) sequences suffice to describe the joint sources (X^n, Y^n) (for large n). The fundamental idea of code binning is to uniformly place these $2^{nH(X,Y)}$ sequences into a table of r rows and c columns, where $r \times c = 2^{nH(X,Y)}$. Clearly that requires $log_2 r$ bits to index the rows, $log_2 c$ bits to index the columns, and a total of $log_2 r + log_2 c = nH(X,Y)$ bits to uniquely identify a sequence in the table. The SW theorem states that as long as $2^{nH(X|Y)} \leq r \leq 2^{nH(X)}$ and $2^{nH(Y|X)} \leq c \leq 2^{nH(Y)}$, there *exists* an arrangement of these X-Y jointly typical sequences (and corresponding row/column indexes), such that one can unambiguously find the row index of an jointly typical sequence by looking only at the X component, and find the column index by looking only at the Y component. The question is how to attain such an arrangement for any valid pair of c and r?

The problem appears to be difficult in general, except for the two boundary cases which correspond to asymmetric compression. Consider the boundary case when $r = 2^{nH(X|Y)}$ and

$c=2^{nH(Y)}$. To start, we note that there are about $2^{nH(X)}$ typical X sequences, about $2^{nH(Y)}$ typical Y sequences, and about$2^{nH(X,Y)}$ jointly typical sequences. Not all pairs of typical X^n and typical Y^n are also jointly typical, but a jointly typical sequence must be formed from a typical X^n and a typical Y^n. Let us assign the $2^{nH(Y)}$ typical Y^n's to $c=2^{nH(Y)}$ columns, one for each column. For a given typical Y^n, there are about $2^{nH(X|Y)}$ typical X sequences that are jointly typical with it; hence, these typical X^n's can take the $r=2^{nH(X|Y)}$ distinct rows pertaining to that Y^n column. On the other hand, for a given typical X^n, there are about $2^{nH(Y|X)}$ typical Y^n's that are jointly typical with it; hence, a typical X^n sequence will (re)appear in the same row in about $2^{nH(Y|X)}$ different column positions. Put another way, each column (with $r=2^{nH(X|Y)}$ row positions) will host one unique typical Y^n, which takes up all the row positions; whereas each row (with $c=2^{nH(Y)}$ column positions) will host $2^{I(X;Y)}$ different typical X^n's, each occupying $2^{nH(Y|X)}$ column positions pertaining to their respective *jointly typical* Y^n's. It can be easily verified that such an arrangement guarantees that the table contains only the X-Y jointly typical sequences, and that the row index is only a function of the X component and the column index is only a function of the Y component, thus enabling *separate* encoding or mapping of a sequence to its row/column index.

The practical implementation of this *random* binning approach (for asymmetric compression) explores the properties of linear channel codes and their cosets. For binary symmetric sources with BSC(p) correlation, Y^n can be losslessly transmitted at a full rate of $R_y = H(Y) = 1$ bit/sample. X^n can be compressed using the coset structure of the (n,k) linear channel code. Specifically, we can take X^n's as virtual codewords, cosets as bins, and syndromes as bin-indexes. By mapping the length n source sequences to the length $n-k$ syndromes/bin-indexes, a compression ratio of $n:n-k$ is achieved. If the (n,k) channel code has sufficient error correction capability to support a reliable transmission through the virtual BSC(p), then a losslessly recovery of X^n can be guaranteed [1]. Additionally, if the channel code achieves the highest possible transmission rate promised by the Shannon theory (i.e. $k/n = 1-H(p)$), then the source code beats the lowest compression rate promised by the SW theory (i.e. $R_x = (n-k)/n = H(p) = H(X|Y)$ bit/sample).

III. The General Framework

This section details the proposed constructive framework for efficient symmetric and asymmetric SW compression. We start with a brief discussion on the *asymmetric SF-ISF framework*, where the concept of syndrome former and inverse syndrome former was introduced [8]. The more general *symmetric SF-ISF framework* is then presented with a rigorous discussion on its ability to achieve an arbitrary point in the SW rate region. Examples using practical linear channel codes are provided in the next section. Unless otherwise stated, below we will use an (n,k) linear channel code with a valid pair of syndrome former and inverse syndrome former. Let $\mathbf{x}$ and $\mathbf{y}$ denote the n-bit source sequences, $\mathbf{z} = \mathbf{x} \oplus \mathbf{y}$ the n-bit difference pattern, and $\mathbf{s}_x$, $\mathbf{s}_y$ and $\mathbf{s}_z$ the respective syndromes of length $n-k$ bits, all in column vectors.

A. The Asymmetric SF-ISF Framework (ASIF)

The asymmetric SF-ISF framework, first proposed in [8], is a direct exploitation of the binning idea discussed in the previous section. The system structure is presented in Figure 2, where the source encoder compresses $\mathbf{x}$ to $\mathbf{s}_x$ through a syndrome former, and the source decoder decompresses $\mathbf{s}_x$ (with the help of side information $\mathbf{y}$) through an inverse syndrome former and the original channel decoder. The key elements here are a matching pair of syndrome former and inverse syndrome former, whose roles are to systematically "bin" and "de-bin" the source sequences. Consider an (n,k) linear channel code with 2^{n-k} cosets, each indexed with a unique syndrome of length $n-k$. The syndrome former maps the codeword space $\mathcal{X}^n$ to the syndrome space $\mathcal{S}^{n-k}$ by retrieving *the* syndrome/bin-index for a given

codeword sequence, and the matching inverse syndrome former does the reverse mapping by retrieving *an arbitrary* sequence associated with that syndrome. Since there are 2^k codewords in each coset, there thus exist 2^k matching ISFs for any given SF, each producing a different set of outputs. Further, there is no particular rule on the association between syndromes and cosets, except that the all-zero syndrome has to index the "elementary" coset, i.e. the coset that contains all the valid codewords of the channel code. This leads to $(2^{n-k}-1)!$ possible choices for SF, and consequently $(2^{n-k}-1)!2^k$ possible choices for a matching SF-ISF pair! As discussed in [8], these SF-ISF pairs are functionally equivalent as far as SW coding is concerned, but the induced complexity may differ.

Detailed discussion on ASIF's ability to achieve the corner points of the SW boundary can be found in [8]. One key advantage of this framework is that it reduces the problem of constructing SW encoder/decoder to a much easier one of finding a matching SF-ISF pair. Let $\mathbf{H}_{(n-k)\times n}$ be the parity check matrix of a general linear channel code. From the coding theory, one realizes that the syndrome former is an $(n-k)\times n$ matrix in the form of

$$SF = \mathbf{PH}, \tag{3}$$

and a matching inverse syndrome former is an $n\times(n-k)$ matrix satisfying

$$SF \times ISF = \mathbf{I}, \tag{4}$$

where $\mathbf{P}$ is an arbitrary $(n-k)$ full-rank square matrix and $\mathbf{I}$ is an $(n-k)$ identity matrix.

For linear block codes including low-density parity-check codes where $\mathbf{H}$ comes in handy, $\mathbf{H}$ can simply be taken as the SF and its right inverse $\mathbf{H}^{-1}$ as the ISF. For convolutional codes with a generator polynomial/matrix $\mathbf{G}(D)$ (in the $\mathcal{D}$-domain), the transpose of the transfer matrix $\mathbf{T}^T(D)$, where $\mathbf{GT}=\mathbf{0}$, can serve as the syndrome [8]. For parallelly and/or serially concatenated codes where a single parity check or generator matrix is less available, layered or cascade structures can be exploited to efficiently build the SF-ISF pair of the compound code from those of the component codes. For detailed discussion of the latter and especially SF-ISF construction for parallel/serial turbo codes, please refer to [8].

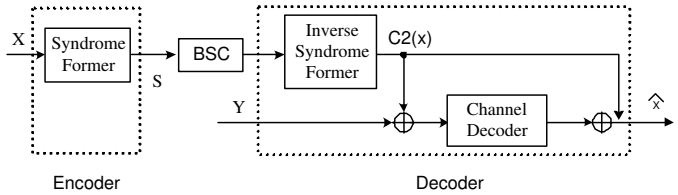

Fig. 2. The asymmetric SF-ISF framework. S denotes the syndrome (output of the SF), and $c2(s)$ denotes an arbitrary codeword that is associated with syndrome s (output of the ISF).

B. The Symmetric SF-ISF Framework (SSIF)

Clearly, the asymmetric SF-ISF framework discussed above is but a general approach to achieve the corner points of the SW boundary. To attain an arbitrary point in the SW region, consider the framework in Figure 3. Without loss of generality, let $\mathbf{H}$ be the parity check matrix of the (n,k) linear channel code, and $SF=\mathbf{H}$ and $ISF=\mathbf{H}^{-1}$ be a matching SF-ISF pair, where $\mathbf{HH}^{-1}=\mathbf{I}$. As shown in Figure 3, the two sources $\mathbf{x}$ and $\mathbf{y}$ will each transmit a syndrome of length $n-k$,

$$\mathbf{s}_x = \mathbf{Hx}, \quad \text{and} \quad \mathbf{s}_y = \mathbf{Hy}, \tag{5}$$

as well as *complementary* subsets of the first k source bits, $\mathbf{x}_1^{k_1}$ (length k_1) and $\mathbf{y}_{k_1+1}^{k}$ (length $k-k_1$), where $0\le k_1\le k$. If the channel code is capacity-achieving on BSC(p), then

$$k/n = 1-H(p) = 1-H(Y|X) = 1-H(X|Y). \tag{6}$$

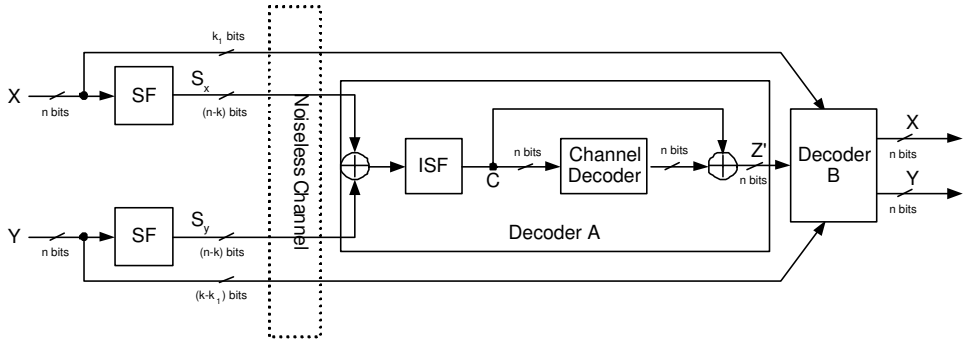

Fig. 3. General scheme for any rate partition between X and Y with an arbitrary channel code in BSC correlation channel.

Since the two sources have transmitted $(n-k+k_1)$ and $(n-k_1)$ bits respectively, the total amount of transmission is thus $2n-k=nH(X,Y)$ bits; further, depending on the value of k_1 $(0\leq k_1\leq k)$, the compression rate $(R_x,R_y)=((n-k+k_1)/n,(n-k_1)/n)$ can achieve any point in the line between $(H(X|Y)=(n-k)/n,H(Y)=1)$ and $(H(X)=1,H(Y|X)=(n-k)/n)$, which is the SW bound.

The source decoder performs the joint decompression in two steps: "Decoder A" (see Figure 3) will first retrieve the difference pattern, $\mathbf{z}=\mathbf{x}\oplus\mathbf{y}$, from syndromes $\mathbf{s}_x$ and $\mathbf{s}_y$; and "Decoder B" will subsequently recover sources $\mathbf{x}$ and $\mathbf{y}$ from the difference sequence $\mathbf{z}$, the syndromes $\mathbf{s}_x$ and $\mathbf{s}_y$, and the complimentary subsets $\mathbf{x}_1^{k_1}$ and $\mathbf{y}_{k_1+1}^{k}$.

Decoder A

Before discussing how Decoder A works, let us present a few basic facts about linear channel codes.

An (n,k) linear channel code can be described in the form of coset codes; see Figure 4. Let $\mathbf{s}$ be an $(n-k)$-bit syndrome, and $\mathbf{c}(\mathbf{s})$ be an n-bit codeword having syndrome $\mathbf{s}$. Let subscripts u,v and i,j denote the row indexes and column indexes of the coset table, respectively, where $0\leq u,v\leq 2^{n-k}-1$ and $0\leq i,j\leq 2^k-1$. Without loss of generality, assume $\mathbf{s}_0$ is the all-zero syndrome and $\mathbf{c}_0(\mathbf{s}_0)$ the all-zero codeword.

- *Fact 1:* Each row in the coset table (see Figure 4) represents a coset consisting of 2^k codewords having the same syndrome:

  $$\mathbf{c}_0(\mathbf{s}_u),\mathbf{c}_1(\mathbf{s}_u),\cdots,\mathbf{c}_{2^k-1}(\mathbf{s}_u).$$

  Specifically, the first row/coset (i.e. $\mathbf{c}_i(\mathbf{s}_0)$'s) contains all the valid codewords of this channel code.
- *Fact 2:* Each column in the coset table represents the set of outputs from a particular inverse syndrome former:

  $$\mathbf{c}_i(\mathbf{u}_0),\mathbf{c}_i(\mathbf{u}_1),\cdots,\mathbf{c}_i(\mathbf{u}_{2^{n-k}-1}).$$

- *Fact 3:* Every four codewords in rectangular positions are related in the following manner:

  $$\mathbf{c}_i(\mathbf{s}_u)\oplus\mathbf{c}_i(\mathbf{s}_v)=\mathbf{c}_j(\mathbf{s}_u)\oplus\mathbf{c}_j(\mathbf{s}_v). \tag{7}$$

  Specifically, for $v=0$ and $j=0$,

  $$\mathbf{c}_i(\mathbf{s}_u)\oplus\mathbf{c}_i(\mathbf{s}_0)=\mathbf{c}_0(\mathbf{s}_u)\oplus\mathbf{c}_0(\mathbf{s}_0)=\mathbf{c}_0(\mathbf{s}_u). \tag{8}$$

- *Fact 4:* The valid codeword closest (in Hamming distance) to $\mathbf{c}_i(\mathbf{s}_u)$ is $\mathbf{c}_i(\mathbf{s}_0)$. In other words, given a (noisy) sequence $\mathbf{c}_i(\mathbf{s}_u)$, a maximum likelihood (ML) decoder will output $\mathbf{c}_i(\mathbf{s}_0)$ (or the k-bit information sequence corresponding to $\mathbf{c}_i(\mathbf{s}_0)$).

- *Fact 5:* Let $\mathbf{z}$ be an n-bit noise vector of a binary symmetric channel. For a given (n, k) linear channel code, treat $\mathbf{z}$ as a virtual codeword with syndrome $\mathbf{s}_z$, i.e. $\mathbf{z} = \mathbf{c}_i(\mathbf{s}_z)$ for some i between 0 and $2^k - 1$. If the channel code is sufficient to support this BSC, then $\mathbf{z} = \mathbf{c}_0(\mathbf{s}_z)$.

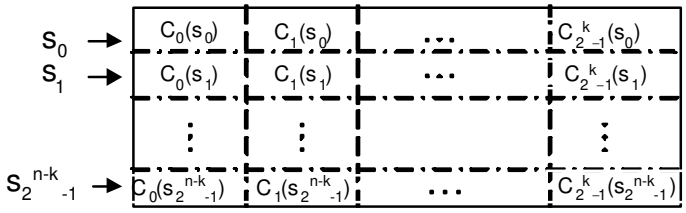

Fig. 4. Coset partition.

The first four facts follow directly from the basic properties of a linear channel code. Fact 5 can be easily proved by contradiction. Recall that a sufficiently powerful channel code on a BSC should recover the all-zero sequence, i.e. the valid codeword $\mathbf{c}_0(\mathbf{s}_0)$, from the noise vector $\mathbf{z}$. If $\mathbf{z} = \mathbf{c}_i(\mathbf{s}_z)$ such that $i \neq 0$, then the channel decoder will produce $\mathbf{c}_i(\mathbf{s}_0)$ (Fact 4), rather than $\mathbf{c}_0(\mathbf{s}_0)$.

Equipped with these facts of a linear channel code, we will now demonstrate how decoder A in Figure 3 can successfully retrieve the difference sequence $\mathbf{z}$.

Upon receiving the syndromes of the source sequences, $\mathbf{s}_x$ and $\mathbf{s}_y$, the decoder will first add them together to get the syndrome of the difference sequence, $\mathbf{s}_z$. This is due to the linearity of the syndrome former:

$$\mathbf{s}_x \oplus \mathbf{s}_y = \mathbf{H}\mathbf{x} \oplus \mathbf{H}\mathbf{y} = \mathbf{H}(\mathbf{x} \oplus \mathbf{y}) = \mathbf{H}\mathbf{z} = \mathbf{s}_z. \tag{9}$$

Next, the decoder passes $\mathbf{s}_z$ through the inverse syndrome former to obtain an n-bit codeword $\mathbf{c}_i(\mathbf{s}_z)$ (Fact 2), and subsequently through the channel decoder to obtain a valid codeword $\mathbf{c}_i(\mathbf{s}_0)$ (Fact 4). Finally, the decoder combines $\mathbf{c}_i(\mathbf{s}_0)$ and $\mathbf{c}_i(\mathbf{s}_z)$ to capture the difference pattern $\mathbf{s}_z$:

$$\begin{aligned} \mathbf{c}_i(\mathbf{s}_0) \oplus \mathbf{c}_i(\mathbf{s}_z) &= \mathbf{c}_0(\mathbf{s}_0) \oplus \mathbf{c}_0(\mathbf{s}_z) \qquad \text{(Fact 3)} \\ &= \mathbf{c}_0(\mathbf{s}_z) \\ &= \mathbf{s} \qquad\qquad\qquad \text{(Fact 5)} \end{aligned} \tag{10}$$

Decoder B

Decoder B recovers the sources $\mathbf{x}$ and $\mathbf{y}$ from $\mathbf{z}$, $\mathbf{s}_x$, $\mathbf{s}_y$, $\mathbf{x}_1^{k_1}$ and $\mathbf{y}_{k_1+1}^{k}$ by means of syndrome former partitioning.

First, the missing parts of the first k bits of both sources can be easily recovered using the difference pattern (see Figure 5 where the gray area represents the bits known before Decoder B):

$$\mathbf{x}_{k_1+1}^{k} = \mathbf{y}_{k_1+1}^{k} \oplus \mathbf{z}_{k_1+1}^{k}, \tag{11}$$

$$\mathbf{y}_{1}^{k_1} = \mathbf{x}_{1}^{k_1} \oplus \mathbf{z}_{1}^{k_1}. \tag{12}$$

Next, notice that the syndrome former, $\mathbf{H}$, can be partitioned to two sub-matrices:

$$\mathbf{H}_{(n-k)\times n} = \Big[\mathbf{A}_{(n-k)\times k}, \mathbf{B}_{n-k}\Big], \tag{13}$$

where $\mathbf{B}$ is a square matrix. Without loss of generality, assume $\mathbf{B}$ is full-rank. Since

$$\mathbf{s}_x = \mathbf{H}\mathbf{x} = \big[\mathbf{A}, \mathbf{B}\big] \left[\begin{array}{c} \mathbf{x}_1^k \\ \mathbf{x}_{k+1}^n \end{array} \right] = \mathbf{A}\mathbf{x}_1^k \oplus \mathbf{B}\mathbf{x}_{k+1}^n, \tag{14}$$

$$\mathbf{s}_y = \mathbf{H}\mathbf{y} = \big[\mathbf{A}, \mathbf{B}\big] \left[\begin{array}{c} \mathbf{y}_1^k \\ \mathbf{y}_{k+1}^n \end{array} \right] = \mathbf{A}\mathbf{y}_1^k \oplus \mathbf{B}\mathbf{y}_{k+1}^n \tag{15}$$

we can then recover the remaining $n-k$ source bits using:

$$\mathbf{x}_{k+1}^n = \mathbf{B}^{-1}\Big(\mathbf{s}_x \oplus \mathbf{A}\mathbf{x}_1^k\Big), \tag{16}$$

$$\mathbf{y}_{k+1}^n = \mathbf{B}^{-1}\Big(\mathbf{s}_y \oplus \mathbf{A}\mathbf{y}_1^k\Big). \tag{17}$$

Alternatively, after recovering one source, we can recover the other using the difference pattern $\mathbf{z}$.

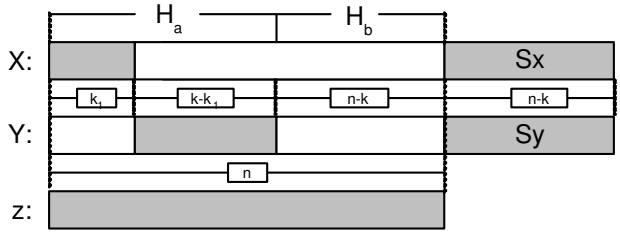

Fig. 5. Illustration for Decoder B. The gray areas represent the bits known before Decoder B.

IV. Examples Using LDPC Codes

In this section, we demonstrate the feasibility and efficiency of the symmetric SF-ISF framework using a popular and powerful class of linear channel codes, namely, LDPC codes.

Consider an (n, k) LDPC code with parity check matrix $\mathbf{H}_{(n-k)\times n} = [\mathbf{A}_{(n-k)\times k}, \mathbf{B}_{(n-k)}]$. Without loss of generality, assume that the columns of $\mathbf{H}$ have been pre-arranged such that the right part $\mathbf{B}$ is a full rank square matrix. By letting $\mathbf{P} = \mathbf{I}$ in (3), we get the syndrome former:

$$SF = \mathbf{H} = [\mathbf{A}, \mathbf{B}]_{(n-k)\times n}. \tag{18}$$

A matching inverse syndrome former can be obtained by solving (4). A possible solution is:

$$ISF = \left[\begin{array}{c} \mathbf{0} \\ \mathbf{B}^{-1} \end{array} \right]_{n\times(n-k)}. \tag{19}$$

Equipped with this SF-ISF pair and a message-passing LDPC decoder, SW compression using the SSIF framework becomes rather straight-forward.

In our simulation, we take two independent and identically distributed (i.i.d.) binary sources X and Y with a BSC(p) correlation. We consider rate 1/2 regular (n, k) LDPC codes with a constant column weight of 3. This means that two sources of length n bits each will be compressed to $2n - k = 1.5n$ bits altogether. The information block lengths we tested were $k = 1000, 2000$ and 3000 bits, respectively. The experimental results are obtained based on the compression of at least 10^8 bits. Figure 6 plots the LDPC SSIF performance of a symmetric compression case where $R_x = R_y = 0.75$ bit/symbol. The X-axis denotes different correlation level p and the Y-axis denotes the average (normalized) Hamming distortion, averaged over the two sources. We see that the compression quality improves with the block length, as one would expect from the behavior of LDPC codes.

To see whether the proposed SSIF framework performs equally well for symmetric and asymmetric compression alike, we test the LDPC code with $k = 5000$ bits for different

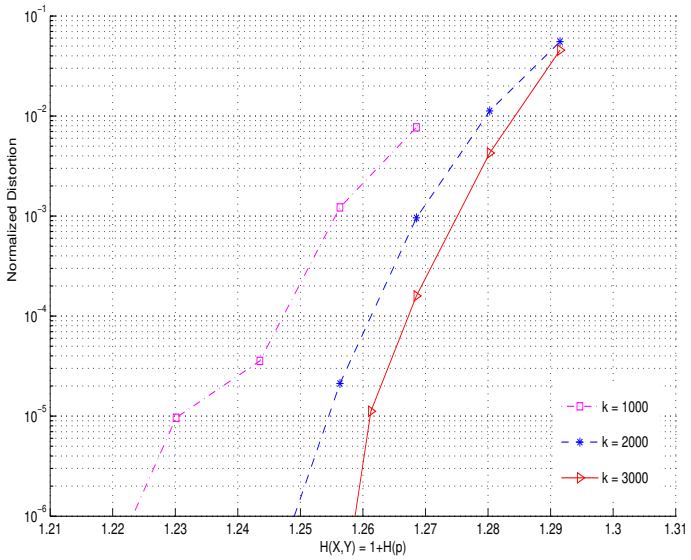

Fig. 6. The averaged distortion of SSIF using four LDPC codes with length of $k = 1000, 2000, 3000$.

allocations of compression rates: $R_x : R_y = 2:1,\ 7:5,\ 1:1,\ 5:7,\ 1:2$, and examine their respective gaps to the SW limit. It should be noted that although we target lossless compression, the imperfectness of the channel code will introduce a small number of errors, or a slight distortion, in the recovered sources. Following the convention, we consider a normalized Hamming distortion of 10^{-6} and below as near-lossless. The experimental results of the achievable rate pairs are shown as hexagrams in Figure 7. We note that all the five rate allocations support exactly the same crossover probability of $p = 0.070$. Hence, their rate pairs fall in a straight line that is parallel to the SW bound, demonstrating a robust performance that is insensitive to $R_x : R_y$ rate allocations. Due to the short block size and therefore the relatively weak performance of the LDPC code, the gaps between the actual rate pairs and the SW bound are not very small. We have not had time to conduct a lengthy simulation on long and powerful LDPC codes, but the results demonstrated in Figure 7) are sufficient to show that that the SSIF can uniformly approach the SW bound. In light of the facts that the SSIF incurs no rate loss when converting a linear channel code to a SW code and that long LDPC codes are capacity-approaching channel codes, it is fair to say that the SSIF is capable of getting arbitrarily close to any point in the SW bound.

V. Conclusions

We have proposed a constructive framework, termed the symmetric SF-ISF framework, for efficient Slepian-Wolf coding of memoryless binary symmetric sources. The framework is an extension and generalization of the asymmetric SF-ISF framework we proposed earlier, and can now achieve the entire rate region promised by the theory. The key advantages of the new framework include:

1) The framework is a simple but powerful one that can be easily applied to a general linear channel code including systematic codes and non-systematic codes;
2) The framework can achieve an arbitrary point in the SW rate region (including all the points in the boundary) by simply choosing the appropriate channel code and by adjusting the appropriate subsets of the source bits to be transmitted;
3) While we have focused the discussion on the probabilistic correlation model (i.e. BSC model), the framework generalizes to other constrained correlation model such as the Hamming correlation model [13].

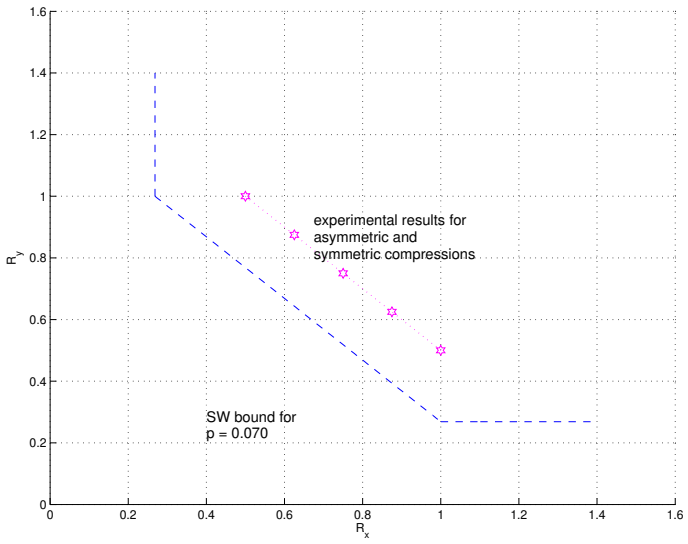

Fig. 7. Achievable rate pairs of the LDPC-SSIF and the SW bound. All the LDPC codes have length $k = 5000$.

Examples using LDPC codes are provided to demonstrate the practicality and efficiency of the framework. We see that the compression performances improve with the increase of the block length, and they uniformly approach the SW limit regardless of rate allocation between the sources. This result is consistent with the error correction performances of LDPC codes, and reflects the lossless conversion property of the framework. For future research, we will consider binary asymmetric sources and sources with memory. We will examine the optimality of the framework in the new context, and search for ways to exploit the additional information, i.e. the *a prior* probability and the temporal correlation of the sources, in compression.

REFERENCES

[1] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol.19, pp.471-480, July 1973.

[2] S. S. Pradhan, and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Trans. Inform. Theory*, vol. 49, no.3, pp. 626-643, March 2003.

[3] A. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Letters*, pp. 440-442, Oct. 2002.

[4] T. P. Coleman, A. H. Lee, M. Medard, and M. Effros, "On some new approaches to practical Slepian-Wolf compression inspired by channel coding," *Proc. of IEEE Data Compression Conf.*, pp.282-291, Snowbird, UT, March 2004.

[5] A. D. Liveris, Z. Xiong, and C. N. Georghiades. "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," *Proc. of IEEE Data Compression Conf.*, March 2003.

[6] A. Aaron and B. Girod, "Compression with side information using turbo codes," *Proc. of IEEE Data Compression Conf.*, pp. 252-261, April 2002.

[7] J. Li, Z. Tu, and R. S. Blum, "Slepian-Wolf coding for nonuniform sources using turbo codes," *Proc. IEEE Data Compression Conf.*, pp. 312-321, Snowbird, UT, March 2004.

[8] Z. Tu, J. Li, and R. S. Blum, "An efficient SF-ISF approach for the Slepian-Wolf source coding problem," *Eurasip J. of Advanced Signal Processing - Special Issue on Turbo Processing*, no. 6, pp. 961-971, May 2005.

[9] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Letters*, pp. 417-419, Oct. 2001.

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

[11] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Design of Slepian-Wolf codes by channel code partitioning," *Proc. IEEE Data Compression Conf.*, pp. 302-311, 2004.

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

[13] P. Tan, and J. Li, "A general constructive framework to achieve the entire rate region for Slepian-Wolf coding," *Eurasip Signal Processing Journal - Special Issue on Distributed Source Coding*, accepted, 2006.