

Turbo décodage de codes produits par un algorithme basé sur la propagation de croyance

Christophe JEGO

GET/ENST Bretagne, CNRS TAMCIC UMR 2872,
Brest, France.
christophe.jego@enst-bretagne.fr

Warren J. GROSS

ECE Department, McGill University,
Montreal, Quebec, Canada.
warren.gross@ece.mcgill.ca

Résumé – Dans cet article, nous présentons l'algorithme *modified-Adaptive Belief Propagation (m-ABP)*, une nouvelle technique pour le turbo décodage de codes produits à base de codes en blocs linéaires de type BCH ou Reed-Solomon. L'algorithme original *Adaptive Belief Propagation* de Jiang et Narayanan est simplifié en effectuant l'adaptation de la matrice de parité avant le processus itératif. Des performances en termes de Taux d'Erreur Binaire (TEB) sont données pour ce nouvel algorithme. Les simulations pour le turbo décodage de codes produits BCH montrent des performances similaires à celle obtenues avec l'algorithme Chase-Pyndiah. Par ailleurs, l'algorithme *m-ABP* a de meilleures performances que l'algorithme Chase-Pyndiah en particulier au niveau de la convergence pour le turbo décodage de codes produits de type Reed-Solomon.

Abstract – *This paper introduces the Modified Adaptive Belief Propagation (m-ABP) algorithm, an innovative method for the turbo decoding of product codes based on BCH or Reed-Solomon component codes. The Adaptive Belief Propagation algorithm of Jiang and Narayanan is simplified by moving the parity check matrix adaptation step outside of the iteration loop, significantly reducing the complexity. Performance in terms of the bit-error-rate (BER) of the novel turbo decoding algorithm is given. Simulation results for the turbo decoding of BCH product codes show that compared to the Chase-Pyndiah algorithm no significant BER deviation is observed. Besides, a significant gain in terms of BER performance is obtained with the m-ABP algorithm for the turbo decoding of Reed-Solomon product codes.*

1. Introduction

Les codes correcteurs d'erreurs sont une des solutions permettant d'améliorer la qualité des communications numériques. Le principe du codage canal est d'introduire de la redondance dans la séquence d'information binaire afin de corriger les erreurs de transmission durant la réception de l'information. Deux grandes classes de codes correcteurs d'erreurs existent : les codes convolutifs et les codes en blocs linéaires. Au début des années 90, une nouvelle famille de codes correcteurs d'erreurs a été découverte par C. Berrou [1] : les turbocodes. Cette famille de codes correcteurs d'erreurs est construite par concaténation de codes convolutifs. Les turbocodes convolutifs sont le résultat de deux innovations majeures: la concaténation de deux codes pour le codage et le décodage itératif. Le décodage itératif est appliqué à des décodeurs élémentaires à entrées et à sorties pondérées (EPSP). Le turbo décodage a de bonnes performances et une complexité calculatoire raisonnable. Le concept général de décodage itératif appliqué à des décodeurs élémentaires EPSP a ensuite été étendu aux codes produits [2] et aux codes LDPC [3].

Le décodage itératif basé sur la propagation de croyance (*BP* pour *Belief Propagation* en anglais) [4] est un algorithme de décodage à entrées et sorties pondérées dédié aux codes en blocs linéaires. L'algorithme *BP* est

basé sur la propagation d'information à travers les nœuds d'un graphe bipartite défini à partir de la matrice de parité d'un code en blocs linéaire. Un graphe bipartite est un graphe constitué de nœuds appartenant à deux groupes distincts. Dans le cas des codes en blocs linéaires, le graphe se compose de nœuds variables et de nœuds parités. L'algorithme de décodage *BP* a récemment connu un regain d'intérêt avec la redécouverte des codes LDPC en 1995. Cet algorithme est considéré comme étant l'algorithme de référence pour le décodage des codes en blocs linéaires de type LDPC. Cependant, il est aussi communément admis que l'algorithme *BP* n'est pas approprié si la matrice de parité du code en bloc linéaire n'est pas creuse. Une matrice de parité est dite creuse si elle est composée d'un faible nombre de 1. Or, les codes en blocs linéaires tels que les codes BCH et les codes Reed-Solomon n'ont pas de matrices de parité creuses. Pour cette raison, l'algorithme *BP* n'est pas utilisé pour le décodage des codes BCH et des codes Reed-Solomon.

Cependant, un algorithme *BP* adapté, pour lequel nous utiliserons l'acronyme *ABP* pour *Adaptive Belief Propagation*, a été récemment proposé pour le décodage de codes Reed-Solomon [5] [6]. Le principe consiste à adapter la matrice de parité au cours du processus itératif de l'algorithme *BP* en fonction de l'évolution de la fiabilité des bits d'information. L'adaptation de la matrice de parité revient en fait à diminuer le nombre de 1 des colonnes de la

matrice associées aux bits d'information les moins fiables. L'objectif est de favoriser la propagation de l'information jugée fiable au sein du graphe bipartite. L'algorithme *ABP* est donc particulièrement attrayant pour le décodage des codes en blocs linéaires dont la matrice de parité n'est pas creuse.

Dans ce contexte, nous présentons dans cet article, les performances d'un algorithme qui s'inspire de l'algorithme *ABP* original pour le décodage de codes produits de type BCH ou Reed-Solomon. L'algorithme initial est simplifié en effectuant l'adaptation de la matrice de parité avant l'algorithme *BP* proprement dit et non plus au cours de cet algorithme. Cette simplification permet une diminution significative de la complexité de l'algorithme original en particulier pour des codes produits. De plus, cette simplification permet d'améliorer les performances initiales de décodage en termes de taux d'erreur binaire avec une diminution du nombre d'itérations dans le cas des codes produits. Les résultats de simulation montrent que l'algorithme que nous proposons sous le terme *m-ABP* pour *modified-Adaptive Belief Propagation* a des performances similaires à celles de l'algorithme Chase-Pyndiah [7] pour le turbo décodage de codes produits de type BCH. Un gain en termes de taux d'erreur binaire est quant à lui obtenu avec l'algorithme *m-ABP* pour le turbo décodage de codes produits de type Reed-Solomon.

2. L'algorithme *m-ABP* pour le turbo décodage de code produits

Un code produit est un code performant obtenu à partir de deux codes en blocs linéaires ayant de faibles distances de Hamming minimales. En effet, sa distance de Hamming minimale est alors égale au produit des distances de Hamming minimales des codes élémentaires utilisés et son rendement au produit des rendements élémentaires. Si on considère deux codes en blocs élémentaires $C(n, k)$, le code produit se présente sous la forme d'une matrice à n lignes et à n colonnes. Le turbo décodage d'un tel code consiste à faire un décodage à entrées et sorties pondérées de toutes les lignes puis de toutes les colonnes de la matrice. Le décodeur élémentaire utilisé au cours du processus de turbo décodage est défini par le schéma de la Figure 1. \mathbf{R} correspond à l'information reçue du canal, $\mathbf{R}_{(it)}$ à l'information qui vient de la demi-itération précédente et $\mathbf{R}_{(it+1)}$ à l'information envoyée à la demi-itération suivante. La sortie de chaque demi-itération est égale à $\mathbf{R}_{(it)}$ plus l'information extrinsèque, $\mathbf{W}_{(it)}$ multipliée par $\alpha_{(it)}$. Cette information extrinsèque correspond à l'apport du décodeur. Elle est obtenue par différence entre la sortie pondérée $\mathbf{F}_{(it)}$ du décodeur et l'entrée pondérée de ce même décodeur. Traditionnellement, le décodage élémentaire EPSP est réalisé par l'algorithme Chase-Pyndiah [7]. Nous proposons donc de le remplacer par un nouvel algorithme de décodage : l'algorithme *m-ABP*.

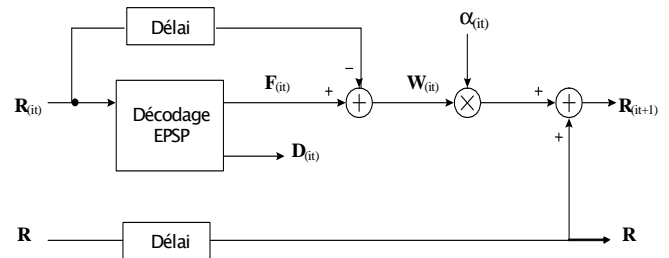


FIG. 1 : schéma du décodeur élémentaire EPSP

Considérons un code en bloc linéaire de longueur n avec k bits d'information. Ce code a une matrice de parité $[\mathbf{H}]$ de $n-k$ lignes et de n colonnes. L'algorithme *m-ABP* s'effectue en deux temps : l'adaptation de la matrice de parité du code en blocs retenu suivie de l'algorithme itératif *BP* traditionnel. Cette approche est motivée par le coût élevé de l'adaptation de la matrice de parité qui nécessite en particulier un pivot de Gauss. L'algorithme proposé est composé de 4 étapes. Tout d'abord, une opération de tri du vecteur d'information $\mathbf{R}_{(it)}$ est effectuée pour déterminer les symboles les moins fiables du vecteur reçu. Il s'agit d'une opération similaire à celle que l'on trouve dans l'algorithme Chase-Pyndiah. Puis, les $n-k$ colonnes de la matrice de parité originale $[\mathbf{H}]$ qui correspondent aux $n-k$ symboles les moins fiables du vecteur reçu $\mathbf{R}_{(it)}$ sont réduites pour obtenir une matrice identité. Pour ce faire, la méthode du pivot de Gauss est appliquée. L'objectif est de diminuer le nombre de 1 dans la partie de la matrice de parité qui est associée aux symboles les moins fiables. Ainsi, la propagation des informations les plus fiables est favorisée à travers les nœuds du graphe bipartite lors du décodage itératif basé sur la propagation de croyance. Il s'agit de l'étape suivante de l'algorithme que nous proposons.

Un algorithme de décodage EPSP standard *BP* est donc ensuite effectué. Nous ne détaillons pas l'algorithme itératif basé sur la propagation de croyance dans cet article. Cependant, le lecteur peut le cas échéant consulter les références [8] [9] qui traitent de cet aspect. Il est à noter que les expérimentations que nous avons menées montrent que le nombre d'itérations nécessaire à l'algorithme *BP* est faible, c'est à dire entre 3 et 5 dans notre approche. Nous utilisons l'expression itération locale pour désigner les itérations du décodage *BP* et ainsi les différencier des itérations du processus de turbo décodage que nous appelons itération globale. La dernière étape de l'algorithme *m-ABP* est le calcul de la décision $\mathbf{D}_{(it)}$. Elle est réalisée uniquement au cours de la dernière itération globale du processus de turbo décodage.

L'innovation majeure de l'algorithme proposée par rapport à l'algorithme *ABP* original pour le turbo décodage de code produit tient dans le fait que l'adaptation de la matrice de parité est faite avant un processus *BP* standard. La matrice de parité adaptée est ainsi identique pour toutes les itérations locales. Cette caractéristique permet une diminution notable de la complexité. Elle favorise également les performances en termes de taux d'erreur

binnaire du turbo décodeur dans le cas de codes produits. Enfin, un coefficient de pondération n'est pas nécessaire alors que c'est le cas dans l'algorithme *ABP* original [6].

3. Comparaison d'algorithmes basés sur la propagation de croyance

Une comparaison des performances de différents algorithmes basés sur la propagation de croyance pour le turbo décodage de codes produits a été effectuée. Pour ce faire, un code produit BCH étendu $(16,11)^2$, un mapping de type *BPSK* et un canal Gaussien ont été choisis. Les algorithmes *BP* standard, *ABP* standard et *m-ABP* proposé ont été successivement considérés. La Figure 2 présente les performances en termes de TEB des différents algorithmes. Cinq itérations locales et six itérations globales sont nécessaires au processus de turbo décodage utilisant l'algorithme *BP* standard. Par contre, le nombre d'itérations locales passe de cinq à dix dans le cas de l'algorithme *ABP* standard pour un même nombre d'itérations globales. Cette augmentation est due à l'insertion de l'étape d'adaptation de la matrice de parité au sein de l'algorithme basé sur la propagation de croyance. Par contre, seules trois itérations locales avec six itérations globales sont nécessaires à l'algorithme *m-ABP* que nous proposons. Ce dernier est donc le moins coûteux en nombre d'itérations. Pour comparaison, les performances obtenues pour un algorithme Chase-Pyndiah traitant seize vecteurs de test dans un processus de six itérations globales sont également tracées sur la Figure 2. De même, la courbe non codée ainsi que la courbe représentant la limite corrigée sont fournies.

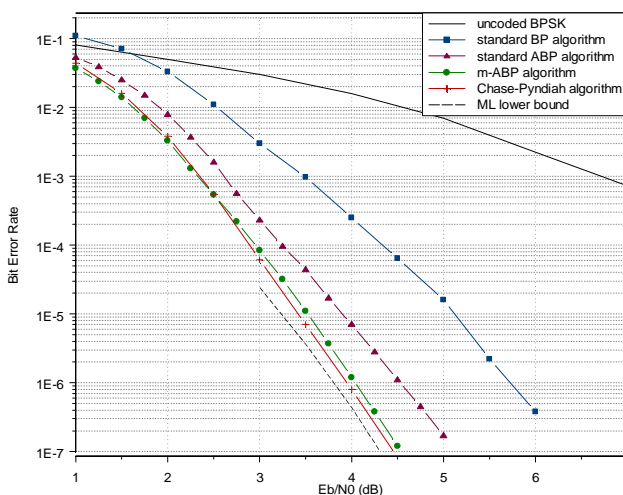


FIG. 2 : TEB pour le turbo décodage du code produit BCH $(16,11)^2$ sur un canal Gaussien

Comme attendu, les performances de l'algorithme *BP* standard sont médiocres lorsqu'il est utilisé pour le turbo décodage de codes produits. L'algorithme *ABP* standard quant à lui, a de meilleures performances que l'algorithme *BP* standard. Cependant, une dégradation de 0,5 dB pour un TEB de 10^{-6} subsiste si ses performances sont

comparées à celles de l'algorithme Chase-Pyndiah. Enfin, l'algorithme *m-ABP* proposé est l'algorithme basé sur la propagation de croyance offrant les meilleures performances pour le turbo décodage de codes produits. En effet, il n'y a pas d'écart notable en termes de TEB si il est comparé à l'algorithme Chase-Pyndiah. Il est à noter que des comparaisons similaires pour d'autres codes produits de type BCH ont abouti aux mêmes conclusions.

Une étude analogue sur le turbo décodage de codes produits constitués de codes en blocs linéaires de type Reed-Solomon a également été effectuée. Les performances en termes de TEB pour le code produit Reed-Solomon $(7,5)^2$ sont tracées sur la Figure 3. Cinq itérations locales ont été choisies pour les algorithmes *BP* et *ABP*. Par contre, trois itérations locales sont suffisantes pour l'algorithme *m-ABP* proposé. Les nombres d'itérations globales nécessaires sont respectivement de huit, dix et six pour les algorithmes *BP*, *ABP* et *m-ABP*. Là encore, l'algorithme *m-ABP* est celui qui nécessite le moins d'itérations locales et globales. De plus, il est le seul à avoir des performances asymptotiques équivalentes à celles obtenues avec l'algorithme Chase-Pyndiah traitant seize vecteurs de test dans un processus de huit itérations globales. Nous pouvons enfin constater que l'algorithme *m-ABP* présente une meilleure convergence que l'algorithme Chase-Pyndiah.

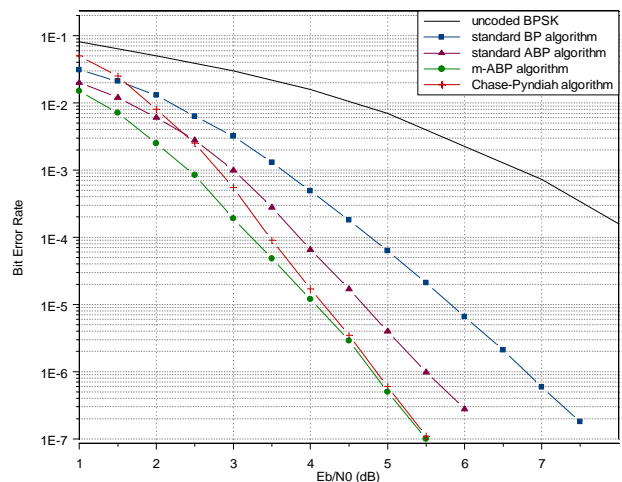


FIG. 3 : TEB pour le turbo décodage du code produit Reed-Solomon $(7,5)^2$ sur un canal Gaussien

4. Décodage itératif de codes produits de type BCH et Reed-Solomon

Dans cette dernière section, nous présentons des performances en termes de taux d'erreur binaire pour le turbo décodage de codes produits ayant des rendements plus élevés. Les performances de l'algorithme *m-ABP* proposé sont comparées à celles de l'algorithme Chase-Pyndiah. Un canal Gaussien et un mapping de type *BPSK* ont été choisis. La Figure 4 présente les résultats pour des codes produits étendus BCH $(64,57)^2$ et BCH $(64,51)^2$ qui ont des pouvoirs de correction de respectivement 1 et 2. Trois itérations locales et huit itérations globales sont

retenues pour l'algorithme m -ABP. De même, huit itérations globales sont réalisées dans le cas de l'algorithme Chase-Pyndiah avec seize vecteurs de test. Les deux algorithmes présentent alors des performances similaires. En effet, l'écart relevé le plus important est de 0.125 dB à un taux d'erreur binaire de 10^{-6} en faveur de l'algorithme Chase-Pyndiah.

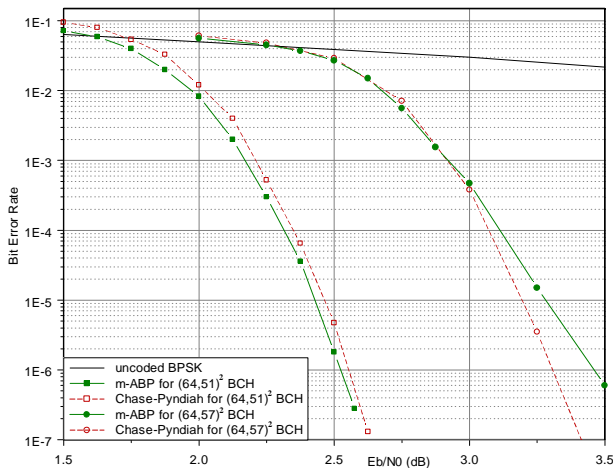


FIG. 4 : TEB pour des codes produits de type BCH sur un canal Gaussien

La Figure 5 présente quant à elle les résultats pour les codes produits Reed-Solomon $(31,29)^2$ et Reed-Solomon $(31,27)^2$ qui ont des pouvoirs de correction de respectivement 1 et 2. Pour les codes produits de type Reed-Solomon, cinq itérations locales et huit itérations globales sont retenues pour l'algorithme m -ABP et huit itérations globales avec seize vecteurs de test pour l'algorithme Chase-Pyndiah. Dans ce cas, l'algorithme m -ABP a de meilleures performances que l'algorithme Chase-Pyndiah en particulier au niveau de la convergence. Ainsi, des écarts de respectivement 0.770 dB et 0.575 dB sont observés à des taux d'erreur binaire de 10^{-3} et de 10^{-6} en faveur de l'algorithme m -ABP pour le turbo décodage du code produit Reed-Solomon $(31,27)^2$

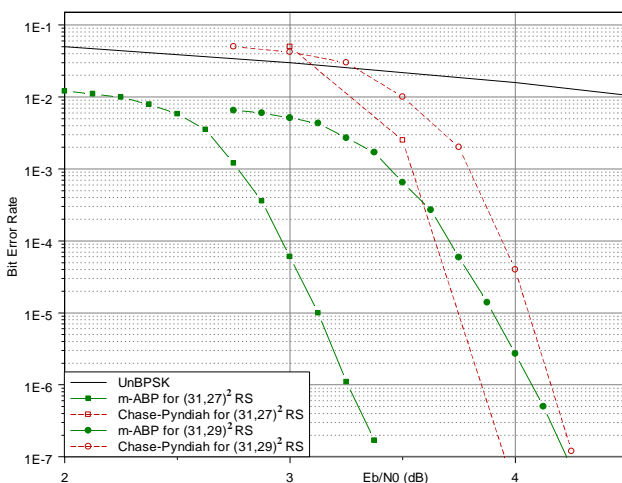


FIG. 5 : TEB pour des codes produits de type Reed-Solomon sur un canal Gaussien

Conclusion

Nous avons proposé un nouvel algorithme m -ABP pour le turbo décodage de codes produits. Cet algorithme reprend le principe de l'algorithme ABP original dédié aux codes en blocs linéaires. L'algorithme m -ABP a une complexité inférieure et requiert moins d'itérations que la version originale dans le cas des turbocodes en blocs. Par ailleurs, les comparaisons en termes de performances effectuées avec l'algorithme Chase-Pyndiah montrent son efficacité pour le turbo décodage de codes produits. Il apparaît donc comme une alternative à l'algorithme Chase-Pyndiah pour répondre aux besoins de turbo décodage à des très hauts débits. En effet, son principe fondé sur la propagation d'information à travers les nœuds d'un graphe bipartite permet un traitement complètement parallèle.

Références

- [1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding : Turbo-codes (1)," *IEEE Int. Conf. on Comm. ICC' 93*, vol 2/3, May 1993, pp. 1064-1071.
- [2] R. Pyndiah, A. Glavieux, A. Picart, S. Jacq , "Near optimum decoding of product codes", *IEEE Int. Conf. GLOBECOM94*, Nov. 1994, pp. 339-343.
- [3] R. G. Gallager, "Low Density Parity Check Codes", *IRE Trans. Inform. Theory*, 1962.
- [4] F. R. Kschischang, B. J. Frey, H.-A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, Feb. 2001, pp. 498-519.
- [5] Jing Jiang, Krishna R. Narayanan, "Iterative soft decoding of Reed-Solomon codes", *IEEE Communication Letters*, vol. 8, April 2004, pp. 244-246.
- [6] J. Jiang, K. R. Narayanan, "Iterative Soft-Input-Soft-Output decoding of Reed-Solomon codes by adapting the parity check Matrix", *IEEE Transaction on Information Theory*, vol. 52, issue 8, August 2006. pp. 3746- 3756.
- [7] R. Pyndiah, "Near optimum decoding of product codes: Block Turbo Codes", *IEEE Trans. On communications*, vol. 46, August 1998, pp. 1003-1010.
- [8] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices", *IEEE Trans. Inf. Theory*, vol. 45, March 1999, pp. 399-431.
- [9] J. Chen, M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes", *IEEE Trans. Communications*, vol. 50, March 2002, pp. 406-414.