

# Optimisation de la Consommation d'Énergie pour des Codes LDPC Construits à Partir de Protographes

Mohamed YAOUNI<sup>1</sup>, Elsa DUPRAZ<sup>1</sup>, François LEDUC-PRIMEAU<sup>1,2</sup>, Frederic GUILLOUD<sup>1</sup>

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

<sup>2</sup>Polytechnique Montréal, Canada

mohamed.yaoumi@imt-atlantique.fr, elsa.dupraz@imt-atlantique.fr  
francois.leduc-primeau@imt-atlantique.fr, frederic.guilloud@imt-atlantique.fr

**Résumé** – Dans cet article, nous nous intéressons à l'optimisation de la consommation d'énergie de codes LDPC construits à partir de protographes. Nous considérons que le décodage s'effectue avec un décodeur de type Min-Sum quantifié, et nous introduisons deux modèles différents pour prédire la consommation d'énergie de ce décodeur. Le premier modèle prend en compte la complexité des opérations de calcul, tandis que le deuxième modèle évalue la consommation d'énergie liée aux accès mémoires en écriture. Nous optimisons ensuite les protographes des codes LDPC, de manière à minimiser la consommation d'énergie des décodeurs, tout en maintenant un niveau de performance satisfaisant.

**Abstract** – In this paper, we optimize the energy consumption of LDPC codes constructed from protographs. We consider a quantized Min-Sum decoder, and we introduce two models to predict the decoder energy consumption. The first model takes into account the decoding complexity. The second model evaluates the memory energy consumption required by the decoder. We then optimize the protographs in order to minimize the decoder energy consumption, while maintaining a good level of decoding performance.

## 1 Introduction

La consommation d'énergie des systèmes électroniques est devenue un enjeu crucial, tant pour diminuer l'impact environnemental de ces systèmes, que pour augmenter les durées de vies de leurs batteries. Dans les systèmes de communication, l'approche classique consiste à minimiser l'énergie nécessaire à l'émission de l'information. Cependant, il est montré dans [1] que l'énergie nécessaire à la réception et au décodage de l'information est souvent loin d'être négligeable, en particulier dans les systèmes de transmission à courte distance. Dans ce contexte, cet article s'intéresse à la conception de codes correcteurs d'erreurs à la fois performants et efficaces en énergie. Nous nous intéressons ici aux codes de type Low Density Parity Check (LDPC), qui ont été retenus dans le processus de standardisation de la 5G.

Des études récentes se sont intéressées à ce problème. Les travaux de [1] considèrent des décodeurs à décision dure, et identifient deux types de modèle de consommation d'énergie. Le premier modèle s'intéresse à la complexité des opérations de calcul, tandis que le deuxième modèle considère la consommation d'énergie nécessaire à la transmission de l'information d'un point à un autre du circuit. De plus, [2] considère des décodeurs travaillant sur des alphabets discrets de messages, et cherche à réduire la consommation d'énergie nécessaire au stockage de ces messages. Enfin, [3] propose d'optimiser les distributions de degrés des codes LDPC, à la fois pour obtenir de bonnes performances, et pour minimiser la complexité

du décodage. La notion d'énergie n'est cependant pas étudiée explicitement dans [3].

Dans cet article, nous nous intéressons à des décodeurs LDPC de type Min-Sum quantifié, qui peuvent s'implémenter facilement sur circuit. Nous considérons des codes LDPC construits à partir de protographes. En effet, il est possible d'obtenir facilement des codes de type quasi-cycliques, très pratiques également pour l'implémentation, à partir de protographes [4]. Nous introduisons deux modèles différents pour évaluer la consommation d'énergie d'un décodeur de type Min-Sum quantifié pour un code obtenu à partir d'un protographe donné. Le premier modèle évalue la consommation d'énergie en prenant en compte le nombre d'opérations de calcul nécessaires au décodage. Le deuxième modèle s'intéresse à la consommation d'énergie due à l'écriture en mémoire des messages échangés dans le décodeur. À partir de ces modèles d'énergie, nous proposons une méthode d'optimisation des protographes de manière à minimiser la consommation d'énergie du décodeur, tout en maintenant un niveau de performance de décodage satisfaisant.

## 2 Codes LDPC

Dans cet article, nous supposons que la transmission se fait sur un canal de type Bruit Additif Blanc Gaussien (BABG), de variance  $\sigma^2$ . On appelle  $SNR$  le rapport signal à bruit d'un tel canal. Dans cette partie, nous décrivons la construction de

codes LDPC à partir de protographes, puis nous introduisons le décodeur Min-Sum quantifié. Nous présentons ensuite une méthode d'évaluation des performances du décodeur pour un protographe donné.

## 2.1 Construction de codes LDPC à partir de protographes

Un code LDPC peut être représenté par sa matrice de parité binaire  $H = (h_{c,v})$  de dimension  $(N-K) \times N$  et de rendement  $R = K/N$ . La matrice de parité peut se représenter sous la forme d'un graphe de Tanner qui connecte  $N$  noeuds variables à  $M = N - K$  noeuds checks. Le  $v$ -ième noeud variable et le  $c$ -ième noeud check sont connectés si la composante  $h_{c,v}$  est égale à 1. Dans la suite, nous notons  $\mathcal{N}_v$  l'ensemble des noeuds checks connectés au noeud variable  $v$ , et  $\mathcal{N}_c$  l'ensemble des noeuds variables connectés au noeud check  $c$ . On note aussi  $d_v$  le degré du noeud variable  $v$ , et  $d_c$  le degré du noeud check  $c$ .

Les codes LDPC peuvent être construits à partir de degrés réguliers, c'est à dire que le degré  $d_v$  est le même quelque soit le noeud variable, et le degré  $d_c$  est le même quelque soit le noeud check. On peut également spécifier des distributions de degrés irrégulières [5].

Dans cet article, nous nous intéressons à la construction de matrices de parité à partir de protographes. Un protographe  $S$  de dimension  $m \times n$  peut-être représenté sous la forme d'un graphe de Tanner comportant  $m$  noeuds checks et  $n$  noeuds variables. Pour obtenir une matrice de parité à partir du protographe, on fixe tout d'abord un facteur d'extension  $Z$  tel que  $M = Zn$  et  $N = Zm$ . Le protographe est alors dupliqué  $Z$  fois, puis les connections dans le graphe de Tanner résultant sont permutées en respectant le modèle du protographe original. En pratique, pour obtenir un code de type quasi-cyclique qui permettra une implémentation efficace de l'encodage et du décodage sur un circuit, on effectue plutôt une extension en deux étapes [4], la deuxième étape permettant d'obtenir la structure quasi-cyclique. A chacune des deux étapes, on cherche à minimiser le nombre de cycles courts dans le code résultant, de manière à améliorer les performances de décodage.

## 2.2 Décodeur Min-Sum quantifié

Ici, nous nous intéressons à un décodeur LDPC de type Min-Sum quantifié, qui peut être implémenté facilement sur un circuit [6]. Le décodeur Min-Sum est un algorithme itératif qui consiste à échanger des messages entre les noeuds variables et les noeuds checks. Dans le décodeur quantifié, les messages sont stockés sur  $q$  bits et peuvent prendre des valeurs entre  $-Q$  et  $Q$ , avec un pas de quantification  $s$ . La fonction de quantification est donnée par  $q(x) = \text{sgn}(x) \min \left( s \left\lfloor \frac{|x|}{s} + \frac{1}{2} \right\rfloor, Q \right)$ .

Dans le décodeur, les messages sont initialisés avec le rapport de vraisemblance

$$r_0 = \alpha \log \left( \frac{\Pr(x = 1|y)}{\Pr(x = -1|y)} \right) = \alpha \frac{2y}{\sigma^2} \quad (1)$$

où  $\alpha$  est un paramètre d'échelle. Le nombre total d'itérations du décodeur est noté  $L$ . A l'itération  $\ell$ , on calcule la somme des messages entrants  $m_v^{(\ell)}$  aux noeuds variables  $v$  :

$$m_v^{(\ell)} = q(r_0) + \sum_{c \in \mathcal{N}_v} m_{c \rightarrow v}^{(\ell-1)} \quad (2)$$

Dans ces expressions, les  $m_{c \rightarrow v}^{(\ell-1)}$  représentent les messages des noeuds checks aux noeuds variables. La valeur de  $m_v^{(\ell)}$  est stockée sur  $q + q_s$  bits, où  $q_s = \lceil \log_2(\max(d_v) + 1) \rceil$ , pour éviter des problèmes de saturation des messages.

Ensuite, le message  $m_{c \rightarrow v}^{(\ell)}$  transmis du noeud check  $c$  au noeud variable  $v$  est calculé en deux étapes :

$$m_{v \rightarrow c}^{(\ell)} = m_v^{(\ell)} - m_{c \rightarrow v}^{(\ell-1)} \quad (3)$$

$$m_{c \rightarrow v}^{(\ell)} = \left( \prod_{v' \in \mathcal{N}_c \setminus \{v\}} \text{sign} \left( m_{v' \rightarrow c}^{(\ell-1)} \right) \right) \times \max \left( \min_{v' \in \mathcal{N}_c \setminus \{v\}} |m_{v' \rightarrow c}^{(\ell-1)}| - \lambda, 0 \right). \quad (4)$$

où  $\lambda$  est un paramètre d'offset. Pour implémenter le décodeur, on utilise un séquençement en série de type C [7].

## 3 Évaluation des performances du décodeur à longueur finie

Avant de pouvoir optimiser l'énergie du décodeur, il faut tout d'abord évaluer sa performance de décodage. Cette performance dépend notamment du choix du protographe  $S$  qui permettra d'obtenir la matrice de parité  $H$ . Pour évaluer cette performance, on utilise une méthode d'évolution de densité (ED) [8] adaptée aux protographes, que l'on implémente avec une approche du type multi-edge quantifié. Pour un protographe donné  $S$ , la méthode d'ED permet de prédire le seuil d'un ensemble de codes construits à partir de  $S$ . Pour un canal Gaussien, le seuil est la valeur minimum de SNR qui permet d'assurer une probabilité d'erreur de décodage  $P_e$  égale à 0, lorsque la taille  $N$  du mot de code tend vers l'infini.

Malheureusement, l'ED ne permet pas de prédire la performance d'un code à longueur finie. Il est important dans notre cas de pouvoir faire cette prédiction, puisque la consommation d'énergie du décodeur dépendra de la taille du code. Le décodage d'un code de taille finie se distingue de celui d'un code de taille infinie à cause de deux facteurs, le premier étant l'incertitude liée au canal de transmission, et le second l'effet des cycles dans le graphe de Tanner. Pour prendre en compte l'incertitude liée au canal, nous proposons d'utiliser la méthode suivante, initialement introduite dans [9].

On introduit un paramètre  $p_0 = \frac{1}{2} (1 - \text{erf}(R \times \text{SNR}))$ , directement lié à la variance  $\sigma^2$  du canal BABG. Pour estimer le taux d'erreur binaire  $B_N$  pour une longueur de mot de code  $N$ , on utilise l'équation suivante :

$$B_N(p_0) = \int_0^{\frac{1}{2}} B_\infty(p_{obs}) \phi_{\mathcal{N}} \left( p_{obs}; p_0, \frac{p_0(1-p_0)}{N} \right) dp_{obs}. \quad (5)$$

Dans cette expression,  $\phi_N(x; \mu, \sigma^2)$  correspond à la densité de probabilité d'une variable aléatoire Gaussienne de moyenne  $\mu$  et de variance  $\sigma^2$ , et  $B_\infty(p_{obs})$  représente la probabilité d'erreur du décodeur obtenu par l'ED adaptée au décodeur Min-Sum quantifié.

Ce calcul de taux d'erreur binaire prend en compte l'incertitude liée au canal à travers le paramètre  $p_0$ . En revanche, il ne permet pas de prendre en compte l'effet des cycles présents dans le code. Comme nous le verrons dans la partie simulations, cette méthode évalue malgré tout avec précision la performance à longueur finie du décodeur pour un protographe donné, lorsqu'on considère des codes relativement longs.

## 4 Modèles d'énergie

Dans cet article, nous introduisons deux modèles d'énergie différents : un modèle qui prend en compte la complexité du décodeur, et un modèle qui prend en compte le nombre d'accès en écriture dans la mémoire.

### 4.1 Modèle de complexité

Pour établir ce premier modèle, nous allons évaluer le nombre d'opérations de calcul nécessaires à chaque itération du décodeur. On considère tout d'abord un noeud variable de degré  $d_v$ . Pour mettre à jour le message  $m_v^{(\ell)}$ , on a besoin de deux opérations : une pour l'ajout de  $m_{c \rightarrow v}^{(\ell)}$ , et une pour la soustraction de  $m_{c \rightarrow v}^{(\ell-1)}$ . Pour un noeud check de degré  $d_c$ , il est nécessaire d'effectuer  $2d_c - 1$  opérations de type XOR pour calculer le bit de signe, et  $\lfloor \frac{d_c}{2} \rfloor + 2(\lceil \frac{d_c}{2} \rceil - 1)$  comparaisons pour obtenir les deux valeurs de minimum à partir d'un circuit de type merge-sort [10].

Pour obtenir le modèle d'énergie, on fait le total des opérations de chaque type, multiplié par le coût en énergie correspondant. Pour un protographe donné, le modèle de complexité est donc donné par

$$E_c = LN \left( 2E_{add}(q + q_s)\tilde{d}_v + E_{xor}(2\tilde{d}_c - 1) + \frac{3}{2}E_{comp}(q - 1)(\tilde{d}_c - 1) \right), \quad (6)$$

où  $\tilde{d}_v = \frac{1}{n} \sum_{i=1}^n d_{v_i}$  et  $\tilde{d}_c = \frac{1}{m} \sum_{j=1}^m d_{c_j}$ . Dans cette expression,  $E_{add}$ ,  $E_{xor}$ ,  $E_{comp}$  représentent les coûts en énergie par bit pour une addition, une opération de type XOR, et une comparaison, respectivement.

### 4.2 Modèle d'écriture en mémoire

L'écriture en mémoire consomme également de l'énergie. Pour le prendre en compte, nous établissons maintenant un second modèle d'énergie. Il s'agit dans un premier temps de calculer le nombre de bits à écrire en mémoire à chaque itération. Pour un noeud variable, on doit stocker le message  $m_v^{(\ell)}$  sur  $q + q_s$  bits. Pour un noeud check de degré  $d_c$ , nous aurons besoin de stocker  $d_c$  bits pour les signes des messages, et  $2q - 2$  bits pour stocker les deux minimums.

Finalement, le modèle d'écriture en mémoire est donné par

$$E_m = LE_{bit}N \left( (q + q_s)\tilde{d}_v + (1 - R)(\tilde{d}_c + 2q - 2) \right). \quad (7)$$

Dans cette expression,  $E_{bit}$  représente le coût en énergie pour le stockage d'un bit en mémoire.

## 5 Optimisation

On souhaite maintenant optimiser la consommation d'énergie du décodeur tout en maintenant un certain niveau de performance de décodage. Pour formuler ce problème d'optimisation, nous allons supposer que les paramètres  $L$ ,  $N$ ,  $n$ , et  $R$  sont fixés. Nous allons ensuite chercher à optimiser le protographe  $S$  qui minimise la contrainte d'énergie du décodeur. La contrainte de performance est la suivante : pour un protographe  $S$  donné, on recherche le SNR minimum

$$\text{SNR}_{\min}(S) = \min_{\alpha, \lambda} \text{SNR} \quad \text{tel que} \quad B_N < B_{\max},$$

que l'on peut obtenir en minisant le seuil SNR par rapport aux paramètres  $\alpha$  et  $\lambda$  du décodeur, où  $B_N$  est donné par (5).

Nous allons donc chercher à résoudre le problème d'optimisation suivant

$$\min_S E \quad | \quad \text{SNR}_{\min}(S) < \text{SNR}_c. \quad (8)$$

Dans cette expression, l'énergie  $E$  peut-être donnée soit par le modèle de complexité, soit par le modèle d'écriture en mémoire. Pour résoudre ce problème d'optimisation, on utilise un algorithme génétique appelé évolution différentielle [11], que l'on adapte au problème d'optimisation de protographes. A l'origine, cette méthode avait été introduite pour optimiser des paramètres réels. Pour optimiser des protographes, nous l'appliquons sur des paramètres discrets que sont les coefficients du protographe ayant des valeurs entre 0 et  $S_{\max}$ . Nous ajoutons également la contrainte que  $d_{v_i} > 1$  et  $d_{c_j} > 1, \forall i, j$ . Cette contrainte permet d'éviter les noeuds de degrés 0 ou 1 qui pourraient engendrer des matrices avec de mauvaises performances [12].

## 6 Résultats de simulation

Nous présentons maintenant des résultats d'optimisation et de simulation. Nous avons fixé les paramètres  $n = 4$ ,  $N = 10000$ ,  $R = 1/2$ ,  $S_{\max} = 6$  et appliqué la méthode d'évolution différentielle pour résoudre le problème d'optimisation décrit dans (8) avec les deux modèles d'énergie. Pour les valeurs d'énergie, nous avons fixé  $E_{add} = E_{comp} = 3.13$  fJ,  $E_{xor} = 1.56$  fJ,  $E_{bit} = 0.156$  pJ. Nous présentons les résultats de ces optimisations dans la Table 1. Nous fournissons trois protographes  $S_c$ ,  $S_m$ , et  $S_0$ , ainsi que les valeurs de SNR et d'énergie  $E_c$  et  $E_m$  correspondantes.  $S_c$  correspond au protographe obtenu avec le modèle de complexité, tandis que  $S_m$  provient du modèle d'écriture en mémoire.  $S_0$  a été obtenu en optimisant le seuil du décodeur, sans contrainte d'énergie.

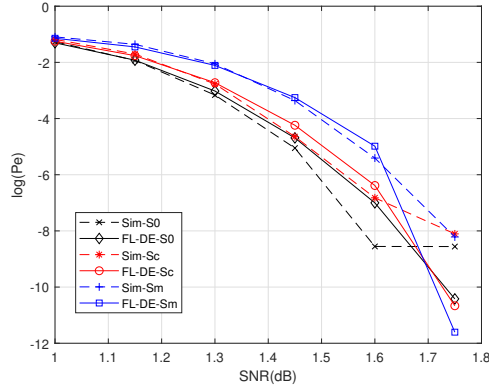


FIGURE 1 – Performance des décodeurs pour trois codes construits à partir des protographe  $S_0$ ,  $S_m$ ,  $S_c$ .

Protographe	SNR	$E_m$	$E_c$
$S_m = \begin{bmatrix} 3 & 2 & 1 & 2 \\ 0 & 1 & 1 & 4 \end{bmatrix}$	1.21 dB	20.1 nJ	523 nJ
$S_c = \begin{bmatrix} 0 & 1 & 2 & 5 \\ 2 & 2 & 0 & 2 \end{bmatrix}$	1.20 dB	19.7 nJ	533 nJ
$S_0 = \begin{bmatrix} 2 & 1 & 3 & 2 \\ 5 & 1 & 1 & 0 \end{bmatrix}$	1.15 dB	33.5 nJ	883 nJ

TABLE 1 – Protographe obtenus par la méthode d’optimisation avec les deux modèles d’énergie.

Chaque protographe minimise bien le critère correspondant à l’optimisation réalisée. On constate en particulier que le protographe  $S_0$  présente les plus grandes valeurs d’énergie, pour les deux modèles. Plus intéressant, il n’y a pas un protographe qui minimise à la fois  $E_c$  et  $E_m$ , ce qui confirme l’utilité de considérer les deux critères.

Pour valider expérimentalement l’approche employée, nous avons construit les codes à partir des trois protographe, à partir de la méthode décrite dans [4]. Nous avons comparé leur taux d’erreur binaire avec le taux d’erreur binaire prédit par la méthode décrite en partie 3. Les résultats sont présentés en Figure 1. Pour les codes  $S_c$  et  $S_m$ , on voit que la méthode décrite en partie 3 prédit de manière précise la performance des décodeurs à longueur finie. En revanche, le code construit à partir du protographe  $S_0$  présente un plancher d’erreur important, qui ne peut être prédit par cette méthode.

## 7 Conclusion

Dans cet article, nous avons proposé une méthode d’optimisation de l’énergie nécessaire au décodage de codes LDPC construits à partir de protographe. Nous avons introduit deux modèles d’énergie : un modèle de complexité, et un modèle d’écriture en mémoire. Nos travaux futurs s’intéresseront à l’évaluation de l’énergie effective du décodeur à partir de mesures sur circuit.

## Références

- [1] K. Ganesan, P. Grover, J. Rabaey, and A. Goldsmith, “Towards approaching total-power-capacity : Transmit and decoding power minimization for ldpc codes,” *arXiv preprint arXiv : 1504.01019*, 2015, 2015.
- [2] T. T. Nguyen-Ly, K. Le, V. Savin, D. Declercq, F. Ghafari, and O. Boncalo, “Non-surjective finite alphabet iterative decoders,” in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.
- [3] W. Yu, M. Ardakani, B. Smith, and F. Kschischang, “Complexity-optimized low-density parity-check codes for gallager decoding algorithm b,” in *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005*. IEEE, 2005, pp. 1488–1492.
- [4] D. G. Mitchell, R. Smarandache, and D. J. Costello, “Quasi-cyclic LDPC codes based on pre-lifted protographe,” *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 5856–5874, 2014.
- [5] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE transactions on information theory*, vol. 47, no. 2, pp. 619–637, 2001.
- [6] C. K. Ngassa, V. Savin, E. Dupraz, and D. Declercq, “Density evolution and functional threshold for the noisy min-sum decoder,” *IEEE Transactions on Communications*, vol. 63, no. 5, pp. 1497–1509, 2015.
- [7] E. Sharon, S. Litsyn, and J. Goldberger, “Efficient serial message-passing schedules for ldpc decoding,” *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4076–4091, 2007.
- [8] T. Richardson, R. Urbanke *et al.*, “Multi-edge type LDPC codes,” in *Workshop honoring Prof. Bob McEliece on his 60th birthday, California Institute of Technology, Pasadena, California*, 2002, pp. 24–25.
- [9] F. Leduc-Primeau and W. J. Gross, “Finite-length quasi-synchronous ldpc decoders,” in *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*. IEEE, 2016, pp. 325–329.
- [10] F. Leduc-Primeau, F. R. Kschischang, and W. J. Gross, “Modeling and energy optimization of ldpc decoder circuits with timing violations,” *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 932–946, 2017.
- [11] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [12] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, “Capacity-approaching protographe codes,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 6, pp. 876–888, 2009.