Slepian-Wolf Coding for Nonuniform Sources Using Turbo Codes

Jing Li (Tiffany), Zhenyu Tu and Rick S. Blum
Department of Electrical and Computer Engineering
Lehigh University, Bethlehem, PA 18105

Abstract

The recently proposed turbo-binning scheme is shown to be both efficient and optimal for uniform source Slepian-Wolf coding problem [1]. This paper studies the case when sources are i.i.d but nonuniformly distributed. It is firstly shown that any algebraic binning scheme based on linear codes is optimal for nonuniform sources only asymptotically. Next two modifications are proposed to improve the performance of the turbo-binning scheme for nonuniform sources. The first is to carefully design the constituent encoder structures to maximally match the turbo code to the nonuniform source distribution, and the second is to use variable-length syndrome sequences to index the bins. Simulations show that the combination of both strategies can lead to an improvement of as much as 0.22 bit/symbol in overall compression rate for highly nonuniform sources.

I. Introduction

The lossless distributed source coding (DSC) problem, also known as the Slepian-Wolf problem, is an interesting problem that has close connections to a number of (network) information processing problems, like sensor networks, communication on broadcast channels, watermarking and data hiding. To the best of the authors' knowledge, the DSC research work so far has all assumed uniform sources. This is not always true in real life. For example, many uncompressed binary images (e.g. facsimile images) may have a source distribution as biased as $p_0 = 0.96$ and $p_1 = 0.04$ [2]. Hence, there is a strong need for studying nonuniform sources. This paper tackles the asymmetric-compression DSC problem for nonuniform i.i.d. sources using turbo codes.

Consider two binary i.i.d. sources X and Y that are content-correlated but physically-separated (i.e. no communication between the sources). The goal of distributed source coding is to devise efficient ways to separately encode (i.e. compress) but jointly decode (i.e. decompress) the sources to reach the same compression rate achievable when the sources are jointly encoded and jointly decoded. The famous Slepian-Wolf theorem [3] determines the 2-source DSC rate region: $R_x \geq H(X|Y)$, $R_y \geq H(Y|X)$, and $R_x + R_y \geq H(X,Y)$. It has been shown that the Slepian-Wolf boundary is achievable both

This material is based on research supported by the Air Force Research Laboratory under agreement No. F49620-03-1-0214 and by the National Science Foundation under Grant No. CCR-0112501.

asymptotically and with finite-length sequences [4]. Specifically, the corner points of the Slepian-Wolf boundary, where one source is losslessly available at the decoder (e.g. Y compressed to $H(Y)$ via a conventional entropy-compression method) and the other is maximally compressed utilizing the statistical correlation between the two sources (X compressed to $H(X|Y)$), may be modeled as an equivalent channel coding problem with decoder side information (SI) where the "equivalent transmission channel" is specified by the correlation of the sources. To get close to the theoretical limit, two key issues need to be resolved: (i) finding a capacity-approaching channel code for the equivalent transmission channel and (ii) bridging the practice and solution of channel coding with that of source coding. Although closely related, the two issues reflect different aspects of the DSC problem. While the former can take advantage of the rich literature available on channel coding, the latter is much less studied. One of the most successful approaches that bring the solution of channel coding to serve the problem of source coding is the coset/syndrome/binning approach [3].

The binning approach (Section II), originated from the proof of the Slepian-Wolf theorem, has been widely deployed with a variety of practical channel codes including block codes and coset/lattice codes (e.g. [5][6][7]). The beauty of the algebraic binning approach is the exploitation of the uniformity (regularity) of the code space of a linear code to group source sequences into bins. By transmitting the (short) bin-index/syndrome instead of the (long) source sequence, compression is achieved. While this binning scheme is applicable to all linear codes and optimal for uniform i.i.d. sources, it is unfortunately suboptimal for nonuniform sources with any finite length.

This work investigates the nonuniform-source DSC problem with a focus on the turbo-binning scheme proposed in [8][1]. The scheme was shown to be simple, general, and better performing (with uniform sources) than other existing turbo-DSC approaches in literature [9][10][11][12]. However, as we will see later, its performance with nonuniform sources is much less desirable. We first discuss the reasons behind its sub-optimality (with nonuniform sources), and then find ways to improve it. Specifically, two approaches are proposed: the first is to *design constituent encoder structures to better match the turbo code to the nonuniform source distribution*, and the second is to *employ variable-length bin-index assignment to mitigate the gap between the optimal bin-index assignment and the current (suboptimal) fix-length bin-index assignment*. Simulations show that the combination of the two methods leads to a significant improvement of the compression rate especially for highly biased sources.

The paper is organized as follows. Section II presents the technological background. Section III discusses the suboptimality of the turbo-binning scheme in [1] for nonuniform sources. Section IV addresses turbo code design and variable-length bin-index assignment.

Section V presents simulation results and Section VI concludes the paper.

II. Background

A. System Model of the Nonuniform Source DSC Problem

Consider the asymmetric compression scenario, where Y is treated as the decoder side information, and X is compressed utilizing its correlation with Y. In a general setup, correlation between two binary i.i.d nonuniform sources requires two parameters to describe (e.g. $P(Y \neq X|X=0)$ and $P(Y \neq X|X=1)$) and the equivalent virtual channel between X and Y is thus a binary asymmetric channel (BAC). Since content-dependent crossover probabilities make the channel tricky and difficult to analyze, for convenience, we consider a subset of the general problem by imposing a symmetry condition on the source correlation. This translates the virtual channel to a binary symmetric channel (BSC), i.e., $Y = X + Z$, where $q = P(Z=1)$ denotes the crossover probability and $p_0 = P(x=0)$ the nonuniform source distribution.

B. Algebraic Binning Schemes Based on Linear Codes

Before discussing the turbo-binning scheme, we first summarize the key points of the algebraic binning scheme based on (n, k) binary linear channel codes:

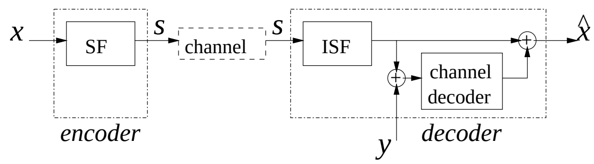

Fig. 1. Encoder and decoder structure using the syndrome/binning approach.

Constructing Bins: Partitioning the codeword space $\{0,1\}^n$ into 2^{n-k} subspaces (disjoint sets, bins or cosets) such that each subspace $\{0,1\}^n \backslash 2^{n-k}$ contains 2^k codewords of length n and *the same distance properties are preserved in each subspace*. Such a partitioning is possible and not unique. In fact, any (n, k) binary linear channel code automatically defines a partition where codewords having the same syndrome belong to the same subspace, and the 2^{n-k} syndromes of length $n-k$ each can naturally be used to index the subspace/bins.

Encoding and Decoding: The encoder is essentially a *syndrome former* (SF) which maps a codeword sequence to its syndrome/bin-index, and thus achieves a compression rate of $n : (n-k)$. The decoder employs a combination of an *inverse syndrome former* (ISF) and the original channel decoder. The role of the ISF is to find an arbitrary codeword, $\tilde{X}^n$, that is associated with the given syndrome. The combination of the SI Y^n and $\tilde{X}^n$ is then treated as a noise-corrupted version of a valid codeword and passed to the channel decoder. If the channel code is sufficiently strong (for the equivalent virtual channel), the channel decoder will produce the valid codeword with (near-)zero error probability. Subtracting $\tilde{X}^n$ from the output of the channel decoder will recover the original source sequence X^n. The exact structure is illustrated in Fig. 1, and the validity is warranted by the preservation of the same distance properties in each bin. Detailed discussion and proof can be found, for example, in [1].

III. Turbo-Binning Scheme and Its Sub-Optimality with Nonuniform Sources

Given the above algebraic binning scheme, design of an algorithm for compression with decoder SI becomes a well-defined two-step process: finding a well-performing channel code with rate $k/n \approx H(X|Y)$, and constructing an SF-ISF pair for this code. This work investigates turbo codes, since (i) turbo codes are among the best-known channel codes, (ii) a turbo encoder is cheap to implement (thus appealing for applications like sensor networks where the computation on the transmitter side needs to be minimized) and the length of a turbo code can be easily changed, making it possible to adapt to the varying correlation between sources. However, the random interleaver in a turbo code makes SF/ISF construction less obvious. Fortunately, [1][8] proposed an efficient method to get around the random scrambling effect. Below we briefly describe the main result therein.

A. Constructing Syndrome Former and Inverse Syndrome Former for Turbo Codes

Consider a parallel turbo code whose component codes are recursive systematic convolutional (RSC) codes with generator matrix G. As discussed in [1][8], the SF and ISF of a turbo code can be constructed from those of the component codes. First, let us note that, akin to the generator matrix and parity-check matrix of a linear block code in the binary domain, a generator matrix and a transfer matrix can be defined for a convolutional code which offer similar functions but operate in the D-domain. The generator matrix is composed of generator polynomials (e.g. $G(D)=[1, U(D)/V(D)]$ for a rate 1/2 RSC code) which specify the (sub) space containing all the valid codewords; and the transfer matrix specifies its null space ($GH^T = \mathbf{0}$).

<u>SF and ISF for the Constituent RSC Code:</u> It can be easily shown that the transfer matrix fulfills the function of an SF and, by taking the left inverse of the SF, a matching ISF is obtained [13]. Specifically, for a (typical) constituent RSC code with generator matrix $G(D)=[1, U(D)/V(D)]$, we can select (superscript T denotes matrix transpose):

$$\text{SF}: \;\; H^T = [U(D)/V(D), \; 1], \qquad \text{ISF}: \;\; (H^{-1})^T = [0, \; 1]. \tag{1}$$

Apparently, selection of an SF-ISF pair is not unique. Different SF-ISF pairs result in different ways of assigning syndrome sequences to bins (the role of SF) as well as which specific codeword to output for a given bin-index (the role of ISF). It should be emphasized that the specific choice of the SF-ISF pair in (1) is important for turbo codes, since the ISF therein always finds the codeword with the all-zero systematic bits in the bin for any given bin-index. This effectively avoids the potential misalignment of systematic bits (due to the random interleaver) for turbo codes and consequently reduces complexity [1].

SF and ISF for the Parallel Turbo Code: Unlike the case of RSC codes, the SF and ISF of a turbo code do not have closed-form expressions. They can nevertheless have simple structures and be linear-time computable. The structures of an SF and its matching ISF proposed in [1][8] are illustrated in Fig. 2, where H^T and $(H^{-1})^T$ respectively denote the SF and ISF of the constituent RSC codes, and π denotes the random interleaver used in the turbo code.

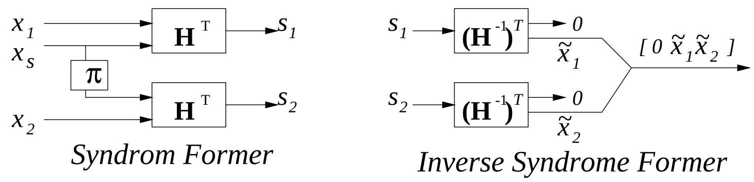

Fig. 2. The SF-ISF pair for a turbo code.

What is discussed above refers to the typical case where the parallel turbo code is built from rate 1/2 constituent RSC codes. The general case where the constituent code can be of any rate k/n is discussed in [8][1]. Further, a similar but slightly more involved treatment for serial turbo codes can be found in [1].

B. Suboptimality of the Turbo-Binning Scheme

While the turbo-binning scheme in [1] is optimal with uniform sources, it is unfortunately not optimal with nonuniform sources except for the asymptotic case. Recall that the fundamental idea in the binning approach is to use approximately $2^{nH(X,Y)}$ sequences to describe i.i.d. sources (X^n, Y^n) where the $2^{nH(X,Y)}$ sequences will be placed in $2^{nH(X|Y)}$ bins with $2^{nH(Y)}$ sequences in each bin. $nH(X|Y)$ bits will be used to specify a particular bin and $nH(Y)$ bits will be used to specify a particular sequence in the bin. Clearly, in order to achieve an overall optimality, the process of assigning bin-indexes (i.e. computing syndromes) need to achieve "entropy compression" for the bins, where each bin is associated with a given probability: the cumulative probability of all codewords in the bin. Since the algebraic binning scheme based on linear codes uses fixed-length syndromes as bin-indexes, the bin-indexes are an optimal assignment only when the bins are "balanced". Here we mean balanced in the sense that each bin contains exactly 2^k codewords and the cumulative probability of all codewords in any bin is 2^{k-n}. Whereas this is the case with uniform sources, it is sadly untrue for nonuniform sources, except for the asymptotic case where there are infinite number of sequences in the bin.

Aside from the above factor, part of the performance loss of the turbo-binning scheme also stems from the fact that turbo codes are not optimal codes for nonuniform sources. As illustrated in Fig. 1, the sum of the side information Y and the intermediate codeword $\tilde{X}$ (at the output of the ISF) is treated as the noise-corrupted version of a valid codeword and fed into the channel decoder. Clearly, the channel code in use (in this case the turbo code) needs to be sufficiently strong to ensure a (near) error free detection of the correct codeword, which will then enable a lossless recovery/decompression of the original sequence X. Due to the nonuniform distribution of sources, some codewords will have a larger chance of occurrence. This calls on some form of unequal error protection (UEP) where highly probable codewords are better protected and thus have a smaller chance of making an error. However, a turbo code, or more generally any linear code, maintains a geometrical uniformity in their codeword space, where the "set of distances" from a particular (valid) codeword to the rest of (valid) codewords is the same for every codeword and, hence, all codewords are equally protected. It is interesting to note that this uniformity property, while preventing a linear code from capacity-achieving for nonuniform sources, is required and exploited in constructing the bins (Section II). This seems to suggest an irresolvable dilemma in the binning practice, i.e. the need for linear codes and the sub-optimality of linear codes for nonuniform sources.

IV. Turbo-Binning Scheme for Nonuniform Sources

Although it is unclear how to optimally conduct DSC for nonuniform sources, there are ways we can take to improve the current algebraic binning scheme. Specifically, we propose (i) to optimize turbo codes to better match with the nonuniform source distribution and (ii) to employ variable-length bin-indexes.

A. Optimizing Turbo Codes for Nonuniform i.i.d. Sources

Despite the fundamental sub-optimality, linear codes can, subject to the individual code space mapping, exhibit very different error correcting behaviors with nonuniform sources. Specifically, [14] and [15] show that it is possible for a turbo code to fall behind its peer (of similar complexity) with uniform sources but well outperform it with nonuniform sources (on AWGN and fading channels). This implies that code selection for nonuniform sources need to adopt different criteria from those of uniform sources.

Before conducting encoder optimization, we note that the turbo decoder needs to, first of all, be optimally matched to the nonuniform source distribution. Recall that the APP (maximal *a posteriori*) decoder of the constituent RSC code takes $L_{ap}(d_k)$, the *a priori* information of bit d_k, and $L_{ch}(d_k)$, the reliability information from the channel, at the input, and, after running the BCJR algorithm, computes the overall LLR (log-likelihood ratio) information $\Lambda(d_k)$. The "add-on" value, also termed the extrinsic information, is computed as $L_{ex}(d_k) = \Lambda(d_k) - L_{ch}(d_k) - L_{ap}(d_k)$, and passed to the other constituent decoder for subsequent decoding. For BSCs with a crossover probability q, the channel reliability information is given by $L_{ch}(d_k) = \log\frac{1-q}{q}(1-2r_k)$ where $r_k \in \{0,1\}$ is the received bit for d_k. Detailed discussion on the BCJR algorithm can be found in [16]. Here we emphasize the computation of the *a priori* information $L_{ap}(d_k)$. As the name suggests, $L_{ap}(d_k)$ needs to account for all knowledge that is available *a priori* to the decoder. Hence, in addition to the extrinsic information from the other constituent decoder, it should also include knowledge on the source distribution, i.e.

$$L_{1,ap}^{(l)}(d_k) = \log \frac{p_0}{1-p_0} + L_{2,ex}^{(l-1)}(d_k), \qquad L_{2,ap}^{(l)}(d_k) = \log \frac{p_0}{1-p_0} + L_{1,ex}^{(l)}(d_k), \tag{2}$$

where superscript (l) denotes the l_{th} decoding iteration, and subscripts 1 and 2 denote the respective constituent codes. The inclusion of the term $\log \frac{p_0}{1-p_0}$ *at every iteration* is important to ensuring an optimal performance at the decoder.

The job of encoder optimization is more involved. A purely analytical approach is difficult. Here, we use a similar method as in [14] and pursue a constraint search for better constituent encoders for nonuniform sources on BSC channels.

First, we note that [14] attributes the reason why the $(37, 21)$ Berrou code performs poorly for nonuniform sources to the fact that some states in the trellis are rarely reached. We conjecture that the above observation holds true in general, that is, for a code to be well-performing (for a given source distribution), all states in the trellis need to be accessed regularly (i.e. at a reasonable probability). While it is hard to prove or quantitatively analyze this argument, a tentative explanation is that, when some states are rarely accessed, the number of "active states" is decreased. This can be equivalently viewed as a reduction in the "effective memory size"[1], which in turn leads to performance degradation.

Although the above observation helps to provide an intuitive judgment of the code performance, the actual search nevertheless relies on computer simulation. Considering the large number of possible generator polynomials, instead of performing an exhaustive (and inefficient) search, we first constrain both the feed-forward and feed-back polynomials to be in the form of $G(D) = 1 + \sum_{i=1}^{m-1} g_i D + D^m$, where m is the memory size and $g_i \in \{0, 1\}$ is the coefficient. Further, observe that a convolutional code with a generator matrix $[G_1(D), G_2(D)]$ has the same performance as its "reciprocal code" whose generator matrix is given by $[D^m G_1(\frac{1}{D}), D^m G_2(\frac{1}{D})]$. Hence, we fold the search region in half, or equivalently, we impose a second constraint on the feed-back polynomial such that its coefficient g satisfies $g \le g_{m-i}$ for $i = 0, 1, \cdots, \lfloor \frac{m}{2} \rfloor$. Next we conduct a similar iterative search as in [14], namely, fix the feed-forward polynomial and search for the best feed-back polynomial, then fix the feed-back polynomial and search for the best feed-forward polynomial, and so on. The search results are reported in the simulation section.

B. Variable-Length Bin-Index Assignment

As discussed above, the structured binning scheme is no longer optimal for nonuniform sources, since fixed-length syndromes/bin-indexes are assigned to bins with unequal probabilities. The concept of variable-length bin-indexes may serve as an extension of and a

[1] In a typical binary case, we have $S = 2^m$ where S is the number of states and m is the memory size of the binary convolutional code.

remedy to the current algebraic binning scheme. A one-step approach to compute variable-length syndromes/bin-indexes is difficult, but a two-step approach is straightforward. The fixed-length syndrome sequence computed directly from the channel code can be treated as the "intermediate syndrome", and be further compressed by a conventional method before being sent to the receiver. The receiver first decompresses the received sequence to recover the intermediate syndrome, and then proceeds with the conventional binning approach.

The question then is how much can be further compressed, or whether it is worth the trouble. To illustrate this, we take the $(31, 23)$ turbo code, the winning code from our computer search, as an example. Tab. I lists the entropy of the syndrome bits obtained using the SF-ISF pair presented in (1). Due to the space limitation, we omit the discussion on the computation of syndrome entropy, but focus on the result. As can be seen from the table, when the source distribution is near uniform ($p_0 \to 0.5$), the syndrome entropy is close to the maximum value of 1 and fixed-length syndrome/bin-index assignment is sufficient for any practical purpose. However, when the source is highly biased ($p_0 \to 1$), syndrome bits contain a significant amount of redundancy that can be removed. For example, in the case of $p_0 = 0.95$, an optimal variable-length syndrome/bin-index assignment can achieve an additional compression rate of $1 : 0.4529$ over its fixed-length counterpart!

It should be noted that the entropy listed in the table corresponds to that of the individual *syndrome bit*, $H(S)$, where syndrome bits are treated as if they were i.i.d.. Due to the correlation among syndrome bits, the entropy of the *syndrome sequence*, $H(S^{n-k})$, is actually lower than $(n-k)H(S)$. Hence, the compression gain will be even larger in theory. However, since it is very difficult, if not impossible, for a practical compression method to exploit the correlation in a syndrome sequence, the entropy of the syndrome bit serves as a fair evaluation of how much gain variable-length syndromes/bin-indexes have over their fixed-length counter-parts.

TABLE I

THE ENTROPY OF SYNDROME BITS OF THE $(31, 23)$ TURBO CODE FOR DIFFERENT SOURCE DISTRIBUTION p_0

p_0	0.5000	0.5500	0.6000	0.6500	0.7000	0.7500	0.8000	0.8500	0.9000	0.9500
$H(S)$	1.0000	0.9999	0.9988	0.9941	0.9815	0.9544	0.9044	0.8191	0.6801	0.4529

V. SIMULATIONS

Although the $(37, 21)$ Berrou code exhibits remarkable waterfall region performance with uniform sources, results from computer search reveal that the best constituent codes for nonuniform sources on BSC channels are the $(25, 23)$ code and the $(21, 23)$ code. Fig. 3 plots the performance curves of these codes on BSC channels with different source distributions p_0. The interleaver size is 1000. X-axis denotes the crossover probability q and y-axis denotes the bit error rate. In the 4_{th} subplot, the performances of the $(25, 23)$ and $(21, 23)$ turbo codes are compared with that of the $(37, 21)$ Berrou code at $p_0 = 0.7$.

Noticeable performance gain is observed at low bit error rate. The gain is expected to be even larger when the source becomes more nonuniform.

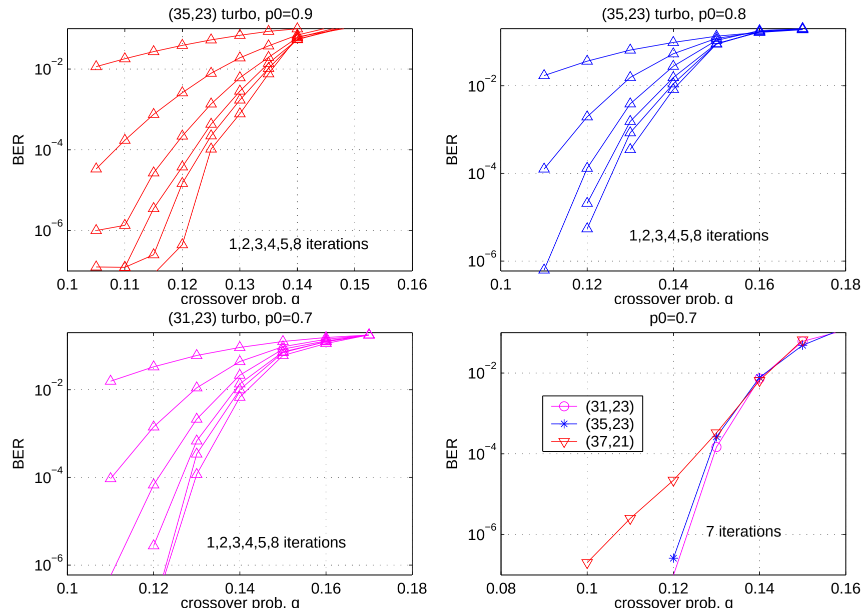

Fig. 3. Performance of $(35, 23)$ and $(31, 23)$ turbo codes on BSC(q) channels with nonuniform sources p_0.

Tab. II summarizes the results of the turbo-binning scheme using optimized turbo codes and variable-length bin-indexes. Code rate is 1/3 and interleaver size is 16k. For comparison, we also include the performance of the Berrou code with fixed-length bin-indexes. We assume a distortion of 10^{-6} is near-lossless. "Attainable q" in the table refers to the maximum crossover probability of the equivalent BSC (the amount of correlation between X and Y) that the code can support. Gap A, B and C refer to the gap between the theoretical limit and the achievable compression rate of turbo-binning schemes not using proposed strategies (i.e. Berrou codes and fix-length bin-indexes), using optimized turbo codes only, and using both optimized turbo codes and variable-length bin-indexes, respectively. Clearly, employing the proposed strategies has significantly improved the overall performance. Specifically, for the highly nonuniform source like $p_0 = 0.9$, the two strategies combined has resulted in a compression rate of only 0.1444 bit/symbol away from the theoretic limit, which has closed the gap by as much as 0.3632-0.1444=0.2188 bit/symbol!

VI. Conclusion

This paper extends the turbo-binning scheme proposed in [1] to the nonuniform source DSC problem. We consider the scenario where source Y is losslessly available at the decoder, source X is nonuniformly distributed, and the equivalent channel from X to Y is a binary symmetric channel.

We attribute the suboptimality of the turbo-binning scheme to two reasons: the suboptimal performance of the turbo code (as a channel code) on BSC channels for nonuniform sources and the suboptimality of the fixed-length bin-index assignment in the algebraic binning scheme. Correspondingly, we propose two ways to improve the performance, namely, to search for the best turbo code matched to the nonuniform source and to employ variable-length bin-index assignment. Combining both strategies, we show that the compression rate of the turbo-binning scheme can be improved for as much as 0.22 bit/symbol for highly nonuniform sources, which is quite impressive.

TABLE II

THE ENTROPY OF SYNDROME BITS OF THE $(31, 23)$ TURBO CODE FOR DIFFERENT SOURCE DISTRIBUTION p_0

source dist.	Berrou Code + Fixed-Length Bin-indexes			Optimal Code (+ Variable-Length bin-indexes)			
p_0	Attainable q	$H(X\|Y)$	**Gap A**	Attainable q	$H(X\|Y)$	**Gap B**	**Gap C**
0.7	0.139	0.5239	**0.1427**	0.143	0.5330	**0.1337**	**0.1213**
0.8	0.139	0.4435	**0.2232**	0.143	0.4507	**0.2159**	**0.1522**
0.9	0.136	0.3035	**0.3632**	0.141	0.3090	**0.3574**	**0.1444**

REFERENCES

[1] Z. Tu, J. Li, and R. Blum, "An efficient turbo-binning approach for the Slepian-Wolf source coding problem," *submitted to Eurasip Jour. on Applied Signal Processing - Special Issue on Turbo Processing*, 2003.

[2] J. Kroll and n. Phamdo, "Source-channel optimized trellis codes for bitonal image transmission over awgn channels," *IEEE Trans. Image Processing*, pp. 899–912, July 1999.

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

[4] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *Proc. of Data Compression Conference*, March 1999.

[5] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inform. Theory*, pp. 1250–1276, June 2002.

[6] A. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, pp. 440–442, Oct. 2002.

[7] S. S. Pradhan and K. Ramchandram, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Tran. Inform. Theory*, pp. 626–643, Mar. 2003.

[8] Z. Tu, J. Li, and R. S. Blum, "Compression of a binary source using side information with concatenated convolutional codes," *submitted to IEEE Intl. Conf. Commun.*, June 2004.

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

[10] J. Bajcsy and P. Mitran, "Coding for the slepian-wolf problem with turbo codes," *Proc. of IEEE Globecom*, Nov 2001.

[11] A. Aaron and B. Girod, "Compression with side information using turbo codes," *Proc. of Data Compression Conference*, April 2002.

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

[13] JR G. D. Forney, "Trellis shaping," *IEEE Trans. Inform. Theory*, pp. 281–300, Mar. 1992.

[14] G.-C. Zhu and F. Alajaji, "Turbo codes for nonuniform memoryless sources over noisy channels," *IEEE Communication Letters*, vol. 6, pp. 64–66, Feb. 2002.

[15] G.-C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, "Non-systematic turbo codes for non-unifrom iid sources over awgn channels," *Proc. CISS*, March 2002.

[16] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, p. 248, Mar. 1974.