

RAPPORT COMMUNICATIONS NUMERIQUES AVANCEES

**COMPRÉHENSION DES CODES LDPC
&
IMPLÉMENTATION DES DÉCODEURS HARD ET SOFT**

Groupe 4

BEN LTAIFA Adam

I. Préliminaires :

1.1 À quelle famille de codes le LDPC appartient-il ?

Les codes LDPC font partie de la famille des codes en blocs linéaires. Ce sont des codes correcteurs d'erreurs qui ajoutent de la redondance aux données à envoyer. C'est cette redondance qui permet la correction des erreurs. On parle de "linéaire" car chaque bit du code block est une combinaison linéaire de chaque bit du bloc.

Les codes LDPC sont reconnaissables grâce à leurs matrices de contrôle de parité particulières. Ces codes présentent plusieurs avantages dont celui d'être extrêmement bien adaptés au parallélisme informatique.

De plus, les codes LDPC donnent des performances en décodage qui égalent quasiment celles d'algorithmes complexes.

1.2 A quoi fait référence le nom LDPC :

Le nom "LDPC" (Low Density Parity Check) fait référence aux matrices de parité. En effet, on parle de matrice de parité à faible densité, car dans ces dernières, la proportion de 1 est très faible devant celle des 0. Idéalement, ces matrices ont des tailles très grandes.

Chaque ligne de la matrice de parité représente une équation de parité qui participera au décodage.

1.3 La représentation graphique :

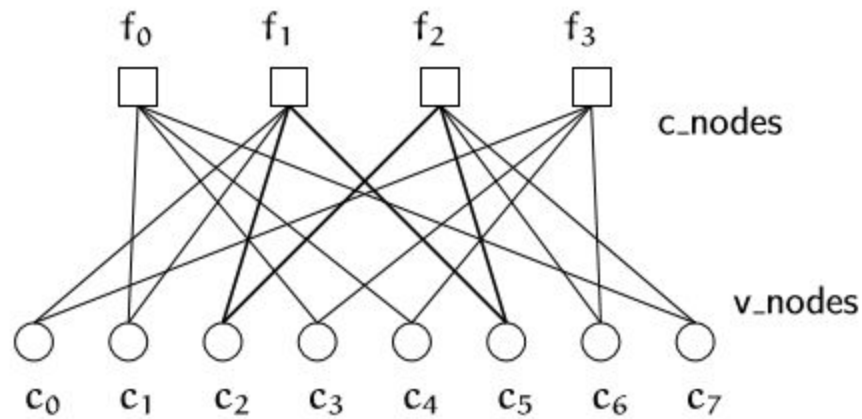
La représentation graphique des codes LDPC, aussi appelée graphe de Tanner, apporte plusieurs niveaux de compréhension et facilite l'assimilation de la méthode de décodage.

Cette représentation prend la forme d'un graphe composé de deux types de noeuds différents. Nous avons d'un côté les noeuds dits variables (v_nodes) et de l'autre les noeuds de vérification (ou noeuds de parité c_nodes).

Il y a autant de noeuds variables que de bits dans le mot-code à décoder, ainsi chaque noeud variable représente un bit du mot code.

Chaque noeud de vérification correspond à une équation de parité ; il y en a autant que d'équations. Autrement dit, le nombre de noeuds de vérification est le nombre de lignes dans la matrice de parité.

Les liens entre les c_nodes et les v_nodes s'obtiennent de manière simple. Chaque fois que nous avons un 1 dans la matrice à la position (i,j) nous pouvons représenter le lien entre le c_node j et de v_node i .



Ainsi, en regardant cette représentation graphique on peut interpréter plusieurs notions:

- Chaque c_node est relié aux v_nodes qui doivent vérifier l'équation de parité.
- Chaque v_node est relié aux c_nodes qui participent à la correction du bit impliqué.
- En regardant le graph, on peut connaître le nombre d'opérations qu'effectuent les c_nodes pour vérifier les équations de parité en comptant le nombre de liens sortant des c_nodes .
- En regardant chaque v_node , on peut connaître le nombre de bits reçus qui participeront à la correction du bit correspondant dans le mot code, en comptant le nombre de liens entrants dans chaque v_node .

II. Études des deux décodeurs :

2.1 La différence entre le hard décodeur et le soft décodeur

La principale différence entre le hard décodeur et le soft réside dans la nature des messages transmis entre les noeuds. En effet, dans le *hard decoder*, les noeuds s'échangent de l'information sous forme binaire afin d'influer directement sur la décision des bits du mot code.

Alors que dans le *soft decoder*, les valeurs échangées entre les noeuds sont des probabilités. Elles représentent un degré de confiance sur la nature du bit en question. Chaque noeud variable envoie aux noeuds de contrôle le degré de confiance que le bit soit un 0 et également le degré de confiance que le bit soit un 1. Les noeuds de contrôle répondent également en degré de confiance (probabilités). C'est la grande différence entre les deux méthodes de décodage. Nous avons également trouvé d'autres différences intéressantes suites à nos recherches. Elles sont synthétisées dans le tableau ci-dessous (Figure 1).

<i>Hard decoder</i>	<i>Soft decoder</i>
<ul style="list-style-type: none"> · Les entrées sont sous forme binaire · Décodeur déterministe · Il a deux niveaux de décision · Un décodeur moins fiable en décision · Implémentation matérielle simple · Capacité de correction d'erreur relativement faible 	<ul style="list-style-type: none"> · Les entrées sont quantifiées en nombres entiers · Décodeur stable · Une meilleure quantification grâce aux différents niveaux de décision (plus de 2 niveaux) · Implémentation matérielle complexe · Forte capacité de correction d'erreurs

Figure 1 : Tableau comparatif des résolutions HARD et SOFT

2.2 En déduire pourquoi un décodeur soft a plus de chance d'être performant qu'un décodeur hard ?

Le *hard decoder* est un décodeur de force brute. En effet, le fait que les messages soient uniquement binaires n'introduit pas la notion de confiance. Ainsi, lors des étapes de *bit flipping* (opération où l'on change la valeur d'un bit suivant ce qu'on pense avoir reçu), nous pouvons modifier un bit correct et ainsi choisir le mauvais lors de l'étape de vote majoritaire. Ce qui est important de retenir, c'est qu'il n'y a aucune priorité entre les bits. Tandis que dans le

soft decoder, les probabilités viennent pondérer ces décisions en mettant un poids plus ou moins important sur ces dernières si les probabilités sont plus ou moins élevées.

Lorsque le *soft decoder* commence, il a en entrée un vecteur de probabilités a priori, qui lui permet déjà d'avoir une idée de la valeur des bits ; au fur et à mesure des itérations, il calcule les probabilités a posteriori et affine ses pondérations ce qui lui permet d'obtenir un résultat beaucoup plus performant que le hard décodeur !

De plus, afin d'optimiser le *soft decoder*, nous pouvons implémenter une solution qui utilise les rapports de vraisemblance logarithmiques (LLR) afin de transformer nos opérations de multiplications en additions qui sont moins coûteuses en termes de complexité algorithmique.

4 Evaluation des décodeurs

Nous pouvons considérer que nous sommes dans un problème de détection. Nous ne connaissons pas le mot-code qui est envoyé mais nous observons un résultat à travers un canal bruité. On cherche alors à établir une règle de décision, qui va permettre de minimiser les erreurs de décision pour chaque élément du mot code.

Dans le cas du *soft decoder*, on utilise pour chaque bit du mot code un détecteur MAP qui minimise l'erreur *a posteriori*.

Afin d'évaluer les performances des *hard decoder* et *soft*, nous pouvons imaginer une modulation QAM avec un mot-code, de taille considérable N, traversant un canal discret sans mémoire (DMC).

Si on souhaite évaluer des décodeurs SOFT entre eux alors on pourrait observer leur courbes ROC et choisir celle qui est la plus optimale. On peut également faire varier le niveau de bruit du canal et observer les taux d'erreur binaires (BER).

BIBLIOGRAPHIE

Introducing Low-Density Parity-Check Codes, Sarah J. Johnson

Hard decision and soft decision decoding algorithms for LDPC and QC-LDPC codes, Er. Sonia, Er. Swati Gupta

LDPC Decoding: VLSI Architectures and Implementations, Ned Varnica

LDPC-in-SSD, Kai Zhao, Hongbin Sun