# Encoding for the Blackwell Channel with Reinforced Belief Propagation 

Alfredo Braunstein<br>Institute for Scientific Interchange<br>Villa Gualino, viale S.Severo 65<br>10133, Turin, Italy<br>braunstein@isi.it

Farbod Kayhan<br>Institute for Scientific Interchange, Politecnico di Torino,<br>10129, Turin, Italy<br>kayhan@isi.it

Guido Montorsi<br>Politecnico di Torino<br>Dipartimento di Elettronica<br>10129, Turin, Italy<br>Email: montorsi@polito.it

Riccardo Zecchina<br>International Center for Theoretical Physics,<br>Politecnico di Torino, Institute for Scientific Interchange<br>Strada Costiera 11<br>I-34100, Trieste, Italy<br>Email: zecchina@ictp.it


#### Abstract

A key idea in coding for the broadcast channel (BC) is binning, in which the transmitter encode information by selecting a codeword from an appropriate bin (the messages are thus the bin indexes). This selection is normally done by solving an appropriate (possibly difficult) combinatorial problem. Recently it has been shown that binning for the Blackwell channel -a particular BC- can be done by iterative schemes based on Survey Propagation (SP). This method uses decimation for SP and suffers a complexity of $\mathcal{O}\left(n^{2}\right)$. In this paper we propose a new variation of the Belief Propagation (BP) algorithm, named Reinforced BP algorithm, that turns BP into a solver. Our simulations show that this new algorithm has complexity $\mathcal{O}(n \log n)$. Using this new algorithm together with a non-linear coding scheme, we can efficiently achieve rates close to the border of the capacity region of the Blackwell channel.


## I. Introduction

Broadcast channels (BC) were first introduced and analyzed by Cover [7]. The general BC with $t$ receivers is depicted in Fig. 1. In a BC, a single transmitter sends simultaneously independent information to multiple receivers.

Coding for each receiver independently with a normal point-to-point code and sending the $t$ messages sequentially -by allocating proportions of time to each receiver- is known as time sharing strategy. It is shown in [7] that jointly optimized codes can have a larger capacity region for error-free communication than that of time sharing codes [1], [7], [8].

A key idea in coding for the BC is the binning strategy, which allows the transmitter to encode information by selecting a codeword from an appropriate bin. In this paper we deal with practical implementation of random binning for the BC. Existing practical binning schemes for BC are often based on structured codes and maximum likelihood algorithms. Martinian and Yedidia in [11] have used for the first time the random codes on graphs for quantization of a binary erasure source. Still their method works only for erasure sources and is not applicable to the general BC.


Fig. 1. A single sender and $t$ receivers broadcast channel.

Recently, Wei Yu and M. Aleksic [18] showed that the binning problem for a particular BC , namely the Blackwell Channel (BWC), when coding is performed by random lowdensity parity-check like codes, can be thought as a constraint satisfaction problem. They proposed an iterative encoder that works well at rates close to the border of the BC capacity region.

The main difference of this problem with that of decoding classical codes is that this combinatorial problem admits many solutions. In fact in these cases the application of BP allows to compute the cardinality of the solution space but fail to find a particular solution.

In [18] they use Survey Propagation (SP) algorithm for encoding, fixing one variable after each convergence (decimation). The main drawback of this method is the encoding complexity which grows as $\mathcal{O}\left(n^{2}\right)$. Also the decimation works well only when the connectivity of XOR nodes are very small ( $c=2,3$ and 4 ).

In this paper we use a modified version of BP, called Reinforced Belief Propagation (RBP), originally proposed in the context of perceptron learning [3], which effectively turns BP into a solver. Experiments show that RBP does not converge for factor graphs with XOR function nodes. To overcome this, we propose a new class of sparse non-linear codes. These two modifications result in a more efficient encoding complexity (from $\mathcal{O}\left(n^{2}\right)$ to $\mathcal{O}(n \log n)$ ) and a lower Frame Error Rate (FER), i.e., the probability of not finding a solution to the
encoding problem.
This paper is organized as follows. In the next section we introduce the general framework of broadcast channels and their capacity regions. In section III we present the iterative updates for BP and RBP algorithms. Our scheme for coding for the BWC using non-linear nodes is explained in section IV. Our results are presented in section V. The final section is devoted to conclusions and outlooks.

## II. Notations and Basic Concepts

In this section we first introduce the basic concepts and then briefly review some results on capacity region for deterministic broadcast channels.

Definition 2.1: A broadcast channel consists of an input alphabet $\mathcal{X}$, two output alphabets $\mathcal{Y}_{1}$ and $\mathcal{Y}_{2}$ and a probability transition function $P\left(\mathbf{y}_{1}, \mathbf{y}_{2} \mid \mathbf{x}\right)$. The channel is said to be memoryless if

$$
P\left(\mathbf{y}_{1}, \mathbf{y}_{2} \mid \mathbf{x}\right)=\prod_{i=0}^{n} P\left(y_{1 i}, y_{2 i} \mid x_{i}\right)
$$

A $\left(\left(2^{n R_{1}}, 2^{n R_{2}}\right), n\right)$ code for a BC with independent information consists of an encoder

$$
\mathcal{E}: 2^{n R_{1}} \times 2^{n R_{2}} \rightarrow \mathcal{X}^{n}
$$

and two decoders

$$
\mathcal{D}_{1}: \mathcal{Y}_{1}^{n} \rightarrow 2^{n R_{1}} \quad, \quad \mathcal{D}_{2}: \mathcal{Y}_{2}^{n} \rightarrow 2^{n R_{2}}
$$

We assume that the transmitted message pair $\left(W_{1}, W_{2}\right)$ is uniformly distributed over the set $2^{n R_{1}} \times 2^{n R_{2}}$. The probability of error $P_{e}^{n}$ is defined to be

$$
P_{e}^{n}=P\left(W_{1} \neq \hat{W}_{1} \quad \text { or } \quad W_{2} \neq \hat{W}_{2}\right) .
$$

Definition 2.2 (Capacity Region): A rate pair $\left(R_{1}, R_{2}\right)$ is called achievable for the BC if there is a sequence of $\left\{\left(\left(2^{n R_{1}}, 2^{n R_{2}}\right), n\right)\right\}_{n}$ codes with $P_{e}^{n} \rightarrow 0$ as $n \rightarrow \infty$. The capacity region of the broadcast channel is the closure of the set of achievable rates.

A broadcast channel is deterministic if the channel transition probabilities are deterministic, i.e., $P\left(\mathbf{y}_{1}, \mathbf{y}_{2} \mid \mathbf{x}\right)$ is a $0-1$ function. The largest achievable rate region for a general BC using the binning strategy is known as the Marton's region [12]. This region is proved to be the capacity region for a discrete deterministic channels [13].

A well-known example of a deterministic BC is the BWC (see Fig. 2). The BWC has one input with three symbols and two outputs each one with two symbols. Given two messages $W_{1}$ and $W_{2}$, the goal is to find the codewords $\mathbf{y}_{1} \in 2^{n R_{1}}$ and $\mathbf{y}_{2} \in 2^{n R_{2}}$ such that $\left(y_{1 i}, y_{2 i}\right) \neq(1,1)$ for $i=1,2, \ldots, n$. The other three combinations are allowed and they can be reached by selecting one of the three input symbols of the channel. Even though this channel is not realistic, it is a non-trivial BC which illustrates the conflict between transmitting information to first receiver and transmitting to second receiver [10].

Since the channel is deterministic we have $H(X)=$ $H\left(Y_{1}, Y_{2}\right)$. In the rest we assume a uniform probability


Fig. 2. The Blackwell channel.


Fig. 3. Rate region for the BWC with uniform distribution.
distribution over $X$. With this input distribution the capacity region for BWC becomes

$$
\begin{aligned}
R_{1} & \leq H\left(\frac{1}{3}\right) \\
R_{2} & \leq H\left(\frac{1}{3}\right) \\
R_{1}+R_{2} & \leq \log _{2} 3
\end{aligned}
$$

This capacity region is shown in Fig. 3.

## III. BP and RBP algorithms

Let $g: S \subset \mathbb{R}^{n} \rightarrow \mathbb{R}$ be a real valued function over the domain $S$ and

$$
\begin{equation*}
g\left(x_{1}, x_{2}, \ldots, x_{n}\right) \propto \prod_{j \in M} f_{j}\left(X_{j}\right) \tag{1}
\end{equation*}
$$

where $X_{j}$ is a subset of the set of variables.
Definition 3.1: A factor graph of a function $g$ factorized as in (1) is a bipartite graph with $n$ vertex in one part (variable nodes) and $M$ vertex in the second part (factor nodes). An edge connects variable node $x_{i}$ to factor node $f_{j}$ if and only if $x_{i}$ is an argument of the local function $f_{j}$, i.e., $x_{i} \in X_{j}$.

We show the $i$ th marginal function associated with $g\left(x_{1}, x_{2}, \ldots, x_{n}\right)$ by

$$
g_{i}\left(x_{i}\right) \propto \sum_{\sim\left\{x_{i}\right\}} g\left(x_{1}, x_{2}, \ldots, x_{n}\right)
$$

where the symbol $\sim\left\{x_{i}\right\}$ indicates the set of all variable configurations with the $i$-th variable fixed to $x_{i}$.

Calculating the marginal functions in general is a hard task. BP is an efficient and exact algorithm to calculate all marginal functions $g_{i}\left(x_{i}\right)$ when the factor graph of $g$ is cyclefree. It is possible to use BP also in the presence of loops. The resulting algorithm will be iterative and calculates the


Fig. 4. The modified factor graph for RBP. The black squares are dynamic nodes which their value is a function of the marginal of the related variable at a preceding iteration.
marginals approximately. In the rest of this section, first we review the BP update rules and then present a generalization of BP called Reinforced BP algorithm (RBP) [4].

Let $\mu_{x \rightarrow f}^{\ell}(x)$ denotes the message sent form variable node $x$ to factor node $f$ at the $\ell$ th iteration. Similarly, $\mu_{f \rightarrow x}^{\ell}(x)$ denotes the message sent from factor node $f$ to variable node $x$ at the iteration $\ell$. Also, let

$$
\begin{aligned}
\mathcal{N}\left(x_{i}\right) & \triangleq\left\{j \mid x_{i} \in X_{j}\right\} \\
\mathcal{M}\left(f_{j}\right) & \triangleq\left\{i \mid x_{i} \in X_{j}\right\}
\end{aligned}
$$

then the BP algorithm messages can be expressed as follows: Local Function to Variable:

$$
\begin{equation*}
\mu_{f_{j} \rightarrow x_{i}}^{\ell}\left(x_{i}\right) \propto \sum_{\sim\left\{x_{i}\right\}}\left(f_{j}\left(X_{j}\right) \prod_{l \in \mathcal{M}\left(f_{j}\right) \backslash\{i\}} \mu_{x_{l} \rightarrow f_{j}}^{\ell}\left(x_{l}\right)\right) \tag{2}
\end{equation*}
$$

## Variable to Local Function:

$$
\begin{equation*}
\mu_{x_{i} \rightarrow f_{j}}^{\ell+1}\left(x_{i}\right) \propto \prod_{l \in \mathcal{N}\left(x_{i}\right) \backslash\{j\}} \mu_{f_{l} \rightarrow x_{i}}^{\ell}\left(x_{i}\right) \tag{3}
\end{equation*}
$$

For $\ell=1$, we initialize the messages $\mu_{x \rightarrow f}^{\ell}(x)$ randomly. These updating rules tell us how to produce locally outgoing messages from incoming messages. We define the marginal function of variable $x_{i}$ at iteration $\ell+1$ as

$$
\begin{equation*}
g_{i}^{\ell+1}\left(x_{i}\right) \propto \prod_{l \in \mathcal{N}\left(x_{i}\right)} \mu_{f_{l} \rightarrow x_{i}}^{\ell}\left(x_{i}\right) \tag{4}
\end{equation*}
$$

The algorithm converges after $t$ iterations if and only if for all variables $x_{i}$ and all function nodes $f_{j}$

$$
\mu_{f_{j} \rightarrow x_{i}}^{t+1}\left(x_{i}\right)=\mu_{f_{j} \rightarrow x_{i}}^{t}\left(x_{i}\right)
$$

In practice we need to predefine maximum number of iterations $\ell_{\max }$ and a precision parameter $\epsilon$ as the input to the algorithm.

BP has been generalized/modified in many ways [2], [4], [5], [15], [19]. BP and its generalizations have proven to be efficient when the variables are biased toward a solution. Unfortunately when this condition is not fulfilled marginal themselves are not sufficient to find a solution to the combinatorial problem and one has to resort to some decimation
techniques ( [2], [18]), resulting in a high computational complexity.

We will show here the RBP equations [3] that turn BP into an efficient solver. The idea is to introduce a new set of reinforcement messages which drive the equations toward a single solution. First we modify the original factor graph by adding to each variable node a new function node. In Fig. 4 these new function nodes are depicted by black squares. These function nodes are dynamic and at the $\ell$ th iteration take the value $\left(g_{i}^{\ell-1}\left(x_{i}\right)\right)^{\gamma(\ell-1)}$, i.e, a power of the marginal of the variable $x_{i}$ at the preceding iteration. $\gamma(\ell)$ is a non decreasing function in $[0,1]$ with $\gamma(0)=0$. While the updating rule (2) at function nodes does not change for RBP, the variable to function messages should be modified as below.

Variable to Local Function for RBP:

$$
\begin{equation*}
\mu_{x_{i} \rightarrow f_{j}}^{\ell+1}\left(x_{i}\right) \propto\left(g_{i}^{\ell}\left(x_{i}\right)\right)^{\gamma(\ell)} \prod_{l \in \mathcal{N}\left(x_{i}\right) \backslash\{j\}} \mu_{f_{l} \rightarrow x_{i}}^{\ell}\left(x_{i}\right) \tag{5}
\end{equation*}
$$

In this paper we deal only with binary constraint satisfaction problems, where $\mathbf{x} \in\{0,1\}^{n}$ and the local functions $f_{j}\left(X_{j}\right)$ are 0-1 indicator functions. A vector $\left(x_{1}, x_{2}, \ldots, x_{n}\right)$ satisfies $f_{j}\left(X_{j}\right)$ if $f_{j}\left(X_{j}\right)=1 .\left(x_{1}, x_{2}, \ldots, x_{n}\right)$ is called a solution of the constraint satisfaction problem if all local functions are satisfied, i.e., $\prod_{j \in M} f_{j}\left(X_{j}\right)=1$. It is easy to show that if RBP converges, it converges to a solution of our problem (all messages completely polarized to delta functions). This simple modification provides us with a solver with complexity $\mathcal{O}(n)$ (assuming roughly constant convergence time). Note that the number of iteration of RBP depends also on the choice of $\gamma(\ell)$ in (5). As our experiments show, choosing an optimal $\gamma$ can dramatically decrease the number of iterations of RBP at least for the binning problem. For the rest of this paper we will set

$$
\begin{equation*}
\gamma(\ell)=1-\gamma_{0} \gamma_{1}^{\ell} \tag{6}
\end{equation*}
$$

where $\gamma_{0}, \gamma_{1}$ are in $[0,1]$.

## IV. Coding for the Blackwell Channel Using Non-Linear Nodes

One of the main coding strategies for deterministic broadcast channel is binning. The idea is to generate $2^{n H\left(Y_{1}\right)}$ codewords $\mathbf{y}_{1}$ and $2^{n H\left(Y_{2}\right)}$ codewords $\mathbf{y}_{2}$ and randomly assign them into $2^{n R_{1}}$ and $2^{n R_{2}}$ bins. To transmit a particular pair of bin indices $(i, j)$, the transmitter looks for a pair of codeword $\left(\mathbf{y}_{1}, \mathbf{y}_{\mathbf{2}}\right) \in(i, j)$ such that they are jointly typical.

For the BWC, the joint typicality of $\mathbf{y}_{1}$ and $\mathbf{y}_{2}$ is equivalent to being consistent with the channel constraints. Therefore, we are looking for efficient ways to finding a pair $\left(\mathbf{y}_{1}, \mathbf{y}_{2}\right)$ such that $\left(y_{1 i}, y_{2 i}\right)=(1,1)$ does not occur for $i=1,2, \ldots, n$.

Wei Yu and Marko Aleksic in [18] have suggested a random binning method for BWC based on low-density paritycheck like codes. In this section we first review their results and then modify their scheme using non-linear nodes and RBP algorithm. As we will see in the next section, these modifications imply a better encoding complexity and a lower FER for large function node connectivity.


Fig. 5. Factor graph for LDPC like encoding for Blackwell channel.

Fig. 5 illustrates the graphical structure (factor graph) of the coding scheme used in [18]. $n$ circles denote the variable nodes, $n R_{1}+n R_{2}$ squares denote the parity check nodes and $n$ crossed squares denote the product constraints (ensuring the $(1,1)$ pair does not occur). The encoding process is as follows. The information bits $n R_{1}$ and $n R_{2}$ are placed at the parity checks. These values are actually the bin indices. The goal is to find the set of variable assignments that satisfy the paritychecks and product constraints simultaneously. These two sets of constrains ensure the typicality of the pair $\left(\mathbf{y}_{1}, \mathbf{y}_{2}\right)$. When $n$ is large an exhaustive search is not feasible and practical algorithms are desirable.

In [18] the survey propagation algorithm is suggested for this encoding problem. The main drawback of using SP/BP is the complexity which grows as $\mathcal{O}\left(n^{2}\right)$ because of the decimation process. As it was also mentioned in [18], this method works only for small function nodes connectivity.

On the other hand, the RBP algorithm, introduced in the section III, do not converge -even for rates not close to the capacity- for linear codes. To overcome this, we substitute parity check nodes with non-linear (random) functions. These kind of gates have been analyzed with methods from statistical physics [6]. Intuitively, the reason for which random gates may show a better performance with respect to the linear nodes can be explained as follows. Strong symmetry properties of XOR functions do not allow a decimation procedure to choose a good decimation path that preserves the uncorrelation hypothesis needed for BP; indeed, in any decimation step with XOR gates, undecided variables have all equal probability of taking 0 or 1 .

Given $c$ variable input nodes we choose a non-linear function node randomly from all $2^{c}$ possible balanced truth-tables. We eliminate from this choice fully-canalized nodes, i.e., random nodes for which a particular value of one of their variables determine the output. For our code constructions we have used 4 to 8 different random nodes for each connectivity $c$. Note that the complexity of updating messages on a random node with degree $c$ is of order $2^{c}$. In this paper we confine ourselves to a constant degree $c=6$ and hence ignore this


Fig. 6. Entropy as a function of rate $\left(R_{1}=R_{2}\right)$ for different function node connectivity $c$. At any given rate the entropy of codes with non-linear factor nodes increases with $c$ and approaches the entropy of linear codes.

| Rate | 0.5 | 0.6 | 0.7 | 0.72 | 0.73 | 0.74 | 0.75 |
| ---: | :---: | :---: | :---: | :---: | :---: | :---: | :--- |
| FER | 0 | 0 | 0.03 | 0.1 | 0.35 | 0.825 | 0.975 |
| BER | 0 | 0 | 0.00011 | 0.0013 | 0.00425 | 0.0119 | 0.0347 |
| $\gamma_{1}$ | 0.99 | 0.995 | 0.999 | 0.9995 | 0.9999 | 0.99999 | 0.999995 |

TABLE I
BER AND FER OF NON-LINEAR LDPC LIKE ENCODERS AT A GIVEN RATE
( $R_{1}=R_{2}$ ) AND CONNECTIVITY $c=6$.
factor in the rest.
In order to show the suitability of non-linear nodes to the problem at hand we compute the normalized size of the solution space, defined as $H_{S}=\log \left(N_{s}\right) / n$, where $N_{s}$ is the number of solutions. An approximation to $H_{S}$ can be computed directly from the BP messages at a fixed point [19].

In Fig. 6 we plot $H_{S}$ as a function of rate using linear nodes and non linear nodes for different values of the function node connectivity $c$. The entropy of codes with non-linear function nodes increases with $c$ and approaches the entropy of linear codes. Note that for linear codes the entropy does not change with connectivity. A connectivity $c=6$ thus guarantees a solution space with cardinality near to those of LDPC codes when using non-linear type nodes. This value of connectivity has then been chosen for the code construction.

## V. Results

Table I shows the FER and BER of our constructed nonlinear codes for the BWC with $n=1000$ and constant connectivity $c=6$ at different rates. The last line reports the values we chose for $\gamma_{1}$.

We estimated the algorithmic complexity of the presented coding scheme in a series of experiments described below. In particular, we will show how the convergence time changes as a function of $n$ and $\gamma_{1}$. The RBP algorithm was run with an estimated optimal value of $\gamma_{1}$, and we have chosen a cutoff time of $\frac{1}{\left(1-\gamma_{1}\right)}$ to measure the bit and frame error rates.

Fig. 7 shows the average number of iterations needed (in the case of success) for a rate $R_{1}=R_{2}=0.70$ as a function of


Fig. 7. The average number of needed iterations as a function of $\gamma_{1}$ at rate $R_{1}=R_{2}=0.70$ for different values of $n$. Note that for smaller $\gamma_{1}$ we need less number of iterations but both BER and FER are larger (see Fig. 8 and table I).


Fig. 8. Bit error rate as a function of $\gamma_{1}$ at rate $R_{1}=R_{2}=0.70$ for different values of $n$. Error bars are smaller than symbols size in this scale.
$n$ and $\gamma_{1}$ and for 160 encoding operations. These simulations indicate that the number of iterations increase as $\mathcal{O}(\log n)$. Although the number of iterations increase (exponentially) with $\gamma_{1}$, both the BER and FER decrease (exponentially) as it can be seen in Fig. 8. Note that for rates closer to the capacity bound ( $R_{1}=R_{2} \approx 0.785$ ) a value of $\gamma_{1}$ closer to 1 (and larger number of iterations) is needed.

Although the results depicted in Fig. 7 indicate a logarithmic increase in the number of iterations as a function of $n$, this result may be due to a not optimized choice of $\gamma(\ell)$. For example by choosing $\gamma_{0}=0.8$ in (6) it is possible to reduce the number of iterations for $n=4000$ and $\gamma_{1}=0.999$ by nearly $25 \%$. In other words, one can avoid approximately the first 200 iterations of RBP without loosing in performance.

## VI. Conclusion and Outlooks

We have introduced a novel variation of the BP algorithm, called reinforced BP, that turns it into an efficient solver for
non-linear problems even when they have a large solution space. The algorithm have the same complexity of BP and thus considerably smaller than the decimation approach applied to BP/SP proposed in [18].

Using RBP we have constructed a general and rather efficient encoding scheme for the BWC. Our codes provide good encoding performances for rates up to 0.72 . This result can be possibly improved by optimizing the function $\gamma(\ell)$ and the degree distributions of the code.

Our scheme compares well with existing ones: for linear codes with $R=R_{1}=R_{2}=0.75$ and decimation, as it was reported also in [18], one can get the bit error rate of $5.10^{-3}$. Still, simulations show that the FER in this case is 0.9 and it does not improve for smaller rates like $R=0.72$ with the same connectivity. On the other hand it works only for low function node connectivity. Our scheme is much more flexible and provides a comparable FER and BER at $R=0.75$ with lower computational complexity. For smaller rates our codes outperform the existing linear encoding schemes.

## REFERENCES

[1] P. P. Bergmans, Random Coding Theorem for Broadcast Channels with Degraded Components, IEEE Trans. Inform. Theory, vol. 19, Mar. 1973.
[2] A. Braunstein, M. Mezard and R. Zecchina, Survey Propagation: An Algorithm for Satisfiability, Random Structures and Algorithms, 27, 201226, 2005.
[3] A. Braunstein and R. Zecchina, Learning by Message-Passing in Networks of Discrete Synapses, Phys. Rev. Lett. 96, 030201, 2006.
[4] J. Chavas, C. Furtlehner, M. Mezard and R. Zecchina, Survey Propagation Decimation Through Distributed Local Computations, Jour. Stat. Mech. P11016, 2005.
[5] J. Chen, M. P. C. Fossorier, Density Evolution for Two Improved BPBased Decoding Algorithms of LDPC codes, IEEE Communications Letters. vol. 6. no. 5 May 2002.
[6] S. Ciliberti, M. Mezard, R. Zecchina, Lossy Data Compression with Random Gates, Phys. Rev. Lett. 95, 038701, 2005.
[7] T. M. Cover, Broadcast Channels, IEEE Trans. Info. Theory, vol 18, no. 1, Jan. 1972.
[8] T. M. Cover and J. A. Thomas, Elements of Information Theory, Wiley, 1991.
[9] U. Erez and S. tenBrink, A Close-to-Capacity Dirty Paper Coding Scheme, IEEE Trans. Info. Theory, vol. 51, no. 10, Oct. 2005.
[10] S. I. Gelfand, Capacity of one broadcast channel, Probl. Inform. Transm. July-Sept. 1977.
[11] E. Martinian and J. Yedidia, Iterative Quantization Using Codes on Graphs, Allerton Conf. Comm. Control and Comuting, Oct. 2003.
[12] K. Marton, A Coding Theorem for the Discrete Memoryless Broadcast Channles, IEEE Trans. Inform. Theory, vol. 25, May 1979.
[13] K. Marton, The Capacity Region of Deterministic Broadcast Channles, in Trans. Int. symp. Inform. Theory, France, 1977.
[14] M. Mezard and R. Zecchina, The Random K-SAT Problem: from an analytic solution to an efficient algorithm, Phys. Rev. E66, 056126, 2002
[15] T. Murayama, Thouless-Anderson-Palmer Approach for Lossy Compression, Phys. Rev. E 69, 035105, 2004.
[16] K. Nakamura, Y. Kabashima, R. M. Zaragoza and D. Saad, Statistical Mechanics of Broadcast Channels Using Low Density Parity Check Codes,International Symposium on Information Theory (ISIT), 2003.
[17] T. J. Richardson, A. Shokrollahi and R. Urbanke, Design of Capacity Approaching Irregular Low-Density Parity Check Codes, IEEE Trans. Inform. Theory, 47(2001), pp. 599-619.
[18] Wei Yu and M. Aleksic, Coding for the Blackwell Channel: A Survey Propagation Approach, International Symposium on Information Theory (ISIT), 2005.
[19] J. Yedidia, W. T. Freeman and Y. Weiss, Generalized Belief Propagation, Advances in Neural Information Processing Systems, vol 13, MIT Press, 2001.

