Décodage des codes LDPC par le CCCP

Jean-Christophe SIBEL, Sylvain REYNAL, David DECLERCQ

Laboratoire ETIS / ENSEA / Univ. de Cergy-Pontoise / CNRS UMR 8051
6 avenue du Ponceau, CS 20 707 CERGY, 95014 Cergy-Pontoise Cedex , France
jean-christophe.sibel@ensea.fr
reynal@ensea.fr, declercq@ensea.fr

Résumé – L'équivalence entre les points fixes du Belief Propagation (BP) et les points stationnaires de l'énergie libre variationnelle de Bethe est une propriété largement démontrée dans la littérature. Cependant, les équations de message-passing du BP ne sont pas strictement orientées vers la minimisation de cette énergie. Ainsi, le décodage des codes LDPC par le BP présente des problèmes de convergence et de performance tels qu'il est difficile de s'approcher au plus près du Maximum Likelihood Decoding (MLD), le décodeur optimal. L'algorithme ConCave-Convex Procedure (CCCP), basé sur une résolution par optimisation lagrangienne stricte, permet de résoudre cette difficulté. Très peu connu dans le domaine des codes correcteurs d'erreurs, le CCCP n'a quasiment jamais été utilisé en tant qu'algorithme de décodage. Dans cet article, nous donnons les équations de message-passing pour des codes LDPC de degrés quelconques, et nous présentons les performances en termes de convergence et de taux d'erreur binaire. Le CCCP a un temps de calcul supérieur à celui du BP étant donné que l'algorithme repose sur l'utilisation de deux boucles imbriquées. Cependant, la précision du CCCP en termes de taux d'erreur binaire est significativement plus bas que celui du BP, ce qui en fait un décodeur plus puissant.

Abstract – Equivalence between Belief Propagation (BP) fixed points and Bethe stationnary points has been widely proved in litterature. However, update equations of the BP algorithm are not strictly computed according to the minimization of this energy. Therefore, decoding Low-Density Parity-Check (LDPC) codes exhibits such lacks of convergence and performance that it appears difficult to reach the optimal Maximum Likelihood Decoding. ConCave-Convex Procedure (CCCP) is an algorithm based on a strict minimization according to the Lagrange theory that allows us to solve this problem. CCCP is a little known decoding method then it has almost never been used as a decoder. In this paper we provide update equations of CCCP for any LDPC code, and we present associated performance in terms of convergence and binary error rate. CCCP is made of two nested loops therefore the corresponding computation time for a decoding process is larger than BP computation time. However, the binary error rate of CCCP outperforms the binary error rate of BP, making it an efficient decoder.

1 Introduction

Les codes Low-Density Parity-Check (LDPC) [1] permettent d'approcher asymptotiquement la borne de Shannon et offrent, par leur faible densité, des possibilité de décodage de faible complexité. Le Belief Propagation (BP) est l'algorithme servant de référence pour les décoder. Introduit originellement en tant que solution au problème d'inférence sur les arbres et réseaux bayésiens [2], il a été étendu à l'inférences statistique sur la plupart des graphes tels que les champs de Markov aléatoires et les graphes factoriels [3]. Le BP est donc applicable à tout problème d'inférence modélisable graphiquement, il est utilisé pour les réseaux de neuronnes [4], le traitement d'images et vidéos [5], les verres de spins [6], le décodage [7].

Le BP est un algorithme itératif de message-passing, son fonctionnement consiste à utiliser les arêtes du graphe comme support pour la propagation de messages entre les noeuds du graphe. Un message est une approximation de la distribution de probabilité du destinataire conditionnelement à l'expéditeur. L'agorithme retourne au final les approximations des marginales de chaque noeud du graphe, appelées beliefs, dont on peut extraire les états les plus probables. Dans le cadre des codes LDPC, ces états correspondent à l'estimé du mot de code envoyés à travres le canal de transmission.

Il a été montré que les points fixes du BP sont équivalents aux états stationnaires de l'Energie Libre Variationnelle de Bethe (ELVB) [3]. Cette équivalence n'est vrai qu'à l'équilibre, en effet les équations de mises à jour des messages pour le BP ne correspondent pas strictement à une minimisation lagrangienne sous contraintes de l'ELVB. De plus l'ELVB n'est pas une fonctionnelle convexe des beliefs. Cette propriété est intimement liée au fait que les graphes factoriels ne sont que très rarement des arbres, *i.e.* ils présentent de nombreux cycles et combinaisons de cycles. Le BP ayant été inventé pour l'inférence bayésienne sur des arbres, la méthode n'est alors plus exacte, elle est sous-optimale au sens du maximum de vraisemblance. Le caractère cyclique implique certains comportements du BP bien différents de la convergence vers des points fixes tels que des oscillations voire du chaos [8].

Des études ont été menées [9] pour extraire des conditions de convexité de l'ELVB et pour la « convexifier ». Cependant les conditions pour appliquer cette méthode sont très restrictives. Une méthode alternative consiste à minimiser l'ELVB par l'algorithme ConCave-Convex Procedure (CCCP) [10]. Cette

méthode, également appelée double-loop, assure une minimisation sous contrainte itérative, par conséquent elle s'avère plus précise que le BP. Initialement créée sur des graphes type pairwse ($d_c = 2$), il n'apparaît aucune raison de ne pas étendre cet algorithme à tout graphe factoriel de degrés quelconques.

Dans cet article nous donnons les équations de mises à jour du CCCP pour des graphes factoriels de degrés quelconques, et nous présentons des résultats de performances sur des codes LDPC en termes de taux d'erreur binaire et de convergence.

2 Codes LDPC

2.1 Définition

Un code LDPC binaire transforme tout vecteur binaire **u** de taille k en un mot de code **x** de taille N. Pour k fixé, nous pouvons donc compter 2^k mots de code, dont l'ensemble est noté $\mathcal{C}$. Chaque bit x_i, $1 \leq i \leq N$, est corrélé à ces semblables par M équations de parité, représentées par une matrice dite *matrice de parité* H.

Exemple : code de Hamming ($N = 7, k = 4$)

A tout vecteur **u** de taille $k = 4$ est associé un mot de code **x** de taille $N = 7$ tel que :

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad H.\mathbf{x} = \mathbf{0}$$

Toute ligne H_a d'une matrice de parité H est représentée par un facteur c_a. L'état de ce facteur reflète la validité des bits $\mathbf{X}_a$ dans l'état $\mathbf{x}_a$, *i.e.* les arguments de l'équation H_a :

$$c_a(\mathbf{x}_a) = \begin{cases} 1 & \text{si l'équation de parité est satisfaite} \\ 0 & \text{sinon} \end{cases}$$

2.2 Lien avec la physique statistique

A tout vecteur binaire **x** de taille N est associée une probabilité d'occurence :

$$p(\mathbf{x}) \propto \prod_a c_a(\mathbf{x}_a) \tag{1}$$

de telle sorte que :

$$p(\mathbf{x}) = \begin{cases} 2^{-k} & \text{si } \mathbf{x} \in \mathcal{C} \\ 0 & \text{sinon} \end{cases}$$

Cette formulation permet de définir un code LDPC comme un graphe factoriel [3]. Les théorèmes de la physique statistique [6] indiquent alors que la distribution de probabilité d'un code LDPC est également celle d'un champs de Boltzmann :

$$p(\mathbf{x}) \propto e^{-E(\mathbf{x})} \tag{2}$$

où $E(\mathbf{x})$ est appelé *fonction d'énergie* du système dans l'état **x**. Il est alors prouvé [9] que rechercher l'état fondamental

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} E(\mathbf{x})$$

est équivalent à minimiser l'Energie Libre Variationnelle de Bethe F_β à l'aide de distributions tests $\{b_i\}_i, \{b_a\}_a$, appelées *beliefs*, sur les marginales $\{p_i\}_i, \{p_a\}_a$:

$$\{\{p_i\}_i, \{p_a\}_a\} = \arg\min_{\{b_i\}_i, \{b_a\}_a} F_\beta(\{b_i\}_i, \{b_a\}_a) \tag{3}$$

3 Concave-Convex Procedure

3.1 Energie libre

Nous notons F_β l'ELVB, elle se calcule comme la différence entre l'énergie moyenne et l'entropie. Nous notons $\{X_1, \ldots, X_N\}$ l'ensemble des N variables aléatoires qui composent le graphe factoriel, et $\{c_a, \ldots, c_M\}$ l'ensemble des M facteurs qui lient ces variables entre elles. Les beliefs sur les variables et les potentiels sont respectivement notés :

$$\{b_i(x_i)\} \equiv b_i(X_i = x_i)\}_{i,x_i} \tag{4}$$

$$\{b_a(\mathbf{x}_a) \equiv b_a(\mathbf{X}_a = \mathbf{x}_a)\}_{a,\mathbf{x}_a} \tag{5}$$

où $\mathbf{X}_a$ est l'ensemble des variables aléatoires qui sont reliées au potentiel c_a. L'ELVB est [3] :

$$\begin{aligned} &F_\beta(\{b_i(x_i)\}_{i,x_i}, \{b_a(\mathbf{x}_a)\}_{a,\mathbf{x}_a}) \\ &= \sum_{a=1}^{M} \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \log \frac{b_a(\mathbf{x}_a)}{\psi_a(\mathbf{x}_a)} \\ &+ \sum_{i=1}^{N} (1 - n_i) \sum_{x_i} b_i(x_i) \log \frac{b_i(x_i)}{\phi_i(x_i)} \end{aligned} \tag{6}$$

où n_i est le nombre de potentiels liés à X_i, les quantités $\{\phi_i(x_i)\}_{i,x_i}$ sont les vraisemblances issues d'observations comme la sortie du canal de transmission pour les codes LDPC, et $\{\psi_a(\mathbf{x}_a) = c_a(\mathbf{x}_a) \prod_{X_i \in \mathbf{X}_a} \phi_i(x_i)\}_{a,\mathbf{x}_a}$. La minimisation de l'ELVB est soumise à des contraintes sur les beliefs :

– (C1) normalisation : $\forall a \in \{1, \ldots, M\}$,

$$\sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) - 1 = 0 \tag{7}$$

– (C2) marginalisation : $\forall a \in \{1, \ldots, M\}, \forall X_i \in \mathbf{X}_a, \forall x_i$,

$$\sum_{\mathbf{x}_a \cup x_i} b_a(\mathbf{x}_a) - b_i(x_i) = 0 \tag{8}$$

Ainsi le lagrangien de l'ELVB est $\mathcal{L}_\beta = F_\beta + \mathcal{C}_\beta$ où :

$$\begin{aligned} &\mathcal{C}_\beta(\{b_i(x_i)\}_{i,x_i}, \{b_a(\mathbf{x}_a)\}_{a,\mathbf{x}_a}) \\ &= \sum_{a=1}^{M} \gamma_a \left(\sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) - 1 \right) \\ &+ \sum_{a=1}^{M} \sum_{X_i \in \mathbf{X}_a} \sum_{x_i} \lambda_{ai}(x_i) \left(\sum_{\mathbf{x}_a \cup x_i} b_a(\mathbf{x}_a) - b_i(x_i) \right) \end{aligned} \tag{9}$$

avec $\{\gamma_a\}_a, \{\lambda_{ai}(x_i)\}_{a,i,x_i}$ les multiplicateurs de Lagrange.

3.2 Défaut du Belief Propagation

Les codes LDPC sont très souvent décodés à l'aide de l'algorithme du Belief Propagation (BP) [2]. Il a été souvent montré [3] que les points fixes de cet algorithme sont équivalents aux états stationnaires de l'ELVB. Cependant, la recherche de ces états stationnaires pose problème au sens où l'ELVB est rarement une fonctionnelle convexe des beliefs [9]. Plus précisément, la présence de cycles et de combinaisons de cycles dans les codes LDPC affectent le paysage énergétique de l'algorithme [8] entraînant une minimisation difficile.

Ainsi dans [10] est proposée une approche plus rigoureuse reposant sur le problème de convexité de F_β, la méthode CCCP.

3.3 Algorithme

Il est possible d'écrire le lagrangien de l'ELVB comme la somme d'une fonction convexe F_{vex} et d'une fonction concave F_{cav}. De plus, les contraintes étant linéaires, elles s'incorporent naturellement à l'une ou l'autre fonctionnelle. On obtient :

$$\mathcal{L}_\beta = F_{vex} + F_{cav} \tag{10}$$

avec :

$$\begin{aligned} &F_{vex}(\{b_i(x_i)\}_{i,x_i}, \{b_a(\mathbf{x}_a)\}_{a,\mathbf{x}_a}) \\ &= \sum_{a=1}^{M} \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \log \frac{b_a(\mathbf{x}_a)}{\psi_a(\mathbf{x}_a)} + \sum_{i=1}^{N} \sum_{x_i} b_i(x_i) \log \frac{b_i(x_i)}{\phi_i(x_i)} + \mathcal{C}_\beta \end{aligned} \tag{11}$$

et :

$$\begin{aligned} &F_{cav}(\{b_i(x_i)\}_{i,x_i}, \{b_a(\mathbf{x}_a)\}_{a,\mathbf{x}_a}) \\ &- \sum_{i=1}^{N} n_i \sum_{x_i} b_i(x_i) \log \frac{b_i(x_i)}{\phi_i(x_i)} \end{aligned} \tag{12}$$

Les propriétés de convexité et concavité sont démontrables par les propriétés inhérentes à la divergence de Kullback-Liebler [10] utilisée pour formuler l'ELVB. Le CCCP consiste à calculer itérativement les beliefs tels que $\nabla \mathcal{L}_\beta = \mathbf{0}$, *i.e.* tels que :

$$\begin{aligned} &\nabla F_{vex}(\{b_i^{(k+1)}(x_i)\}_{i,x_i}, \{b_a^{(k+1)}(\mathbf{x}_a)\}_{a,\mathbf{x}_a}) \\ &= -\nabla F_{cav}(\{b_i^{(k)}(x_i)\}_{i,x_i}, \{b_a^{(k)}(\mathbf{x}_a)\}_{a,\mathbf{x}_a}) \end{aligned} \tag{13}$$

En posant pour tout couple (c_a, X_i), pour tout x_i, $\Lambda_{ai}(x_i) = e^{-\lambda_{ai}(x_i)}$ et pour tout potentiel c_a, $\Gamma_a = e^{\gamma_a}$, on obtient les équations suivantes sur les beliefs, dites variables primales :
$\forall a \in \{1, \dots, M\}, \forall \mathbf{x}_a$,

$$b_a^{(k+1)}(\mathbf{x}_a) = \psi_a(\mathbf{x}_a) \frac{\prod_{X_i \in \mathbf{X}_a} \Lambda_{ai}(x_i)}{e^1 \Gamma_a} \tag{14}$$

$\forall i \in \{1, \dots, M\}, \forall x_i$,

$$b_i^{(k+1)}(x_i) = \phi_i(x_i) \left(\frac{b_i^{(k)}(x_i)}{\phi_i(x_i)} \right)^{n_i} \frac{e^{n_i - 1}}{\prod_{\mathbf{X}_a \ni X_i} \Lambda_{ai}(x_i)} \tag{15}$$

En utilisant (C1) et (C2) on obtient les équations itérative pour les multiplicateurs de Lagrange, dites variables duales :

$\forall a \in \{1, \dots, M\}$,

$$\Gamma_a^{(\tau+1)} = e^{-1} \sum_{\mathbf{x}_a} \psi_a(\mathbf{x}_a) \prod_{X_i \in \mathbf{X}_a} \Lambda_{ai}^{(\tau)}(x_i) \tag{16}$$

$\forall a \in \{1, \dots, M\}, \forall X_i \in \mathbf{X}_a, \forall x_i$,

$$\begin{aligned} &(\Lambda_{ai}^{(\tau+1)}(x_i))^2 \\ &= \frac{e^{n_i} \phi_i(x_i) \left(\frac{b_i(x_i)}{\phi_i(x_i)} \right)^{n_i} \Gamma_a^{(\tau)}}{\prod_{\mathbf{X}_b \neq \mathbf{X}_a \ni X_i} \Lambda_{bi}^{(\tau)}(x_i) \sum_{\mathbf{x}_a \cup x_i} \psi_a(\mathbf{x}_a) \prod_{X_j \in \mathbf{X}_a \setminus X_i} \Lambda_{aj}^{(\tau)}(x_j)} \end{aligned} \tag{17}$$

Ainsi on obtient un algorithme à deux boucles, une boucle interne pour calculer les variables duales $\{\Gamma_a\}_a, \{\Lambda_{ai}\}_{a,i}$ afin d'obtenir dans la boucle externe des variables primales $\{b_i\}_i$ qui respectent les contraintes. L'utilisation des codes LDPC permet d'insérer dans la boucle externe un critère d'arrêt supplémentaire. Les vecteurs issus d'un encodeur LDPC, les *mots de code*, représentent un ensemble fini $\mathcal{C}$. Ainsi, vérifier si le mot le plus probable issu des beliefs appartient à cet ensemble définit un critère pour sortir de la boucle externe.

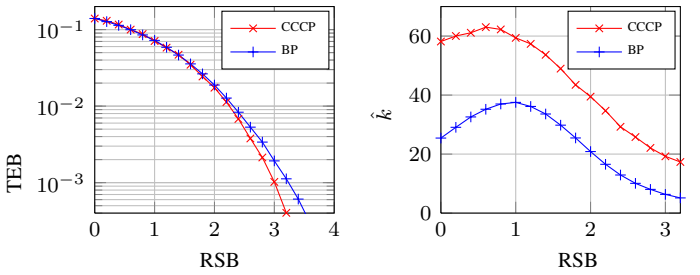
Algorithme 1 : CCCP pour code LDPC

input : Vraisemblances pour un code LDPC de taille N
output : N beliefs + estimé $\hat{\mathbf{x}} = [\hat{x}_1 \dots \hat{x}_N]$

boucle externe;
for $k \leftarrow 1$ **to** K **do**
  initialisation des multiplicateurs $\{\Gamma_a^{(0)}\}_a, \{\Lambda_{ai}^{(0)}\}_{a,i}$;
  boucle interne;
  for $\tau \leftarrow 1$ **to** T **do**
    foreach *facteur* c_a **do**
      $\Gamma_a^{(\tau)} \leftarrow$ équation (16);
    foreach *facteur* c_a, *variable* $X_i \in \boldsymbol{X}_a$ **do**
      $\Lambda_{ai}^{(\tau)} \leftarrow$ équation (17);
    if $\{\Lambda_{ai}^{(\tau)}\}_{a,i} == \{\Lambda_{ai}^{(\tau-1)}\}_{a,i}$ **then**
      convergence i.e. fin de la boucle interne;
  foreach *variable* X_i **do**
    $b_i^{(k)} \leftarrow$ équation (15);
    $\hat{x}_i = \arg\max_{x_i} b_i^{(k)}(x_i)$;
  if $\{b_i^{(k)}\}_i == \{b_i^{(k-1)}\}_i$ **then**
    convergence i.e. fin de la boucle externe;
  else if $\hat{\boldsymbol{x}} \in \mathcal{C}$ **then**
    fin de la boucle externe;

3.4 Résultats

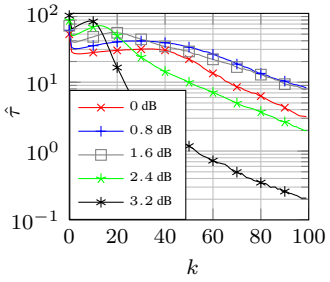
Nous comparons les performances du CCCP avec le BP dans le cas des codes LDPC. Les potentiels sont alors les équations de parité, *i.e.* un potentiel c_a vaut 1 si l'équation de parité correspondante est vérifée, 0 sinon. Nous considérons le code de Tanner de taille $N = 155$, où chaque variable possède trois facteurs voisins, et chaque facteur possède cinq variables voisines. Le canal utilisé est à bruit blanc gaussien additif dont le rapport signal à bruit est noté RSB.



(a) Taux d'Erreur Binaire en fonction du RSB

(b) Nombre d'itérations externes en fonction du RSB

La figure Fig.1(a) montre que le décodage par CCCP est significativement plus efficace que par le BP à partir de 2.0dB environ. Pour un RSB de 3.5dB en effet le taux d'erreur binaire du CCCP est dix fois moindre que celui du BP. On émet alors l'hypothèse, à confirmer dans de futurs travaux, que le décodage par CCCP permet d'amoindrir les dégradations du taux d'erreur binaire telles que l'*error floor* [11] pour de forts RSB.



(c) Nombre d'itérations internes du CCCP en fonction de k

FIGURE 1 – Décodage par BP et CCCP

En revanche le nombre d'itérations de la boucle externe du CCCP est en moyenne deux fois le nombre d'itérations du BP comme nous pouvons le constater Fig.1(b). Le phénomène de résonnance du BP que nous observons aux alentours de 1.0dB est dépendant de conditions sur la dynamique de décodage [8] que nous n'étudions pas dans cet article. Néanmoins, nous constatons que le CCCP présente aussi cette particularité. Une étude sur le comportement dynamique du CCCP analogue à [8] s'avère ainsi nécessaire pour mieux comprendre cet algorithme, ce qui fera l'objet de futurs travaux.

Nous remarquons que chaque itération k de la boucle externe du CCCP demande un nombre d'itérations non négligeables de la boucle interne, Fig.1(c). À titre d'exemple, entre 2.0dB et 3.0dB, pour $\hat{k} \leq 40$, la valeur de $\hat{\tau}$ peut atteindre 50. Pour de tels paramètres, le BP ne demande pas plus de 20 itérations, ce qui est bien plus faible. Il apparaît également un nouveau phénomène de résonnance, dépendant du RSB et de k, qui n'est pas exploité dans la littérature. Une étude sur le comportement dynamique dans cette boucle interne pourrait donner une explication de cette particularité, ce qui constituera de prochaines études. On peut cependant observer pour un RSB$\geq$ 3.2dB que la valeur de $\hat{\tau}$ atteint des valeurs suffisamment faibles pour rendre le CCCP compétitif face au BP. Dans [12] a été proposée une approche calculatoire de la complexité du CCCP en fonction des paramètres du code tels que sa taille et ses degrés. Cependant nos résultats montrent une grande corrélation entre la complexité et les paramètres d'entrées de l'algorithme donc il conviendrait d'effectuer une étude plus large sur la complexité.

4 Conclusion

Les résultats que nous avons obtenus sur le CCCP nous permettent d'affirmer que cet algorithme de décodage est efficace et de complexité raisonnable à forts RSB pouvant surpasser le BP. Le point principal à approfondir reste la complexité dont la valeur apparaît fortement corrélée à la fois aux paramètres du code mais aussi aux RSB, ce qui fera l'objet de futurs publications.

Références

[1] R.G. Gallager. *PhD Thesis : Low-Density Parity-Check Codes*. MIT, 1963.

[2] J. Pearl. *Probablistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[3] J.S. Yedidia, W.T. Freeman et Y. Weiss. *Constructing free energy approximations and Generalized Belief Propagation algorithms*. IEEE Trans. on Inf. Th., Vol.51, pages 2282–2313. 2004.

[4] T. Ott et R. Stoop. *The Neurodynamics of Belief Propagation on Binary Markov Random Fields*. In Proc. of NIPS, 2006.

[5] M. R. Naphade, I. V. Kozintsev et T. S. Huang. *A Factor Graph Framework for Semantic Video Indexing*. IEEE Trans. on Circuits and Syst. for Video Tech., Vol.12. 2002.

[6] M. Mézard et A. Montanari. *Information, Physics and Computation*. Oxford University Press, 2009.

[7] D.J. MacKay et R.M. Neal. *Good codes based on very sparse matrices*. In Cryptography and Coding, 5thIMA Conference, number 1025 in Lecture Notes in Computer Science, pages 100–111. Springer, 1995.

[8] J. C. Sibel, S. Reynal et D. Declercq. *Evidence of chaos in the Belief Propagation for LDPC codes*. To appear in the Chaotic Modeling and Simulation Journal, 2013.

[9] T. Heskes. *Stable fixed points of loopy belief propagation are minima of the Bethe free energy*. Advances in Neural Information Processing Systems 15, pages 359–366. MIT Press, 2003.

[10] A. L. Yuille et A. Rangarajan. *The Concave-Convex Procedure (CCCP)*. Neural Computation, Vol.15, pages 915–936. 2003.

[11] T.J. Richardson. *Error floors of LDPC codes*. In Proc. 41st Annual Allerton Conf. on Communications, Control and Computing. 2003.

[12] T. Shibuya et K. Sakaniwa. *Performance of a Decoding Algorithm for LDPC Codes Based on the Concave-Convex Procedure*. IEICE Trans. Fundamentals, Vol.E86-A. 2003.