

# Construction de Codes LDPC Compatibles en Rendement pour le Codage de Sources avec Information Adjacente

Fangping YE<sup>1</sup>, Elsa DUPRAZ<sup>1</sup>, Zeina MHEICH<sup>2</sup>, Karine AMIS<sup>1</sup>

<sup>1</sup>IMT Atlantique, Lab-STICC, UBL, 29238 Brest, France

<sup>2</sup> University of Surrey , United Kingdom

fangping.ye@imt-atlantique.fr, elsa.dupraz@imt-atlantique.fr  
z.mheich@surrey.ac.uk, karine.amis@imt-atlantique.fr

**Résumé** – Dans cette communication, nous nous intéressons au problème de codage de sources avec information adjacente au décodeur. Nous considérons le cas où le modèle statistique entre la source et l’information adjacente peut varier d’une transmission à l’autre. Pour construire un schéma de codage pratique pour ce problème, on peut utiliser des codes de type Low Density Parity Check (LDPC) compatibles en rendement. Nous proposons ici une nouvelle méthode de construction de codes LDPC compatibles en rendement pour le codage de sources avec information adjacente. Notre construction permet de réduire le nombre de cycles courts dans tous les codes utilisés aux différents débits, et montre ainsi un gain en performance appréciable par rapport aux solutions existantes.

**Abstract** – In this paper, we consider the problem of source coding with side information at the decoder. We focus on the case where the statistical model between the source and the side information can vary from one transmission to another. In order to construct a practical coding scheme for this problem, standard approaches use rate-compatible Low Density Parity Check (LDPC) codes. We propose a novel rate-compatible LDPC code construction for source coding with side information. The proposed construction allows to reduce the amount of short cycles in all the codes that are used at all the considered rates. It shows a great performance improvement compared to standard solutions.

## 1 Introduction

Dans le problème de codage de sources de Slepian-Wolf [1], on souhaite transmettre sans pertes une source  $X$ , sachant que le décodeur a accès à une information adjacente  $Y$ , corrélée avec  $X$ . Ce problème a de nombreuses applications récentes, comme le codage de sources distribué [2], la télévision interactive [3], ou l’accès massif et aléatoire à de grandes bases de données [4]. Par exemple, pour la télévision interactive, la source  $X$  représente la vue courante à transmettre à l’utilisateur, et la source  $Y$  représente la prédiction de cette vue à partir des vues précédemment reçues par l’utilisateur.

Pour construire un schéma de codage efficace pour ce problème, on utilise généralement des codes correcteurs d’erreurs comme les codes de type Low Density Parity Check (LDPC) [5]. Cependant, dans les applications précédentes, le modèle statistique représenté par la distribution conditionnelle  $P(Y|X)$  peut varier d’un utilisateur à l’autre ou d’une transmission à l’autre. Utiliser un code LDPC de longueur fixe risque alors de causer soit une perte de débit, soit une erreur de décodage. Pour éviter cela, on peut utiliser des codes LDPC compatibles en rendement.

En codage de sources, plusieurs méthodes de construction compatibles en rendement ont été proposées. L’approche Rateless [6] consiste partant d’un code de bas débit, et à augmenter progressivement le débit en transmettant des bits d’information

supplémentaires. Le défaut de cette approche est qu’il est difficile de construire un code de bas débit performant, ce qui peut affecter la performance à tous les débits. A l’opposée, l’approche LDPC Accumulate (LDPCA) [7] consiste à effectuer un poinçonnage sur des bits d’information accumulés. Cependant, pour permettre la compatibilité en rendement, l’accumulateur a une structure fixe, et ne peut pas être optimisée pour améliorer la performance du code, par exemple en limitant le nombre de petits cycles dans les codes de plus bas rendement. Une solution intermédiaire est proposée dans [8]. Elle consiste à construire un code de rendement  $1/2$ , puis à utiliser rateless pour les débits supérieurs et LDPCA pour les débits inférieurs à  $1/2$ . Cette méthode permet de palier aux inconvénients de la méthode Rateless, mais les problèmes du LDPCA persistent.

Dans cette communication, nous proposons une nouvelle méthode de construction compatible en rendement pour pallier aux inconvénients du LDPCA. Notre méthode permet un contrôle précis de la structure du code aux différents débits, et en particulier de limiter le nombre de cycles courts, ce qui permet d’améliorer de manière significative la performance des codes.

La méthode que nous décrivons ici a été initialement proposée dans [9]. Les résultats de simulation de [9] comparent uniquement la performance de notre approche avec la méthode LDPCA, pour des débits inférieurs à  $1/2$ . Dans cette communication, nous étudions l’ensemble des débits possibles, en utilisant l’approche rateless pour les débits supérieurs à  $1/2$ .

## 2 Problème de codage de sources avec information adjacente

Dans le problème de codage de sources avec information adjacente [1] décrit en Figure 1 (a), une source  $X$  génère une suite de symboles  $X_1, \dots, X_n$  indépendants et identiquement distribués (i.i.d.) suivant une distribution de probabilité  $P(X)$ . Le décodeur a accès à une information adjacente  $Y$  qui génère une suite de symboles  $Y_1, \dots, Y_n$  i.i.d. suivant une distribution conditionnelle  $P(Y|X)$ . D'après [1], le débit minimum atteignable pour le codage de sources sans pertes de la source  $X$  est donné par l'entropie conditionnelle  $H(X|Y)$ .

Une extension de ce problème a été étudiée dans [10]. Dans le schéma de codage décrit sur la Figure 1 (b), on suppose que le décodeur a accès à une information adjacente  $Y^{(t)}$  qui vient d'un ensemble  $\{Y^{(1)}, \dots, Y^{(T)}\}$ . Dans ce cas, il a été montré dans [4] qu'il était possible de construire une unique description de la source  $X$ , à un débit  $S = \max_t H(X|Y^{(t)})$ , de laquelle on peut extraire  $T$  descriptions différentes. Lorsque l'information adjacente  $Y^{(t)}$  est disponible au décodeur, on extrait une description de débit  $R_t = H(X|Y^{(t)})$ . Dans un schéma pratique, on peut réaliser cela en utilisant des codes LDPC compatibles en rendement.

## 3 Schéma de codage pratique

Dans cette partie, nous décrivons la construction de codes LDPC pour le codage de sources avec information adjacente.

### 3.1 Codes LDPC pour le codage de sources

Nous commençons par décrire le cas où une unique information adjacente est disponible au décodeur. On souhaite transmettre un vecteur binaire  $\mathbf{x}^n$  de longueur  $n$ . On appelle  $H$  la matrice de parité de dimension  $m \times n$  du code LDPC de débit  $R = m/n$ <sup>1</sup>. Le mot de code  $\mathbf{s}^m$  est obtenu par la relation

$$\mathbf{s}^m = H\mathbf{x}^n. \quad (1)$$

Dans ce calcul, toutes les opérations sont effectuées dans GF(2).

Au décodeur, on applique un algorithme de type Somme-Produit pour reconstruire le vecteur  $\mathbf{x}^n$  à partir du mot de code  $\mathbf{s}^m$  et vecteur d'information adjacente  $\mathbf{y}^n$  [5].

### 3.2 Solutions existantes pour la construction de codes compatibles en rendement

Nous supposons maintenant qu'une information adjacente  $Y^{(t)}$  de l'ensemble  $\{Y^{(1)}, \dots, Y^{(T)}\}$  est disponible au décodeur. Dans ce cas, la solution consiste à construire un code LDPC compatibles en rendement.

En codage de canal, plusieurs solutions existent pour construire des codes LDPC compatibles en rendement, comme le poinçonnage [11] ou l'extension de la matrice de parité [12]. En

1. Cette définition est différente du débit d'un code LDPC pour le codage de canal, pour lequel  $R = (n - m)/n$ .

codage de sources, il faudrait appliquer le poinçonnage au mot de code  $\mathbf{s}^m$  qui est transmis au décodeur. Malheureusement, un tel poinçonnage est connu pour détruire très rapidement la structure du code et donne donc des performances très dégradées [7]. La méthode d'extension nécessite d'augmenter à la fois les paramètres  $n$  et  $m$ , ce qui n'est pas possible en codage de sources puisque la dimension  $n$  est déterminée par la taille du vecteur d'information à transmettre.

Pour ces raisons, des méthodes spécifiques au codage de sources ont été développées. Avec la méthode Rateless [6], on commence par construire un code LDPC de bas débit. Pour augmenter le débit du code, on transmet petit à petit des bits supplémentaires de  $\mathbf{x}^n$ . Le problème de cette approche est qu'il est difficile de construire un code LDPC de bas débit, ce qui risque de grandement fragiliser la construction compatible en rendement.

A l'inverse, la méthode LDPCA part d'un code LDPC de haut débit. On construit ensuite un mot de code accumulé  $\mathbf{a}^n$  à partir de  $\mathbf{s}^m$  à partir des relations

$$\begin{aligned} a_1 &= c_1, \\ a_i &= a_{i-1} + s_i, \quad \forall i = \{2, \dots, m\}, \end{aligned} \quad (2)$$

Pour obtenir un débit inférieur à  $m/n$ , on transmet uniquement une partie des valeurs accumulées  $a_i$ . Le problème de l'approche LDPCA est que la structure d'accumulation est fixe, ce qui ne permet pas d'optimiser la structure des codes de plus faible rendement. On souhaiterait par exemple éliminer les cycles courts dans les codes de débit inférieur à  $m/n$ , de manière à améliorer leurs performances de décodage.

Une approche intermédiaire proposée dans [8] consiste à partir d'un code de rendement 1/2, puis à utiliser LDPCA pour les débits plus faibles et Rateless pour les débits plus élevés. Cette approche, que nous employons dans cet article, permet d'éliminer l'inconvénient majeur de la méthode Rateless. Pour pallier aux inconvénients du LDPCA, nous proposons une nouvelle méthode de construction qui permet d'optimiser la structure des codes à tous les débits, et en particulier de réduire le nombre de cycles dans les codes de débit plus faible que 1/2.

## 4 Nouvelle méthode de construction

Nous décrivons maintenant notre nouvelle méthode de construction de codes LDPC compatibles en rendement. Dans cette partie, nous décrivons tout d'abord le passage d'un code de débit  $R_1$  à un code de débit  $R_2$ . La méthode se généralise ensuite à un grand nombre de débits.

### 4.1 Construction compatible en rendement

On considère une matrice mère  $H_1$  de dimension  $m_1 \times n$  et de débit  $R_1 = m_1/n$ , et on souhaite construire une matrice fille  $H_2$  de dimension  $m_2 \times n$ , avec  $m_2 < m_1$  et  $R_2 = m_2/n$ . On note  $\mathbf{s}^{m_1} = H_1\mathbf{x}^n$  le mot de code obtenu à partir de la matrice  $H_1$ , et  $\mathbf{u}^{m_2} = H_2\mathbf{x}^n$  le mot de code obtenu à partir de la matrice  $H_2$ .

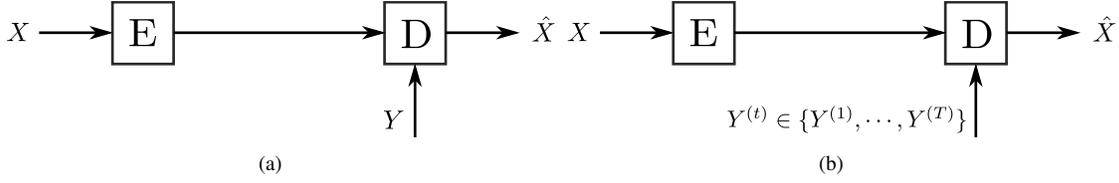


FIGURE 1 – (a) Codage de sources avec information adjacente, (b) Codage de sources avec plusieurs informations adjacentes différentes

Dans la construction que nous proposons, la matrice fille  $H_2$  et la matrice mère  $H_1$  sont liés par une matrice intermédiaire  $H_{1 \rightarrow 2}$  de dimension  $m_2 \times m_1$  telle que

$$H_2 = H_{1 \rightarrow 2} H_1. \quad (3)$$

La construction LDPCA est un cas particulier de cette construction. La matrice intermédiaire  $H_{1 \rightarrow 2}$  doit être construite de manière à permettre une bonne performance de décodage pour  $H_2$ . Elle doit aussi permettre une compatibilité en rendement entre  $H_1$  et  $H_2$ , comme nous le décrivons maintenant.

## 4.2 Condition de compatibilité en rendement

Dans notre construction, on utilise une règle de transmission incrémentale. Pour transmettre un débit  $R_2$ , on envoie le mot de code  $\mathbf{u}^{m_2}$  et le décodage est réalisé avec la matrice  $H_2$ . Pour transmettre un débit  $R_1$ , on envoie le mot de code  $\mathbf{u}^{m_1}$ , et un sous-ensemble  $\mathcal{C}'$  de taille  $m_1 - m_2$  des bits de  $s^{m_1}$ . Pour pouvoir utiliser la matrice  $H_1$  pour le décodage, le récepteur doit pouvoir calculer le syndrome  $\mathbf{c}^{m_1}$  à partir de  $\mathbf{u}^{m_2}$  et de  $\mathcal{C}'$ .

S'il est toujours possible de calculer  $\mathbf{c}^{m_1}$  à partir de  $\mathbf{u}^{m_2}$  et de  $\mathcal{C}'$ , on dit que le code est compatible en rendement. La proposition suivante donne une condition simplifiée pour vérifier que la matrice intermédiaire  $H_{1 \rightarrow 2}$  permet d'obtenir un code compatible en rendement.

**Proposition 1.** *Si la matrice  $H_{1 \rightarrow 2}$  est de rang plein, alors il existe un ensemble  $\mathcal{C}'$  de taille  $m_1 - m_2$  tel que  $(H_1, H_{1 \rightarrow 2}, \mathcal{C}')$  est un code compatible en rendement.*

La preuve est donnée dans [9].

En plus de la condition précédente, il est nécessaire de construire  $H_{1 \rightarrow 2}$  de manière à ce que la matrice  $H_2$  donne de bonnes performances de décodage. Pour cela, nous introduisons maintenant l'algorithme Circle, qui permet de minimiser le nombre de cycles courts présents dans  $H_2$ .

## 4.3 Algorithme Circle

On peut construire la matrice intermédiaire  $H_{1 \rightarrow 2}$  en indiquant les lignes de  $H_1$  que l'on doit combiner pour obtenir la matrice  $H_2$ . Ici, nous allons supposer que les combinaisons se font toujours entre deux lignes de  $H_1$ , et que chaque ligne n'intervient que dans une seule combinaison. Cela revient à considérer que chaque ligne de  $H_{1 \rightarrow 2}$  contient soit une, soit deux composantes non-nulles. Cela permet d'obtenir une matrice  $H_2$  plus creuse, ce qui limitera naturellement le nombre

de cycles courts dans  $H_2$ . De plus, un ensemble  $\mathcal{C}'$  assurant la compatibilité en rendement peut être construit très simplement : pour chaque combinaison de deux lignes, il suffit de transmettre l'une des deux valeurs de  $s^{m_1}$  correspondantes.

Pour passer de  $H_1$  à  $H_2$ , il est donc nécessaire de combiner  $m_1 - m_2$  lignes. On effectue les combinaisons une par une. Quand on souhaite créer une nouvelle combinaison, on sélectionne une première ligne de  $H_1$  au hasard, et  $K$  autres lignes à nouveau au hasard. On cherche alors à combiner la première ligne choisie avec les  $K$  autres lignes, et on choisit la combinaison qui : 1) n'élimine pas de noeud variable dans le code résultant, 2) ajoute le moins de cycles courts au code résultant. La première condition permet de préserver la structure du code, en évitant que certains noeuds variables disparaissent du code. La deuxième condition permet d'améliorer la performance de décodage de  $H_2$ . Pour l'appliquer, on compte en général les cycles de longueur 4 et 6.

## 4.4 Généralisation à plusieurs débits

La méthode précédente se généralise très facilement à n'importe quelle suite de débits  $R_T < R_{T-1} < \dots < R_2 < R_1$ . L'algorithme Circle permet de limiter le nombre de cycles courts dans n'importe lequel des codes construits à ces différents débits. Il permet également d'obtenir facilement les ensembles de bits à transmettre pour passer d'un mot de code d'un certain débit à un autre. On peut même obtenir une granularité de  $1/n$ , c'est à dire que  $R_t - R_{t-1} = 1/n$ , en effectuant une seule combinaison de lignes entre deux débits.

## 5 Résultats de simulation

Nous évaluons maintenant les performances de notre méthode de construction, et la comparons avec LDPCA. Dans toute cette partie, on suppose que la source  $X$  est distribuée uniformément. La source  $X$  est obtenue à partir d'un canal binaire symétrique d'entrée  $X$  et de paramètre  $p$ . On construit tout d'abord deux codes mère  $H_1$  de débits  $R = 1/2$ . Le premier code mère, nommé O1, est de longueur  $n = 512$  et construit à partir du protographe

$$S = \begin{bmatrix} 0 & 2 & 3 & 1 \\ 2 & 0 & 3 & 2 \end{bmatrix}, \quad (4)$$

en appliquant la méthode en deux étapes décrite dans [13] qui permet d'obtenir des codes quasi-cycliques. Le deuxième code-

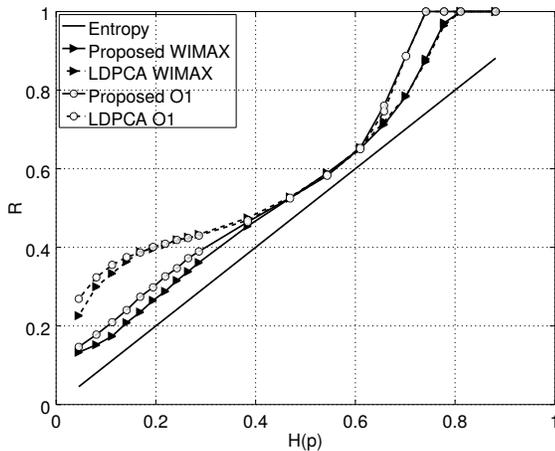


FIGURE 2 – Débits obtenus en fonction de l’entropie, pour deux codes mères : O1 et WIMAX.

mère de longueur  $n = 192$  est obtenu à partir du code WIMAX-Like fourni dans [14]. Pour chacun des codes mères, on construit deux familles de codes compatibles en rendement avec des granularités de  $1/n$ . La première famille de codes est obtenue en appliquant Rateless pour les débits supérieurs  $1/2$  et LDPCA pour les débits inférieurs à  $1/2$ . La deuxième famille est obtenue en utilisant la méthode décrite dans cet article à la place de LDPCA. Lors de l’application de l’algorithme Circle, on utilise  $K = 50$ .

Pour évaluer les performances des différentes familles de code, nous appliquons la méthode suivante, initialement proposée dans [7]. Pour différentes valeurs de  $p$ , on génère 1000 vecteurs  $\mathbf{x}^n$  et  $\mathbf{y}^n$ . Pour chaque couple de vecteurs  $(\mathbf{x}^n, \mathbf{y}^n)$ , on cherche la valeur minimum de débit qui permet de décoder  $\mathbf{x}^n$  sans erreurs. On effectue ensuite la moyenne de ces débits.

Les résultats sont présentés en Figure 2. Pour les débits inférieurs à  $1/2$ , on observe un gain significatif de notre méthode comparé à l’approche LDPCA. On peut également observer la performance de la méthode Rateless pour les débits supérieurs à  $1/2$ . On observe un décrochage de la courbe pour les débits les plus élevés, ce qui suggère que la méthode Rateless pourrait également faire l’objet d’une amélioration.

## 6 Conclusion

Dans cet article, nous avons présenté une nouvelle méthode de construction de codes LDPC compatibles en rendement pour le codage de sources avec information adjacente. La solution proposée permet de réduire le nombre de cycles courts dans les codes utilisés aux différents débits. Nos travaux futurs s’intéresseront à l’amélioration des performances de l’approche Rateless, et au problème de codage de sources avec pertes.

## Références

- [1] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.
- [2] Z. Xiong, A. D. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, 2004.
- [3] M. Tanimoto, “Free-viewpoint television,” in *Image and Geometry Processing for 3-D Cinematography*. Springer, 2010, pp. 53–76.
- [4] E. Dupraz, T. Maugey, A. Roumy, and M. Kieffer, “Rate-storage regions for massive random access,” *arXiv preprint arXiv:1612.07163*, 2016.
- [5] A. D. Liveris, Z. Xiong, and C. N. Georghiades, “Compression of binary sources with side information at the decoder using LDPC codes,” *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, Oct 2002.
- [6] A. W. Eckford and W. Yu, “Rateless Slepian-Wolf codes,” in *Asilomar Conference on Signals, Systems and Computers*, 2005, pp. 1757–1761.
- [7] D. Varodayan, A. Aaron, and B. Girod, “Rate-adaptive codes for distributed source coding,” *Signal Processing*, vol. 86, no. 11, pp. 3123–3130, 2006.
- [8] K. Kasai, T. Tsujimoto, R. Matsumoto, and K. Sakaniwa, “Rate-compatible Slepian-Wolf coding with short non-binary LDPC codes,” in *Data Compression Conference*, 2010, pp. 288–296.
- [9] F. Ye, E. Dupraz, Z. Mheich, and K. Amis, “Optimized rate-adaptive protograph-based ldpc codes for source coding with side information,” *accepted at IEEE Transactions on Communications*, 2019.
- [10] A. Sgarro, “Source coding with side information at several decoders,” *IEEE Transactions on Information Theory*, vol. 23, no. 2, pp. 179–182, 1977.
- [11] J. Ha, J. Kim, and S. W. McLaughlin, “Rate-compatible puncturing of low-density parity-check codes,” *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2824–2836, 2004.
- [12] T. Van Nguyen, A. Nosratinia, and D. Divsalar, “The design of rate-compatible protograph LDPC codes,” *IEEE Transactions on Communications*, vol. 60, no. 10, pp. 2841–2850, 2012.
- [13] D. G. Mitchell, R. Smarandache, and D. J. Costello, “Quasi-cyclic LDPC codes based on pre-lifted protographs,” *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 5856–5874, 2014.
- [14] M. Helmling, S. Scholl, F. Gensheimer, T. Dietz, K. Kraft, S. Ruzika, and N. Wehn, “Database of Channel Codes and ML Simulation Results,” [www.uni-kl.de/channel-codes](http://www.uni-kl.de/channel-codes), 2017.