# Algorithmes Simplifiés pour le Décodage de Codes LDPC non-binaires

Adrian VOICILA[1,3], David DECLERCQ[1], Marc FOSSORIER[2], François VERDIER[1]

[1]ETIS ENSEA/UCP/CNRS UMR-8051, 95014 Cergy-Pontoise, (France),

[2]Dept. Electrical Engineering, Univ. Hawaii at Manoa, Honolulu, HI 96822, (USA),

[3]STMicroelectronics, Crolles, (France)

voicila@ensea.fr, declercq@ensea.fr

marc@aravis.eng.hawaii.edu, verdier@ensea.fr

**Résumé –** Dans cet article, nous présentons un algorithme de décodage simplifié des codes LDPC non-binaires, appelé EMS. Cet algorithme est une simplification de l'algorithme somme-produit dans le domaine logarithmique, qui permet des gains en terme de complexité et de stabilité numérique. Nous proposons également différentes variantes d'implantation de l'algorithme EMS. Les différentes variantes présentées sont comparées en termes de courbes de performances (FER) et complexité.

**Abstract –** In this paper, we introduce a simplified log-domain decoding algorithm for LDPC codes over GF(q), named EMS. While this algorithm is a simplification of the conventional sum-product algorithm, the log-domain decoding has advantages in terms of implementation, computation complexity and numerical stability. Further, we propose some variants of implementation for the EMS algoritm, yielding a lower computational complexity. The different variants are compared both in terms of simulated FER performance and computational complexity.

## 1 Introduction

Pour des tailles modérées des mots de code, une des solutions les plus efficaces pour obtenir de bonnes performances d'erreur est d'utiliser des codes LDPC construits sur des corps de Galois non binaires d'ordre $q$ (GF($q$)) [2]. Cependant, les décodeurs associés ont le désavantage d'avoir une complexité variant en $O(q^2)$ [2, 4]. En conséquence, aucun corps d'ordre plus grand que $q = 16$ ne peut être considéré pour une implantation hardware. Dans cet article nous présentons un algorithme proposé dans [5], qui réduit le nombre d'opérations nécessaires pour la mise à jour des noeuds de parité, tout en conservant des performances proches du décodage par propagation de croyances (BP). L'algorithme proposé est basée sur une généralisation de l'algorithme Min-Sum utilisé pour les décodeurs LDPC binaires, et a été appellé EMS (Extended Min-Sum). Nous étudions en détail dans cet article la complexité algorithmique du décodeur EMS, et présentons des résultats de simulation exhaustifs permettant de quantifier précisément la complexité nécessaire à une mise en œuvre numérique. Ces résultats donnent des indications quant à la complexité requise ainsi que le choix des paramètres du codeur et du décodeur qu'il faut utiliser en fonction des performances désirées. Les résultats de décodage obtenus sont à notre connaissance les meilleurs que l'on puisse obtenir pour des trames courtes et pour une complexité de décodage raisonnable.

## 2 Notations et position du problème

Un code LDPC dans $GF(q)$ est défini par une matrice de parité $H$ creuse de dimension $M \times N$ ($R = \frac{M-N}{N}$ est le rendement) dont les éléments non nuls appartiennent à $GF(q)$. Les multiplications effectuées dans le corps $GF(q)$ seront notées $\otimes$ et tout élément d'un corps GF($q$) peut être décrit en fonction d'un élément primitif du corps $\alpha$. Le corps de Galois est donc composé des éléments $\{0, \alpha^0, \alpha^1, ..., \alpha^{q-2}\}$.

La représentation graphique des codes LDPC non-binaires consiste à connecter l'ensemble des $N$ nœuds de variables aux $M$ nœuds de parité. Dans le cas de codes LDPC réguliers, les nœuds de variables sont connectés à $d_v$ branches et les nœuds de parité à $d_c$ branches. Le décodage par propagation de croyances du code LDPC consite en la mise a jour des messages circulant sur les branches de cette représentation graphique [2, 3]. Les messages circulant sur les branches du trelli représentent les densités de probabilité des symboles du code et sont donc des vecteurs de taille $q$. On va noter les vecteurs des messages rentrant (respectivement sortant) dans un noeud de parité par $U$ (respectivement $V$).

L'algorithme de propagation de croyances consiste en deux étapes de calcul. Premièrement le passage par les nœuds de données est une multiplication termes à termes des $d_v - 1$ messages entrants. Deuxièmement, le passage par les nœuds de parité consite en une marginalisation des $d_c$ messages entrant $\{U_t\}_{1..d_c-1}$ conditionnellement à l'équation de vérification de parité:

$$\sum_{t=1}^{d_c} h_t \otimes c_t = 0 \qquad (1)$$

où $h_t$ sont des éléments non-nuls de la matrice $H$ et $c_t$ les symboles correspondants du mot de code. Cette deuxième étape est très complexe et représente l'obstacle majeur au décodage de code LDPC dans des corps d'ordre élévés. La complexité par nœud de parité est de l'ordre de $O(q^2)$, et peut être ramenée à une complexité moindre $O(q * log(q))$ lorsqu'on effectue le calcul dans le domaine de Fourier [2, 3]. Cependant, sous cette

forme l'algorithme de propagation de croyance requiert des multiplications et des divisions, ce qui est dommageable à toute implantation pratique. En vue d'une implantation hardware des codes LDPC non-binaires, il est donc nécessaire de proposer un décodeur utilisant des log-rapports de vraisemblance comme messages, ne nécessitant que des additions.

# 3 L'algorithme Extended Min-Sum dans le domaine logarithmique (EMS)

## 3.1 Présentation générale de l'algorithme

L'algorithme EMS est une généralisation de l'algorithme min-sum utilisé fréquemment pour le décodage des codes binaires, et fait partie de la vaste classe des algorithme sous-optimaux, dont le but est de simplifier la complexité des calculs réalisés sur l'ensemble des noeuds de parité.

Nous utiliserons par la suite une représentation logarithmique des messages (Log-Density-Ratio: LDR). Un message $U = \{U[k]\}_{k=0..q-1}$ est défini par $q-1$ rapports comme suit :

$$U[k] = \log\left(\frac{prob(c=\alpha^{k-1})}{prob(c=0)}\right)$$

A l'aide des notations de la FIG. 1, les 3 étapes d'une itération de décodage de l'algorithme EMS sont décrites dans la suite de cette section:

– *1. Mise à jour des noeud variable de degré $d_v$:*

$$U_{vp}[k] = L[k] + \sum_{v=1, v\neq c}^{d_v} V_{pv}[k] - offset \quad k = 0 \ldots q-1$$

où L représente les LDR des symboles provenant du canal. La méthode consistant à introduire un facteur de correction (offset) est régulièrement utilisée pour compenser la perte due à l'utilisation d'algorithmes sous-optimaux [5].

– *2. Etape de permutations des messages:*

Cette étape est une conséquence directe de l'équation de parité (1). En effet, l'algorithme de propagation de croyance employé pour la mise à jour des noeuds de parité utilise le produit entre les symboles du mot de code et leur valeur $h_t$ correspondante. Cette multiplication se traduit par une permutation des LDR des symboles.

$$U_{pc}[h_t \otimes \alpha^k] = U_{vp}[\alpha^k] \qquad k = 0 \ldots q-1$$

La transformation du graphe factoriel FIG.1 permet l'écriture du processus de mise à jour de noeud de parité sous la forme d'un produit de convolution, similaire au cas binaire.

La transformation en sens inverse ($V_{cp} \rightarrow V_{pv}$) est faite à l'aide des valeurs $h_t^{-1}$.

– *2. Mise à jour des noeud de parité de degré $d_c$:*

Le principe de l'algorithme EMS est de n'utiliser qu'une partie des valeurs des messages $U_{pc}$, afin de construire une fonction fiabilité sous-optimale. Celle-ci est utilisée pour la mise à jour des noeud de parité, elle doit être très peu coûteuse en nombre d'opérations et dépend de l'équation de vérification de parité. L'algorithme EMS peut être implanté d'une manière efficace via la méthode récursive proposée dans [2] pour la mise à jour

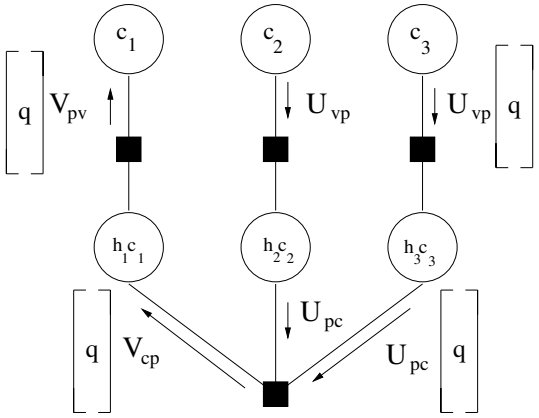


FIG. 1: *Structure en graphe factoriel d'un noeud de parité pour un code LDPC non-binaire*

des noeuds de parité.
Cette méthode s'interprète comme une implantation type "Forward/Backward" du calcul des fiabilités. A chaque étape de récursion, appelée étape élémentaire, on considère seulement deux messages entrants pour effectuer le calcul des fiabilités FIG.2. Notre approche pour présenter les détails de l'étape de mise à jour des noeud de parité est basée sur le principe de la méthode récursive, chaque processus intermédiaire de vérification de parité comportant deux entrées.
Ainsi deux types de messages sont considérés dans la description de l'algorithme. Les messages de type U sont des messages $U_{pc}$, venant du graphe, les messages I sont des messages intermédiaires nécessaires à l'implantation récursive. Nous verrons dans la section suivante que cette distinction de deux types de messages est utile à l'étude de complexité.

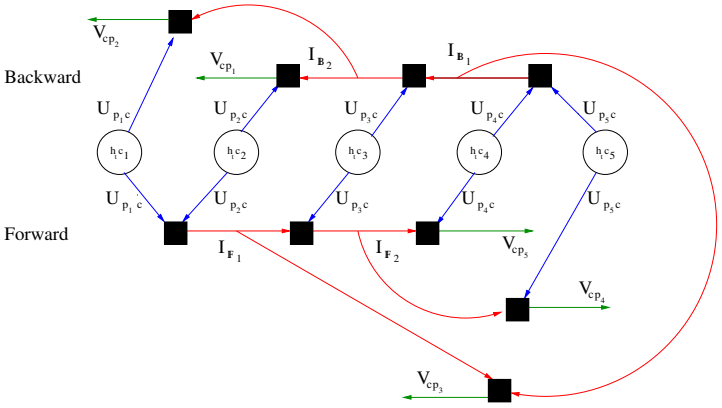


FIG. 2: *La structure récursive de l'algorithme EMS pour un noeud de parité de degrée $d_c = 5$*

Conformément à la représentation en graphe factoriel de la méthode récursive FIG.2, pour des valeurs de $d_c > 4$ on distingue trois types d'étapes élémentaires:

– *Les deux messages entrants sont de type U: étape U-U.*

– *Un des messages entrant est de type U et l'autre de type I: étape U-I.*

– *Les deux messages entrants sont de type I: étape I-I.*

### 3.2 Détails d'une étape élémentaire

Soient $\{X_t\}_{t=1,2}$ les messages entrant dans un nœud de parité élémentaire, c'est-à-dire une des étapes décrites ci dessus (U-U), (U-I) ou (I-I), et $Y_3$ le message sortant. L'algorithme EMS commence par sélectionner dans les messages $X_t$ les $\{r_t\}_{t=1,2}$ plus grandes valeurs que l'on note $x_t^{(k_c)}$, $k_c = 1...r_t$. On note de plus $\alpha_t^{(k_c)}$ les valeurs des éléments de GF($q$) correspondant à ces valeurs maximales.
A l'aide de ces valeurs $\alpha_t^{(k_c)}$ et des contraintes de parité du codes LDPC, on construit l'ensemble de configurations :

$$\begin{aligned} \mathrm{S}(\alpha_3^{(i)}) &= \Big\{ \Big( x_1^{(k_c)}, x_2^{(k_c)} \Big) \\ &\quad : \ \alpha_1^{(k_c)} + \alpha_2^{(k_c)} + \alpha_3^{(i)} = 0 \ \ \forall k_c = 1 \dots r_t \Big\} \end{aligned}$$

A chacune des configurations, on associe une fiabilité définie comme suit:

$$L(\alpha_3^i) = \sum_{t=1}^{2} x_t^{(k_c)} \qquad i = 0 \dots q-1$$

On précise que pour certaines valeurs particulières de $\{r_t\}_{t=1,2}$, il existe des ensembles $S(\alpha_3^{(i)})$ qui peuvent être vides, ce qui implique un problème de convergence pour l'EMS. Une solution à ce problème est l'utilisation de l'ensemble de configurations suivant:

$$\begin{aligned} S_0(\alpha_3^{(i)}) &= \Big\{ \Big( x_1^{(k)}, x_2^{(k)} \Big) \mid \alpha_1^{(0)} + \alpha_2^{(k)} + \alpha_3^{(i)} = 0 \\ &\quad : \ \alpha_1^{(k)} + \alpha_2^{(0)} + \alpha_3^{(i)} = 0 \ \ \forall k = 0 \dots q-1 \Big\} \end{aligned}$$

Cet ensemble n'est jamais vide et garantit la mise à jour de toutes les valeurs de $Y_3[\alpha_3^i], \forall \alpha_3^i = 0 \dots q-1$.
Les valeurs $L(\alpha_3^i)$ expriment la fiabilité, que l'équation de parité soit satisfaite lorsque $h_3 \otimes c_3$ est supposé égal à $\alpha_3^i$. Dans ces conditions, l'équation de mise à jour correspondant à une étape élémentaire est:

$$Y_3[\alpha_3^i] = \max_{S(\alpha_3^{(i)}) \bigcup S(\alpha_0^{(i)})} L(\alpha_3^i) \qquad i = 1 \dots q-1$$

la fonction maximum servant à choisir la configuration la plus fiable.

## 4 Evaluation de la complexité de l'algorithme EMS

Nous évaluons la complexité de l'algorithme EMS en comptant le nombre d'opérations basiques (additions réelles, additions dans GF($q$), comparaisons) par nœud de parité.
Le nombre d'opérations par étape élémentaire est $3r_1r_2$ dans le cas de messages entrants à $r_1$ et $r_2$ valeurs maximales (une addition réelle, une addition dans GF(q) et une comparaison).
Pour la totalité d'un noeud de parité de degré $d_c$, la complexité de la méthode récursive s'exprime comme suite:

$$\mathrm{C}_{rec} = 3 \sum_{i=1}^{3(d_c-2)} \min\left(r_1 r_2, q^2\right) \qquad (2)$$

L'intérêt de l'implantation récursive de l'algorithme EMS repose sur la présence des messages intermédiaires (I), qui au niveau compromis complexité - performances nous offre des degrés de liberté suplimentaires, notamment sur le choix des paramètres $r_1$ et $r_2$.
La complexité de l'algorithme EMS dépend des valeurs prises par $r_1$ et $r_2$ dans les différent (U-U, U-I, I-I) étapes de l'implantation récursive.
Ainsi, le choix suivant:

$$\begin{cases} \text{U-U} & : \ (r_1 = r_2 = q) \\ \text{U-I} & : \ (r_1 = r_2 = q) \\ \text{I-I} & : \ (r_1 = r_2 = q) \end{cases}$$

Correspond à l'algorithme présenté en [4], et également à la complexité la plus grande (infaisable pour des corps d'ordre trop élevés) pour l'algorithme EMS.
Une version moins coûteuse de l'algorithme correspond au choix de paramètres suivants:

$$(\mathcal{A}) \begin{cases} \text{U-U} & : \ (r_1 = r_2 = n_m \ll q) \\ \text{U-I} & : \ (r_1 = n_m r_2 = q) \\ \text{I-I} & : \ (r_1 = r_2 = q) \end{cases}$$

Avec ce choix de paramètres, on se ramène à la même complexité que l'implantation bloc (non-récursive) proposée dans [5]. La complexité globale d'un noeud de parité est dans ce cas approximativement [1]:

$$\mathrm{C} = 3\left\{2n_m^2 + (2d_c - 4)n_m q + (d_c - 4)q^2\right\} \qquad (3)$$

Nous avons fait une étude exhaustive de la comparaison de complexité basé sur l'équation (3).
Pour quantifier la performance des algorithmes de décodage, nous calculons le seuil $\delta$ de convergence du décodeur en utilisant un algorithme d'évolution de densité, outil désormais classique pour l'étude des codes LDPC [1]. Ce seuil correspond au $(E_b/N_0)$ minimal au delà duquel un code de taille infini atteind une probabilité d'erreur nulle. Ainsi, plus le seuil $\delta$ est faible, meilleures sont les performances. Nous utiliserons le code régulier $(d_v = 2, d_c = 4)$ de rendement 0.5 comme base d'étude, et ce pour des ordres de corps de Galois $q = 64, 128, 256$.

La figure FIG.3 trace le logarithme de la complexité par nœud de parité de l'approche récursive (3) en fonction de la valeur du seuil de convergence $\delta$. Très logiquement, la complexité requise augmente de façon très importante lorsqu'on tente de se rapprocher du seuil de convergence optimal, donné par le seuil de l'algorithme BP non-simplifié sur GF(256). Notons également que les codes dans GF(64) sont les plus simples à décoder, mais qu'il sont limités en performances puisque leur seuil minimal se trouve à 0.15dB de la limite (seuil BP). Enfin, ces courbes montrent qu'à performance donnée, un plus gros effort est nécessaire pour passer de GF(128) à GF(256) que pour passer de GF(64) à GF(128). La conclusion est donc que si le code le permet, pour une performance donnée, il est plus avantageux de considérer le plus petit corps possible.

1. On néglige la complexité due à $S_0(\alpha_3^{(i)})$ car l'utilisation de cet ensemble peut être réduite à seulement certaines étapes

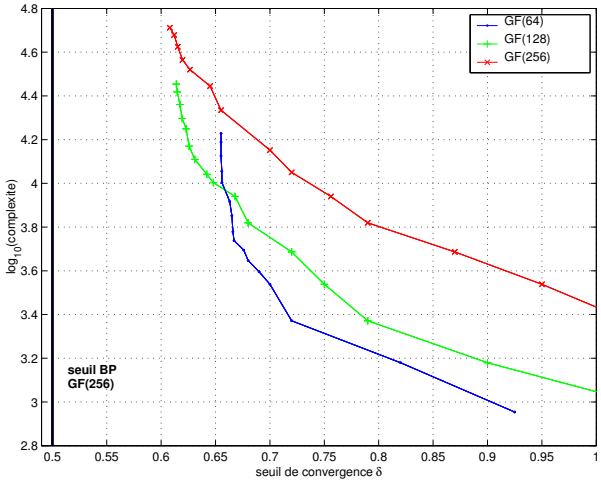


FIG. 3: *Comparaison de complexités de l'algorithme EMS-($\mathcal{A}$) pour différants ordre de corps*

# 5 Algorithme EMS à complexité minimale

Pour un choix de $n_m$ (nombre de maximums) donné, la complexité minimale de l'implantation récursive est obtenue avec les paramètres :

$$(\mathcal{B}) \begin{cases} \text{U-U} & : \quad (r_1 = r_2 = n_m) \\ \text{U-I} & : \quad (r_1 = r_2 = n_m) \\ \text{I-I} & : \quad (r_1 = r_2 = n_m) \end{cases}$$

La complexité de cette approche est d'ordre O($n_m^2$) et la valeur exacte est donnée par:

$$\text{C} = 3 \times 3(d_c - 2)n_m^2 \tag{4}$$

L'un des intérêts de la méthode proposée ci-dessus réside dans la complexité bien moindre que celle donnée par la formule (3). Dans la suite, on présente une comparaison complexité - performances entre deux versions de complexités différentes de l'algorithme EMS. La première version, est basée sur l'hypothèse que chaque message (U ou I) utilisé pour le calcul d'un noeud de parité prend en compte seulement $n_m$ valeurs (choix paramètres $(\mathcal{B})$), la deuxième version de complexité correspond au choix des paramètres $(\mathcal{A})$ .
La FIG.4 présente la perte de performances entre les deux versions de complexité, la moins complexe figure en trait plein et la plus complexe en pointillé, pour un code LDPC non-binaire sur GF(256) $(d_v = 2, d_c = 4)$ comportant K=1504 bits d'information.
L'écart entre les deux algorithmes s'explique par l'utilisation partielle de l'information apportée par les messages U et I. Cet écart est fortement dépendant de l'ordre du corps mais semble insensible à la variation du rapport signal à bruit $\left(\frac{E_b}{N_0}\right)$.
Suite à cette observation, nous nous sommes intéressés à l'influance de $n_m$ sur les performances de décodage de l'algorithme EMS, pour réduire l'écart entre les courbes tout en conservant un important gain de complexité.
Dans le cadre de l'exemple de la figure FIG. 4, il suffit de choisir $n_m = 20$ pour l'algorithme EMS à complexité minimale (EMS-($\mathcal{B}$)) pour récupérer toute la perte de performance. Même avec cette valeur plus grande de $n_m$, le gain de complexité entre l'algorithme EMS-($\mathcal{B}$) et EMS-($\mathcal{A}$) est d'un ordre de grandeur 10.

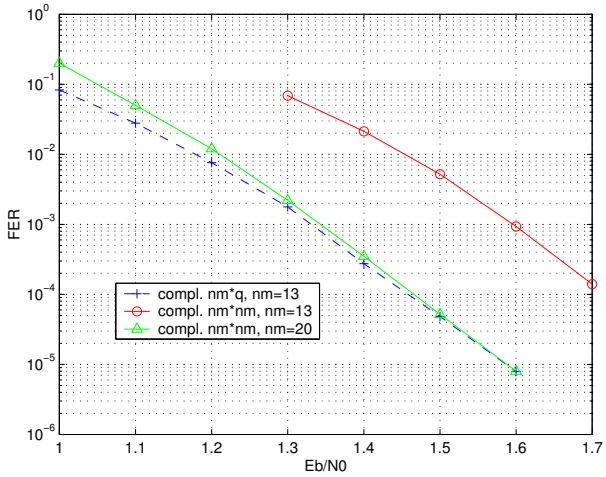


FIG. 4: *Performances comparées des algorithmes EMS-($\mathcal{B}$) et EMS-($\mathcal{A}$) pour différants valeurs de $n_m$*

# 6 Conclusion

Dans cet article, nous avons proposé une implantation récursive de l'algorithme EMS [5] qui nous a permis de développer de nouvelles versions de l'algorithme EMS à complexité réduite (EMS-($\mathcal{B}$)). L'algorithme EMS-($\mathcal{B}$) est un bon candidat pour une éventuelle implantation hardware d'un décodeur LDPC non-binaire en raison de sa complexité inférieure à celle des autres algorithmes existants. De plus la degradation des performances constatée est négligable.

# Références

[1] T.J. Richardson, M.A. Shokrollahi and R.L. Urbanke, "Design of Capacity-Approaching Low-Density Parity Check Codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619-637, Feb. 2001.

[2] M. Davey and D.J.C. MacKay, "Low Density Parity Check Codes over GF($q$)," *IEEE Commun. Lett.*, vol. 2, pp. 165-167, June 1998.

[3] L. Barnault and D. Declercq, "Fast Decoding Algorithm for LDPC over GF($2^q$)," *The Proc. 2003 Inform. Theory Workshop*, Paris, France, pp. 70-73, Mar. 2003,

[4] H. Wymeersch, H. Steendam and M. Moeneclaey, "Log-Domain Decoding of LDPC Codes over GF($q$)," *The Proc. IEEE Intern. Conf. on Commun.*, Paris, France, June 2004, pp. 772-776.

[5] D. Declercq and M. Fossorier, "Extended MinSum Algorithm for Decoding LDPC Codes over GF($q$)", *ISIT'05*, Adelaide, Australia, Sept. 2005.