



**HAL**  
open science

## A novel region graph construction based on trapping sets for the Generalized Belief Propagation

Jean-Christophe Sibel, Sylvain Reynal, David Declercq

► **To cite this version:**

Jean-Christophe Sibel, Sylvain Reynal, David Declercq. A novel region graph construction based on trapping sets for the Generalized Belief Propagation. IEEE ICCS 2012, Nov 2012, Singapore. pp.1. hal-00736234

**HAL Id: hal-00736234**

**<https://hal.science/hal-00736234>**

Submitted on 18 Dec 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A novel region graph construction based on trapping sets for the Generalized Belief Propagation

Jean-Christophe Sibel  
 ETIS/ENSEA/Univ. of Cergy-Pontoise  
 CNRS UMR-8051/INRIA  
 Email: jean-christophe.sibel@ensea.fr

Sylvain Reynal  
 ETIS/ENSEA/Univ. of Cergy-Pontoise  
 CNRS UMR-8051  
 Email: reynal@ensea.fr

David Declercq  
 ETIS/ENSEA/Univ. of Cergy-Pontoise  
 CNRS UMR-8051  
 Email: declercq@ensea.fr

**Abstract**—The Belief Propagation (BP) is an inference algorithm used to estimate marginal probability distributions for any Markov Random Field (MRF). In the realm of Low-Density Parity-Check (LDPC) codes that can be represented by MRF called Tanner graphs, the BP is used as a decoding algorithm to estimate the states of bits sent through a noisy channel. Known to be optimal when the Tanner graph is a tree, the BP suffers from suboptimality when the Tanner graph has a loop-like topology. Furthermore, combinations of loops, namely the trapping sets, are particularly harmful for the decoding. To circumvent this problem were proposed other algorithms, like the Generalized Belief Propagation (GBP) that comes from statistical physics. This algorithm allows to absorb topological structures inside new nodes called regions. An advantage is that the resulting graph, the region graph, is not unique then according to its construction this region graph is a media for the GBP that can provide more accurate estimates than the BP.

In this paper, we propose novel constructions of the region graph for the famous Tanner code of length  $N = 155$  by making use of the trapping sets as basis for the regions.

## I. INTRODUCTION

A well-known problematic of the Markov Random Fields (MRF) is to extract their marginal probability distributions. This inference problem has numerous applications in computer vision, neural networks, image processing, statistical physics and channel coding. In [1] has been proposed the Belief Propagation (BP), known to provide accurate estimates. In the realm of error-correcting coding theory, the BP is a decoding algorithm that helps to recover a sequence of bits sent through a noisy channel. To this end the BP is associated to Low-Density Parity-Check (LDPC) codes that consist in applying artificial constraints between the bits, namely the parity-check equations. The sets of the bits joined with the set of the constraints form an MRF, called the Tanner graph, used as a media for the BP [2]. It is an accurate inference algorithm, meaning that it has relevant performance in terms of the bit-error rate (BER). However the Tanner graphs of most LDPC codes have loop-like topologies, that leads the BP to wrong estimates, making this algorithm suboptimal [3]. In particular, the combination of short loops, as the trapping sets [4], [5], are very harmful. To circumvent this phenomenon, we bring out that the BP is equivalent to the Bethe approximation [6], a method used in statistical physics to estimate averages of thermodynamics quantities by neglecting – to some extent – correlations between interacting variables [7]. As the BP, this technique suffers from the loop-like topology of the graphical representation of the problem, like spin glasses. It has been generalized to the Kikuchi approximation, a technique that consists in gathering interacting variables to absorb loops [8]. The resulting graph, called the region graph, is a media for a new message-passing algorithm, the Generalized Belief Propagation (GBP) that turns out to be more accurate than the BP provided that the region graph has absorbed damageable structures. Along the whole paper, we focus on the Tanner code [9] whose main property is that it can be entirely described by a set of known trapping sets. Then the

region graph can be built by working on them, that allows to lower BER of the GBP comparing with the BP one.

In [6] has been highlighted that the update equations of the GBP does not provide convergence of the algorithm therefore it was proposed to relax the update with the memory by the introduction of a uniform weighting. However, this choice seems quite restrictive. Here we propose a general point of view of the relaxation by the use of a damping factor that depends on the iteration, and we make reveal that it provides relevant properties of the GBP.

The section II deals with preliminaries about the LDPC codes, the BP and the trapping sets. In the section III, are exposed the origins of the GBP, the region graph construction rules. We resume with a classical construction and two novel constructions adapted to the Tanner code. We also introduce the general damping factor and the different evolutions we propose with associated results. In the section IV, we expose results about our novel constructions.

## II. PRELIMINARIES

### A. LDPC Codes – Belief Propagation

We consider  $N$  bits  $\mathbf{X} = \{x_1, \dots, x_N\}$  sent through a noisy channel. In the whole paper, we use the additive white gaussian channel whose power is given by the Signal-to-Noise Ratio (SNR). An LDPC code of length  $N$  is modeled by  $M < N$  parity-check equations  $\mathbf{C} = \{c_1, \dots, c_M\}$  such that:

$$\forall j \in \{1, \dots, M\}, \quad c_j(x_1, \dots, x_N) = \sum_{x_i \in \mathcal{N}_j} x_i$$

where the sum is computed over  $\text{GF}(2)$ , and  $\mathcal{N}_j$  is a set of bits that are the neighborhood of  $c_j$  determined by the code. We define  $\mathcal{N}_i$  the neighborhood of  $x_i$  by the set  $\{c_j | x_i \in \mathcal{N}_j\}_j$ . A couple  $\{x_i, c_j \in \mathcal{N}_i\}$  forms an edge  $e_{ij}$  between a variable node associated to  $x_i$  and a check node associated to  $c_j$  in a bipartite graph  $G = \{\mathbf{X} \cup \mathbf{C}\} \cup \{e_{ij}\}_{ij}$ . We call this graph the Tanner graph. The BP is an inference algorithm that consists in passing messages along the edges of  $G$ . We denote  $m_{ij}^{(k)}$  (resp.  $n_{ji}^{(k)}$ ) the message from the variable node  $x_i$  (resp. the check node  $c_j$ ) to the check node  $c_j$  (resp. the variable node  $x_i$ ) at iteration  $k$ , where a message is the probability distribution of the receiver conditioned to the emitter. For any edge  $e_{ij}$  the update equations are [2]:

$$\begin{aligned} n_{ji}^{(k)}(0) &= \frac{1}{2} + \frac{1}{2} \prod_{x_{i'} \in \mathcal{N}_j \setminus x_i} \left( 2m_{i'j}^{(k-1)}(0) - 1 \right) \\ n_{ji}^{(k)}(1) &= 1 - n_{ji}^{(k)}(0) \\ m_{ij}^{(k)}(x_i) &= \frac{l_i(x_i)}{Z_{ij}} \prod_{c_{j'} \in \mathcal{N}_i \setminus c_j} n_{j'i}^{(k)}(x_i) \end{aligned} \quad (1)$$

$$m_{ij}^{(k)}(x_i) = \frac{l_i(x_i)}{Z_{ij}} \prod_{c_{j'} \in \mathcal{N}_i \setminus c_j} n_{j'i}^{(k)}(x_i) \quad (2)$$

where  $Z_{ij}$  is a normalization factor such that:  $m_{ij}^{(k)}(0) + m_{ij}^{(k)}(1) = 1$ . To initialize the algorithm, we use the likelihoods  $\{l_i(x_i) = p_i(y_i|x_i)\}_i$  computed with observable nodes  $\{y_1, \dots, y_N\}$ :  $\forall e_{ij}, m_{ij}^{(0)} = l_i(x_i)$ . The estimate of the marginal probability distribution on  $x_i$ , i.e., the belief on a variable  $x_i$  is:

$$b_i^{(k)}(x_i) = \frac{l_i(x_i)}{Z_i} \prod_{j \in \mathcal{N}_i} n_{ji}^{(k)}(x_i) \quad (3)$$

where  $Z_i$  is a normalization factor such that:  $b_i(0) + b_i(1) = 1$ . The estimate of the sent bits are:  $\forall i \in \{1, \dots, N\}$ ,  $\hat{x}_i^{(k)} = \arg \max_{x_i} b_i^{(k)}(x_i)$ . The algorithm is stopped as soon as all the parity-check equations are verified by the output bits or if the convergence is reached:  $\forall x_i, \forall c_j \in \mathcal{N}_j$ ,  $n_{ji}^{(k+1)} = n_{ji}^{(k)}$ .

### B. Trapping-sets

When  $N$  is infinite, it has been proved [4] that the BP reaches the optimal solution given by the maximum likelihoods decoder because the Tanner graph is a tree. However, in practical cases,  $N$  is finite, and most LDPC codes contain loops inside their Tanner graph making the BP suboptimal [3]. Furthermore, the short loops and the combinations thereof are very harmful. The trapping sets exposed in [5] are well-known topological structures that harm the decoding. A trapping set  $TS(a,b)$  is a Tanner graph that contains  $a$  variable nodes and  $b$  unverified check nodes. We consider the case  $a = 5$  and  $b = 3$  whose Tanner graph is represented on Fig.1.

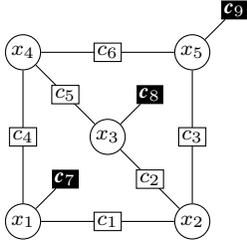


Fig. 1. Tanner graph of the TS(5,3)

Such a structure takes place in many LDPC codes, including the Tanner code [9] of length  $N = 155$  with  $M = 93$ . The interesting property of this LDPC code is that it is entirely covered by a set  $\mathcal{T}$  of 155 trapping sets, i.e.:

$$\forall c_j \in \mathbf{C}, \exists t \in \mathcal{T}, c_j \in t \quad (4)$$

$$\forall x_i \in \mathbf{X}, \exists t \in \mathcal{T}, x_i \in t \quad (5)$$

The BER of the Tanner code suffers from an error floor [4] for high SNR values, especially due to the TS(5,3). A current problematic is then to lower the error floor. The finite alphabet iterative decoders are good challengers to solve this problem but we do not deal with these algorithms, we invite the interested reader to pay attention to the article [10] for more details.

## III. GENERALIZED BELIEF PROPAGATION

### A. Origins

In statistical physics, spin glasses are well-known Markov Random Fields (MRF) in which a problem is to minimize the free energy. This optimization has been demonstrated to be equivalent to the inference problem [6]. Most methods rely on region-based approximations, where the MRF is clustered such that the belief  $b(\mathbf{x})$  is a product of the beliefs  $\{b_r(\mathbf{x}_r)\}_r$  over the clusters:

$$b(\mathbf{x}) = \prod_{r \in \mathcal{R}} b_r^{c_r}(\mathbf{x}_r) \quad (6)$$

where  $\{c_r\}_r$  are overcounting numbers provided to account only once each variable of the MRF [11]. The Bethe Approximation (BA) [7] is a region-based approximation that takes into account the short-range correlations between the spins. It can be modeled by a factor

graph that is identical to a Tanner graph if and only if the factors define hard constraints, which is the case of LDPC codes. Therefore, with computations detailed in [12], it has been concluded that the BA was equivalent to the BP. This equivalence provides a solution to the problem of the loop-like structure of the Tanner graph. Indeed, the BA has been extended to the Kikuchi Approximation (KA) [8], a method to minimize the free energy as the BA with the slight difference that it considers larger ranges of correlation, thereby can absorb loops. The associated inference algorithm is the GBP.

### B. Region-graph

The GBP is a message-passing algorithm that runs on a Bayesian network, namely the region graph, whose nodes are called regions. Each region  $r$  is composed of a set of check nodes  $\mathbf{C}_r$  and a set of variables  $\mathbf{X}_r \neq \emptyset$  with the following condition: if  $\mathbf{C}_r$  is not an empty set then  $\forall c_j \in \mathbf{C}_r, \mathcal{N}_j \subseteq \mathbf{X}_r$ . Here we expose the rules of construction then we present three different constructions, two of which are our novel constructions suited to the Tanner code.

1) *Rules of construction:* The region graph is a graph of  $L$  levels. We present the construction in two steps. The first step is the construction of the first level: it is the choice of the initial regions set  $\mathcal{R}_0 = \{r_k^0\}_k$ , i.e. the choice of the harmful structures of the Tanner graph that need to be neutralized.  $\mathcal{R}_0$  is valid if:

$$\begin{aligned} \forall c_j \in \mathbf{C}, \exists r_k^0 \in \mathcal{R}_0, \quad c_j \in \mathbf{C}_{r_k^0}, \mathcal{N}_j \subseteq \mathbf{X}_{r_k^0} \\ \forall x_i \in \mathbf{X}, \exists r_k^0 \in \mathcal{R}_0, \quad x_i \in \mathbf{X}_{r_k^0} \end{aligned}$$

The second step is the construction of the other levels  $\{\mathcal{R}_1, \dots, \mathcal{R}_{L-1}\}$ . To build  $\mathcal{R}_l$  we search for the common variables and checks between the regions of the previous level  $\mathcal{R}_{l-1}$ . Thus a region  $r_k^l$  belongs to  $\mathcal{R}_l$  if we can find an associated parent set  $\mathcal{P}_k^l \subset \mathcal{R}_{l-1}$  such that:

$$\begin{aligned} \forall c_j \in \mathbf{C}_{r_k^l}, \forall r_j^{l-1} \in \mathcal{P}_k^l, \quad c_j \in \mathbf{C}_{r_j^{l-1}} \\ \forall x_i \in \mathbf{X}_{r_k^l}, \forall r_j^{l-1} \in \mathcal{P}_k^l, \quad x_i \in \mathbf{X}_{r_j^{l-1}} \end{aligned}$$

The region graph reaches its last level  $\mathcal{R}_{L-1}$  when there is no intersection anymore between the regions of  $\mathcal{R}_{L-1}$ . To each region  $r$  is associated an overcounting number  $c_r \in \mathbb{Z}$  such that a graph of regions is a true region graph if:

$$\forall r \in \mathcal{R}, \sum_{\substack{p \in \mathcal{R} \\ r \subset p}} c_p = 1 \quad (7)$$

where  $r \subset p$  means that  $\mathbf{X}_r \subseteq \mathbf{X}_p$  and  $\mathbf{C}_r \subseteq \mathbf{C}_p$ .

2) *Classical construction:* Proposed in [6], it consists in including only one check node per initial region:

$$\begin{aligned} \forall c_j \in \mathbf{C}, \exists! r_k^0 \in \mathcal{R}_0, \quad \mathbf{C}_{r_k^0} = c_j \\ \forall (r_k^0, r_j^0) \in \mathcal{R}_0^2, \quad \mathbf{C}_{r_k^0} \cap \mathbf{C}_{r_j^0} = \emptyset \end{aligned}$$

The next levels are constructed as exposed previously. As an example we display on Fig.2(a) the Tanner graph of the Hamming code, and on Fig.2(b) the region graph from this method. In this case, the region graph has its own topology. However, for most LDPC codes, the sparsity of the Tanner graph makes this method generate a region graph whose topology is identical to the Tanner graph one. Thus the GBP provides the same solution as the BP, which is not relevant.

3) *Novel constructions:* In the case of the Tanner code, we use the trapping sets TS(5,3) introduced previously to create the level  $\mathcal{R}_0$ . The complexity of the GBP grows as the number of variable nodes contained in the regions is increased. A complete TS(5,3), i.e. the TS(5,3) together with the neighborhoods of all the check nodes, is

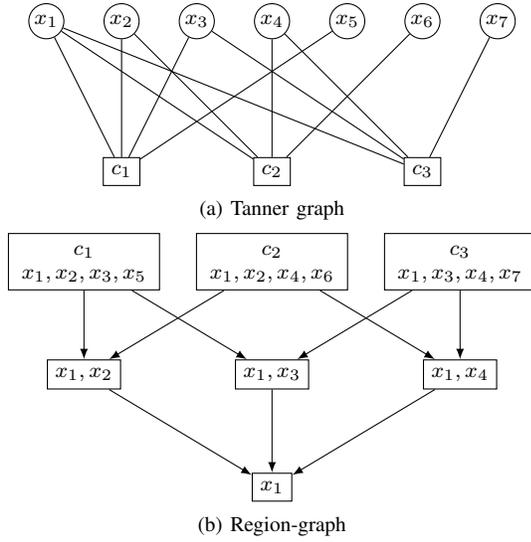


Fig. 2. Hamming code

compounded of 35 variable nodes then any message from  $\mathcal{R}_0$  to  $\mathcal{R}_1$  amounts to  $2^{35}$  operations, which is not fair at all. Therefore, we must use smaller initial regions to lower the complexity.

The first idea we propose is presented on Fig.3, where each TS(5,3) is split into three initial regions, each of them containing two check nodes, one unverified check node, and their associated neighborhoods. We call this construction a triplets construction.

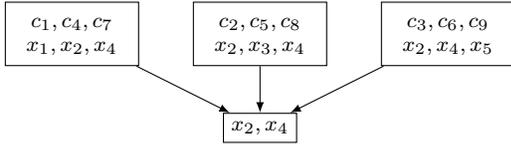


Fig. 3. Region-graph of the TS(5, 3) – Triplets split

Each initial region contains 13 variable nodes, making the computation of a message from  $\mathcal{R}_0$  to  $\mathcal{R}_1$  of  $2^{13}$  operations, which is dramatically less than the complexity given by the previous proposition. Locally, the region graph of a TS(5,3) is loop free, that makes the decoding optimal. In [13] is introduced that a region graph can offer a GBP more accurate than the BP simply by adding to the Tanner graph super regions. Therefore, we only link the triplets to the check nodes of the Tanner graph considering that a region constituted by a check  $c_j$  is actually  $\{c_j \cup \mathcal{N}_j\}$ . Unfortunately, when we consider the whole code, the trapping sets are strongly correlated each other to each other. Thus, the region graph is not loop free, making the decoding not optimal. However, the construction provides a GBP that can offer better properties than the BP as we will see in the results part.

The second proposition is presented on Fig.4, where each TS(5,3) is split into three initial regions, each of them containing four check nodes, two unverified check nodes and their associated neighborhoods. We call this a triangles construction. Each initial region contains 24 variable nodes, making the computation of a message from  $\mathcal{R}_0$  to  $\mathcal{R}_1$  of  $2^{24}$  which is greater than what we got from the previous construction, but still lower than  $2^{35}$ . The advantage of this construction is that it absorbs the short loops of the TS(5,3) and the unverified check nodes. As for the triplets construction, locally the region graph is loop free but embedded to the whole Tanner graph, a

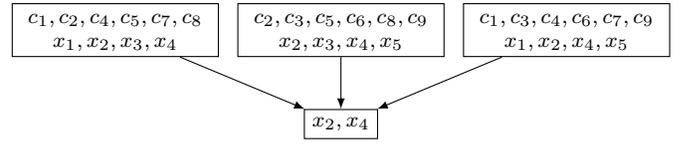


Fig. 4. Region-graph of the TS(5, 3) – Triangles split

TS(5,3) has complex links with its fellow beings. However, we can hope that it won't be affected by trapping sets effects. We expose the results of these construction in the next section.

### C. Message-passing

The GBP consists in passing messages along the edges of the region graph. Unlike the Tanner graph, this one is a directed graph because of the parent links. Thus, we compute a message  $m_{r,q}^{(k)}$  if and only if  $r$  is a parent of  $q$ , i.e.  $q$  is a child of  $r$  denoted  $q \in \mathcal{C}_r$ . We define  $\mathcal{F}_r$  the family of  $r$  as  $\{q | \mathcal{C}_q \subseteq \mathcal{C}_r, \mathbf{X}_q \subseteq \mathbf{X}_r\}_q$ , and its descendants  $\mathcal{D}_r = \mathcal{F}_r \setminus r$ . According to [6], [11] the messages from a region  $r$  to a region  $q$  at iteration  $k$  are:

$$m_{r,q}^{(k)}(\mathbf{x}_q) = \frac{\sum_{\mathbf{x}_{r \setminus q}} L_{r \setminus q}(\mathbf{x}_{r \setminus q}) c_{r \setminus q}(\mathbf{x}_r) \prod_{\substack{p \in \mathcal{R} \setminus \mathcal{F}_r \\ s \in \mathcal{F}_r \setminus \mathcal{F}_q}} m_{ps}^{(k-1)}(\mathbf{x}_s)}{Z_{r,q} \prod_{\substack{p \in \mathcal{D}_r \setminus \mathcal{F}_q \\ s \in \mathcal{D}_q}} m_{ps}^{(k)}(\mathbf{x}_s)} \quad (8)$$

where:

- $\mathbf{x}_{r \setminus q}$  is the state of  $\mathbf{X}_{r \setminus q} = \mathbf{X}_r \setminus \mathbf{X}_q$ ,
- $L_{r \setminus q}(\mathbf{x}_{r \setminus q}) = \prod_{x_i \in \mathbf{X}_{r \setminus q}} l_i(x_i)$ ,
- $c_{r \setminus q}(\mathbf{x}_r) = \prod_{c_j \in \mathcal{C}_r \setminus \mathcal{C}_q} c_j(\mathbf{x}_{\mathcal{N}_j})$ ,
- $Z_{r,q}$  is a normalization factor to ensure that:  $\sum_{\mathbf{x}_q} m_{r,q}^{(k)}(\mathbf{x}_q) = 1$ .

In [6] is assumed that the iteration indices of the messages at the denominator must be the same as the one of  $m_{r,q}$  to keep relevant performance. Though, it prevents from using an efficient parallelisation of the computation. Simulations have shown that it is possible to replace at the denominator the messages  $m_{ps}^{(k)}$  by  $m_{ps}^{(k-1)}$ , without damaging the results of the algorithm, it helps to get a faster parallel algorithm. For any region  $r$  at iteration  $k$  we compute the beliefs:

$$b_r^{(k)}(\mathbf{x}_r) = \frac{1}{Z_r} L_r(\mathbf{x}_r) c_r(\mathbf{x}_r) \prod_{\substack{p \in \mathcal{R} \setminus \mathcal{F}_r \\ s \in \mathcal{F}_r}} m_{ps}^{(k)}(\mathbf{x}_s) \quad (9)$$

where  $Z_r$  is a normalization factor to ensure that  $\sum_{\mathbf{x}_r} b_r^{(k)}(\mathbf{x}_r) = 1$ . To get the beliefs for the single variables we only need to compute the marginal probability distributions of the regions beliefs. Thus for any region  $r$ :

$$\forall x_i \in \mathbf{X}_r, b_i^{(k)}(x_i) = \frac{1}{Z_i} \sum_{\mathbf{X}_{r \setminus i}} b_r^{(k)}(\mathbf{x}_r) \quad (10)$$

where  $Z_i$  is a normalization factor to ensure that  $\sum_{x_i} b_i^{(k)}(x_i) = 1$ . The estimates  $\{\hat{x}_1, \dots, \hat{x}_N\}$  are computed then as for the BP.

### D. Damping factor

In [6] it has been highlighted the fact that the GBP suffers from a poor convergence. Therefore the authors introduced a constant uniform damping factor  $w = 0.5$  that helps to balance between the memory  $\{m_{r,q}^{(k-1)}\}_{r,q}$  of the algorithm and the update  $\{F_{r,q}\}_{r,q}$  by the following relaxation:

$$\forall q \in \mathcal{R}, \forall r \in \mathcal{P}_q, m_{r,q}^{(k)} = \frac{1}{2} F_{r,q} + \frac{1}{2} m_{r,q}^{(k-1)} \quad (11)$$

Here we propose a more general point of view by writing:

$$\forall q \in \mathcal{R}, \forall r \in \mathcal{P}_q, \quad m_{rq}^{(k)} = w_k F_{rq} + (1 - w_k) m_{rq}^{(k-1)} \quad (12)$$

where  $w_k$  is a decreasing damping factor. The last value  $w_K$  must be a null value to ensure that at the end:

$$\forall q \in \mathcal{R}, \forall r \in \mathcal{P}_q, \quad m_{rq}^{(K)} = m_{rq}^{(K-1)}$$

The challenge is to find a suited sequence  $\{w_0, \dots, w_K\}$  that provides both convergence and relevant performance. We consider four different functions displayed on the Fig.5:

- (C) constant:  $w_k = w_0$ ,
- (A) affine:  $w_k = w_0(1 - \frac{k}{K})$ ,
- (P) parabolic:  $w_k = w_0(1 - \frac{k}{K})^n, n > 1$ ,
- (IP) inverse parabolic:  $w_k = w_0(1 - (\frac{k}{K})^n), n > 1$ .

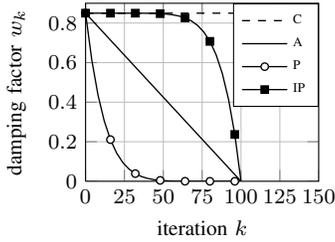


Fig. 5. Damping factors,  $w_0 = 0.85$

The balance between the memory  $m_{rq}^{(k-1)}$  and the update  $F_{rq}$  is different from case to case. In the case of a parabolic function, the memory is quickly highlighted at the expense of the update, whereas in the case of an inverse parabolic function the memory is practically ignored for a long time. The affine coefficient enables a progressive oversight of the update up to a complete use of the memory. The constant function is what is proposed in [6] with different values. We present on Fig.6 the evolution of the average Bit-Error Rate  $BER_k$  along the iterations with these different damping factors for four different initializations  $w_0$ . The simulations have been realized with an SNR of 2.90 dB, i.e. at the end of the waterfall where the trapping sets are really harmful. The first observation to do from 6(d) is that a constant damping factor to the value  $w_k = 1.00$  provides the worst  $BER_k$  meaning that the pure update is not efficient at all as it was indicated previously. From Fig.6(a) joined with Fig.6(b), it appears that a low value of  $w_0$  is recommended to get fast convergence of the GBP. In addition, we see that whatever the decreasing rule of  $w_k$ , the value  $w_0 = 0.5$  lower the BER to a steady value at  $k = 19$  unlike the value  $w_0 = 0.25$  has a longer transient phase before convergence at  $k=22$ . Thus, the initial uniform distribution between the memory and the update is the more relevant choice. Furthermore, on Fig.6(c) and 6(d) we observe that for high  $w_0$  the parabolic  $w_k$ , that gives quickly a low  $w_k$ , fosters the fastest convergence compared with the constant, the affine and the inverse parabolic one. As an example, we see that the parabolic GBP, i.e. the GBP with the parabolic  $w_k$ , converges at  $k = 22$ , whereas the inverse parabolic GBP converges at  $k = K$  i.e. when the convergence is completely forced. Although the parabolic GBP is dramatically faster, the steady value  $BER_K$  are the same for both, meaning that  $w_k$  is a convergence booster. The second convergence booster is the affine one, which is not surprising because it is a particular case of the parabolic function with  $n = 1$ . Then one could assume that increasing  $n$  provides faster convergence. However other simulations that we do not present here do not confirm this, from  $n = 2$  the  $BER_k$  are indistinguishable for the parabolic law. On the four figures, we can extract, among other, a common property: whatever  $w_0$  the parabolic GBP always converges nearly  $k = 20$ . This invariance is crucial because it brings out that  $w_0$  is a degree of freedom for the parabolic GBP, unlike the other decreasing laws.

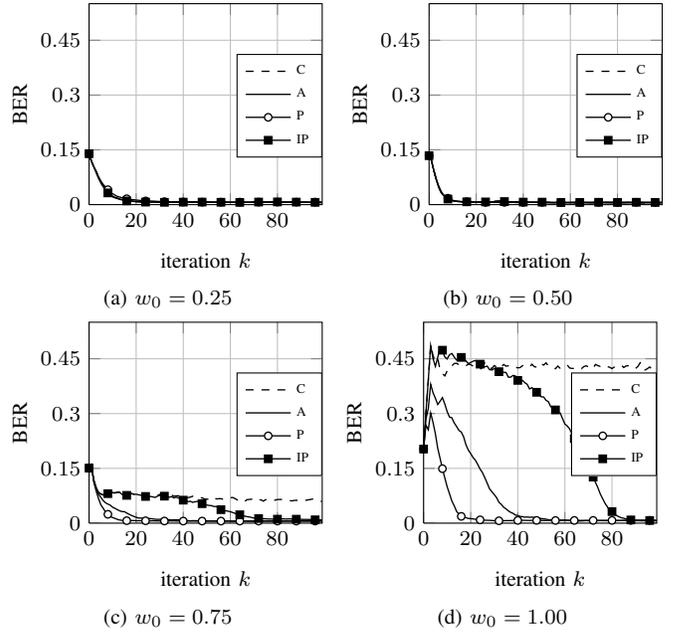


Fig. 6.  $BER_k$  of the GBP on the Tanner code with the triplets construction,  $K = 100$

The surprising information from these observations is that they indicate that a fast and accurate GBP is enhanced by the memory and not by the update that defines the GBP itself. Thus, an assumption is that the GBP acts like a starter we must forget as quick as possible, it seems more relevant to use only a small percentage of the update for small value of  $k$  and to forget it as quick as possible. We do not display the result of the triangles construction here to keep a reasonable paper length, but simulations carried on this case highlight the same assumption. Simulations on other values of the SNR and other codes like the Hamming code made also reveal the same conclusion. It seems that the property of choosing a parabolic decreasing of  $w_k$  with  $w_0 < 0.5$  to obtain relevant performance with few iterations belongs to the GBP and not to the code.

#### IV. RESULTS

In this section we expose results of simulations in terms of the BER between the BP and the GBP. We present first the results on the Hamming code to evaluate the classical construction. Then we present the results on the Tanner code with our two constructions to bring out their efficiency.

##### A. Hamming Codes

On Fig.7 is displayed the average BER along the SNR values for the BP and the GBP, used on Hamming codes of length  $N = 7$  and  $N = 15$ .

On Fig.7(a) we displayed the GBP with parabolic damping factor. It appears that the GBP is not very different to the BP except for high SNR values for which we see that the BER of the GBP is lower than the BP one. This indicates that the classical construction brings only a slight improvement in this case. Concerning Fig.7(b), the GBP reveals better performance for an affine damping factor, which is a parabolic one with  $n = 1$ , than for a constant damping factor  $w = 1$ . Once again, it is a relevant illustration of the poor result of the GBP pure update. Actually, these examples are useful to highlight the fact that the GBP turns out to be performant with the classical construction if the region graph is of different topology

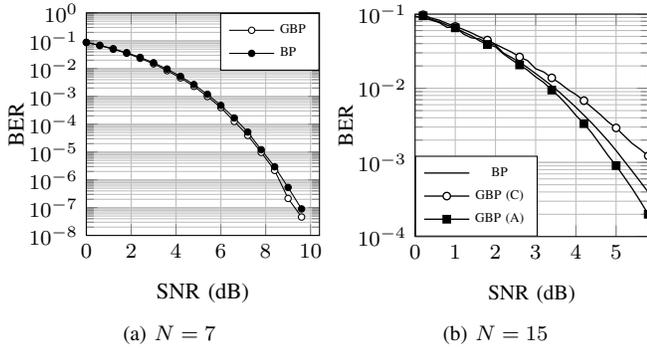


Fig. 7. BER of the BP and the GBP with the classical construction on Hamming codes

compared with the Tanner graph. Moreover, for most LDPC codes, the sparsity of the Tanner graph implies a region graph whose topology is the same as the Tanner graph one. Thus in these cases, there is not any improvement.

### B. Tanner Code

On Fig.8 we display the  $BER_k$  for a particularly error event of 7 errors that makes the decoding by the BP oscillate because of the TS(5,3). We also have simulated on other noise realizations that harm the estimation for the same reason and we obtain the same results as what is presented. First of all, we can see that the GBP by the triplets construction has  $BER_k$  values lower than that of the BP. Furthermore, it appears on Fig.8(c) that the inverse parabolic function implies errors from  $k = 22$ , unlike the parabolic and the affine ones.

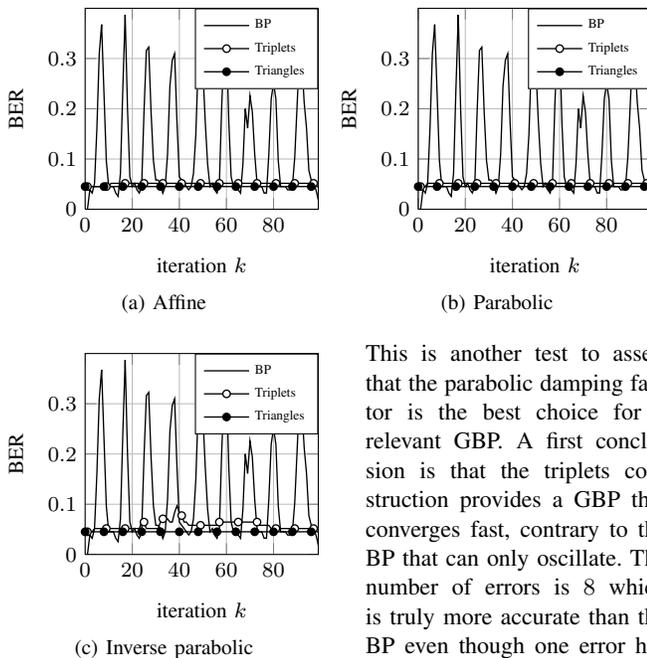


Fig. 8.  $BER_k$  of the BP and the GBPs on the Tanner code

result, we see that the convergence is quickly reached even for the inverse parabolic function. In addition, the number of errors by this construction is 7 i.e. there is no more errors than in the error event. Thus the GBP has a more powerful error correction by

this construction than the triplets construction. One would assert that periodically, the BP corrects more errors than the GBP, whatever the construction. Indeed, every 9 iterations in average, the BP reaches a  $BER_k$  of 0.0322581, meaning that there are only 5 errors left. However, this result cannot be accurately predicted because the periodical aspect is not perfect. Moreover, the difference between this minimum value and the maximum average  $BER = 0.32$  is too large to use this pseudo-periodical pattern as a result to assert that around 9 iterations the lowest  $BER_k$  is reached. Therefore, the GBP is truly more reliable than the BP, because the  $BER_k$  does not change. In addition, other simulations for other SNR values with different error events made us convinced that the GBP always gets a BER equal or lower than the BP one, making it an accurate algorithm, provided that it is run on suited region graph constructions as we proposed.

### V. CONCLUSION

In this paper, we have exposed novel constructions of the region graph for the Tanner code, based on its trapping sets. The simulations we presented indicated that the performance of the GBP were improved by these constructions, raising that a triangles construction absorb the trapping sets better than a triplets construction. We have showed that the use of a damping factor is necessary to ensure relevant performance in terms of the BER, and that the classical proposition is not the best choice. More generally, our work enabled to understand that the GBP needs extensive research on its region graph to give interesting performance, that can go beyond the BP. Finally, the constructions we proposed can be adapted to any LDPC code whose Tanner graph can be represented by a set of trapping sets, making our studies relevant.

### ACKNOWLEDGMENT

This work was supported by the French ANR Defis program under contract ANR-08-EMER-003 (COCQ project).

### REFERENCES

- [1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [2] F.R. Kschischang, B. J. Frey and H.A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. on Inf. Theory*, vol. 47, 2001, NO. 2.
- [3] K.P. Murphy, Y. Weiss and M.I. Jordan, "Loopy belief propagation for approximate inference: an empirical study," in *Proc. Uncertainty in AI*, 1999.
- [4] T.J. Richardson, "Error floors of LDPC codes," in *Proc. 41st Annual Allerton Conf. on Comm. Cont. and Comp.*, 2003.
- [5] S. Sankaranarayanan, S. K. Chilappagari, R. Radhakrishnan and B. Vasic, "Failures of the Gallager B Decoder: Analysis and Applications," in *ITA Workshop UCSD*, 2006.
- [6] J.S. Yedidia, W.T. Freeman and Y. Weiss, "Constructing free energy approximations and Generalized Belief Propagation algorithms," *IEEE Trans. on Inf. Theory*, vol. 51, pp. 2282–2313, 2004.
- [7] H.A. Bethe, "Statistical Theory of Superlattices," in *Proc. Roy. Soc. London*, 1935.
- [8] R. Kikuchi, "A Theory of Cooperative Phenomena," *Phys. Rev.*, vol. 81, pp. 988–1003, 1951.
- [9] R.M. Tanner, R. Michael, D. Sridhara and T. Fuja, "A Class of Group-Structured LDPC Codes," 2001.
- [10] L. Danjean, D. Declercq, S.K. Planjery and B. Vasic, "On the Selection of Finite Alphabet Iterative Decoders for LDPC codes on the BSC," in *Proc. IT Workshop*, 2011.
- [11] P. Pakzad et V. Anantharam, "Estimation and marginalization using Kikuchi approximation methods," *Neural Computation*, vol. 17, pp. 1836–1873, 2003.
- [12] A. L. Yuille, "CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation," *Neural Computation*, vol. 14, 2002.
- [13] M. Welling, "On the Choice of Regions for Generalized Belief Propagation," in *Proc. UAI*, 2004.