

Trapping sets bruités pour la sélection des décodeurs multi-niveaux des codes LDPC

Ludovic DANJEAN^{1,2}, David DECLERCQ¹, Shiva K. PLANJERY^{2,1}, Bane VASIĆ²

¹Equipes Traitement de l'Information et Systèmes (ETIS)
ENSEA / Université de Cergy-Pontoise / CNRS UMR 8051
6, avenue du Ponceau, 95000 Cergy-Pontoise, France

²Department of Electrical and Computer Engineering,
University of Arizona, Tucson, AZ 85721, U.S.A.

{danjean, declercq}@ensea.fr, {shivap, vasic}@ece.arizona.edu

Résumé – Les performances de correction d’erreurs des codes LDPC se dégradent dans les zones de fort rapport signal à bruit (RSB) en raison de certaines structures topologiques du graphe de Tanner. Ce phénomène est appelé plancher d’erreur. Dans cet article nous présentons une nouvelle classe d’algorithmes de décodage itératif appelés décodeurs multi-niveaux. Ces décodeurs parviennent à corriger des événements d’erreurs de poids faible responsable du plancher d’erreur avec une faible complexité. En effet, ce type de décodage itératif consiste à travailler sur des alphabets de messages de taille finie, les fonctions de mises à jour étant de simples tables de vérité. Nous proposons également l’extension de la notion de trapping set grâce aux *trapping sets bruités* afin de rendre l’analyse du décodage sur ces structures soit plus proche du décodage réel sur le graphe complet.

Abstract – The error floor is the abrupt degradation of the slope of the error rate performance of LDPC codes that appears in the high signal-to-noise regime due to small topologies in the Tanner graph. In this paper we present a new class of iterative decoding called multilevel decoders. These decoders succeed in correcting low weight error patterns responsible for the appearance of the error floor. Indeed this kind of iterative decoding evolves on finite message alphabet, the update rules are then simple look-up tables. We propose also the extension of the notion of trapping set with the *noisy trapping sets* in order to analyse in a more realistic way the decoding on those structures in the complete Tanner graph.

1 Introduction

Les codes LDPC (Low Density Parity Check) [1] sont au coeur de la recherche des codes correcteurs d’erreurs en raison de leur excellente performance de décodage en utilisant un algorithme de décodage itératif. Un code LDPC est défini par une matrice de vérification de parité de faible densité, ou son graphe associé dit de Tanner [2]. Les codes LDPC tirent leurs excellentes performances du décodage itératif de type "message-passing" comme le décodage par propagation de croyances (Belief Propagation-BP). Ce type de décodage consiste à échanger itérativement des messages entre les noeuds du graphe de Tanner via des règles de mises à jour afin de décoder le mot reçu en provenance du canal. L’algorithme de décodage BP a des performances proche de la limite décrite par C. Shannon asymptotiquement en la taille du code, ce qui induit un graphe de Tanner sans cycles. Pour les codes LDPC de longueur finie, utilisés en pratique dans les systèmes de communications, la présence de cycles dans le graphe de Tanner est inévitable. Ainsi le décodeur BP va échouer sur des vecteurs d’erreurs de poids faible que le décodeur au sens du maximum de vraisemblance corrige, provoquant une dégradation des performances dans la zone de fort rapport signal-à-bruit (RSB), appelée plancher d’erreur [3]. Dans le cas du canal binaire

symétrique (CBS), le concept de *Trapping Set* proposé dans [3] est la meilleure caractérisation [4] du plancher d’erreur. Les trapping sets des codes LDPC sont des structures topologiques du graphe de Tanner composé de plusieurs cycles, tel qu’un décodeur donné peut échouer sur cette structure ou à son voisinage. Récemment un nouveau type d’algorithme de message-passing appelé *décodeur itératif à alphabet fini* ou *décodeur multi-niveaux* a été proposé pour les codes LDPC sur le CBS [5]. Les messages peuvent prendre un nombre fini d’états, et les fonctions de mise à jour des noeuds (variable ou parité) se font par simple lecture de table de vérité et sont choisies afin d’améliorer les performances de correction d’erreurs dans la région du plancher d’erreur.

Dans cet article nous présentons tout d’abord ce nouveau type de décodeur présenté dans [5] avec un exemple de décodeur sur 5 niveaux. Ensuite nous montrons que les outils existant pour l’analyse des trapping sets ne sont pas suffisants afin de refléter les performances du décodeur sur un type de trapping set. En effet les outils existants ne prennent pas en compte le voisinage des trapping sets, c’est pourquoi nous présentons ensuite l’outil développé afin d’analyser au mieux le décodage sur ces trapping sets avec la notion de *trapping set bruité*. Ainsi avec ce nouvel outil, nous pouvons mieux sélectionner un bon décodeur capable d’avoir de meilleures performances de cor-

rection d'erreurs dans la zone de plancher d'erreur que les algorithmes classiques.

2 Les décodeurs multi-niveaux

Dans le cas général d'un code LDPC binaire de taille N , le graphe de Tanner associé contient N noeuds de variable (\bullet), chacun associé à un bit, et M noeuds de vérification de parité (\square). On considère un code LDPC régulier : tous les noeuds de variable ont d_v noeuds de parité voisins, et tous les noeuds de parité ont d_c noeuds de variable voisins. Un trapping set $TS(a, b)$ est un sous-graphe du graphe de Tanner représentant un ensemble de a bits dont le graphe induit possède a noeuds de variables et b noeuds de parité non vérifiés (\blacksquare).

L'algorithme de décodage multi-niveaux est un décodeur itératif de type "message-passing" introduit dans [5] pour lequel les messages qui circulent sur les branches du graphe de Tanner appartiennent à un alphabet \mathcal{M} de taille finie N_s . Dans le cas de N_s niveaux de quantification avec N_s impair $\mathcal{M} = \{0, \pm L_k : 1 \leq k \leq \lfloor \frac{N_s-1}{2} \rfloor\}$ où le signe de L_k représente la valeur du bit (soit un "0" ou un "1"), et l'amplitude $|L_k|$ représente la confiance ou la croyance correspondant à cette valeur. La valeur de l'observation provenant du canal $\{y_i\}_{i=0,1,\dots,N-1}$ appartient à l'alphabet $\mathcal{Y} = \{-C, +C\}$ dans le cas du CBS. Par convention les valeurs positives dans \mathcal{M} représentent un 0 et les valeurs négatives représentent un 1. On utilise le mot de code identiquement nul, ainsi une erreur est donc synonyme d'une valeur négative sur les messages.

Les règles de mises à jour des messages sont également définies sur les noeuds de variable et de parité du graphe de Tanner afin de calculer les messages de sortie de ces noeuds en fonction des messages entrant sur ces noeuds. Soient Φ_v et Φ_c les fonctions qui représentent ces règles de mise à jour sur un noeud de variable et de parité respectivement. Sur un noeud de parité avec $m_1, m_2, \dots, m_{d_c-1}$ les messages entrant dans le noeud, la règle de mise à jour pour notre décodeur multi-niveaux est donné par

$$\Phi_c(m_1, \dots, m_{d_c-1}) = \left(\prod_{j=1}^{d_c-1} \text{sign}(m_j) \right) \min_{j \in \{1, \dots, d_c-1\}} (|m_j|)$$

où sign représente la fonction signe. Cette fonction de mise à jour Φ_c correspond à la fonction de mise à jour sur les noeuds de parité de l'algorithme original de décodage "Min-Sum"[6]. La règle de mise à jour Φ_v sur un noeud de variable dépend des messages entrant $m_1, m_2, \dots, m_{d_v-1}$ et de la valeur du canal y_i . Φ_v peut-être vu comme une fonction linéaire ou non linéaire (voir [5] pour plus de détails), mais il est plus aisé de voir cette fonction grâce à une table de vérité de taille $(N_s)^{d_v-1}$ pour chaque valeur de sortie du canal. Le message de sortie est ainsi donné par $\Phi_v(m_1, \dots, m_{d_v-1}, y_i)$. Pour un code régulier avec $d_v = 3$, et dans le cas du CBS, la fonction Φ_v est limitée à deux tables de vérité bi-dimensionnelles. En effet, la sortie du canal CBS étant binaire, il existe une table de vérité conditionnellement à la première valeur du canal $y_i = -C$, et une deuxième table de vérité conditionnellement à la deuxième valeur du canal $y_i = +C$. Le tableau 1 montre un exemple d'une table

TAB. 1: Exemple de Φ_v^* pour $y_i = -C$

m_1/m_2	$-L_2$	$-L_1$	0	$+L_1$	$+L_2$
$-L_2$	$-L_2$	$-L_2$	$-L_2$	$-L_2$	0
$-L_1$	$-L_2$	$-L_2$	$-L_1$	$-L_1$	$+L_1$
0	$-L_2$	$-L_1$	$-L_1$	0	$+L_1$
$+L_1$	$-L_2$	$-L_1$	0	$+L_1$	$+L_2$
$+L_2$	0	$+L_1$	$+L_1$	$+L_2$	$+L_2$

de vérité Φ_v à $N_s = 5$ états. Le nombre de décodeurs possibles grossissant exponentiellement en la taille de l'alphabet N_s il est nécessaire d'avoir un outil permettant l'analyse de ces décodeurs sur les trapping sets afin de faciliter la sélection d'un bon décodeur sur un code avec une distribution de trapping sets donnée. En effet, pour $N_s = 5$ il existe 28 314 décodeurs possibles, et pour $N_s = 7$ ce nombre croit à 6 392 620.

3 Nombres critiques

Dans [4] le concept de *nombre critique* est présenté afin de caractériser la contribution d'un trapping set donné à la zone de plancher d'erreur. Le nombre critique m d'un trapping set \mathcal{T} est le nombre minimal de bits à être en erreur dans le trapping set conduisant à un échec de décodage. Il est théoriquement possible de sélectionner un bon décodeur basé sur la connaissance des nombres critiques sur l'ensemble des trapping sets d'un code. En pratique, le nombre de trapping sets présents dans un code augmentant exponentiellement avec le nombre de noeuds de variables, il n'est pas possible de les obtenir tous, ainsi donc que leur nombres critiques. Nous devons donc nous limiter aux trapping sets de petites tailles, d'où la nécessité d'avoir un outil d'analyse performant pour la sélection d'un bon décodeur en utilisant uniquement cet ensemble de trapping sets.

Le calcul du nombre critique est réalisé sur le sous-graphe induit par le trapping set en le considérant comme isolé du reste du graphe de Tanner ce qui signifie que les messages entrant dans le trapping set n'influencent d'aucune façon les messages à l'intérieur du trapping set. Le voisinage du trapping set n'est donc pas pris en compte dans ce calcul ce qui explique la différence qu'il peut y avoir entre le nombre d'erreurs qu'un décodeur peut corriger sur un trapping set isolé, et son comportement réel sur le code entier. En effet ce voisinage peut s'avérer néfaste pour le décodage en raison des chemins existant dans le graphe de Tanner autour d'un trapping set qui ont pour conséquence de dégrader les messages qui entrent dans le trapping set en quelques itérations. Ainsi l'hypothèse de graphe isolé n'est, en général, valide que durant quelques itérations de décodage.

Afin de montrer l'influence du voisinage d'un trapping set dans le décodage, le tableau 2(a) présente l'évolution des messages entrant sur le trapping set (5,3) de la figure 1 dans le code de Tanner lors du décodage de 5 erreurs sur ce trapping set avec le décodeur du tableau 1. On constate bien l'influence du voisinage aux itérations 5 et 7 où les messages entrant correspondent à un effacement de l'information. En ajoutant une sixième erreur dans le graphe complet, on constate (tableau 2(b)) qu'il y a même des messages négatifs entrant, ce qui veut dire qu'un message en erreur arrive dans le trapping set.

TAB. 2: Evolution des messages entrant dans un TS(5,3)

(a) Cas 1				(b) Cas 2			
Iter.	$\theta_1^{(l)}$	$\theta_2^{(l)}$	$\theta_3^{(l)}$	Iter.	$\theta_1^{(l)}$	$\theta_2^{(l)}$	$\theta_3^{(l)}$
1	+L ₁	+L ₁	+L ₁	1	+L ₁	+L ₁	+L ₁
2	+L ₂	+L ₂	+L ₂	2	+L ₂	+L ₂	+L ₁
3	+L ₂	+L ₂	+L ₂	3	+L ₂	+L ₁	+L ₂
4	+L ₁	+L ₁	+L ₁	4	+L ₂	-L ₁	+L ₁
5	0	0	0	5	0	+L ₁	0
6	+L ₁	+L ₁	+L ₁	6	+L ₂	0	+L ₂
7	0	0	0	7	+L ₂	+L ₁	+L ₁
8	+L ₁	+L ₁	+L ₁	8	+L ₁	0	+L ₁
9	+L ₁	+L ₁	+L ₁	9	+L ₁	+L ₁	+L ₁
10	+L ₁	+L ₁	+L ₁	10	+L ₁	0	-L ₁

4 Nombres critiques bruités

Afin de remédier à cette insuffisance, nous proposons le calcul du *nombre critique bruité*, c'est-à-dire que nous prenons en compte le voisinage du trapping set en le supposant isolé, mais avec des messages entrant dans le trapping set. On définit ainsi la notion de *trapping set bruité*. Ce bruit est modélisé par un vecteur d'initialisation de messages entrant sur les noeuds de parité non vérifiés du trapping set (■).

Soit N_c le nombre de noeud de parité non vérifiés dans un TS(a,b).

Definition 1. Un vecteur d'initialisation d'un TS(a,b) est défini comme un vecteur $\Theta_{\mathbf{I}}^{(1)} = \{\theta_1^{(1)}, \dots, \theta_{N_c}^{(1)}\}$ où $\theta_i^{(1)} \in \mathcal{M}$ tel que pendant le décodage du TS(a,b) avec un décodeur donné, le message entrant sur le i^e noeud de parité à l'itération l est $\theta_i^{(l)}$.

Definition 2. Un TS(a,b) initialisé par un vecteur d'initialisation $\Theta_{\mathbf{I}}^{(1)}$ est appelé un *trapping set bruité*.

Le but de cette initialisation est de simuler le "retour" de messages bruités dans le trapping set et ainsi de démontrer la robustesse d'un décodeur à ce bruit, c'est à dire la capacité de correction d'erreur avec une influence dans les messages échangés provenant du voisinage du trapping set. Nous simulons donc le voisinage du trapping set, comme la figure 1 le montre sur un TS(5,3) qui, une fois initialisé par le vecteur $\Theta_{\mathbf{I}}^{(1)}$, va simuler une structure plus grande dans laquelle est présent le TS(5,3) comme le TS(8,2) de la figure 2.

Pour savoir si un décodeur est bon dans la zone de plancher d'erreur, nous proposons de vérifier sa capacité de correction d'erreurs sur un ensemble de trapping sets en considérant un ensemble de K_v vecteurs d'initialisation donné Δ . On définit

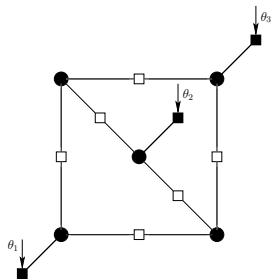


FIG. 1: Initialisation d'un TS (5,3)

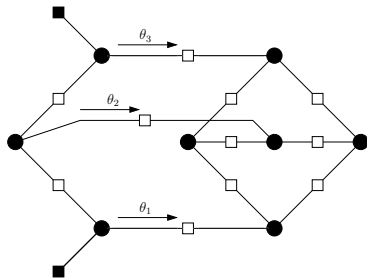


FIG. 2: TS(5,3) + 3 noeuds de variable = TS(8,2)

ainsi la notion de *vecteur des nombres critiques bruités* d'un trapping set $\mathcal{T}(a,b)$.

Soient \mathcal{D}_i un décodeur multi-niveaux sur N_s niveaux et $\Delta = \{\Theta_{\mathbf{I}}(k)\}_{k=1\dots K_v}$ un ensemble de K_v vecteurs d'initialisation statique d'un TS(a,b).

Definition 3. Le vecteur des nombres critiques bruités d'un TS(a,b) décodé avec \mathcal{D}_i est défini comme un vecteur

$$\mathcal{N}_{\mathcal{D}_i}(\mathcal{T}(a,b), \Delta) = \{\tilde{m}_{\mathcal{D}_i,k}\}_{k=1\dots K_v}$$

où $\tilde{m}_{\mathcal{D}_i,k}$ est le nombre d'erreurs minimal que \mathcal{D}_i ne peut pas corriger, lorsque le TS(a,b) est initialisé par $\Theta_{\mathbf{I}}(k)$.

5 Utilisation pour l'analyse du décodage et la sélection des décodeurs

L'utilisation des vecteurs des nombres critiques bruités pour l'analyse des décodeurs multi-niveaux est multiple. Tout d'abord, comme l'avantage de cet outil est que le comportement de décodage est plus proche du comportement réel sur un graphe complet, nous pouvons mieux prévoir le nombre d'erreurs qu'un décodeur multi-niveaux pourra corriger sur un trapping set donné. Le tableau 3 présente les nombres critiques minimum et maximum que le décodeur Φ_v^* peut corriger dans les deux cas extrêmes d'initialisation. Le premier étant lorsque tous les messages entrant sont égaux à $+L_2$. Il s'agit du meilleur cas possible où les messages indiquent qu'un 0 a bien été transmis avec la plus forte croyance (sous réserve du mot de code nul transmis). Ainsi les nombres critiques bruités correspondant sont donc le nombre maximum de bits que ne peut pas corriger Φ_v^* . Le deuxième cas est le cas opposé, à savoir lorsque tous les messages sont à $-L_2$, c'est-à-dire lorsque tous les messages indiquent fortement une erreur. Ainsi les nombres critiques bruités obtenus donnent donc le nombre de bits minimum que ne peut pas corriger Φ_v^* . Comme il s'agit du pire cas, le décodeur ne pourra pas échouer davantage. Par exemple, nous constatons que Φ_v^* va donc corriger entre 3 et 6 erreurs sur le TS(7,5). Ainsi nous allons pouvoir connaître la borne inférieure de correction d'erreur sur les trapping sets d'un décodeur donné ce qui est un premier élément pour la sélection de bon décodeur dans la zone du plancher d'erreur. Un algorithme de sélection basé sur ces nombres critiques bruités est possible afin de réaliser une extraction fine des décodeurs – par exemple 5 décodeurs parmi les 28 314 possibles pour les décodeurs à 5 niveaux en utilisant les $\mathcal{N}_{\mathcal{D}_i}$ pour un ensemble de petits trapping sets.

6 Résultats et Conclusion

Nous présentons sur les figures 3 et 4 les performances de taux d'erreur de l'algorithme BP ainsi que des meilleurs décodeurs multi-niveaux sur 5 et 7 états sur trois codes LDPC de tailles et rendements différents. Nous constatons que les décodeurs extraits ne présentent pas de plancher d'erreur pour des valeurs de taux d'erreurs allant jusqu'à 10^{-6} , et donc qu'il

TAB. 3: Nombres critiques bruités minimum et maximum pour les trapping sets du code de Tanner

θ_{c_1}	...	θ_{c_b}	TS(5,3)	TS(6,4)	TS(7,3)	TS(7,5)	TS(7,5)	TS(8,2)	TS(8,4)	TS(8,4)	TS(8,4)	TS(8,6)	TS(8,6)
$-L_2$...	$-L_2$	4	5	4	4	6	4	4	5	5	4	7
$+L_2$...	$+L_2$	∞	∞	4	7	∞	4	4	5	6	8	∞

surpasse bien le BP dans la zone de plancher d'erreur. La complexité et les temps de simulation des algorithmes multi-niveaux sont bien moindres que le BP. En effet les équations de mises à jour du décodeur BP demandent un accès mémoire conséquent alors qu'il s'agit ici de lectures de tables de vérité. Les décodeurs à 5 et 7 états simulés ont été extraits avec cette approche de nombre critique bruité sur un ensemble de trapping sets donné. Nous présentons également les simulations pour les décodeurs multi-niveaux possédant le meilleur seuil (δ) de la fonction d'évolution de densité discrète. Bien que meilleur dans la zone de "waterfall", ces décodeurs ont des performances moyennes voire catastrophiques dans la zone de plancher d'erreur. Ce paramètre seul n'est donc pas suffisant pour la sélection des décodeurs, il sert néanmoins d'indicateur car les décodeurs extraits ont des seuils très bons.

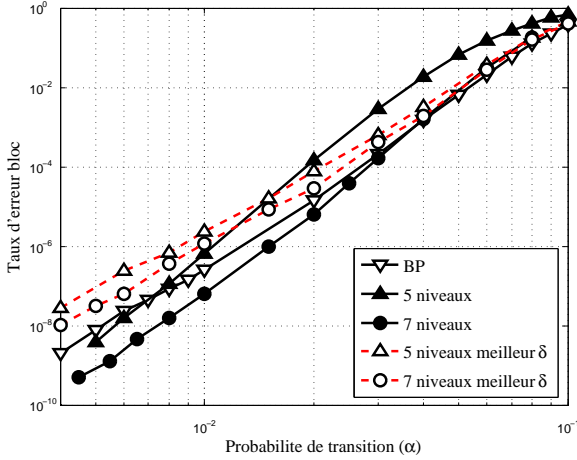


FIG. 3: Code de Tanner (N=155, R=0.41)

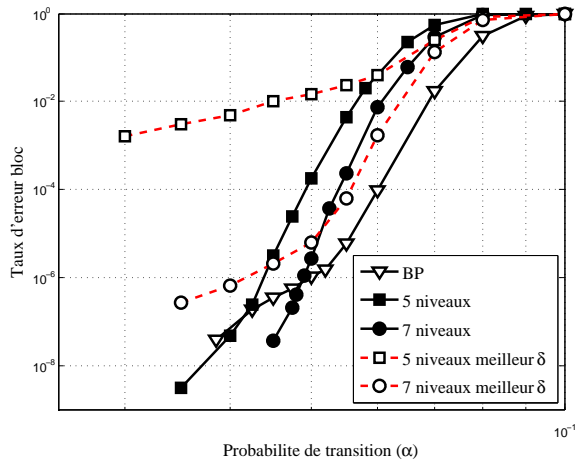


FIG. 4: Code Margulis (N=2640, R=0.5)

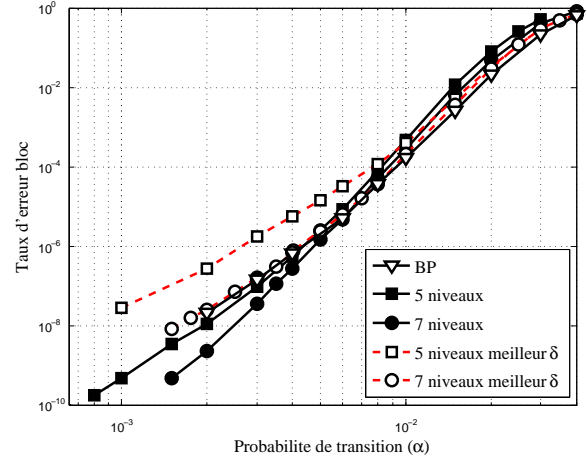


FIG. 5: Code LDPC QC (N=530, R=0.7)

Remerciements

Les auteurs remercient D. V. Nguyen pour avoir fourni le code quasi-cyclique de taille 530. Ce travail est financé par le Projet Nano2012, le GdR-ISIS, et les bourses NSF CCF-0830245 and CCF-0963726.

Références

- [1] D. J. C. MacKay and R. M. Neal. Good codes based on very sparse matrices. In *Cryptography and Coding. 5th IMA Conference*, number 1025 in Lecture Notes in Computer Science, pages 100–111. Springer, 1995.
- [2] R. Tanner. A recursive approach to low complexity codes. *IEEE Trans. on Inf. Theory*, 27(5):533–547, 1981.
- [3] T. Richardson. Error Floors of LDPC Codes. *Proc. 41st Annual Allerton Conf. on Comm. Cont. and Comp.*, 2003.
- [4] S. K. Chilappagari, S. Sankaranarayanan, and B. Vasić. Error floors of LDPC codes on the binary symmetric channel. In *Proc. IEEE Int. Conf. on Comm.*, pages 1089–1094, 2006.
- [5] S. K. Planjery, D. Declercq, S. K. Chilappagari, and B. Vasić. Multilevel decoders surpassing belief propagation on the binary symmetric channel. In *Proc. Int. Symp. on Inf. Theory*, pages 769–773, June 2010.
- [6] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X. Y. Hu. Near optimal reduced-complexity decoding algorithms for LDPC codes. In *IEEE Int. Symp. on Inf. Theory*, 2002.