Adaptive Slepian-Wolf Decoding using Particle Filtering based Belief Propagation

Samuel Cheng, Shuang Wang, and Lijuan Cui
School of Electrical and Computer Engineering
University of Oklahoma, Tulsa, OK, 74135
Email: {samuel.cheng, shuangwang, lj.cui}@ou.edu

***Abstract*—A major difficulty that plagues the practical use of Slepian-Wolf coding (and distributed source coding in general) is that the precise correlation among sources need to be known *a priori*. To resolve this problem, we propose an adaptive Slepian-Wolf decoder using particle filtering based belief propagation. We show through experiments that the proposed algorithm can simultaneously reconstruct a compressed source and estimate the joint correlation between the sources. Further, comparing to the conventional Slepian-Wolf coder based on standard belief propagation, the proposed approach can achieve higher compression under varying correlation statistics.**

I. Introduction

Slepian-Wolf (SW) coding refers to the lossless distributed compression of correlated sources. Consider N correlated sources $X_1, X_2, \cdots, X_N$. Assuming that encoding can only be performed separately that N encoders can see only one of the N sources but the compressed sources are transmitted to a base station and decompressed jointly. To the surprise to many researchers of their time, Slepian and Wolf showed that [1] it is possible to have no loss in sum rate under this restrictive situation. That is, 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 $H(X_1, X_2, \cdots, X_N)$.

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], where the scheme is restricted to two correlated sources with one of them treated as side information. Numerous channel coding based SW coding schemes have been proposed [3], [4], [5], [6], [7], [8]. 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) [9], [10]. However, the fundamental assumption is that the correlation statistics needs to be known accurately *a priori*.

In this paper, we propose an adaptive LDPC code based SW decoder that combines Particle Filtering (PF) with belief propagation (BP) to simultaneously reconstruct a compressed source and estimate the joint correlation between the sources. The PF algorithms, also known as sequential Monte Carlo algorithms, are sophisticated modeling techniques for estimation based on Monte Carlo simulations [11]. The main objective of PF is to estimate posterior probability distribution of an object of interest by sampling a list of random particles with associated weights. Our proposed algorithm is carried out based on factor graph [12], [13], which affords great flexibility in modeling systems. We show that the proposed algorithm no longer depends on the initial estimation of the correlation parameter and offers an accurate real-time estimation of the parameter. For different code rates, our algorithm shows a lower decoding error rate (and thus a more efficient compression) than that of a standard 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 [14]). Unlike in channel coding that channel state information can be estimated with the help of a pilot signal, this trick cannot be used for SW coding and Distributed Source Coding (DSC) in general since sources in DSC are specified by the problems themselves and are not controllable by users. Further, while we only present asymmetric SW coding of two sources (i.e., compressing one source assuming that the other source is available at the decoder as side information), the propose approach can be easily extended to non-asymmetric cases and multiple sources [15].

This paper is structured as follows. In Section II, we describe the precise problem formulation and an overview of our proposed PF based BP algorithm for SW decoding. A brief review of the standard BP algorithm on a factor graph is given in Section III. The BP algorithm based on PF are described in Section IV. Finally, in Section V we present simulation results and in Section VI, we draw the concluding remarks.

II. Problem Formulation

Let X and Y be two correlated binary sources (taking value 0 and 1) and the correlation between them be symmetric in such a way that Y can be considered as the output of X passing through a Binary Symmetric Channel (BSC) with unknown crossover probability p. That is,

$$Y = \begin{cases} X, & \text{with probability } 1-p, \\ X \oplus 1, & \text{with probability } p, \end{cases} \tag{1}$$

where $\oplus$ means "exclusive or" operation. We assume that the crossover probability p may drift over time but will not change too rapidly.

Note that if p is constant over time and is known *a prior*, X can be compressed very close to the SW limit ($H(X|Y)$) using syndrome based approach and LDPC codes [9]. At the SW encoder, the syndrome of a block of X is computed and transmitted to the decoder. At the SW decoder, Y is treated as the output of X passing through a correlation channel. SW decoding is almost identical to conventional LDPC decoding. However, rather than decoding to a codeword, the decoder approximates the estimated block of X as a code vector with the received syndrome.

Just as channel decoding, a block of X can be reconstructed using BP algorithm [16] over the corresponding factor graph [12]. Fig. 1 show the factor graph of the proposed BF based BP for SW coding, which includes the Tanner graph of the standard BP as a subgraph. 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 [12].

For the case of standard BP, a block of X $(X_1, X_2, \cdots, X_N)$ is compressed into M syndrome bits $(S_A, S_B, \cdots, S_M)$, thus resulting in $M : N$ compression. A syndrome factor node f_a, $a \in \{A, B, \cdots, M\}$, which takes into account the constraint imposed by the received syndrome bit s_a, is defined as

$$f_a(\mathbf{x}_a) = \begin{cases} 1, & \text{if } s_a \oplus \bigoplus_{i \in N(a)} x_i = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

On the other hand, a correlation factor node $f_i, i \in \{1, 2, \cdots, N\}$, which handles the correlation between the X_i and Y_i, is defined as

$$f_i(x_i) = \begin{cases} 1-p, & \text{if } x_i = y_i, \\ p, & \text{otherwise.} \end{cases} \tag{3}$$

Note that in the standard BP, the crossover probability p is assumed to be constant and known *a priori*. The main contribution of our approach is to release from these constraints. We assume that p is unknown and varies slowly over time. As shown in Fig. 1, multiple f_i, $i \in \{1, 2, \cdots, N\}$, will connect to the same variable node $p_{i'}$. We call the number of correlation factor nodes connecting to each $p_{i'}$ the connection ratio, which is equal to three in Fig. 1. Since we assume that p only varies slowly over time, the adjacent p should be close in value. This characteristic is captured by the p-factor nodes $f_{1,2}, f_{2,3}, \cdots, f_{N'-1,N'}$ as shown in Region 1 of Fig. 1, where a p-factor node $f_{i'-1,i'}$ is defined as

$$f_{i'-1,i'}(p_{i'-1}, p_{i'}) = \frac{1}{\sqrt{2\pi\lambda}} \exp\left(-\frac{(p_{i'-1} - p_{i'})^2}{2\lambda}\right), \tag{4}$$

where λ is a hyper-prior and can be chosen rather arbitrarily.

III. Belief Propagation In Factor Graphs

BP algorithm is a powerful method for computing approximate marginal probability functions by exchanging messages between adjacent neighboring nodes. Let $m_{a\to i}(x_i)$ denotes the message sent from factor node a to variable node i, and let $m_{i\to a}(x_i)$ denotes the message sent from variable node i to factor node a. In this paper, the proportionality symbol $\propto$ indicates that these messages must be normalized so that they sum to one. The messages updating steps can be expressed as follows:

$$m_{i\to a}(x_i) \propto \prod_{c \in N(i)\backslash a} m_{c\to i}(x_i) \tag{5}$$

and

$$m_{a\to i}(x_i) \propto \sum_{\mathbf{x}_a \backslash x_i} \left(f_a(\mathbf{x}_a) \prod_{j \in N(a)\backslash i} m_{j\to a}(x_j) \right), \tag{6}$$

where $N(i)\backslash a$ denotes the set of all neighbors of a given node i except for node a; $\sum_{\mathbf{x}_a \backslash x_i}$ denotes a sum over all the variable $\mathbf{x}_a$ that are arguments of f_a except x_i. Moreover, the BP algorithm approximates the belief $b_i(x_i)$ at a variable node i as follows

$$b_i(x_i) \propto \prod_{a \in N(i)} m_{a\to i}(x_i). \tag{7}$$

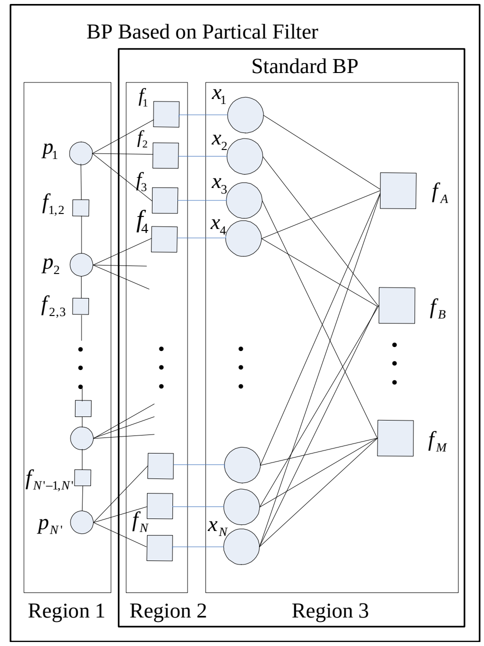

Fig. 1. Factor graph representation of the proposed BF based SW coding.

IV. Belief Propagation Algorithm Based On Particle Filter

Since standard BP can only handle discrete variables, it is futile for estimating the crossover probabilities $p_1, p_2, \cdots, p_{N'}$ (Region 1 in Fig. 1). However, by incorporating PF with BP, we are able to extend BP to handle even continuous variables. To ease exposition, we will briefly review PF in the following. Then we will describe in detail our PF based BP algorithm.

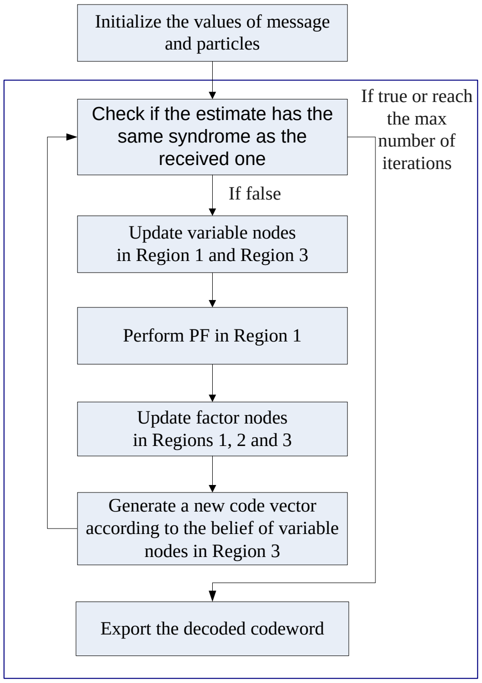

Fig. 2. The workflow of the proposed SW decoder.

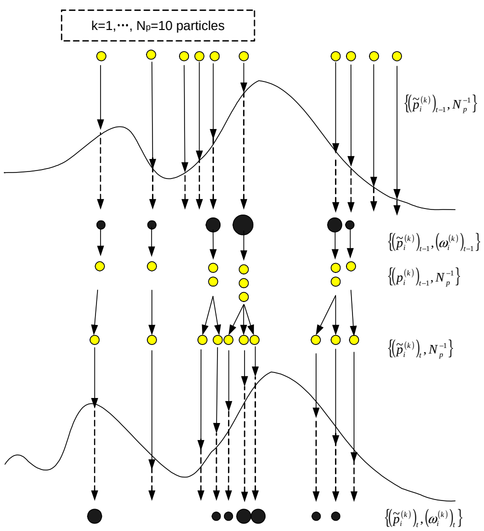

Fig. 3. In this figure, the starting time at $t-1$, a standard PF has 10 unweighted measures $\left\{\left(\tilde{p}_i^{(k)}\right)_{t-1}, N_p^{-1}\right\}$ for a variable node i. Then using the information at time $t-1$, we can get the weighted measure $\left\{\left(\tilde{p}_i^{(k)}\right)_{t-1}, \left(\omega_i^{(k)}\right)_{t-1}\right\}$ by computing the importance weights. The next step is to select the "fittest" particle to obtain the unweighted measure $\left\{\left(p_i^{(k)}\right)_t, N_p^{-1}\right\}$ by using the resampling algorithm. Then by perturbing the congested particles, we can get the unweighted measures $\left\{\left(\tilde{p}_i^{(k)}\right)_t, N_p^{-1}\right\}$. Finally, by computing the weights, we can get $\left\{\left(\tilde{p}_i^{(k)}\right)_t, \left(\omega_i^{(k)}\right)_t\right\}$ at time t and then continue the above steps.

A. Particle Filtering

PF is a technique for optimal numerical estimation when exact solutions cannot be analytically derived [11]. The ap-

k	$\omega^{(k)}$	$i^{(k)}$
1	3/40	1
2	3/40	1
3	2/40	0
4	1/40	0
5	7/40	2
⋮	⋮	⋮

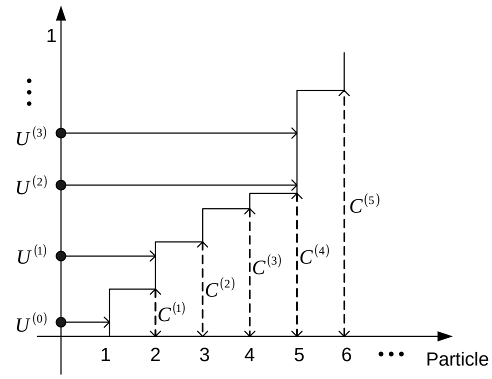

Fig. 4. Systematic resampling process for an example with N_p particles. The weights of the particles are listed in the table above.

proach is used to estimate posterior probability distribution of an interest object by sampling a list of random particles with associated weights. The basic procedure of PF is as follows:

1) Initialize the list of particles from the prior distribution. For each variable node i in Region 1, each particle k is assumed to have a value $p_i^k = \hat{p}$ for estimating the noise and a uniform weight $\frac{1}{N_p}$, where N_p is the number of the particles.
2) Compute the importance weights $\omega_i^{(k)}$ at each iteration. In our model this weight is equal to the belief of each particle, where the belief is obtained by the BP algorithm, which will be presented in Part B of this section.
3) Since the variance of the importance weights increase stochastically over time, a selection (resampling) stage is needed to eliminate particles with negligible weights and to concentrate on particles with large weights. The future particles in domains of higher posterior probability entail improved estimates. The systematic resampling (SR) [11], [17], [18], [19], [20] algorithm, are illustrated graphically in Fig. 4, where the weight of particles are listed in the table. SR first calculates the cumulative sum of the particle weights $C^{(k)} = \sum_{i=1}^{k} \omega^{(i)}$ and updated uniform number $U^{(k)} = U^{(k-1)} + \frac{1}{N_p}$, $k = 1, ..., N_p$, where $U^{(0)}$ is obtained by drawing from the uniform distribution $u\left[0, \frac{1}{N_p}\right]$. Then SR compares $C^{(k)}$ and $U^{(k)}$ to determine the number of replications for particle k by computing the number of time $U^{(k)}$ in the range $[C^{(k-1)}, C^{(k)})$. In Fig. 4, for particle one and two, $U^{(0)}$ belongs to the range $[0, C^{(1)})$ and $U^{(1)}$ belongs to the range $[C^{(1)}, C^{(2)})$ respectively, so particle one and two are replicated once. Particle three and four are

discarded and particle five is replicated twice. Other particles follow the same rule.

4) Perturb particles. After the resampling step, particles congregate round the values with large weights. In order to maintain the diversity of particles for further PF iterations, particles should be perturbed by slight values. In this paper, random walk (RW) algorithm is implemented by adding a Gaussian random noise $N\left(0,\sigma_r^2\right)$ with zero mean and variance σ_r^2 on the current value p_i^k of each new particle generated in step 3.
5) Update weight by resetting to a uniform weight $\frac{1}{N_p}$ for each particle, where N_p is the number of the particles.
6) Iterate the steps 2 to 5 when updating each variable node.

B. Belief Propagation based on the Particle Filter

We applied the BP algorithm [16] to the factor graph in Fig. 1. Since we assumed that the correlation was symmetric, the initial value of the message sent from the factor nodes in Region 2 to variable nodes in Region 3 is given by

$$m_{a\to i}(x_i) = f_a\left(x_i\right) = \begin{cases} 1-\hat{p}, & \text{if } y_i = x_i, \\ \hat{p}, & \text{otherwise}, \end{cases} \tag{8}$$

where $\hat{p}$ is an initial estimated crossover probability. The message passing schedule is illustrated in Fig. 2. First, the messages sent from variable nodes were updated. Then we performed PF. Finally, the messages sent from factor nodes were updated. The message update rules are detailed as follows, where we use i and a denote the variable nodes and factor nodes respectively:

1) Update variable nodes in Region 3 using (5)
2) In the PF algorithms, the messages are represented by N_p particles. Hence, updating variable nodes in Region 1 needs to update information for each particle. The updating equation follows

$$m_{i\to a}\left(p_i^k\right) \propto \prod_{c\in N(i)\backslash a} m_{c\to i}\left(p_i^k\right), \tag{9}$$

where $N\left(i\right)\backslash a$ denotes the set of neighbors of node i except for node a; k denotes the k-th particle and p_i^k is the value of k-th particle for variable node i.

3) Compute the belief of each variable node i in Region 3 being x_i

$$b_i\left(x_i\right) \propto \prod_{a\in N(i)} m_{a\to i}\left(x_i\right) \tag{10}$$

where x_i is equal to 0 or 1.

4) Compute the belief of each particle for each variable node i in Region 1 being p_i^k

$$b\left(p_i^k\right) \propto \prod_{a\in N(i)} m_{a\to i}\left(p_i^k\right) \tag{11}$$

In the model, the belief $b\left(p_i^k\right)$ of each particle k is corresponding to the weight $\left(\omega_i^{(k)}\right)$ shown in Fig. 3.

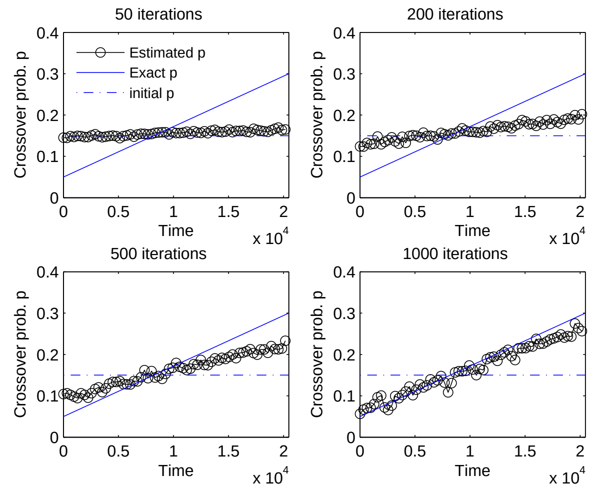

Fig. 5. Estimation of crossover probabilities for linearly changing correlations and 1:16 connection ratio.

5) Update factor node in Region 3 follows (12)

$$m_{a\to i}\left(x_i\right) \propto \sum_{\mathbf{x}_a\backslash x_i}\left(f_a\left(\mathbf{x}_a\right)\prod_{j\in N(a)\backslash i} m_{j\to a}\left(x_j\right)\right), \tag{12}$$

where $f_a\left(\mathbf{x}_a\right)$ is defined in (2)

6) Update factor node in Region 1, which means to update each particle according to (13)

$$\begin{aligned} m_{a\to i}\left(p_i^k\right) &\propto \sum_{p_{i+1}}\left(f_a\left(p_{i+1}^k,p_i^k\right)m_{i+1\to a}\left(p_{i+1}^k\right)\right), \\ m_{a\to i+1}\left(p_{i+1}^k\right) &\propto \sum_{p_i}\left(f_a\left(p_{i+1}^k,p_i^k\right)m_{i\to a}\left(p_i^k\right)\right), \end{aligned} \tag{13}$$

where $f_a\left(p_{i+1}^k,p_i^k\right) = e^{-\frac{\left(p_{i+1}^k-p_i^k\right)^2}{\lambda}}$, and p_i or p_{i+1} means all the particles in the variable nodes i or $i+1$ of Region 1.

7) Update factor node in Region 2
   a) Message from Region 2 to Region 1

   $$m_{a\to i}\left(p_i^k\right) \propto \sum_{x_j\in\{0,1\}}\left(f_a\left(x_j,y_j,p_i^k\right)m_{j\to a}\left(x_j\right)\right), \tag{14}$$

   where $f_a\left(x_j,y_j,p_i^k\right) = \begin{cases} 1-p_i^k & \text{if } y_j = x_j \\ p_i^k & \text{otherwise} \end{cases}$, and j refers to j-th variable node in Region 3.

   b) Message from Region 2 to Region 3

   $$m_{a\to i}\left(x_i\right) \propto \sum_{k\in[1,N_p]}\left(f_a\left(x_i,y_i,p_j^k\right)m_{j\to a}\left(p_j^k\right)\right), \tag{15}$$

   where $f_a\left(x_i,y_i,p_j^k\right) = \begin{cases} 1-p_j^k & \text{if } y_i = x_i \\ p_j^k & \text{otherwise} \end{cases}$, and j refers to the j-th variable node in Region 1.

V. Results

In this section, the decoding performance of standard BP and that of PF based BP algorithm are compared. Moreover,

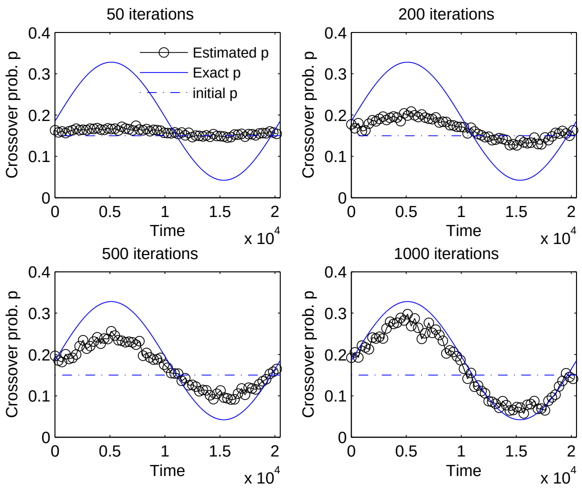

Fig. 6. Estimation of crossover probabilities for sinusoidal changing correlations and 1:16 connection ratio.

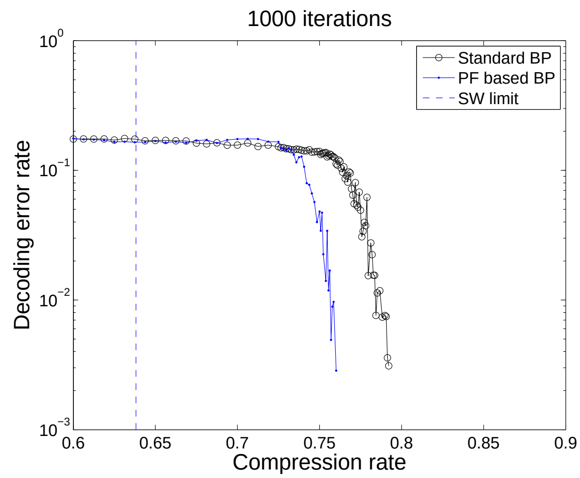

Fig. 7. Decoding error probability for a linearly changing correlation with 1000 iterations and 1:16 connection ratio.

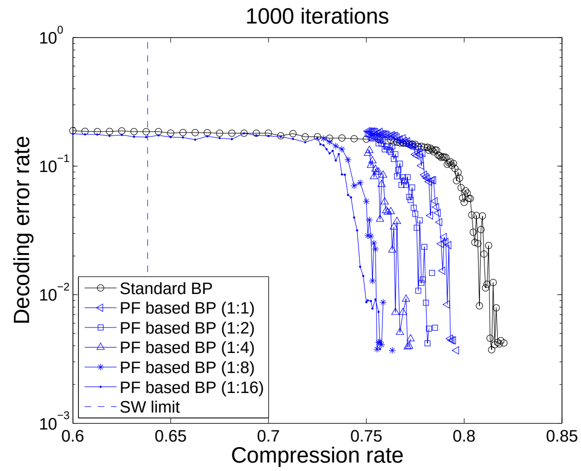

Fig. 8. Decoding error probability for a sinusoidal changing correlation with 1000 iterations, where different connection ratios are used to test the decoding performance.

the estimation accuracy of error probability are also analyzed under the conditions with different numbers of iterations and different connection ratios.

We first investigated the estimated crossover probability obtained by the BF based BP algorithm over a linearly and a sinusoidal changing correlations. Here, SW codes were randomly generated by parity check matrices with 20480 variable nodes and 12288 check nodes, where each variable node connects 3 check nodes. Moreover, 16 particles were assigned to each variable node in Region 1 (see Fig. 1). For the random walk step, we assumed $\sigma_r^2 = 0.0001$. For the factor node update in Region 1, we assumed that $\lambda = 0.001$. The following results were obtained by averaging the estimated error probability of 30 different codewords. Fig. 5 shows the estimated results of a linearly changing correlation, where the crossover probability p increased continuously from 0.05 to 0.3 by the step $V = \frac{0.3-0.05}{20480}$ for each input codeword bit. From Fig. 5, we can see that with the increase of the number of iterations, the estimated crossover probability $\hat{p}$ becomes closer to the exact input crossover probability, even though they started with a far underestimated initial value. Moreover, as shown in Fig. 6, an estimated $\hat{p}$ for SW codes over a sinusoidal changing correlation is presented. The results also verified that our proposed algorithm can generate a good estimation of a complexly changing correlation.

Next we compared the performance results of the standard BP and our proposed algorithm in terms of decoding error. The following performance results were also obtained by averaging 30 independent simulations, where the code length was 20480 and the code rates changed from 0.1 to 0.4. The SW limit[1] is equal to the conditional entropy $H(X|Y) = \frac{1}{20480}\sum_{i\in[1,20480]} H(p_i)$, where $H(p_i) = -p_i \log(p_i) - (1 - p_i)\log(1-p_i)$. The value of crossover probability p_i in the BSC continuously increased from 0.05 to 0.3 by the step $V = \frac{0.3-0.05}{20480}$. The number of particles was 16. Fig. 7 shows that our proposed algorithms obtained better performance than that of standard BP. Similarly, for the sinusoidal time-varying BSC, we assumed that the crossover probability changed according to Fig. 6. In this situation, Fig. 8 also shows that PF based BP algorithm offered a better performance. Moreover, we can see that with the increase of connection ratios from 1:1 to 1:16, the decoding performance of our proposed algorithm became better. Furthermore, in Fig. 9, we shows the results of estimated error probability as a function of connection ratio with 1000 iteration times. We can see that the estimation accuracy increases as the connection ratio increases, since the variable nodes in Region 1 can obtain more channel information from the factor nodes in Region 2 by using a higher connection ratio. When the connection ratio is equal to 1:16, our algorithm shows a perfect matching result in Fig. 9,

[1] Strictly speaking, the achievable compression rate should be even higher than the SW limit since the original SW setup assume that the correlation statistics are known to both the encoder and decoder.

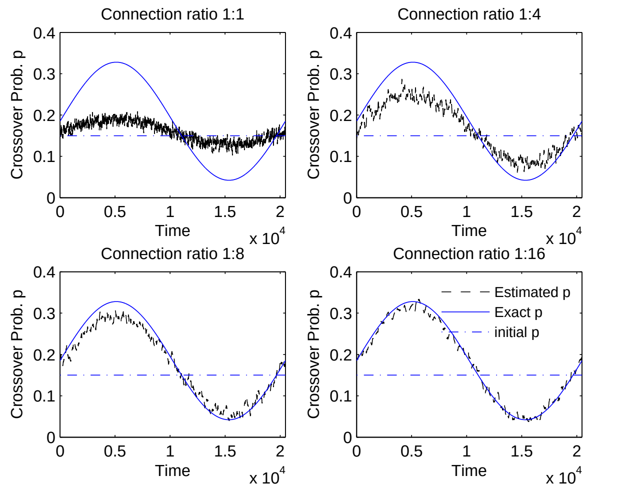

Fig. 9. Estimation of error probability for sinusoidal changing correlation with 1000 iterations and different connection ratios.

which also explained why the 1:16 connection ratio yielded a better decoding performance in Fig. 8.

VI. Conclusion

We proposed an adaptive SW decoding algorithm based on PF and BP on a factor graph. By introducing the PF algorithm in the left-hand side of the factor graph in the standard BP algorithm, the error probability of a BSC will be updated for each variable node step by step. During our experiments, a precise estimation of error probability has been observed by using our PF based BP algorithm. Thus, the decoding performance of our algorithm is no longer sensitive to the initial estimation of the error probability p. Finally, we concluded that our PF based BP algorithm yields a more efficient compression than that does the standard BP algorithm.

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 *Data Compression Conference, 2004. Proceedings. DCC 2004*, 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] J. Garcia-Frias and Y. Zhao, "Near-Shannon/Slepian-Wolf performance for unknown correlated sources over AWGN channels," *Communications, IEEE Transactions on*, vol. 53, no. 4, pp. 555–559, 2005.

[7] J. Chen, D.-k. He, A. Jagmohan, and L. A. Lastras-Montano, "On the reliability function of variable-rate Slepian-Wolf coding," in *45th Annual Allerton Conference*, Urbana-Champaign, IL, 2007.

[8] A. Liveris, C. Lan, K. Narayanan, Z. Xiong, and C. Georghiades, "Slepian-Wolf coding of three binary sources using LDPC codes," in *Proc. Intl. Symp. Turbo Codes and Related Topics*, 2003.

[9] 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.

[10] 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.

[11] A. Doucet and N. De Freitas, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[12] 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.

[13] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2282–2312, 2005.

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

[15] S. Cheng, "Non-asymmetric Slepian Wolf coding using belief propagation," in *Proc. CISS 2009*, Baltimore, MD, 2009.

[16] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, 1996.

[17] 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.

[18] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S. A. Adelaide, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing [see also IEEE Transactions on Acoustics, Speech, and Signal Processing]*, vol. 50, no. 2, pp. 174–188, 2002.

[19] M. Isard and A. Blake, *Active Contours*. Springer-Verlag, 1998.

[20] Y. Zhai, M. Yeary, J. P. Havlicek, J. C. Noyer, and P. Lanvin, "Visual tracking using sequential importance sampling with a state partition technique," in *IEEE International Conference on Image Processing.*, vol. 3, 2005.