How Optimal Is Algebraic Binning Approach: A Case Study of the Turbo-Binning Scheme With Uniform and Nonuniform Sources

Jing Li (Tiffany), Zhenyu Tu and Rick S. Blum
Department of Electrical and Computer Engineering
Lehigh University, Bethlehem, PA 18105
{jingli, zht3, rblum}@ece.lehigh.edu

***Abstract*— This paper investigates the optimality of the binning approach in distributed source coding for both uniform and nonuniform sources. While the algebraic binning scheme is optimal for uniform sources both asymptotically and at finite lengths, it is shown that the optimality holds only asymptotically for nonuniform sources. High-performance turbo codes are used with the binning scheme on several source distributions to quantify how close they can get to the theoretical limit with relatively large block sizes. For nonuniform sources, optimal code design and variable-length bin-indexes are exploited as a useful extension to the conventional binning scheme. It is shown that the two strategies combined can improve the compression rate by as much as 0.22 bit/symbol for highly biased sources.**

I. Introduction

The syndrome/coset/binning scheme used in the proof of the Slepian-Wolf boundary in distributed source coding (DSC) [1] provides a generic approach for asymmetric compression where one source is assumed losslessly available at the decoder (e.g. via conventional entropy-achieving compression method) and the other is compressed as much as possible. This paper studies the optimality of the binning approach with binary memoryless sources that are either uniformly or nonuniformly distributed. That the binning scheme is optimal for uniform sources both asymptotically and at finite lengths is well-established [1][2]. The case of nonuniform sources, however, is much less studied. It should be noted that nonuniform sources are not uncommon in real life. For example, many binary images (e.g.. facsimile images) may contain as much as 76% of redundancy which corresponds to a source distribution of $p_0 = 0.96$ and $p_1 = 0.04$ [3]. For most communication and signal processing problems, it can be assumed that a front-end compression will be performed to get rid of the source redundancy before the intended signal processing and/or transmission. For distribued source coding, however, such a pre-process will either ruin the cource correlation or make the correlation analytically intractable and, hence, is not possible.

We first show that, while the generic *binning concept* does not make any assumption on the underlying source distribution and is in principle optimal regardless the uniformity of the sources, in practice, *the algebraic binning scheme* using linear codes is optimal for nonuniform sources only asymptotically. Specifically, we show that the nonuniformity in the source distribution and the geometry uniformity of a linear code (which is required by the binning construction) present two factors that oppose each other, causing a loss in compression rate unless the length of source sequences goes to infinity. Next, we show that, by exploiting optimal code selection and variable-length bin-indexes, the suboptimality of the binning approach (for nonuniform sources) can be mitigated. To give a quantitative feel of how much can be achieved, we explore high-performance turbo codes with the algebraic binning scheme [4][5] for several source distributions. For uniform sources, as shown in [4][5], the turbo-binning scheme can perform as close as 0.07 bit/symbol from the theoretic limit with fairly large block sizes. For (highly) nonuniform sources, we show that not using the proposed strategies (i.e. optimal channel code and variable-length bin-indexes) sees a huge gap (e.g. 0.36 bit/symbol) between the achievable compression rate and the theoretical limit. Using these remedies can close the gap by as much as 0.22 bit/symbol, but the performance is nevertheless 0.14 bit/symbol away from the limit.

The rest of the paper is organized as follows. Section II introduces the system model and the Slepian-Wolf boundary. Section III discusses the theoretical binning concept and the practical binning scheme, and analyzes their optimality with uniform and nonuniform sources. Section IV discusses the turbo-binning scheme to quantify the gap between the achievable performance and the theoretical results. Finally Section V concludes the paper.

This material is based on research supported by the Air Force Research Laboratory under agreement No. F49620-03-1-0214, by the National Science Foundation under Grant No. CCR-0112501, and by a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development, through the Pennsylvania Infrastructure Technology Alliance (PITA).

II. System Model and the Slepian-Wolf Boundary

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 problem of distributed source coding is to devise efficient ways to separately encode/compress but jointly decode/decompress the sources. Mathematically, this is to find a triple of mappings (f, g, ϕ) where the encoders f and g map X^n and Y^n into some codeword sets $\mathcal{A}$ and $\mathcal{B}$, respectively, and the decoder ϕ maps $\mathcal{A} \times \mathcal{B}$ back into X^n and Y^n. The famous Slepian-Wolf theorem shows that as long as the joint distribution $P_{x,y}$ is known to the encoders, separate encoding can reach the same compression rate as jointly encoding [1]. The achievable rate region is given by the Slepian-Wolf boundary [1]:

$$R_x \geq H(X|Y), \;\; R_y \geq H(Y|X), \;\; R_x + R_y \geq H(X,Y). \tag{1}$$

Specifically, the corner points of the Slepian-Wolf boundary, i.e. asymmetric compression, can be effectively transformed to a channel coding problem with decoder side information (SI). The *equivalent transmission channel* is specified by the correlation between the two sources (e.g. $P(Y|X)$).

In a general setup, correlation between two binary i.i.d. 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 correspondingly a binary asymmetric channel (BAC). Since content-dependent crossover probabilities make the channel difficult to analyze, here we consider a subset of the general problem by imposing a symmetry condition on the source correlation: $P(Y \neq X|X = 0) = P(Y \neq X|X = 1)$. This translates the virtual channel to a binary symmetric channel (BSC). We use $q \overset{\Delta}{=} P(Y \neq X)$ to denote the crossover probability of the equivalent BSC $X \rightarrow Y$, and use $p_0 \overset{\Delta}{=} P(x=0)$ to denote the distribution of source X. We refer to the case of $p_0 = 0.5$ as "uniform source DSC", and "nonuniform source DSC" otherwise. Clearly, the distribution of Y is irrelevant since Y is treated as the side information that is losslessly available at the decoder.

III. The Binning Approach

To solve the DSC problem, in theory, only a codebook that specifies the (optimal) mappings: $f : X^n \rightarrow \mathcal{A}$, $g : Y^n \rightarrow \mathcal{B}$ and $\phi : \mathcal{A} \times \mathcal{B} \rightarrow X^n \times Y^n$, is needed. In practice, however, a pair of practically-implementable encoder and decoder (i.e. with manageable complexity) is also needed. The former can usually be approached using typical sequences. The latter is helped by the algebraic binning scheme first proposed in [1]. The algebraic binning scheme, through the use of linear channel codes, provides a simple and general framework for constructing encoder/decoder pairs as well as defining code-books.

A. The Generic Binning Concept

For binary i.i.d. sources $X \in \{0,1\}^n$, $Y \in \{0,1\}^n$ (n can be either finite or infinite), the combined information content is given by the joint entropy $H(X^n, Y^n) = nH(X, Y)$. The generic binning concept refers to the idea of using 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. Clearly, $nH(X|Y)$ bits are needed to specify a particular bin and $nH(Y)$ bits to specify a particular sequence in the bin.

B. The Algebraic Binning Scheme

The above binning concept is practically implemented by exploiting the uniformity (regularity) of the code space of a linear code. By grouping source sequences (i.e. codewords of the linear code) into bins/cosets and transmitting the short bin-index (i.e. syndrome of the linear code) instead of the long source sequence, compression is achieved. The key steps are summarized below:

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. It follows naturally that the 2^{n-k} syndromes of length $n-k$ each can be used to index the subspaces/bins[1].

Encoder: The encoder (Fig. 1) is essentially a *syndrome former* (SF) which maps a codeword sequence to its syndrome/bin-index, and thus achieves a compression rate of $n : (n-k)$.

Decoder: The decoder (Fig. 1) 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/bin-index. 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 decoder can decode codewords on the equivalent channel with (near-)zero error probability, then its output, when subtracting $\tilde{X}^n$, will almost surely recover the original source sequence X^n. The efficacy

[1] It should be noted that the assignment of syndromes to bins can be random as long as the all-zero syndrome is assigned to the subspace that contains all the valid codewords [5].

of this process is warranted by the fact that the same distance property is preserved in each bin (due to the geometric uniformity of a linear code), and detailed discussion can be found, for example, in [5].

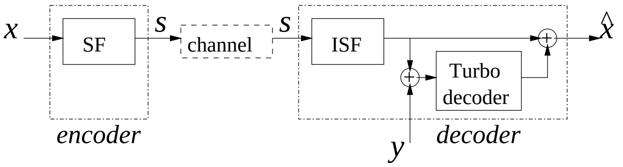


Fig. 1. Encoder and decoder structure using the algebraic binning approach.

C. Optimality Analysis

Comparing the generic binning concept with the algebraic binning scheme, it becomes clear that, to achieve an overall optimality, the process of assigning bin-indexes needs to achieve "entropy compression" for the bins, where each bin is associated with 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 that the cumulative probability of all codewords in any bin is 2^{k-n}.

Uniform Sources: Clearly, the requirement for balanced bins is automatically fulfilled when the source distribution is uniform (i.e. $p_0 = 0.5$) and the algebraic binning approach is therefore optimal (for both finite and infinite lengths). It should be pointed out that "the optimality of the binning approach" and "the achievability of the theoretic limit" are two related but different concepts. To achieve maximal compression rate in a DSC setup using channel codes, two key issues need to be resolved: (i) converting the source coding problem to an equivalent channel coding problem and (ii) finding a capacity-approaching channel code for the equivalent transmission channel. The former refers to the bridging work that brings the solution of channel coding to serving the problem of source coding, and the latter should certainly take advantage of the rich literature available on channel coding research. Apparently, the binning approach is an efficient and general solution for the former. Hence, for uniform sources, the optimality of the binning approach, together with an optimal channel code on the equivalent channel, will guarantee the achievability of the theoretic limit. It should also be noted that none of the above concepts has assumed infinite lengths. In fact, [2] presents an neat example where the algebraic binning scheme using a $(3,1)$ repetition code (finite length) is shown to achieve the compression limit for two i.i.d. binary uniform sources with a particular correlation. Nonuniform Sources: When the algebraic binning scheme is used for nonuniform sources, no matter what linear channel code is used (random or structured), except for the asymptotic case where there are infinite number of codewords in each bin, the cumulative probabilities of codewords in different bins will be different. Hence, the practice of using fixed-length syndromes to index bins, although stems naturally from the structure of a linear code, is suboptimal for any finite length. Instead, variable-length bin-indexes need to be assigned according to the bin probabilities in order to get close to the optimal compression rate ($nH(X|Y)$ for the bins). A one-step implementation of this idea is difficult, but a two-step approach is straight-forward. That is, following the fixed-length bin-index assignment, a conventional entropy-approaching compression method can be used to further compress bin-indexes/syndromes. As we shall see in the turbo-binning example, this second step of compressing fixed-length bin-indexes to their entropy can be critical for highly biased source distributions.

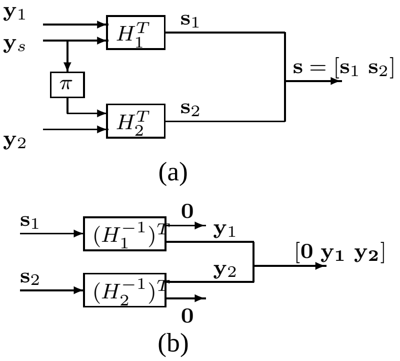
One can also view the optimality issue from the perspective of typical sequences. From the previous discussion on the binning concept, we know that typical sequences are used to describe sources. For uniform sources, all codewords are typical sequences and will thus *all* go into bins. For nonuniform sources, however, only a (small) *subset* of all codewords are typical sequences (those sequences whose possessions of 0's and 1's agree with the respective probabilities in the source). The use of a small typical set to describe the entire space suggests that some form of entropy compression is imperative. This element is unfortunately not inclusive in the algebraic binning practice.

IV. A Case Study of the Turbo-Binning Scheme

A. The Turbo-Binning Scheme

As a useful supplement to the above general discussion, we conduct a case study on turbo codes to give a quantitative feel of how far the practical performance is from the theoretical limit for finite-length uniform and nonuniform sources. The reason for using turbo codes are three-fold: (i) turbo codes are very powerful channel codes which exhibit perform stably and uniformly well on a variety of channels; (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 (iii) the length of a turbo code can be easily changed, making it possible to track and adapt to the varying correlation between sources.

To employ turbo-binning approach, we need to firstly construct a syndrome former and its matching inverse syndrome former. The construction of a valid SF-ISF pair is a straightforward task with most blocks codes and coset codes, but less so with turbo codes due to the random interleaver in the code structure. The turbo-DSC scheme by Liveris, Xiong and Georghiades [6] is the first to overcome the random interleaver problem but does not make explicit use of the binning scheme. The implicit binning approach therein involves merging a principle trellis with a complementary trellis to construct a source coding trellis that contains parallel branches. If the component RSC code has rate $1/k$, the resulting source coding trellis will have 2^{k-1} parallel branches between a pair of states. Encoding is performed by a walk through the trellis (or the corresponding state diagram), and decoding requires a modified turbo decoder to accommodate the time-variant trellis [6]. A simpler and more efficient approach as well as the first approach to explicitly exploit the binning scheme by constructing SF-ISF pairs for parallel turbo codes is given in [4]. The turbo-binning scheme therein does not require redesign of the code structure nor modification of the turbo decoder. It is directly applicable to all existing turbo codes including asymmetric turbo codes[2] [4] and, hence, allows the rich literary available on turbo codes to serve directly the DSC problem at hand.



(a)

(b)

Fig. 2. The SF-ISF pair for a turbo code. (a) Structure of the syndrome former. (b) Structure of the inverse syndrome former.

Fig. 2 illustrates the structure of a valid SF-ISF pair for a parallel turbo code that is based on (sub) SF-ISF pairs of the constituent RSC codes [4][5]. The SFs of the RSC codes, denoted as ${H_1}^T$ and ${H_2}^T$, are simply the transfer polynomials/matrices of the respective convolutional codes [8]. (A transfer matrix/polynomial of a convolutional code, H^T, is defined as $GH^T = \mathbf{0}$, where G is the generator polynomial/matrix of the convolutional code [8].) The ISFs of the RSC codes, denoted as $(H_1^{-1})^T$ and $(H_2^{-1})^T$, are left inverses of their respective SFs, where superscript T refers to the matrix/vector transpose operation, and superscript -1 refers to the left inverse operation of a matrix/vector. Whereas the choice of SF-ISF for a convolutional code is not unique, it should be emphasized that, in order for the simple structure in Fig. 2 to work (i.e., to avoid the potential problem of "systematic bits misalignment" due to the random interleaver), the ISFs of the constituent RSC codes need to always find the codeword with the all-zero systematic bits in the bin for any given bin-index. This restricts the format of the ISF to $(H^{-1})^T = [\mathbf{0},\ \mathbf{J}]$, where J is a square matrix. For example, for the common case where the constituent RSC code has generator polynomial $G(D) = [1, U(D)/V(D)]$, the corresponding SF-ISF pair takes the form of $H^T = [U(D)/V(D),\ 1]$ and $(H^{-1})^T = [0,\ 1]$. Due to the space limitation, detailed discussion is skipped. Interested readers please refer to [4][5].

B. Optimal Code Selection

As discussed above, the channel code in use needs to be carefully selected, since its performs directly affects the overall compression rate. For BSC channels with uniform sources, long turbo codes are known to perform very close to the channel capacity. For nonuniform sources, however, turbo codes are less well performing.

First we note that turbo codes as well as other linear channel codes are inherently suboptimal for nonuniform sources. In a pure channel coding problem, nonuniform sources can be passed through a nonlinear source-shaping code before getting to a linear error correcting code[3]. Such a treatment, however, is not possible with the DSC problem. Recall that the channel code in a DSC-binning approach plays a dual role: (i) to conduct error correcting on the equivalent transmission channel and (ii) to specify how codewords should be grouped into bins. The second role requires the code to be linear in order to preserve the same distance properties in each bin (a pre-requisite for the algebraic binning scheme to work). Clearly, the combination of the source-shaping code and the error correcting code results in an overall nonlinear code and, hence, is not applicable in DSC.

Despite the fundamental sub-optimality, turbo codes can, subject to the individual code space mapping, exhibit different error correcting behaviors with nonuniform sources. Specifically, the work of [9] and [10] reveals 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. This suggests that code selection for nonuniform sources needs to adopt different criteria from that of uniform sources.

Optimal code selection on Gaussian and Rayleigh fading channels is discussed in [9][10] and that on BSC channels is discussed in [11]. The basic method is via computer search. Due to the time and complexity involved, only 16-state turbo codes are considered.

[2] Asymmetric turbo codes have non-identical component codes, and bear certain advantage in terms of joint optimization of both the error floor and the waterfall region [7].

[3] A nonlinear source-shaping code can be implemented, for example, using a table-lookup encoder and a maximum likelihood decoder.

We employ a constrained iterative search, i.e., 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 [11]. Results from this search procedure reveal that, although the $(37, 21)$ Berrou code exhibits remarkable waterfall region performance with uniform sources, the best turbo codes for BSC channels with source distributions $p_0 = 0.7, 0.8, 0.9$ are the $(25, 23)$ code and the $(21, 23)$ code. Their performances, together with that Berrou code, are compared in Fig. 3 for $p_0 = 0.7$. We observe that the two winning codes exhibit very similar simulation performances with $(25, 23)$ marginally better for $p_0 = 0.7$ and $(21, 23)$ marginally better for $p_0 = 0.9$. Hence, we avoid making such statements as which is *the* best code for which source distribution.

In addition to careful selection of the code polynomial, the *a posteriori* probability (APP) decoding of the constituent RSC code (the BCJR algorithm) also needs to account for the (nonuniform) source distribution. That is, the knowledge of the source distribution $P(a_k)$ needs to be exploited as *a priori* information in *every* decoding iteration [11]. The *a priori* message content (in the form of log-likelihood ratio) due to source distribution is computed using $L_{ap}(a_k) = \log \frac{p_0}{1-p_0}$.

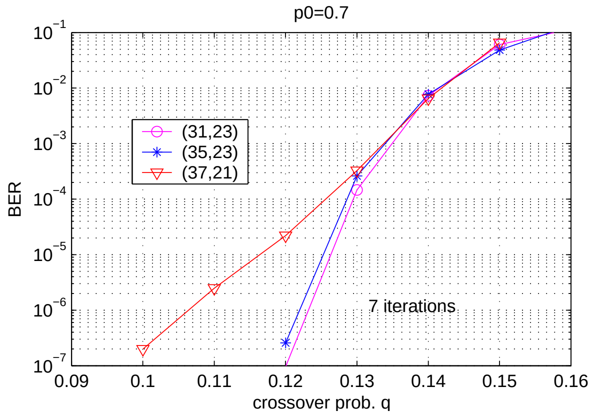


Fig. 3. Performance comparison of $(35, 23)$, $(31, 23)$, and $(37, 21)$ turbo codes on BSC(q) channels with source distribution $p_0 = 0.7$.

C. Variable-Length Bin-Indexes

For uniform sources, fix-length bin-indexes are optimal; for nonuniform sources, variable-length bin-indexes are desirable. The question then is how much gain variable-length bin-indexes have over their fixed-length peers, or whether it is worth the trouble of further compression. The answer to this question clearly depends on a given source distribution and the specific channel code (bin structure) in use. To give a quantitative feel, here we take a $(31, 23)$ turbo code, the winning code from our computer search, as an example. Tab. I lists the entropy rate of its syndrome bits obtained using the SF-ISF pair presented in Fig. 2 [4][5]. Again, due to the space limitation, the computation steps are omitted and only the results are reported.

Tab. I shows that, when source distribution is near uniform ($p_0 \to 0.5$), syndrome entropy is close to 1, the maximum value, and fixed-length syndromes/bin-indexes are sufficient for practical purposes. However, when the source becomes 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 as treated as if they were i.i.d.. Due to the correlation among syndrome bits, the normalized entropy of the *syndrome sequence*, $\frac{1}{n-k}H(S^{n-k})$, is actually lower than $H(S)$. 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 deploy the correlation among syndrome bits, the entropy of the syndrome bits serves as a fair evaluation of how much gain variable-length syndromes/bin-indexes have over their fixed-length peers.

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

D. Simulations with Uniform Sources

For uniform sources, we simulate the performance of the turbo-binning scheme using a rate-1/3, 8-state turbo code with the same constituent codes as in [12][6]: $(18, 13)$. S-random interleavers of length 10^4 and 10^3 are used, ten turbo decoding iterations are performed before the turbo decoder outputs its estimates, and appropriate clip-values are applied to avoid numerical overflows/downflows in the turbo decoder. Tab. II lists the simulation results where n denotes the interleaver length, and q the crossover probability. The interleaving gain can be easily seen from the table. If a normalized distortion of 10^{-6} is considered near-lossless, then this turbo coding scheme can work for a virtual BSC with $q = 0.145$. Since the compression rate is 2/3, there is a gap of only $2/3 - H(0.145) = 0.07$ from the theoretical limit, which is quite impressive [4][5]. This gap is noticeably smaller than those reported [6][12][4]. To get even closer to the limit, a longer turbo code with a larger memory size can be used.

[4] The performance reported in [6] and [12] are 0.09 and 0.15 from the limit, respectively. They have the same interleaver size as what is used in this paper, but different code rate.

E. Simulations with Nonuniform Sources

Tab. III 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, the performance of the Berrou code with fixed-length bin-indexes is also included. In the table, "P_0" specifies the distribution of source X, "Attainable q" refers to the largest $q = H(Y|X)$ (i.e. the amount of the source correlation) that the turbo-binning scheme can support for a compression distortion of 10^{-6} or less, and "$H(X|Y)$" denotes the corresponding theoretical limit for compressing source X (i.e., $H(Y|X) = q_{attainable}$, $P(X = 0) = P_0$ and Y is losslessly available). 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. As can be seen from the table, employing the proposed strategies (optimal code selection and variable-length bin-indexes) has significantly improved the overall compression rate. Specifically, for the highly nonuniform source like $p_0 = 0.9$, the two strategies combined can achieve an additional compression rate of as much as $0.3632 - 0.1444 = 0.2188$ bit/symbol! Nevertheless, the gap to the theoretical limit is in the range of 0.12 to 0.14 bit/symbol for p_0=0.7 to 0.9, which is noticeably larger than that of the uniform source case.

TABLE II

PERFORMANCE OF THE TURBO-BINNING SCHEME WITH UNIFORM SOURCES

Crossover Prob.	Distortion	
q	$n = 10^3$	$n = 10^4$
0.110	1.5×10^{-6}	-
0.140	8.0×10^{-4}	4.1×10^{-7}
0.145	4.0×10^{-3}	6.4×10^{-7}
0.150		8.3×10^{-6}
0.155	3.5×10^{-2}	3.9×10^{-3}

TABLE III

PERFORMANCE OF THE TURBO-BINNING SCHEME WITH NONUNIFORM SOURCES

source dist.	Berrou Code + Fixed-Length Bin-Indexes		
p_0	Attainable q	$H(X\|Y)$	**Gap A**
0.7	0.139	0.5239	**0.1427**
0.8	0.139	0.4435	**0.2232**
0.9	0.136	0.3035	**0.3632**

source dist.	Optimal Code + Variable-Length Bin-Indexes			
p_0	Attainable q	$H(X\|Y)$	**Gap B**	**Cap C**
0.7	0.143	0.5330	**0.1337**	**0.1213**
0.8	0.143	0.4507	**0.2159**	**0.1522**
0.9	0.141	0.3090	**0.3574**	**0.1444**

V. CONCLUSION

We have studied the optimality of the binning approach for asymmetric compression of binary i.i.d. sources. To illustrate exactly how much can be achieved and is yet to be achieved with practical systems, a case study of the turbo-binning scheme is conducted for both uniform and nonuniform sources. Blow summarizes the main results:

- The algebraic binning scheme based on linear codes is optimal at both finite and infinite lengths for uniform sources, but only at infinite lengths for nonuniform sources. The suboptimality with finite-length nonuniform sources is (in part) due to the practice of assigning fixed-length bin-indexes to bins with unequal probabilities.
- It is possible for a turbo code to outperform its peers on uniform sources but falls behind on nonuniform sources. For highly nonuniform sources with distribution $p_0 = 0.7 \sim 0.9$ on binary symmetric channels, $(35, 23)$ and $(31, 23)$ turbo codes are among the best.
- The conventional algebraic binning scheme using turbo codes can get very close to the theoretical limit for uniform sources, but not nearly so for nonuniform. Simple strategies like employing optimal code selection and variable-length bin-index assignment can significantly close up the gap, especially for highly biased sources.

REFERENCES

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

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

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

[4] Z. Tu, J. Li, and R. S. Blum. Compression of a binary source using side information with concatenated convolutional codes. *submitted to IEEE GLOBECOM.*, 2004.

[5] 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.

[6] 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.

[7] O. Y. Takeshita, O. M. Collins, P. C. Massey, and D. J. Costello. A note on asymmetric turbo-codes. *IEEE Commun. Letters*, pages 69–71, Mar. 1999.

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

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

[10] 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.

[11] J. Li, Z Tu, and R. Blum. Slepian-wolf coding for nonuniform sources using turbo codes. *to appear in Proc. Data Compression Conference*, Mar. 2004.

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