Adaptive Binary Slepian-Wolf Decoding using Particle Based Belief Propagation

Lijuan Cui, Shuang Wang, Samuel Cheng and Mark Yeary
School of Electrical and Computer Engineering
University of Oklahoma, Tulsa and Norman, OK
Email: {lj.cui, shuangwang, samuel.cheng, yeary}@ou.edu

***Abstract*—A major difficulty that plagues the practical use of Slepian-Wolf (SW) coding (and distributed source coding in general) is that the precise correlation among sources needs to be known *a priori*. To resolve this problem, we propose an adaptive asymmetric SW decoding scheme using particle based belief propagation (PBP). We explain the adaptive scheme for asymmetric setup in detail and then further extend it to the non-asymmetric setup based on the code partitioning approach. Moreover, we introduce a Metropolis-Hastings (MH) algorithm in the resampling step, which efficiently decreases the number of simulation iterations. We show through experiments that the proposed algorithm can simultaneously reconstruct the compressed sources and estimate the joint correlation among sources. Further, comparing to the conventional SW decoder based on standard belief propagation, the proposed approach can achieve higher compression under varying correlation statistics.**

***Index Terms*—Adaptive decoding, Distributed algorithms, Source coding, Data compression**

I. Introduction

Slepian-Wolf (SW) coding is a technique to losslessly compress correlated remote sources separately and decompress them jointly [1]. To the surprise of many researchers of their time, Slepian and Wolf showed that it is possible to have no loss in sum rate, even though only separate encoding is allowed. Thus, at least in theory, it is possible to recover the source losslessly at the base station even though the sum rate is barely above the joint entropy of the sources.

Wyner is the first who realized that by taking computed syndromes as the compressed sources, error-correcting parity check codes can be used to implement SW coding [2]. The approach was rediscovered and popularized by Pradhan *et al.* more than two decades later [3]. Numerous channel coding based SW coding schemes have been proposed [3], [4], [5]. Noticeably, by using efficient channel codes such as the Low-Density Parity-Check (LDPC) codes, it is possible to compress a joint binary source very closed to the SW limit (i.e., the joint entropy) [6], [7]. However, the fundamental assumption is that the correlation statistics needs to be known accurately *a priori*.

Actually in many real applications, such as a sensor network which is widely used for environmental monitoring of temperature, pressure and humidity, or real-time area video surveillance, the correlation statistics among sensors cannot be obtained easily. In general, the correlations among sensors may vary over both space and time. Since the decoding performance of distributed source coding (DSC) relies on the knowledge of correlation very much, the design of an online estimation scheme of correlation for sensor network becomes a significant task both in theoretical study and practical applications.

In this paper, we propose an adaptive LDPC code based SW decoder using particle based belief propagation (PBP) to simultaneously reconstruct a compressed source and estimate the joint correlation between the sources. Our proposed algorithm works well for the sources which have a slowly changing correlation and is carried out based on factor graph [8], [9], which affords great flexibility in modeling our problems. We show that the proposed algorithm no longer depends on the initial estimation of the correlation parameters and offers accurate real-time estimations of the parameters. For different code rates, our algorithm shows a lower decoding error rate (and thus a more efficient compression) than that of the standard belief propagation (BP) algorithm.

Since the close relationship between SW coding and channel coding, the proposed approach can also be used for channel state estimation (for example, see our prior work in [10]). Unlike in channel coding that channel state information can be estimated with the help of a pilot signal, this pilot method cannot be used for SW coding and DSC in general since sources in DSC are specified by the problems themselves and are not controllable by users.

The rest of the paper is structured as follows. Related work will be discussed in Section II. The precise problem formulation will be described in Section III. We will review the standard BP in Section IV-A and describe the proposed PBP in Section IV-B. The proposed PBP decoding for asymmetric and non-asymmetric SW coding will then be explained in Sections IV-C and IV-D, respectively. In Section V we present simulation results, and we conclude in Section VI.

II. Related Work

Correlation parameter estimation for LDPC-based SW coding is proposed in [11], [12], [13]. In [12], the residual redundancies in LDPC syndromes are used to estimate the crossover probability between two correlated binary sources using Mean-Intrinsic-LLR. However, these algorithms [11], [12] work only for highly correlated sources. In [13], the Expectation Maximization (EM) algorithm was used to estimate the correlation between two sources in SW coding. However, unlike our setup, the parameter is assumed to be constant and does not change within the code block.

Some other related works are based on rate-adaptive approaches, where an encoder transmits few syndrome or parity bits to start with and gradually increases the number of syndrome or parity bits if decoding fails [14]. However, this is possible only when a feedback channel is allowed.

Comparing to the aforementioned prior works, we consider a rather different problem where the decoder tries its best to recover the source even when the correlation is unknown. As a by-product, the correlation is accurately estimated (see our prior work in [15]). This information is useful in applications such as distributed video coding, where it can aid the estimation of motion vectors [16].

Generally, the standard BP algorithm can only handle discrete variables with small alphabet sizes. However, in many problems, variables may have large alphabet sizes or may even be continuous, e.g. a continuous correlation parameter between two sources. In these situations, the standard BP is not applicable. However, integrating particle methods into BP provides a way for BP to handle continuous variables. J. Dauwels *et. al* incorporated particle methods into message passing [17], and also used it for phase estimation in channel coding [18]. Moreover, other particle based message passing algorithms [19], [20], are also studied for continuous variable problems.

The main contribution of this paper is to propose the first adaptive asymmetric and non-asymmetric SW coding schemes that can perform online estimation of the correlation among sources while decoding. Correlation estimation at the decoder is essential for practical implementation of SW coding since the encoders, which have direct access of the original sources, cannot communicate to each other and thus cannot perform correlation estimation.

III. Problem Formulation

Let X and Y be two correlated binary sources (taking values 0 and 1) and the correlation between them be modeled by a binary symmetric channel (BSC) with unknown crossover probability p. Namely, $X = Y \oplus Z$, where $\oplus$ is the bitwise addition ("exclusive or ") operation and

$$Z = \begin{cases} 0, & \text{with probability } 1-p, \\ 1, & \text{with probability } p. \end{cases} \quad (1)$$

Moreover, we assume that the crossover probability p may drift over time but will not change too rapidly.

When only one of the two sources, let say X, is compressed whereas the other source is taken as side information at the decoder, we refer to this case as *asymmetric* SW coding. When both sources are compressed, we refer to this as *non-asymmetric* SW coding [21], [22]. Apparently, asymmetric SW coding is a special case of non-asymmetric SW coding. In either case, there is no loss comparing to joint encoding under some mild conditions, and thus, the total rate required in theory is $H(X,Y)$. In particular, for the asymmetric case, since we expect $H(Y)$ bits per sample are needed to compress the side information Y independently, the rate required for compressing X is $H(X,Y) - H(Y) = H(X|Y)$.

Throughout the paper, we use an upper case letter to represent a random variable and the corresponding lower case letter to indicate the realization of the variable. Bold letters are reserved for vectors. We use factor graphs [8] to formulate our algorithms. Using the usual convention, a variable node that specifies an unknown is denoted by a circle and a factor node that specifies the "correlation" among multiple variable nodes is denoted by a square. The name factor graph comes from the fact that the joint probability function can be expressed as the multiple of the factor functions of the factor nodes. Moreover, we use $N(a)$ to represent the set of neighbors for a node a. For a factor node a, we use $\mathbf{x}_a$ to indicate all variables connecting to a. That is, $\mathbf{x}_a = (x_i | i \in N(a))$.

IV. Application of Particle Based Belief Propagation Algorithm in SW Coding

A. Review of Belief Propagation Algorithm

The BP algorithm is an approximate technique for computing marginal probabilities by exchanging the message between neighboring nodes. Denote $m_{a\to i}(x_i)$ as the message sent from a factor node a to a variable node i, and $m_{i\to a}(x_i)$ as the message sent from a variable node i to a factor node a. Loosely speaking, $m_{a\to i}(x_i)$ and $m_{i\to a}(x_i)$ can be interpreted as the beliefs of node i taking the value x_i transmitting from node a to i and from node i to a, respectively. The message updating rules can be expressed as follows:

$$m_{i\to a}(x_i) \propto \prod_{c\in N(i)\backslash a} m_{c\to i}(x_i) \quad (2)$$

and

$$m_{a\to i}(x_i) \propto \sum_{\mathbf{x}_a \backslash x_i} \left(f_a(\mathbf{x}_a) \prod_{j\in N(a)\backslash i} m_{j\to a}(x_j) \right), \quad (3)$$

where $N(i)\backslash a$ denotes the set of all neighbors of node i excluding node a; f_a is the factor function for factor node a; $\sum_{\mathbf{x}_a \backslash x_i}$ denotes a sum over all the variables in $\mathbf{x}_a$ that are arguments of f_a except x_i. Moreover, the BP algorithm approximates the belief of node i taking x_i as

$$b_i(x_i) \propto \prod_{a\in N(i)} m_{a\to i}(x_i). \quad (4)$$

B. Particle based belief propagation algorithm

In the standard BP algorithm [8], (3) is generally intractable when variables are continuous or the alphabet sizes of variables are large, since the summation in (3) will have infinite number of terms. Thus, we introduce a PBP algorithm to solve this problem by combining BP with particle methods. The key idea of PBP is to model each continuous variable (or variable with large alphabet sizes) with K number of particles with associated weights, which just corresponds to K number of labels in the standard BP. Note that in standard BP only the belief of each label will be updated after each iteration, but in PBP both the value (i.e., location) of each label (i.e., particle) and the corresponding belief of each label will be updated after each iteration. Please note that these changes do not affect the sum-product message update rules described in the standard BP algorithm.

By introducing a distribution $W_j(x_j)$ (corresponding to the particle weights), we can rewrite (3) as an expectation form, which can be considered as importance-sampling transform of (3), as following[1]

$$m_{a\to i}(x_i) \propto \sum_{x_j\in\chi_j} f_a(x_j,x_i)\frac{m_{j\to a}(x_j)}{W_j(x_j)}W_j(x_j) \propto E\left[f_a(x_j,x_i)\frac{m_{j\to a}(x_j)}{W_j(x_j)}\right], \quad (5)$$

where E is the expectation with respect to the distribution $W_j(x_j)$. Then, the above message can be approximated by a list of K particles as

$$\hat{m}_{a\to i}\left(x_i^{(k)}\right) \propto \frac{1}{K}\sum_{l'=1}^{K} f_a\left(x_j^{(l')},x_i^{(k)}\right)\frac{\hat{m}_{j\to a}\left(x_j^{(l')}\right)}{W_j(x_j^{(l')})}. \quad (6)$$

Moreover, the distribution $W_j(x_j)$ can be chosen from the marginal distribution of variable x_j, which corresponds to the belief of this variable (see (4)). Additionally, locations and corresponding weights of particles have to be adjusted over time. This is achieved by using systematic resampling [23] and Metropolis-Hastings (MH) [24] random walk perturbation after each message update. The MH algorithm efficiently reduces the number of simulation iterations by half when comparing to the standard Gaussian random walk. In the following, the workflow of the PBP algorithm is described.

1) First, the weight of a particle $x_i^{(k)}$ will be computed as $b(x_i^{(k)})$, the belief of $x_i^{(k)}$ from standard BP, where $k = 1, 2, \cdots, K$.
2) Then K new samples, $\tilde{x}_i^{(1)}, \cdots, \tilde{x}_i^{(K)}$, will be drawn with probabilities proportional to $b(x_i^{(k)})$ using systematic resampling [23]. As a result, some $x_i^{(k)}$ that have small probabilities will be likely to be discarded whereas those with high probability will be repeatedly drawn.
3) To maintain the diversity of the particles, the particle locations will be perturbed by an MH [24] based Gaussian random walk, which consists of two basic stages. First, let the proposed new K particles at each iteration be $\hat{x}_i^{(k)} = \tilde{x}_i^{(k)} + Z_r$, that is the current value plus a Gaussian random variable $Z_r \sim N(0, \sigma_r^2)$. Second, decide whether the proposed values of new particles are rejected or retained by computing the acceptance probability $a\{\hat{x}_i^{(k)}, \tilde{x}_i^{(k)}\} = min\{1, \frac{p\left(\hat{x}_i^{(k)}\right)}{p\left(\tilde{x}_i^{(k)}\right)}\}$, where $\frac{p\left(\hat{x}_i^{(k)}\right)}{p\left(\tilde{x}_i^{(k)}\right)}$ is the ratio between the proposed particle value and the previous particle value. When the proposed value has a higher posterior probability than the current value $\tilde{x}_i^{(k)}$, it is always accepted; otherwise, it is accepted with probability a.
4) Based on the new particles, update messages and beliefs using standard BP.
5) Iterate steps 2 to 4 unless the maximum number of iterations is reached or other exit condition is satisfied.

[1]For ease exposition, we consider a factor function f_a with only two variables in our analysis. It is easy to extend the analysis to a factor function with more variables.

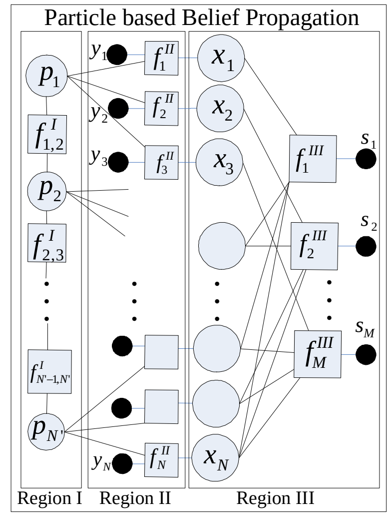

Fig. 1. Factor graph representation of the proposed PBP algorithm, where the superscripts of f indicate the factor nodes in Regions I, II, III, respectively.

C. Particle based BP for Asymmetric SW Coding

Our proposed approach is based on the syndrome based approach using LDPC code [6] as shown in Fig. 1 (see Regions II and III). At the encoder, a block of N input bits, $x_1, x_2, \cdots, x_N$, is compressed into M syndrome bits, $s_1, s_2, \cdots, s_M$, thus resulting in $M : N$ compression. The factor nodes f_l^{III}, $l = 1, 2, \cdots, M$ as shown in Region III of Fig. 1 take into account the constraint imposed by the received syndrome bits. For a factor node f_l^{III} in Region III, we define the corresponding factor function $f_l^{III}(\mathbf{x}_{f_l^{III}})$, as

$$f_l^{III}(\mathbf{x}_{f_l^{III}}) = \begin{cases} 1, & \text{If } s_l \oplus \bigoplus_{i\in N(f_l^{III})} x_i = 0, \\ 0, & \text{otherwise}, \end{cases} \quad (7)$$

where $N(f_l^{III})$ denotes the set of neighbors of factor node f_l^{III}, $\bigoplus_{i\in N(f_l^{III})}$ represents the bitwise sum of all elements x_i with $i \in N(f_l^{III})$, and for a factor node f_l^{III}, $\mathbf{x}_{f_l^{III}}$ indicate all variables connecting to f_l^{III}.

For the conventional SW coding, the correlation between a pair of sources, x_i and y_i, is handled by a correlation factor node f_i^{II}, $i = 1, 2, \cdots, N$ (see Region II of Fig. 1), where the corresponding factor function $f_i^{II}(y_i, x_i, p)$ is defined as

$$f_i^{II}(y_i, x_i, p) = \begin{cases} 1-p, & \text{If } x_i = y_i, \\ p, & \text{otherwise}. \end{cases} \quad (8)$$

With the variable and factor nodes defined and in place, one can estimate the values of $\mathbf{x}$ using the BP algorithm. While the source X can be compressed very closely to the SW limit $H(X|Y)$ in the classic BP approach [6], the crossover probability p is assumed to be constant and known *a priori*. The main contribution of our approach is to relax these constraints. Namely, we assume that p is unknown and varies slowly over time. To model this, we connect the factor node f_i^{II} to a variable $p_{i'}$, where $p_{i'}$ is now a variable instead of a constant. Thus, the factor function $f_i^{II}(y_i, x_i, p)$ in (8) will be updated to

$$f_i^{II}(y_i, x_i, p_{i'}) = \begin{cases} 1-p_{i'}, & \text{if } x_i = y_i, \\ p_{i'}, & \text{otherwise}. \end{cases} \quad (9)$$

We call the number of correlation factor nodes connecting to each $p_{i'}$ the connection ratio, which is equal to three in Fig. 1. The higher the connection ratio, the simpler the model and the fewer the number of hidden parameters [2].

Since we assume that p only varies slowly over time, the corresponding probability of any two variable nodes of $p_{i'}$ and $p_{i'+1}$ in Region I, should be close. This characteristic is captured by the p-factor nodes $f^I_{1,2}, f^I_{2,3}, \cdots, f^I_{N'-1,N'}$ as shown in Region I of Fig. 1, where a p-factor function $f^I_{i',i'+1}(p_{i'}, p_{i'+1})$ is defined as

$$f^I_{i',i'+1}(p_{i'}, p_{i'+1}) = \exp\left(-\frac{(p_{i'} - p_{i'+1})^2}{\lambda_{i',i'+1}}\right), \quad (10)$$

where the estimation of $\lambda_{i',i'+1}$ will be described in Remark 1.

With the factor functions defined in (7), (9), and (10), it may appear that the BP algorithm can be directly applied. However, $p_1, p_2, \cdots, p_{N'}$ are continuous and cannot be handled by standard BP (see also Section IV-B) Nevertheless, by applying PBP, we are able to handle even continuous variables.

As mentioned in Section IV-B, PBP handles continuous variables by modeling each $p_{i'}$ in Region I with K particles $p^{(1)}_{i'}, \ldots, p^{(K)}_{i'}$ and adjusting particle locations and weights according systematic resampling and MH random walk. Region II plays the role of connecting standard BP (Region III) and PBP (Region I) to exchange information between each other. The factor node message update from Region II to Region I can be written as

$$m_{f^{II}_i \to i'}(p^{(k)}_{i'}) \propto \sum_{x_i \in \{0,1\}} f^{II}_i\left(y_i, x_i, p^{(k)}_{i'}\right) m_{i \to f^{II}_i}(x_i), \quad (11)$$

while the factor node message update from Region II to Region III can be written as

$$m_{f^{II}_i \to i}(x_i) \propto \frac{1}{K} \sum_{k=1}^{K} f^{II}_i\left(y_i, x_i, p^{(k)}_{i'}\right) \frac{m_{i' \to f^{II}_i}\left(p^{(k)}_{i'}\right)}{W_{i'}(p^{(k)}_{i'})}, \quad (12)$$

where $f^{II}_i(y_i, x_i, p^{(k)}_{i'}) = \begin{cases} 1 - p^{(k)}_{i'}, & \text{if } x_i = y_i \\ p^{(k)}_{i'}, & \text{otherwise} \end{cases}$, and $W_{i'}(p^{(k)}_{i'})$ corresponds to the belief of particle $p^{(k)}_{i'}$. On one hand, we can see that the message $m_{i \to f^{II}_i}(x_i)$ from Region III is used to update the message $m_{f^{II}_i \to i'}(p^{(k)}_{i'})$ to Region I. Furthermore, the updated message in Region I can be used to update the value of each particle according to the belief $b\left(p^k_{i'}\right) \propto \prod_{f^{II}_i \in N(i')} m_{f^{II}_i \to i'}\left(p^{(k)}_{i'}\right)$. On the other hand, for Region III, not only the message from $m_{i' \to f^{II}_i}\left(p^{(k)}_{i'}\right)$ is used to update the message $m_{f^{II}_i \to i}(x_i)$, but also, more importantly, the updated value of each particle $p^{(k)}_{i'}$, which corresponds to the crossover probability, has played a role for updating the message $m_{f^{II}_i \to i}(x_i)$. Actually, updating message $m_{f^{II}_i \to i}(x_i)$ equals to the update of estimate of source correlation. Finally, by performing the aforementioned scheme iteratively, the source decoding and correlation estimation can be done simultaneously.

[2]To estimate a constant correlation as in other estimation algorithms [11], [12], [13], one can set the connection ratio equal to the code length N. The complexity of the PBP algorithm would be competitive with other aforementioned estimation algorithms in this degenerated case.

Remark 1. Generally, $\lambda_{i',i'+1}$ is taken as a predetermined value to simplify the problem. It may be beneficial to estimate $\lambda_{i',i'+1}$ for each factor node $f^I_{i',i'+1}$ in Region I to improve decoding performance. In our study, we utilize a similar method used for correlation estimation (see section IV-B) to estimate $\lambda_{i',i'+1}$ by sampling K particles $\lambda^{(1)}_{i',i'+1}, \cdots, \lambda^{(K)}_{i',i'+1}$, for each factor node $f^I_{i',i'+1}$. Here, we suppose that the change in $\lambda_{i',i'+1}$ has the same trend as the difference between the averages of the two p connecting to $\lambda_{i',i'+1}$. That is, define $\Delta p_{i',i'+1} = |\bar{p}_{i'} - \bar{p}_{i'+1}|$, where $\bar{p}_{i'} = \frac{1}{K}\sum_{k=1}^{K} p^{(k)}_{i'}$ is the mean location of all the particles in variable node of $p_{i'}$. A larger $\Delta p_{i',i'+1}$ means a greater probability of $\lambda^k_{i',i'+1}$ to take a larger value. To increase the stability and decrease computational overhead, one can perform $\lambda_{i',i'+1}$ estimation once after several number of PBP iterations, whereas $p_{i'}$ estimation is performed at each iteration as described in Section IV-B.

Remark 2. The complexity of BP increases linearly with the degree of a variable node but exponentially with the degree of a factor node. However, we can easily incorporate the "method" of passing log-likelihood ratios $L_{ai} \triangleq \log\frac{m_{a\to i}(0)}{m_{a\to i}(1)}$ instead of probabilities as messages to reduce the complexity for the factor node updates in Region III [25]. The resulting complexity will be linear with respect to code length [26]. Note that the same method cannot be used in general for factor nodes in Regions I and II since the method can only be used to variables with alphabet size of two and there are generally more than two labels for the variable there. For example, we generally use more than two particles to represent $p_{i'}$ (i.e., each $p_{i'}$ can take more than two values). However, this does not have significant impact to the complexity of the overall algorithm since the node degrees of the factor nodes in Regions I and II are only two as shown in Fig. 1.

D. Particle based BP for Non-asymmetric SW Coding

Different attempts have been made to implement non-asymmetric SW coding, which includes: time-sharing, source splitting [5], and code partitioning [4], [7]. However, like all aforementioned work, they assume the correlation statistics between the two sources is constant and known *a priori*.

The code partitioning approach effectively converts a SW coding problem into a channel coding problem. In [7], the code partitioning approach is implemented using irregular repeated accumulat (IRA) codes [27], a special case of LDPC codes. Being a form of LDPC codes, the IRA based SW coding can be decoded using BP, and the proposed PBP method can be directly applied. For completeness, a brief description about code partitioning approach is given as follows.

Let $H = [P|I] = [P_1P_2|I]$ be the parity check matrix of a systematic linear block code, where the widths of P_1 and P_2 are N_1 and N_2, respectively, and I is an identity matrix of size $M \times M$. Therefore, H is of size $M \times N$, where $N = N_1 + N_2 + M$.

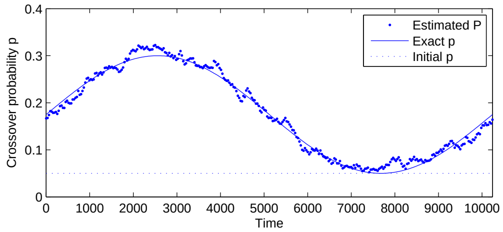

Fig. 2. Estimation of crossover probabilities for sinusoidal changing correlations.

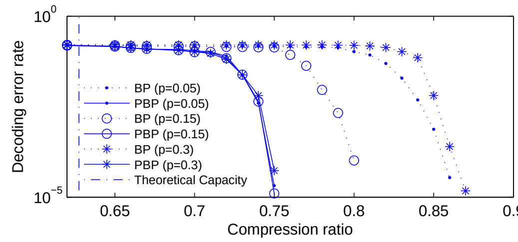

Fig. 3. Decoding bit error rate for a sinusoidal changing correlation.

Now, we can partition the code into two subcodes with parity check matrices $H_1 = \begin{bmatrix} I & 0 & 0 \\ 0 & P_2 & I \end{bmatrix}$ and $H_2 = \begin{bmatrix} 0 & I & 0 \\ P_1 & 0 & I \end{bmatrix}$. Length-$N$ blocks drawn separately from the two correlated sources, $\mathbf{x}$ and $\mathbf{y}$, will be compressed to $\mathbf{u} = H_1\mathbf{x}$ and $\mathbf{v} = H_2\mathbf{y}$, respectively. For the ease of explanation, let us split $\mathbf{x}$ into $\mathbf{x}^1$, $\mathbf{x}^2$ and $\mathbf{x}^3$, where their lengths are N_1, N_2 and M, respectively, and split $\mathbf{u}$ into $\mathbf{u}^1$ and $\mathbf{u}^2$ with lengths N_1 and M. Therefore, we have $\mathbf{u}^1 = \mathbf{x}^1$ and $\mathbf{u}^2 = P_2\mathbf{x}^2 + \mathbf{x}^3$. Similarly, $\mathbf{y}$ is split into $\mathbf{y}^1$, $\mathbf{y}^2$ and $\mathbf{y}^3$, and $\mathbf{v}$ is split into $\mathbf{v}^1$ and $\mathbf{v}^2$. This gives us $\mathbf{v}^1 = \mathbf{y}^2$ and $\mathbf{v}^2 = P_1\mathbf{y}^1 + \mathbf{y}^3$.

At the decoder, the received bits of $\mathbf{u}$ and $\mathbf{v}$ will be rearranged and padded with zeros into $\mathbf{t}^1 = \begin{bmatrix} \mathbf{u}^1 \\ 0_{N_2 \times 1} \\ \mathbf{u}^2 \end{bmatrix}$ and $\mathbf{t}^2 = \begin{bmatrix} 0_{N_1 \times 1} \\ \mathbf{v}^1 \\ \mathbf{v}^2 \end{bmatrix}$. Then, it can be easily verified that $\mathbf{t}' \triangleq \mathbf{t}^1 + \mathbf{t}^2 + \mathbf{x} + \mathbf{y} = \begin{bmatrix} I \\ P \end{bmatrix} \begin{bmatrix} \mathbf{x}^2 \\ \mathbf{y}^1 \end{bmatrix}$. Note that $\begin{bmatrix} I \\ P \end{bmatrix}$ is actually a generator matrix of the original code. Thus, $\mathbf{t}' = \mathbf{t}^1 + \mathbf{t}^2 + \mathbf{x} + \mathbf{y}$ is the codeword encoded from the message $\begin{bmatrix} \mathbf{x}^2 \\ \mathbf{y}^1 \end{bmatrix}$. We can rewrite $\mathbf{t}'$ as $\mathbf{t}' = \mathbf{t} + \mathbf{z}$, where $\mathbf{t} = \mathbf{t}^1 + \mathbf{t}^2$ and $\mathbf{z} = \mathbf{x} + \mathbf{y}$. Therefore, given $\mathbf{t}$ (corresponding to the side information used in asymmetric case), the decoder can recover $\mathbf{t}'$ by taking $\mathbf{t}$ as a corrupted codeword passing through a channel with noise $\mathbf{z}$. Given $\mathbf{x}^2$ and $\mathbf{y}^1$ (obtained from the decoded $\mathbf{t}'$), $\mathbf{x}^3$ and $\mathbf{y}^3$ can be solved accordingly from $\mathbf{u}_2 = P_2\mathbf{x}^2 + \mathbf{x}^3$ and $\mathbf{v}_2 = P_1\mathbf{y}^1 + \mathbf{y}^3$, whereas $\mathbf{x}^1$ and $\mathbf{y}^2$ can be read out from $\mathbf{u}_1$ and $\mathbf{v}_1$ directly. Finally, by combining all the decoded information, both sources $\mathbf{x}$ and $\mathbf{y}$ can be recovered. According to the aforementioned description, we can see the factor graphs used for the non-asymmetric case are the same as the asymmetric case, except replacing the side information $\mathbf{y}$ by $\mathbf{t}$, the decoding codeword $\mathbf{x}$ by $\mathbf{t}'$ and setting all the syndrome bits equal to 0. Then the inference problem for non-asymmetric case can be solved similarly as the asymmetric problem. More details about the implementation of non-asymmetric setup can be found in [28].

V. Experimental Results

We first studied the asymmetric case, where SW codes were randomly generated by a 6000×10240 parity check matrix and the variable node degree is equal to 3. Moreover, 16 particles were assigned to each variable node in Region I. For the random walk step, we assumed $\sigma_r^2 = 0.0001$. The following results were obtained by averaging the estimated crossover probability of 200 different codewords. Fig. 2 shows the estimated results of a sinusoidally changing correlation, where the crossover probability p changes sinusoidally from 0.05 to 0.3 for each input codeword bit. The results verified that our proposed algorithm can generate a good estimation of a complexly changing correlation.

Next we analyzed how different settings of parameters effected the decoding performance of our proposed PBP algorithms. The following performance results were obtained by averaging 10000 independent simulations, where the code length was equal to 10240. Moreover, the theoretical capacity is calculated according to the equation $C = \frac{1}{N}\sum_{i=1}^{N} 1 - H(p_i)$, where $H(p_i) = -p_i \log(p_i) - (1 - p_i)log(1 - p_i)$ and $N = 10240$. The value of crossover probability p_i changed sinusoidally from 0.05 to 0.3 in the BSC. The number of particles was also equal to 16.

In Fig. 3, we compared the decoding performance between our proposed PBP algorithm and standard BP algorithm by using different initial estimations of p, namely, $p = 0.05, 0.15$ or 0.3. We can see that the gain was relatively small when the initial estimation is close to the true value of the crossover probability (e.g. $p = 0.15$, which was roughly equal to the mean of the time changing crossover probability). However, when the initial estimation was far away from the true value, the observed gain was significant. In comparison, we can see that our PBP algorithm is not sensitive to the initial estimation of p, since the results showed that all the PBP simulations yielded similar decoding performance.

We then proceeded to study the non-asymmetric case. We tried to compare the performance of our adaptive decoding algorithm with conventional IRA decoding. We fixed the code rates for both X and Y to be 0.75. We then compared decoding performance of the two schemes while varying the correlation parameter p. Unlike the first case, we let p to be a constant over all samples. Initial estimations of p were set 0.1 and 0.2. As shown in Fig. 4 (a), the gain was relatively small when the estimation p was not too far from its true value. However, when the estimation deviated significantly from its true value, the observed gain was substantial.

Finally, we compared the two algorithms for the case when there was some minor fluctuation in p, where p varies sinusoidally from 0.05 to 0.07. We approximated the sum rate where lossless compression was achieved when the probability of error fell below 10^{-4}. The result is shown in Fig. 4 (b). We can see that the gain is rather significant even when the fluctuation of p is rather small.

VI. Conclusion

We proposed an adaptive decoding scheme for SW coding using particle based BP. The scheme has been tested for

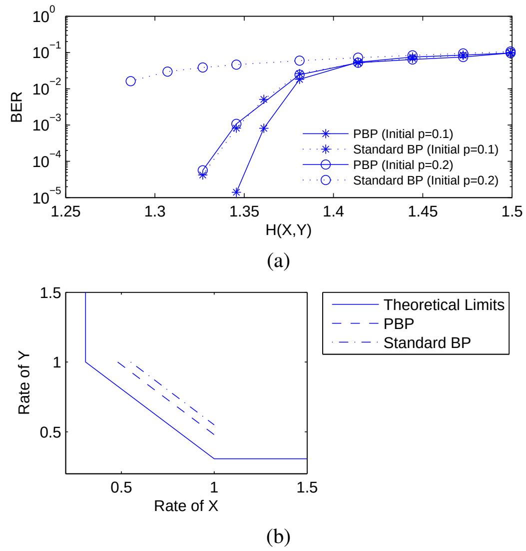

Fig. 4. (a) Change of BER as the correlation parameter p (and thus $H(X,Y)$) varies. Code length $(N) = 10,000$ and the rate pair is $(0.75, 0.75)$. The initial estimations of p are 0.1 and 0.2. (b) Comparing our adaptive decoding algorithm and standard BP with the theoretical limits of SW code, where the correlation is change sinusoidally with max value 0.07 and min value 0.05.

both asymmetric SW coding and non-asymmetric SW coding (with the latter case is attained by incorporating the code partitioning idea). From our experiments, a precise estimation of correlation between the two sources using our adaptive decoding algorithm has been observed. Thus, the decoding performance of our algorithm is no longer sensitive to the initial estimation of the correlation parameter p. Moreover, we have observed a significant gain of our algorithm over the standard BP algorithm even when there is a slight fluctuation of the correlation among sources.

References

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

[2] A. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inform. Theory*, vol. 20, pp. 2–10, Jan. 1974.

[3] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): design and construction," in *Proc. DCC*, 1999, pp. 158–167.

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

[5] B. Rimoldi and R. Urbanke, "Asynchronous Slepian-Wolf coding via source-splitting," in *ISIT'97*, Ulm, Germany, 1997, p. 271.

[6] A. Liveris, Z. Xiong, and C. 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] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades, "On code design for the Slepian-Wolf problem and lossless multiterminal networks," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1495–1507, 2006.

[8] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[9] H. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[10] L. Cui, S. Wang, S. Cheng, and Q. Wu, "Noise adaptive LDPC decoding using particle filters," in *Proc. CISS 2009*, Baltimore, MD, 2009.

[11] Y. Fang and J. Jeong, "Correlation parameter estimation for LDPC-based slepian-wolf coding," *IEEE Communications Letters*, vol. 13, no. 1, pp. 37–39, 2009.

[12] Y. Fang, "Crossover probability estimation using mean-intrinsic-LLR of LDPC syndrome," *IEEE Communications Letters*, vol. 13, no. 9, p. 679, 2009.

[13] A. Zia, J. Reilly, and S. Shirani, "Distributed Parameter Estimation with Side Information: A Factor Graph Approach," in *IEEE International Symposium on Information Theory*, 2007, pp. 2556–2560.

[14] J. Chen, A. Khisti, D. M. Malioutov, and J. S. Yedidia, "Distributed source coding using serially-concatenated-accumulate codes," *Information Theory Workshop, IEEE*, pp. 209–214, 2004.

[15] S. Cheng, S. Wang, and L. Cui, "Adaptive slepian-wolf decoding using particle filtering based belief propagation," in *47th Annual Allerton Conference on Communication, Control, and Computing*, Urbana-Champaign, IL, 2009.

[16] D. Varodayan, D. Chen, M. Flierl, and B. Girod, "Wyner-Ziv coding of video with unsupervised motion vector learning," *EURASIP Signal Processing: Image Communication Journal, Special Section on Distributed Video Coding*, 2008.

[17] J. Dauwels, S. Korl, and H. A. Loeliger, "Particle methods as message passing," in *Proc. IEEE Int. Symp. Inf. Theory*, 2006, pp. 2052–2056.

[18] J. Dauwels and H. A. Loeliger, "Phase estimation by message passing," in *IEEE International Conference on Communications*, vol. 1, 2004.

[19] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky, "Nonparametric belief propagation," 2002. [Online]. Available: citeseer.ist.psu.edu/sudderth02nonparametric.html

[20] E. Sudderth, M. Mandel, W. Freeman, and A. Willsky, "Distributed occlusion reasoning for tracking with nonparametric belief propagation," *Advances in Neural Information Processing Systems*, vol. 17, pp. 1369–1376, 2004.

[21] V. Stankovic, A. Liveris, Z. Xiong, and C. Georghiades, "Design of Slepian-Wolf codes by channel code partitioning," in *Data Compression Conference*. IEEE, 2005, pp. 302–311.

[22] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Non-asymmetric Slepian-Wolf coding of non-uniform Bernoulli sources," *6th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, pp. 304–308, 2010.

[23] M. Bolic, P. M. Djuric, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP Journal on Applied Signal Processing*, vol. 15, no. 5, pp. 2267–2277, 2004.

[24] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings algorithm," *American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[25] A. Liveris, Z. Xiong, and C. Georghiades, "Joint source-channel coding of binary sources with side information at the decoder using ira codes," in *Proc. MMSP'02 IEEE Workshop on Multimedia Signal Process.*, St. Thomas, US Virgin Islands, Dec. 2002.

[26] T. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley-Blackwell, 2005.

[27] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, 2000, pp. 1–8.

[28] S. Cheng, S. Wang, and L. Cui, "Adaptive nonasymmetric slepian-wolf decoding using particle filtering based belief propagation," in *IEEE ICASSP*, Dallas, TX, 2010.