Université Sidi Mohamed Ben Abdellah

Faculté des Sciences et Techniques - Fes

THÈSE

pour obtenir le grade de docteur délivré par

Centre d'Études Doctorales : Sciences et Techniques de l'Ingénieur

Formation doctorale: "Sciences de l'Ingénieur, Sciences Physiques, Mathématiques et Informatique"

présentée et soutenue publiquement par

Abdelilah KADI

le 16 mai 2017

Etude et amélioration des algorithmes de décodage du code LDPC

Jury

Prof. Driss CHENOUNI,	Directeur de l'ENS Fes	Président
Prof. Mostafa BELLAFKIH,	INPT Rabat	Rapporteur
Prof. Farid ABDI,	FST Fes	Rapporteur
Prof. Adnane ADAIM,	ENSA Kénitra	Rapporteur
Prof. Fatiha MRABTI,	FST Fes	Examinateur
Prof. Said NAJAH,	FST Fes	Co-Directeur de thèse
Prof. Mostafa MRABTI,	ENSA Fes	Directeur de thèse

Table des matières

Acronymes **ix**

Remerciement **xi**

Résumé **xii**

Abstract **xiv**

Introduction Générale **xvii**

Publications **xxiii**

1 Théorie de l'information **1**

1.1 Introduction 1

1.2 Chaîne de transmission numérique 1

1.2.1 Codeur de source 2

1.2.2 Codeur canal 3

1.2.3 Modulation et démodulation 4

1.2.4 Canal de transmission 6

1.2.5 Décodeur de canal 7

1.2.6 Décodeur de source 7

1.3 Modèles des Canaux 8

1.3.1 Le canal binaire symétrique 8

1.3.2 Canal à effacement BEC 8

1.3.3 Le canal à bruit blanc gaussien 9

1.4 Formats de modulation 10

1.5 Théorèmes de Shannon 11

1.6 Codes correcteurs d'erreurs 12
1.6.1 Introduction 12
1.6.2 Notions de base des codes correcteurs 13
1.6.3 Notion du code 15
1.6.4 Les codes en blocs les plus courants 16
1.7 Conclusion 18

2 Les Codes LDPC et algorithmes de décodage itératif 19
2.1 Introduction 19
2.2 Codes LDPC 19
2.2.1 Bref historique 19
2.2.2 Définitions 21
2.2.3 Codes LDPC systématiques 23
2.2.4 Les codes LDPC réguliers 25
2.2.5 Les codes LDPC irréguliers 26
2.2.6 Mesure des performances d'un code 27
2.3 Construction des codes LDPC 28
2.3.1 Construction géométrique des codes LDPC 29
2.3.2 Construction des codes LDPC de Gallager 30
2.3.3 Construction par expansion 32
2.3.4 Construction des codes LDPC quasi-cycliques 32
2.3.5 Construction de type *Repeat Accumlate* 34
2.3.6 Construction déterministe 35
2.4 Codage des codes LDPC 36
2.5 Décodage du code LDPC 39
2.5.1 Décodage MLG des codes LDPC 39
2.5.2 Algorithme de décodage à basculement de bit (BF) des codes LDPC 40
2.5.3 Décodages MLG et BF pondérés 40
2.5.4 Décodage itératif par propagation de croyance 41
2.5.5 Algorithme de décodage *Min Sum* 45
2.6 Conclusion 47

3 Amélioration des performances de l'algorithme *Belief Propagation* 49

3.1 Introduction . . . 49

3.2 Problème des cycles courts . . . 51

3.3 Algorithme pour le calcul des cycles courts . . . 51

3.3.1 Introduction . . . 51

3.3.2 Algorithme de Fan & Xiao . . . 52

3.3.3 Algorithme de Halford & Chugg . . . 56

3.4 Algorithme VFAP-BP . . . 57

3.5 l'algorithme EFAP-BP proposé . . . 62

3.6 Résultats & Simulations . . . 66

3.6.1 Structure du code régulier . . . 66

3.6.2 Convergence du code régulier . . . 67

3.6.3 Performances du code régulier . . . 68

3.6.4 Structure du code irrégulier . . . 70

3.6.5 Convergence du code irrégulier . . . 71

3.6.6 Performances du code irrégulier . . . 71

3.7 Conclusion . . . 74

4 Nouvelle méthode pour améliorer les performances du décodeur MS normalisé en utilisant un code LDPC de la norme WIMAX 75

4.1 Introduction . . . 75

4.2 Détermination du cycle court traversant chaque nœud de contrôle . . . 76

4.3 Algorithme NMS . . . 78

4.3.1 Facteur d'optimisation γ . . . 78

4.4 L'algorithme GA-NMS . . . 80

4.4.1 Structure du code LDPC de la norme WIMAX . . . 81

4.4.2 Structure cyclique du code . . . 82

4.4.3 Facteurs d'optimisation pour l'algorithme GA-NMS . . . 83

4.5 Résultats & Simulations . . . 85

4.5.1 Comportement de convergence . . . 85

4.5.2 Performances de correction d'erreurs . . . 85

4.6 Conclusion . . . 87

Conclusion & Perspectives **88**

Bibliographie **98**

Liste des figures

1.1 Chaîne de communications numériques 2
1.2 Schéma d'un codeur de canal 4
1.3 Canal Binaire Symétrique de probabilité d'erreur P_e 8
1.4 Canal binaire à effacement. 9
1.5 Canal binaire à entrée binaire et bruit blanc additif Gaussien. 10
1.6 Canal binaire 11
1.7 Représentation d'un système correcteur d'erreur. 14

2.1 Graphe factoriel d'un code LDPC régulier ($N = 9, d_v = 2, d_c = 3$) de rendement $R = 1/3$. 23
2.2 Graphes de Tanner de codes LDPC régulier ($d_v = 2; d_c = 4$) et irrégulier ($\lambda(x) = (8/14) \cdot x + (6/14) \cdot x^2$) ; ($\rho(x) = (6/14) \cdot x^2 + (8/14) \cdot x^3$). 26
2.3 Illustration des trois régions caractérisant les performances d'un système de codage. 28
2.4 Exemple de construction de la matrice de parité des codes LDPC de Gallager. 31
2.5 Légende courte pour la figure 33
2.6 Illustrations des différentes représentations d'un code de type *Repeat Accumulate*. Ces codes peuvent être vus comme des codes LDPC (a). Ils peuvent aussi être représentés par une concaténation d'un code de parité, alimenté par un organe de répétition (Rep) dont les sorties sont entrelacées (π), et d'un accumulateur (b). Enfin, ces codes peuvent être vus comme la concaténation série d'un répéteur, d'un accumulateur. Les sorties sont alors poinçonnées (Punct) pour obtenir le rendement codage choisi (c) 34
2.7 Forme particulière de la matrice H_p normalisée dans des standards 38
2.8 Illustration de la mise à jour des messages $\beta_{m,n}$ 43

2.9 Illustration de la mise à jour des messages $\alpha_{m,n}$ 44
2.10 Illustration de l'étape de calcul de l'information a posteriori 45

3.1 Exemple de cycles de longueur 4, 6 et 8 52
3.2 Représentation d'un cycle 4 53
3.3 Les différentes forme d'un cycle 6 dans le code 54
3.4 Un chemin *lollipop* $(2, 4)$ où $\alpha_3 = \alpha_7$ alors que α_1,α_2,...,α_6 sont distinct . . 56
3.5 Le modèle graphique représente l'attribution du facteur ρ au nœuds de parité dans un code LDPC lors du décodage VFAP-BP, $\rho_m(m = 0, 1, ..., M - 1) = \{\rho_v, 1\}$ 58
3.6 Structure de la matrice de parité régulière généré par la méthode PEG avec les paramètres $(N = 504, R = 1/2, d_v = 3, d_c = 6)$ 66
3.7 Valeurs du facteur ρ en fonction du nombre des cycles court traversant chaque nœud de contrôle s_m pour le code PEG régulier. 67
3.8 Comparaison du comportement de convergence pour les algorithmes EFAP-BP, VFAP-BP et BP pour le code LDPC régulier de dimension $(252, 504)$, quand le SNR prend les valeurs SNR=[2,4,6]dB avec $k = 4$. 68
3.9 Comparaison des performances d'erreurs de l'algorithme EFAP-BP, l'algorithme VFAP-BP et BP standard pour le code LDPC régulier $(d_v = 3, d_c = 6)$ de dimension $(252, 504)$ avec le maximum d'itération $I_{max} = (10, 60)$. . . 69
3.10 illustration de la matrice de contrôle de parité PEG_{irreg} $(N = 504, R = 1/2)$, les points bleus correspondent aux positions de la valeur 1 dans la matrice de contrôle de parité. 70
3.11 Valeurs du facteur ρ en fonction du nombre des cycles courts traversant chaque nœud de contrôle s_m pour le code PEG irrégulier. 71
3.12 comparaison du comportement de convergence pour les algorithmes EFAP-BP, VFAP-BP et BP pour le code LDPC irrégulier de dimension $(252, 504)$, quand le SNR prend les valeurs $SNR = [2, 4, 6]$ avec $k = 4$. 72
3.13 Comparaison des performances d'erreurs de l'algorithme EFAP-BP, l'algorithme VFAP-BP et BP standard pour le code LDPC irrégulier de paramètres $(252, 504)$ avec le maximum d'itération $I_{max} = (10, 60)$. 72

4.1 Légende courte pour la figure 77

4.2 Variation du FER en fonction du facteur d'optimisation γ quand le SNR=2.5 dB, en utilisant un code LDPC de la norme WIMAX du standard IEEE P802.16e de longueur $N = 576$ et de rendement $R = 1/2$ 79
4.3 Modèle graphique. 81
4.4 Structure en bloc de la matrice de parité du code LDPC avec un rendement 1/2 dans le standard IEEE P802.16e. 82
4.5 Structure de la matrice H du standard IEEE P802.16e, avec les paramètres $(N = 576, R = 1/2)$ 82
4.6 Les valeurs $\gamma_1, \gamma_2, \gamma_3$ trouvé empiriquement. quand SNR=2.5 dB et $I_{max} = 50$. 83
4.7 Comparaison des comportements de convergence des algorithmes GA-NMS, NMS et MS pour le décodage du code LDPC dans le standard IEEE P802.16e, où le SNR est égale à 3 dB, 4 dB, 5 dB et 6 dB 85
4.8 Comparaison des performances BER&FER entre les algorithmes GA-NMS, NMS et le MS pour le décodage du code LDPC de la norme WIMAX, avec 50 itérations max 86

Liste des tableaux

4.1 Structure cyclique de la matrice H 83

Acronymes

APP A Posteriori Probability. 39

AWGN Additive White Gaussien Noise. 9

BBAG Bruit Blanc Additif Gaussien. 2

BCH Bose Chaudhuri et Hocquenghem. 13

BER Bit Error Rate. 15, 27

BF Bit Flipping. 39

BP Belief Propagation. 39

BPSK Binary Phase Shift Keying. 10

CAE Canal À Effacement. 8

CBS Canal Binaire Symétrique. 8

CG Corps de Galois. 31

DVB-S2 Digital Video Broadcasting-Satellite Second Generation. 35

EFAP-BP Exponential Factor Appearance Probability Belief Propagation. 50

EG-LDPC Euclidean Geometry-Low Density Parity Check. 30

EXIT Extrinsic Information Transfert (chart). 28

FER Frame Error Rate. 15, 27

GA-NMS Girth Aware Normalized Min Sum. 76

IP Internet Protocol. 8

JPEG Joint Photographic Experts Group. 3

MLG Majority Logic decoding. 39

MPEG Moving Picture Experts Group. 3

MS Min Sum. 20

NMS Normalized Min Sum. 75

OMS Offset Min Sum. 75

PEG Progressive Edge Growing. 29

PG-LDPC Projective Geometry-Low Density Parity Check. 30

QAM Quadrature amplitude. 10

QC Quasi-Cyclic. 38

QPSK Quadrature phase-shift keying. 10

RS Reed Solomon. 17

SNR Signal Noise Ratio. 5

SP Sum Product. 20

SSD Solid-State Drive. 17

TEB Taux d'erreurs binaire. 3, 27

TET Taux d'Erreur Trame. 27

URW-BP Uniformly Reweighted Belief Propagation. 50

VFAP-BP Variable Factor Appearance Probability Belief Propagation. 50

Remerciement

Je tiens ici à exprimer toute ma reconnaissance aux personnes qui m'ont aidé, encouragé et soutenu pendant ces années de préparation de thèse.

Premièrement, j'aimerais remercier mon directeur de thèse, Monsieur Mostafa MRABTI, pour sa confiance son soutien et sa vision plus haut niveau du sujet qui m'a permis d'avancer dans mon travail de thèse, je remercie également mon co-directeur de thèse Monsieur Said NAJAH pour son accompagnement, sa disponibilité, sa motivation, ses commentaires constructifs sans lesquels ce travail n'aurait pu voir le jour.

J'exprime ma gratitude envers Pr. Driss CHENOUNI qui m'a fait l'honneur d'accepter de présider le jury de cette thèse. Je remercie également Pr. Mostafa BELLAFKIH, Pr. Farid ABDI et Pr. Adnane ADDAIM d'avoir accepté la lourde tâche de rapporter d'une manière détaillée ce mémoire. Je les remercie pour l'intérêt qu'ils ont porté à ce manuscrit ainsi que pour leurs remarques constructives et encourageantes. Je remercie également Pr. Fatiha MRABTI pour avoir examiné ce travail.

Ensuite je voudrais remercier Professeur EL Houssein ABARKAN et professeur Mhammed LAHBABI directeur du laboratoire SSC à la Faculté des Sciences et Techniques – Fès. Je les remercie de m'avoir accueilli au sein du laboratoire SSC.

Je tiens aussi à remercier Monsieur David DECLERCQ Professeur des Universités ENSEA France pour le séjour de recherche qu'il m'a accordé au sein du laboratoire ETIS France, je remercie également Monsieur Emmanuel BOUTILLON professeur de l'université de Bretagne SUD Lorient France pour le stage qu'il m'a donné au sein du laboratoire STICC et qui m'a aider à avancer dans ma thèse et acquérir de nouvelles compétences.

Je salue mon ami Rachid el ALAMI Professeur à la faculté des sciences Dhar el Mahraz avec qui j'ai eu des échanges et des discussions très riches.

Je remercie mes parents pour leur accompagnement et leur soutien moral et financière, je remercie également mes frères, sœurs, mon oncle et toute la famille KADI pour leur

encouragement et leur soutien qui m'ont permis d'arriver jusque-là.

Finalement, je remercie tous ceux qui ont contribué dans la réalisation de ce travail dont j'ai oublié de citer leurs noms.

Résumé

En cette ère numérique, les systèmes de communication modernes jouent un rôle essentiel dans presque tous les aspects de la vie, avec des exemples allant de réseaux mobiles et les communications par satellite à l'Internet et le transfert de données. Malheureusement, tous les systèmes de communication dans le cadre pratique sont bruyants, ce qui indique que nous pouvons soit améliorer les caractéristiques physiques du canal ou trouver une éventuelle solution systématique, à savoir les codes correcteurs d'erreurs.

L'histoire des codes correcteurs d'erreurs remonte à l'année 1948, lorsque Claude Shannon a publié son célèbre ouvrage "*A Mathematical Theory of Communication*", qui a construit un cadre pour le codage de canal, codage de source et de la théorie de l'information. Pour la première fois, nous avons vu des preuves de l'existence de codage canal, qui permettent une communication fiable aussi longtemps que le taux d'information ne dépasse pas ce que l'on appelle la capacité de canal. Néanmoins, dans les 60 années qui suit, aucun des codes qui ont été découverts n'a pu s'approcher de la limite théorique jusqu'à l'arrivée des Turbo codes et la renaissance des codes LDPC. En tant que plus grand concurrent des Turbo codes, les avantages des codes LDPC comprennent la mise en œuvre parallèle des algorithmes de décodage et, plus fondamentalement, la construction graphique des codes. Cependant, il y a aussi quelques inconvénients des codes LDPC, par exemple la dégradation des performances en raison de la présence de cycles courts dans le code, aussi le temps de décodage itératif est long. Dans cette thèse, nous allons nous concentrer sur l'amélioration des algorithmes de décodage pour traiter ces problèmes.

Dans la première partie, durant le processus de décodage, nous étudions de nouvelles stratégies de décodage qui compensent l'effet négatif des cycles courts en pondérant les messages extrinsèques échangés entre les nœuds d'un graphe de tanneur (TG). L'algorithme de propagation de croyance (BP) pondéré résultant vise à mettre en œuvre un décodage efficace, à savoir la reconstruction de signal précis avec une faible latence de

décodage, pour les codes LDPC via la méthode de conception. L'algorithme de décodage proposé appelé *Exponential Factor Appearance Probability Belief Propagation* (EFAP-BP) peut être utilisé pour améliorer de façon significative les performances de décodage pour des codes LDPC réguliers et irréguliers. Plus important encore, la détermination des paramètres de pondération se déroule dans une phase avant le processus de décodage de sorte qu'aucune complexité de calcul supplémentaire n'est ajoutée au cours du processus de décodage en temps réel.

Dans la deuxième partie, nous nous déplaçons vers l'algorithme *Min Sum* (MS) et sa version modifiée *Normalized Min Sum* (NMS), qui sont les plus utilisés dans les systèmes de communication réel. Nous proposons un nouvel algorithme appelé *Girth Aware Normalized Min Sum* (GA-NMS), cet algorithme proposé est une nouvelle version de l'algorithme NMS qui prenne en considération la structure cyclique du code via le facteur d'optimisation de l'algorithme NMS. L'algorithme GA-NMS est évalué en utilisant un code LDPC de la norme WIMAX, les résultats de simulation obtenus montrent que l'algorithme proposé est mieux par rapport aux algorithmes NMS et le MS, en taux d'erreur et en rapidité de convergence avec un minimum de complexité.

Abstract

In this digital era, modern communication systems play an essential part in every aspect of life, with examples ranging from mobile networks and satellite communications to Internet and data transfer. Unfortunately, all communication systems in a practical environment are noisy, which indicates that we can either improve the physical characteristics of the channel or find a possible systematical, i.e. error control coding.

The history of error control coding dates back to 1948 when Claude Shannon published his celebrated work "*A Mathematical Theory of Communication*", which built a framework for channel coding, source coding and information theory. For the first time, we saw evidence for the existence of channel codes, which enable reliable communication as long as the information rate of the code does not surpass the so-called channel capacity. Nevertheless, in the following 60 years none of the codes have been proven closely to approach the theoretical bound until the arrival of turbo codes and the renaissance of LDPC codes. As a strong contender of turbo codes, the advantages of LDPC codes include parallel implementation of decoding algorithms and, more crucially, graphical construction of codes. However, there are also some drawbacks to LDPC codes, e.g. significant performance degradation due to the presence of short cycles or very high decoding latency. In this thesis, we will focus on improving the decoding algorithms to tackle those issues.

firstly, at the decoding process, we study novel decoding strategies which compensate for the negative effect of short cycles by reweighting part of the extrinsic messages exchanged between the nodes of a tanner graph (TG). The proposed reweighted *Belief Propagation* (BP) algorithm aim to implement efficient decoding, i.e. accurate signal reconstruction and low decoding latency, for LDPC codes via design method. An *Exponential factor appearance probability belief propagation* (EFAP-BP) algorithm is proposed which can be employed to enhance decoding performance significantly for regular and irregular LDPC codes. More importantly, the optimisation of reweighting parameters only takes

place in an offline stage so that no additional computational complexity is required during the real-time decoding process.

Secondly we study the *Min Sum* (MS) algorithm and his modified version *Normalized Min Sum* (NMS) which are the most used in the real communication systems. We propose new algorithm called *Girth Aware Normalized Min Sum* (GA-NMS), the algorithm proposed is a new version of the NMS algorithm which take into considération the cyclic structure of the code via the optimization factor of the NMS algorithm. the GA-NMS algorithm is evaluated using WIMAX code, the simulation results obtained show that the proposed algorithm is better compared to the NMS and MS standard algorithms in bit error rate performances and convergence behaviors speeds with minimum of complexity.

Introduction Générale

Contexte & Motivation

Les télécommunications résultent d'un besoin beaucoup plus ancien de l'être humain, ainsi que des autres espèces animales, de communiquer, autrement dit "de mettre en commun, de faire connaître des informations".

La communication peut avoir plusieurs formes auditives, visuelles, chimiques, olfactives, etc. Alors que certaines espèces animales ont développé des formes chimiques ou olfactives, les hommes utilisent surtout la communication auditive et visuelle (voix, sifflements, gestes, peintures, écriture, ...) depuis l'antiquité afin de transmettre leur savoir. Ils ont même laissé des traces de leur passage que nous retrouvons aujourd'hui. Par exemple, des peintures rupestres, vieilles de plus de 26 000 ans, ont été retrouvées dans la grotte Apollo 11 en Afrique Namibie 1970 [1].

Quelle que soit la forme utilisée, la distance le temps et la quantité se sont tout de suite imposés comme des obstacles à surmonter, notamment pour la coordination militaire.

La découverte de l'électricité a permis d'accélérer et d'étendre encore plus les communications au travers du télégraphe découvert par Morse en 1840 [2]. L'invention du transistor en 1947 dans les laboratoires Bell [3] a accéléré le développement technologique de ces dernières décennies. La technologie a ainsi permis d'accélérer et d'étendre les communications avec des systèmes tels que la radio, la télévision ou encore les satellites de communication.

Tous ces moyens de communication sont cependant soumis à des perturbations. Par exemple, dans une rue bruyante, ou lors du passage d'un train, le niveau de bruit est tel que la communication orale entre deux personnes peut s'avérer difficile. De même, à l'écrit, des parties du message peuvent être effacées. Enfin, avec les communications numériques, des interférences peuvent altérer le message transmis, que ce dernier transporte la voix,

des vidéos ou des données quelconques. La correction de ces erreurs apparaît alors comme une nécessité.

Entre deux personnes, la communication est facilitée par le cerveau. En effet, ce dernier compense les perturbations dues aux bruits externes car il a des connaissances sur la langue utilisée ainsi que sur le contexte de la conversation. Il en est de même pour les messages écrits. La connaissance de la langue et du contexte permet de retrouver les mots qui pourraient manquer. En ce qui concerne les communications numériques, les valeurs manipulées sont composées de 0 et de 1.

Par conséquent, lorsqu'une valeur reçue est fausse, alors il y a une erreur qui est irrécupérable. Plusieurs problématiques en découlent. Comment déterminer que le message reçu est vrai ou faux ? Comment corriger les erreurs de transmission ? Pour répondre à ces questions, des codes correcteurs d'erreurs ont été inventés. Ces derniers permettent de rajouter de l'information au message à transmettre ce qui peut correspondre, par analogie, au contexte d'une conversation entre deux personnes.

Différents codes correcteurs d'erreurs sont utilisés dans des multitudes d'appareils comme les téléphones, les CD de musique, les DVD, les disques durs ou encore les paquets transmis par internet. En modélisant mathématiquement l'information transmise sur un canal de transmission, une étape importante fût franchie par Claude Shannon en 1948 [4]. Grâce à l'approche qu'il propose, il devient possible de répondre aux deux questions fondamentales posées par la théorie de l'information :

— Quelles sont les ressources nécessaires à la transmission de l'information ?

— Quelle est la quantité d'information que nous pouvons transmettre de façon fiable ?

Ainsi, depuis que Shannon fixa les performances limites d'une transmission sur un canal bruité, le challenge des chercheurs est de trouver des codes ainsi que des algorithmes du décodage associés qui permettent d'atteindre cette limite. Ainsi deux familles de codes correcteurs d'erreurs pseudo-aléatoires se sont distinguées par leurs performances atteignant la capacité de Shannon, à savoir : les Turbo-codes introduit par Berrou, Glavieux et Thitimajshima [5] en 1993, et les codes *Low-Density-Parity-Check* (LDPC) inventés à l'origine par Robert Gallager en 1963 [6]. Ce dernier a montré que ces codes ont des propriétés théoriques excellentes et il a également fourni un algorithme du décodage.

Cependant, du fait de leur complexité du codage, du décodage et les moyens matériels de l'époque, ces codes n'ont pas suscité suffisamment d'intérêt au sein de la communauté

de la théorie du codage. Cet oubli durera jusqu'à l'introduction des Turbo-codes et du principe itératif. Ainsi en 1996, les codes LDPC redécouverts ensuite par Mackay et al. [7] et Sipser et al. [8]. Il a été démontré que les codes LDPC longs avec un décodage itératif basé sur la propagation de la croyance atteignent une performance d'erreur à une fraction de décibel de la limite de Shannon [9] [10] [11] [12] [13]. En conséquence, ces codes sont devenus de puissants concurrents pour les turbocodes [5] [14] par leur capacité puissante de contrôle d'erreur dans de nombreux systèmes de communication et de stockage numérique où une haute fiabilité est requise.

Les codes LDPC sont des codes correcteurs d'erreurs pseudo-aléatoires basés sur des matrices à vérification de parité de faible densité. Après la redécouverte des codes LDPC par Mackay [7] en 1996, il a montré que l'algorithme du décodage développé par Gallager [6], peut également se décrire comme l'algorithme de propagation de croyance (*Belief Propagation* (BP)) de Pearl [15], communément appelé *Sum-Product* (SP) [16] [11] [17] [18]. Une des contributions les plus significatives depuis 1995 est celle de Luby et al. [19] qui ont introduit et paramétré les codes LDPC irréguliers. Ces derniers ont pour principale caractéristique de présenter de meilleures performances que les codes réguliers à condition que l'irrégularité soit judicieusement choisie.

Afin de bien choisir cette irrégularité, Richardson, Urbanke et al. [20] [11] ont montré qu'il était possible de prédire le comportement des codes LDPC de taille infinie lors de l'étape du décodage et que celui-ci était fonction des paramètres définissant le code LDPC. Cet algorithme, nommé évolution de densité, permet alors de déterminer la distance qui sépare les performances de codes LDPC (fonction de ces paramètres d'irrégularité) de la limite de Shannon. A l'aide de cet outil d'analyse, les auteurs de [20] [11] ont ensuite proposé d'optimiser la structure des codes LDPC de manière à minimiser cet écart à la limite de Shannon, ce qui a permis d'obtenir des codes LDPC atteignent cette limite pour le canal binaire à effacement [21] et le canal à bruit additif blanc Gaussien [22].

Depuis leur redécouverte, les codes LDPC ont été utilisés avec succès dans un certain nombre d'applications, et ils sont proposés pour une utilisation par de nombreux standard de communication.

Problématique & Objectif

Depuis la redécouverte du code LDPC par Mackay en 1995, des recherches ont été fait sur le code LDPC et de son algorithme de décodage itératif optimal *Belief Propagation* (BP) appelé aussi *Sum Product* (SP) afin d'améliorer les performances de corrections des erreurs, réduire la complexité de décodage et augmenter le débit. L'existence des cycles courts dans le code empêche la correction de l'information durant le processus itératif [16] [18], ce problème est appelé dans la littérature "surestimation" ou "excé de confiance".

Dernièrement Liu et de Lamare [23] ont proposé une version pondéré de l'algorithme BP, le facteur de pondération tient en compte la structure cyclique et la distribution des éléments non nuls dans le code, cet algorithme est appelé *Variable Factor Appearance Probability Belief Propagation* (VFAP-BP), il permet d'améliorer les performances en correction des erreurs et en rapidité de convergence pour les codes réguliers et irréguliers.

L'algorithme VFAP-BP a des inconvénients comme :

— La difficulté de calcul du facteur de pondération.

— Le facteur de pondération n'est pas idéal.

— La complexité de calcul ajouté durant le processus itératif.

La solution VFAP-BP reste une solution qui n'est pas idéal, et le problème des cycles courts dans le code est toujours posé, L'objectif de la thèse présenté dans cette mémoire est d'améliorer les performances, en termes de correction des erreurs, et en temps de latence de l'algorithme de décodage sur la base de la connaissance de la structure cyclique du code, et avec un minimum de complexité.

Contribution Scientifique

Pour répondre à ces objectifs, nos travaux de recherche apportent différentes contributions. Elles sont énumérées ci-dessous :

1. Proposition de l'algorithme *Exponential Factor Appearence Probability Belief Propagation* (EFAP-BP) : la solution EFAP-BP proposé est une version modifié de l'algorithme VFAP-BP, elle permet de simplifier le calcul du facteur de pondération ainsi de réduire la complexité de calcul durant le processus de décodage, du côté

performances, l'algorithme EFAP-BP est meilleur en convergence et en correction des erreurs en comparaison avec les deux algorithmes VFAP-BP et le BP standard pour les codes réguliers et irréguliers.

2. Proposition de l'algorithme *Girth Aware Normalized Min Sum* (GA-NMS) : nous avons développé un nouvel algorithme qui fusionne entre la connaissance de la distribution des cycles courts dans le code et l'algorithme *Normalized Min Sum* (NMS), l'évaluation du nouvel algorithme résultant GA-NMS a été effectué en utilisant un code LDPC de la norme WIMAX du standard IEEE P802.16e en comparaison avec les algorithmes de décodage NMS et Min Sum (MS) Standard. Les simulations ont montré que l'algorithme proposé est meilleur en rapidité de convergence et en correction des erreurs avec un minimum de complexité.

Plan du Mémoire

Ce mémoire est composé de quatre chapitres. Les deux premiers introduisent le contexte de recherche ainsi que les algorithmes de décodages dont nous avons besoin. Les deux suivants décrivent les différentes contributions apportées au cours des années de thèse.

Le **premier chapitre** a pour objectif de placer le sujet de thèse dans le contexte des communications numériques. Pour atteindre ce but, nous commençons par étudier les différents blocs constitutifs d'une chaîne de communications numériques afin de donner une vision globale du sujet. Deux blocs particuliers de cette chaîne concernent le codage et le décodage de canal. Ces derniers représentent un moyen de protéger l'information à transmettre contre les erreurs de transmission ou de stockage. Une description du théorème de Shannon a été présentée. La dernière partie de ce chapitre comporte une introduction détaillée et des notions de base sur un code correcteur d'erreur, on s'intéresse particulièrement dans ce chapitre à la famille des codes en bloc dont le code LDPC fait partie.

Le **deuxième chapitre** de cette thèse introduit les différentes notations et concepts nécessaires à la compréhension des chapitres suivants. Commençant par une introduction détaillée des codes LDPC, la partie suivante comporte les principales méthodes de constructions des codes LDPC qui existent dans la littérature. Dans le but de comprendre l'opération de codage une méthode de codage a été décrite finalement une description sur l'opération de décodage avec des détails sur les algorithmes dont nous avons besoin dans

les prochains chapitres SP, MS.

L'existence des cycles courts dans le code empêche la convergence et dégrade les performances en correction des erreurs, le **chapitre trois** présente la première contribution dans cette thèse qui consiste à donner une solution au problème des cycles courts, ce dernier commence par une présentation détaillée du problème des cycles courts ensuite définit comment on peut identifier les cycles courts dans un code, la partie suivante comprenne une description détaillée de l'algorithme qui existe dans la littérature VFAP-BP ensuite la cinquième partie de ce chapitre introduit la nouvelle solution proposée EFAP-BP, à la fin de ce chapitre les résultats de simulation sont présentés qui montrent les performances de l'algorithme EFAP-BP en comparaison avec les deux algorithmes VFAP-BP et le BP standard.

Toujours dans le même axe de trouver une solution aux problèmes des cycles courts durant le décodage itératif, le **chapitre quatre** présente une nouvelle méthode pour améliorer les performances de décodage de l'algorithme NMS en utilisant un code LDPC dans le standard IEEE P802.16e. Ce chapitre commence par présenter une méthode pour le calcul des cycles courts traversant chaque nœud dans le graphe, ensuite une description détaillée de l'algorithme NMS, le paragraphe suivant introduit la solution proposée GA-NMS, à la fin de ce chapitre les résultats de simulation sont présentés et discutés.

Finalement, la conclusion de ce travail synthétisera les différentes idées présentées dans ce document et introduira également les perspectives et travaux futurs.

Publications

[**A1**] A.Kadi, S.Najah, M.Mrabti "Hardware Architecture for LDPC code using Min Sum Algorithm" WOTIC December 2013.

[**A2**] A.Kadi, S.Najah, M.Mrabti "Hardware Design for LDPC code using Sum Product Algorithm" CMT May 2014.

[**A3**] A.Kadi, S.Najah, M.Mrabti "Reduced latency and complexity of (VFAP-BP) algorithm for regular and irregular LDPC codes" ISIVC November 2014.

[**A4**] A.Kadi, S.Najah, M.Mrabti "An Exponential Factor Appearance Probability Belief Propagation algorithm for regular and irregular LDPC codes" AEUE - International Journal of Electronics and Communications Elsevier vol. 69, pp. 933 - 936, 2015

[**A5**] A.Kadi, S.Najah, M.Mrabti, Samir Belefkih "Improving Performances of the Min Sum Algorithm for LDPC Codes" proceeding Medict page 381 – 388, 2015.

[**A6**] A.Kadi, S.Najah, M.Mrabti, E.Boutillon "Novel method for improving performances of Normalized MS decoder using WIMAX code" ACOSIS Oct 2016, pp. 1-5 (Best paper award).

Chapitre 1

Théorie de l'information

1.1 Introduction

Les systèmes de transmission numérique transportent l'information de la source au destinataire en utilisant un support physique comme le câble, la fibre optique ou encore, la propagation sur un canal radioélectrique avec le plus de fiabilité possible. Les signaux transportés peuvent être d'origine numérique, comme dans les réseaux de données, soit d'origine analogique (parole, image...) mais ils doivent être convertis sous une forme numérique. Dans ce chapitre, nous présentons quelques notions de base de la communication numérique. Nous allons dans un premier temps décrire d'une façon générale les différents éléments de la chaîne de transmission, à savoir, l'émetteur, le récepteur et le canal de transmission. Nous introduisons ensuite les différentes familles de codes correcteurs d'erreurs.

1.2 Chaîne de transmission numérique

Depuis l'invention du tube à vide en 1904 par John Ambrose Fleming [24], l'électronique et les télécommunications ont pris de l'importance. Ces applications requièrent le traitement de données, qui a été facilité depuis l'invention du transistor. Celui-ci a été découvert en 1947 [3], par John Bardeen, Walter Brattain, et William Shockley, et a depuis remplacé le tube à vide. Le traitement numérique des données utilise en particulier ces transistors. La transmission d'informations a été modélisée sous la forme d'une chaîne. La figure 1.1 représente les blocs constitutifs d'une chaîne de communications numériques.

Les principaux éléments sont, d'une part l'émetteur, le canal de transmission et le récepteur, d'autre part le codage/décodage de source, dont le but principal est de réduire la quantité d'information à transmettre. Ensuite le codage et décodage canal. Le but n'est pas de réduire la quantité d'information mais de rajouter de la redondance au message à transmettre afin de pouvoir détecter et corriger le plus d'erreurs possibles. Avant d'être transmis ou reçu, le message subit une modification appelée modulation ou démodulation numérique. Ensuite, le message est transmis à travers un canal de transmission (air, fibre optique, fils torsadé) et subit des perturbations. Une modélisation du canal de transmission avec ses perturbations, permet de caractériser un codage canal. Dans ce chapitre nous aborderons seulement trois types de canaux. Le canal binaire symétrique, le canal à effacement et le canal à Bruit Blanc Additif Gaussien (BBAG).

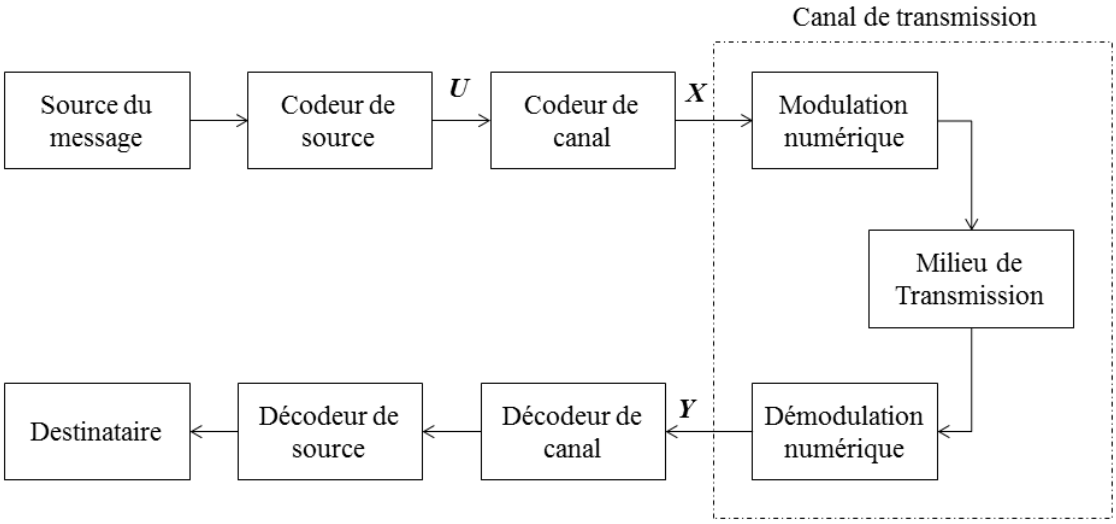


FIGURE 1.1 – Chaîne de communications numériques

1.2.1 Codeur de source

Le codage de source vise à représenter l'information à transmettre sous la forme numérique la plus compacte possible. Il s'agit donc d'une part de numériser les données si elles sont analogiques (son, image, vidéo), d'autre part de compresser les données numériques, de manière à réduire le débit de transmission ou le volume de stockage. Nous distinguerons deux types de compression :

— La compression sans perte, ou conservative, permet de retrouver exactement les données originales après décompression.

— La compression avec perte, ou non-conservative, conduit à une perte d'information. Elle est utilisée pour des objets destinés à être perçus par un humain, en veillant à ce que la perte d'information ne soit pas perceptible. C'est sur ce principe que sont fondés les codeurs perceptifs MPEG (audio et vidéo) et JPEG (image).

1.2.2 Codeur canal

Le bruit et les interférences du canal, qui dégradent les signaux de communication transmis, provoquent des erreurs de détection en réception. Pour un niveau de bruit et d'interférences donné, la probabilité d'erreur peut être réduite en augmentant la puissance d'émission, puisqu'elle est une fonction décroissante de celle-ci. Cependant, cette augmentation de puissance n'est pas toujours souhaitable : d'une part elle se traduit par un accroissement de la consommation électrique du terminal, à éviter pour des terminaux sans-fil ; d'autre part, dans le cas d'une transmission en espace libre, elle augmente les interférences inter-utilisateurs, ce qui accroît la probabilité d'erreur.

Une autre solution est le codage de canal (voir figure 1.2), qui consiste à ajouter au message binaire des bits de redondance, de telle sorte que le message codé ait une structure particulière. En réception, le décodeur de canal vérifie si cette structure est bien respectée. Dans le cas contraire, une erreur est détectée et éventuellement corrigée. Cela rend le système de communication moins vulnérable aux erreurs du canal. Chaque code est caractérisé par le rapport

$$R = \frac{K}{N} < 1 \tag{1.1}$$

appelé le rendement du code. Avec K le nombre de bits d'informations et N le nombre de bits du mot de code. Le codeur de canal est l'élément principal permettant de combattre les erreurs de transmissions résultant des perturbations du canal physique, ou en d'autres termes d'assurer un Taux d'erreurs binaire (TEB) faible sous la contrainte de puissance d'émission. Le principe de codage de canal consiste à insérer parmi les éléments d'informations, des données supplémentaires suivant une loi connue afin de pouvoir en réception, détecter ou corriger les éventuelles erreurs.

Pour tout codeur, il est utile de caractériser son rendement ou le taux de codage défini par :

$$R = \frac{K}{N}$$

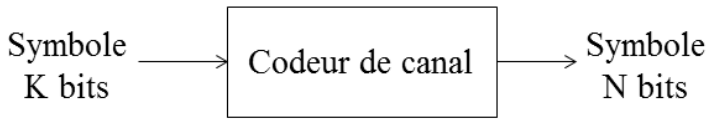


FIGURE 1.2 – Schéma d'un codeur de canal

Ce rapport mesure la proportion de bits utiles transmis par rapport à la totalité de bits envoyés sur le canal. Le codage de canal a pour inévitable contrepartie une augmentation du débit, si l'on souhaite maintenir le débit de données utiles. On cherchera donc des codages offrant la meilleure protection pour une augmentation de débit minimale. Le but fondamental du code correcteur est de maximiser la précision de transmission dans les contraintes d'énergie du signal, bande passante du système et complexité de l'ensemble des circuits. Il y a donc antagonisme entre codage de source et codage de canal, l'objectif du premier étant de diminuer la redondance du message de source, celui du deuxième d'en ajouter dans un but de protection.

1.2.3 Modulation et démodulation

La modulation, désigne l'opération visant à adapter le signal à transmettre aux caractéristiques physiques du canal de transmission utilisé. Dans notre étude, le message à moduler est numérique, nous parlons alors de modulation numérique. Par définition, la modulation numérique est une opération qui consiste à associer une suite numérique binaire appelée signal modulant à un signal analogique appelé porteuse, soit à une fonction à valeurs continues et à temps continu. Le choix d'une modulation s'effectue en considérant les caractéristiques du canal de transmission, et l'efficacité spectrale ciblée par l'application. A la sortie du canal, le démodulateur réalise la fonction inverse, il fournit une fiabilité ferme ou bien délivre une information probabiliste sur la décision qu'il prend (fiabilité). Dans la suite, nous allons définir les principaux critères caractérisants une modulation numérique, à savoir :

— Le débit.

— Le taux d'erreur sur les bits.

— L'efficacité d'occupation spectrale.

— L'efficacité de la puissance émise.

— La simplicité de réalisation.

Débit binaire

Le débit binaire D (bit/s) est le nombre d'informations élémentaires (bits) émises par unité de temps.

$$D = \frac{1}{T_b}(bit/s) \tag{1.2}$$

avec T_b est la durée d'un bit.

Taux d'erreur sur les bits

Les modulations numériques sont évaluées grâce au taux d'erreur par bit. Ce dernier permet de chiffrer l'influence du bruit (grâce au rapport signal sur bruit ou à l'efficacité de puissance) pour chaque modulation et en fonction du nombre d'états de la modulation. Il se définit de la manière suivante :

$$TEB = \frac{\text{Nombre de bits erronés}}{\text{Nombre de bits émis}} \tag{1.3}$$

L'efficacité spectrale

L'efficacité spectrale (en Bit/Seconde/Hertz) d'un signal numérique est le nombre de bits par seconde de données qui peuvent être supportés pour chaque hertz de la bande de fréquence utilisée(B en Hertz) :

$$\eta_s = \frac{D_u}{B} \tag{1.4}$$

On peut aussi noter que pour des applications où la bande passante est limitée par des contraintes physiques, il faut choisir une technique qui donne la plus haute efficacité spectrale, laquelle doit permettre d'obtenir de faibles taux d'erreur sur le bit en sortie du système.

L'efficacité en puissance E_b/N_0

L'efficacité en puissance, se définit par le rapport entre l'énergie moyenne E_b par bit d'information et la densité spectrale d'un bruit blanc $N_0/2$. Nous allons exprimer la relation entre l'efficacité en puissance et le rapport signal sur bruit *Signal Noise Ratio* (SNR). Tout d'abord, définissons le rapport existant entre l'énergie totale E et l'énergie moyenne E_b par bit dans un système à M états.

$$E_b = \frac{E}{log_2 M} \tag{1.5}$$

La puissance S est définie comme le rapport entre l'énergie totale E et la durée d'un bit T_b, nous obtenons la relation suivante pour le rapport signal sur bruit :

$$\frac{S}{N} = \frac{E/T_b}{N_0 B} = \frac{E}{T_b N_0 B} \tag{1.6}$$

1.2.4 Canal de transmission

Le canal de transmission est situé entre la sortie du codeur de canal et l'entrée du décodeur de canal. La partie analogique de la transmission n'est pas décrite en détails dans ce mémoire. Néanmoins, un signal physique est généré à la sortie de la modulation numérique afin de pouvoir le transmettre au travers du milieu de transmission. Un milieu de transmission est un support transportant de l'information comme par exemple l'air, des fils torsadés, un câble coaxial ou encore une fibre optique. Par extension, les supports physiques, comme des disques durs ou des CD, sont des milieux de transmission. De manière générale, un canal de transmission est défini mathématiquement par deux ensembles et une probabilité de transition d'un ensemble vers l'autre :

— un ensemble χ, contenant toutes les entrées possibles,

— un ensemble γ, contenant toutes les sorties possibles,

— une probabilité de transition noté $W(\chi|\gamma)$.

Un canal de transmission subit des perturbations de différents types selon la nature du canal. Ces perturbations peuvent être de type électromagnétique, de type thermique ou des interférences liées à la présence de plusieurs utilisateurs sur le canal. Le canal prend en entrée des données binaires et produit en sortie une estimation des bits originaux. Cette estimation peut être "dure" ou "souple". Une sortie dure implique une prise de décision -$\{0, 1\}$- quant à la valeur des bits de sortie. Une sortie souple implique que la valeur du bit en sortie soit représentée par une probabilité de sa valeur. Des valeurs souples permettent d'obtenir en général de meilleurs résultats lors du décodage du message mais nécessitent une complexité calculatoire supérieure à celle nécessaire pour des valeurs dures. Il existe quatre propriétés principales des canaux :

— Un canal **discret** est un canal de communication qui ne transmet que des nombres entiers ou, plus généralement, un nombre fini de symboles. Le plus souvent, il s'agit d'un canal binaire qui ne transmet donc que 0 ou 1.

— Un canal **continu** possède un alphabet d'entrée et de sortie qui est à valeur dans l'ensemble de réels.

— Un canal **stationnaire** possède une probabilité de transition qui ne dépend pas du temps.

— Un canal **sans effet mémoire** implique qu'un symbole en sortie ne dépend que du symbole d'entrée.

Trois familles de canaux de transmissions sont présentées dans la section suivante 1.3. Les deux premiers modèles de canal sont simples afin d'expliquer les principes de bases. Ensuite, le modèle de canal qui permet de représenter simplement les effets d'un canal réel est détaillé.

1.2.5 Décodeur de canal

Le décodeur de canal transforme la séquence reçue à une séquence symbole appelée la séquence estimée. La stratégie de décodage est basée sur les règles de codage de canal et les caractéristiques de bruit de canal. Plusieurs stratégies peuvent être utilisées par le décodeur de canal.

Détection d'erreurs : Le décodeur observe la séquence reçue (ferme ou pondérée) et détecte la présence éventuelle d'erreurs. Cette détection peut servir à contrôler le taux d'erreur ou à mettre en œuvre des techniques de retransmission (le décodeur demande à l'émetteur de retransmettre la séquence dans laquelle une erreur a été détectée).

Correction d'erreurs : Cette opération nécessite des algorithmes beaucoup plus complexes que la simple détection, et plus de redondance dans la séquence émise. Le décodeur observe la séquence reçue ,détecte et corrige (si cela est possible) les éventuelles erreurs.

1.2.6 Décodeur de source

Basé sur la règle du codeur de source, le décodeur de source transforme la séquence à son entrée en une séquence estimée de la séquence de sortie de la source et la délivre à l'utilisateur. Par le dessin adéquat du système de transmission, ce serait possible d'enlever

ou de réduire les effets d'atténuation et de distorsion et minimiser les effets du bruit. L'impact de bruit ne peut pas être totalement enlevé tant que nous n'avons pas la connaissance complète du bruit.

1.3 Modèles des Canaux

1.3.1 Le canal binaire symétrique

Le Canal Binaire Symétrique (CBS) est un canal discret, stationnaire et sans effet mémoire. Ses ensembles d'entrée $\chi = [\chi_0 = 0, \chi_1 = 1]$ et de sortie $\gamma = [\gamma_0 = 0, \gamma_1 = 1]$ sont de dimensions finies. Puisque ce canal est stationnaire et sans effet mémoire, les probabilités de transitions sont indépendantes du temps et l'élément γ_k ne dépend que de l'élément χ_k. Les probabilités de transitions du canal - probabilité d'obtenir la sortie sachant l'entrée - sont représentées dans la figure 1.3 et peuvent être exprimées comme suit :

$$Pr(y_i = \gamma_0 | x_i = \chi_1) = Pr(y_i = \gamma_1 | x_i = \chi_0) = P_e \tag{1.7}$$

$$Pr(y_i = \gamma_0 | x_i = \chi_0) = Pr(y_i = \gamma_1 | x_i = \chi_1) = 1 - P_e \tag{1.8}$$

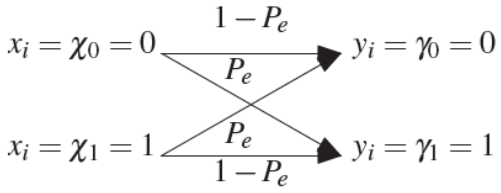


FIGURE 1.3 – Canal Binaire Symétrique de probabilité d'erreur P_e

1.3.2 Canal à effacement BEC

Tout comme le CBS, le *Canal À Effacement* (CAE) est un canal discret, stationnaire et sans effet mémoire. Sa sortie est constituée de l'ensemble $\{0, \epsilon, 1\}$. Le symbole ϵ représente un effacement. Ce dernier correspond à une perte totale d'information permettant de prendre une décision quant à la valeur reçue. Par exemple, dans les communications internet, des paquets *Internet Protocol* (IP) sont utilisés, et certains n'atteignent pas le destinataire car le routeur peut avoir fait une mauvaise redirection. Les probabilités de transitions du canal sont représentées dans la figure 1.4 et peuvent être exprimées comme

suit :

$$Pr(\text{effacement}) = P_e \tag{1.9}$$

$$Pr(y_i = \gamma_0 | x_i = \chi_0) = Pr(y_i = \gamma_1 | x_i = \chi_1) = 1 - P_e \tag{1.10}$$

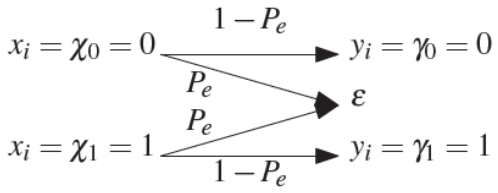


FIGURE 1.4 – Canal binaire à effacement.

1.3.3 Le canal à bruit blanc gaussien

Le canal à BBAG ouen anglais *Additive white Gaussian noise* (AWGN), représenté dans la figure 1.5, est un canal continu, stationnaire et sans effet mémoire. Il est couramment utilisé en communications numériques car il est simple d'utilisation et permet de fournir une première approximation des phénomènes qui s'appliquent sur un canal de transmission réel. Il possède les caractéristiques suivantes :

— il est à entrée, χ_k, discrète et à valeur dans $\{0, 1\}$ dans le cas d'un canal binaire,

— il est à sortie, γ_k, continue,

— il est stationnaire ; il ne dépend pas du temps,

— il est sans effet mémoire ; γ_k ne dépend que de χ_k.

Le BBAG est un bruit dont la densité spectrale de puissance est la même pour toutes les fréquences (bruit blanc). Il est dit additif car il est simplement ajouté au signal entrant. Enfin, il est dit gaussien du fait de sa densité de probabilité de transmission définie comme suit :

$$Pr(x_i|y_i) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x_i - y_i)^2}{2\sigma^2}\right) \tag{1.11}$$

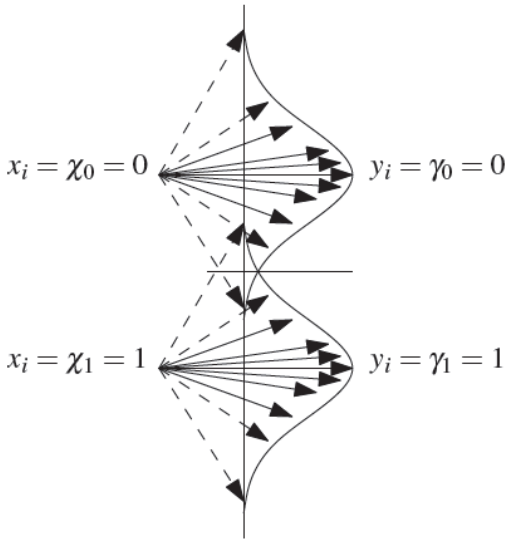


FIGURE 1.5 – Canal binaire à entrée binaire et bruit blanc additif Gaussien.

1.4 Formats de modulation

La taille de l'alphabet transmis A pour le canal AWGN est généralement déterminé par le format de modulation utilisé. Les formats de modulation plus courants sont :

— La modulation *Binary Phase Shift Keying* (BPSK) est une modulation de phase à 2 états de la fréquence intermédiaire. Elle utilise pour la transmission l'alphabet $A = -\sqrt{E}, +\sqrt{E}$, et pour la réception l'alphabet $B = \mathbb{R}$. E représente l'énergie de symbole mesurée au niveau du récepteur.

— *Quadrature phase-shift keying* (QPSK), cette modulation utilise un diagramme de constellation à quatre points, à équidistance autour d'un cercle. Avec quatre phases, QPSK peut coder deux bits par symbole. Elle utilise pour la transmission l'alphabet $A = \sqrt{E/2}\{(-1-i), (-1+i), (+1-i), (+1+i)\}$ avec des symboles complexes, et pour la réception l'alphabet $B = \mathbb{C}$.

— *Quadrature amplitude* (QAM), c'est une généralisation de la modulation QPSK à un ordre supérieur, en utilisant des symboles équidistants.

Dans cette thèse, nous allons adopter la modulation BPSK. Toutefois les méthodes ne sont pas limitées à la modulation BPSK, mais peut carrément être appliqué sur des systèmes utilisant d'autres formats de modulation.

1.5 Théorèmes de Shannon

Dans la théorie de codage de canal, Le rendement d'un code $r = k/n$ mesure la proportion de bits utiles transmis par utilisation du canal de transmission. Le célèbre théorème de codage de canal démontré par C. E. Shannon en 1948 [4] est le suivant :

Let a discrete channel have the capacity C and a discrete source the entropy per second H. If H there exist a coding system such that the output of the source can be transmitted over the channel with an arbitrarily small frequency of errors(or an arbitrarily small equivocation). If $H > C$ it is possible to encode the source so that the equivocation is less than H.

Questions :

1. Peut-on transmettre sans erreurs sur un canal bruité ?
2. Transmettre sans erreurs => avoir un rendement tendant vers 0 ?
3. Y-a-t-il un rendement optimal, pour un niveau de bruit donné ?

Les réponses à ses questions sont apportées par C. Shannon en 1948 dans son papier [4]. C'est le fondement de la théorie de l'information.

a. Il existe une notion de capacité pour un canal, elle est fonction du niveau de bruit, c-à-d. de la loi du canal $P(Y|X)$.

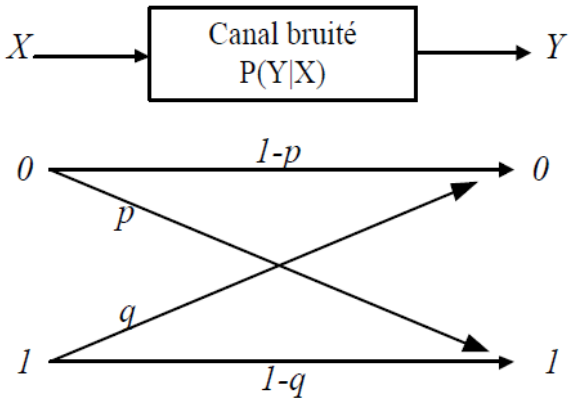


FIGURE 1.6 – Canal binaire

$$C = max_{P(x)} I(X;Y) \tag{1.12}$$

Où $I(X;Y)$ mesure l'information commune entre l'entrée X et la sortie Y du canal (mesurée en "nombre de bits communs en moyenne").

b. Si l'on émet avec un code correcteur de rendement $r > C$, la probabilité d'avoir des bits erronés au décodage reste strictement positive.

c. Pour tout rendement $r < C$ donné, on peut trouver un code de rendement r et dont la probabilité d'erreur binaire est aussi proche de 0 que l'on veut.

Preuve constructive : on prend un code au hasard, 2^k mots de n bits, avec $k/n = r$.

Résultat asymptotique : la longueur n du code doit tendre vers l'infini.

En **résumé**, si l'on émet plus d'information que le canal peut en recevoir, il est certain d'obtenir après décodage des informations erronées. D'après Shannon, il est nécessaire d'utiliser un code aléatoire pour atteindre les performances limites. Malheureusement les codes aléatoires ne sont pas décodables. Tout l'art du codage de canal consiste à effectuer un compromis entre trouver un code presque aléatoire et la possibilité de le décoder.

1.6 Codes correcteurs d'erreurs

1.6.1 Introduction

La probabilité d'erreur dépend du rapport signal sur bruit SNR. Pour réduire cette probabilité, il est possible d'augmenter la puissance du signal, de diminuer la puissance de bruit sur le canal de transmission ou bien d'utiliser un codage correcteur d'erreurs. Ces approches posent néanmoins des problématiques pratiques.

Tout d'abord en ce qui concerne l'augmentation de la puissance du signal. Certaines applications ne permettent pas d'augmenter la puissance du signal significativement comme par exemple les transmissions satellitaires. De plus, augmenter la puissance du signal induit une augmentation de la consommation électrique. Pour les applications à basse consommation cela n'est pas envisageable. De plus, cette augmentation de la consommation entraîne un échauffement plus important. Par conséquent, un système de refroidissement doit être mis en place.

Ensuite, pour augmenter le SNR, la puissance du bruit du canal peut être réduite. Par exemple, au niveau de la réception du signal, l'utilisation d'une technologie plus récente pourrait répondre au problème mais coûte plus cher. Dans le cas des communications téléphoniques, il faudrait changer les caractéristiques de l'environnement afin de réduire la puissance du bruit. Mais cela n'est pas envisageable.

Enfin, un codage correcteur d'erreurs permet de réduire la probabilité d'erreur pour un SNR donné. Cependant, l'utilisation de codes correcteurs d'erreurs implique une complexité supérieure dans le traitement du message reçu. Malgré tout, ces codes apparaissent de plus en plus intéressant grâce aux circuits de codage et de décodage dédiés. De plus, leur utilisation permet de pas modifier la puissance du signal ni celle du bruit.

Les codes correcteurs d'erreurs peuvent être répartis en deux grandes familles, les codes convolutifs et les codes en blocs. Parmi les codes convolutifs on peut citer en particulier les Turbocodes inventés par Claude Berrou en 1995 [25]. Les codes en blocs comprennent, entres autres, les codes Raj **B**ose, D. K. Ray-**C**haudhuri et Alexis **H**ocquenghem (Bose Chaudhuri et Hocquenghem (BCH)), **R**eed-**S**olomon (RS), **R**eed-**M**uller (RM), les Turbocodes en blocs [26] ou bien les codes *Low Density Parity Check* (LDPC) inventé par Robert G. Gallager en 1962 [27], et dernièrement il y a l'apparition d'une nouvelle famille de code appelé codes polaire.

1.6.2 Notions de base des codes correcteurs

Un système de communication numérique peut se décomposer selon le schéma présenté sur la figure 1.1 en plusieurs unités. Nous faisons l'hypothèse que l'émetteur est constitué d'une source binaire dont les bits sont i.i.d (indépendamment et identiquement distribués). Ce message binaire est codé par un codeur de canal. Le codage de canal, appelé aussi codage détecteur/correcteur d'erreurs, consiste en l'introduction d'une redondance associée à l'information utile dans le message à transmettre. La redondance et l'information utile sont liées suivant une loi donnée. La trame binaire résultante est alors transmise sur un canal bruité. En réception, le décodeur de canal exploite la redondance produite par le codeur dans le but de détecter, puis de corriger les erreurs introduites lors de la transmission. L'ajout de la redondance dans le message à transmettre entraîne une perte d'efficacité du système. En effet, les bits de redondance introduits ne véhiculent pas de l'information utile. Cependant, cette perte à mettre en balance avec le gain de qualité obtenu par l'utilisation du codage. La notion et le calcul de ce gain seront illustrés dans les paragraphes suivants.

Considérant le système de correction d'erreurs schématisé dans la figure 1.7. Puisque les codes étudiés dans ce mémoire sont des codes en bloc, les propriétés du système sont formulées en supposant qu'un code en bloc est utilisé. En outre, il est supposé que les

symboles utilisés pour les messages sont des symboles binaires. Un message u avec K bits doit être transmis à travers un canal bruité. Le message est codé en un mot de code c à N bits, où $N > K$. Ensuite le mot de code est modulé au signal analogique en utilisant la modulation BPSK avec une énergie E par bit. Pendant la transmission à travers un canal AWGN, un bruit de densité spectrale N_0 est ajouté au signal pour produire le signal reçu y. Le signal reçu est démodulé pour avoir le vecteur reçu $\tilde{y}$, qui peut contenir soit des bits ou scalaires. Le décodeur de canal est ensuite utilisé pour trouver le mot de code envoyé $\hat{u}$, compte tenu du vecteur reçu $\tilde{y}$.

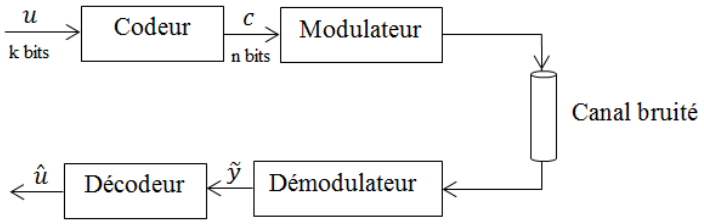


FIGURE 1.7 – Représentation d'un système correcteur d'erreur.

Pour le système décrit en dessus, un certain nombre de propriétés peuvent être définies :

— L'information transmise u est de K bits $u = (u_1, ..., u_K)$.

— La taille de mot de code c est de N bits $c = (c_1, ..., c_N)$. En générale, afin d'obtenir des meilleurs performances de correction d'erreur, N doit être le plus grand possible. Cependant, un mot de code plus grand nécessite un codeur/décodeur plus complexe et augmente le temps de calcul du système. Ainsi, il y'a un compromis entre ces facteurs dans la conception du système du codage et du décodage

— Le taux de codage est $R = K/N$. Évidemment, plus le taux de codage augmente, plus la quantité des informations transmises pour un mot de code de N bits augmente. Cependant, il est également le cas qu'un taux de codage réduit permet de transmettre plus d'informations à un niveau constant de la puissance de l'émetteur. Ainsi, le taux de codage est également un compromis à faire entre les performances de correction d'erreur et la complexité de l'ensemble codeur/décodeur.

— Le rapport signal sur bruit normalisé (*Signal to Noise Ratio* (SNR)), au niveau du récepteur est utilisé au lieu du SNR actuel E/N_0 afin de permettre de faire une comparaison équitable entre des codes de différents taux de codage. Le rapport signal sur bruit normalisé est noté SNR dans le reste de ce mémoire.

— Le taux d'erreur binaire (*Bit Error Rate* (BER) en anglais) décrit le nombre de bits reçus erronés par rapport au nombre de bits transmis moyennée sur plusieurs blocs.

— Le taux d'erreur de trame (*Frame Error Rate* (FER) en anglais) est le nombre de blocs erronés par rapport au total.

En générale, on peut distinguer trois façon dans lesquelles une tentative du décodage peut mettre fin :

— *Succès du décodage* : Le décodeur a trouvé un mot de code valide, et le message $\hat{u}$ correspondant au message transmis u.

— *Erreur du décodage* : Le décodeur a trouvé un mot de code valide, et le message $\hat{u}$ ne correspondant pas au message transmis u.

— *Echec du décodage* : Le décodeur a été incapable de trouver un mot de code valide en utilisant les ressources spécifiées.

Pour les deux derniers cas, erreur et échec du décodage, le décodeur a été incapable de trouver le bon message envoyé u. Cependant, la principale différence est que les défaillances du décodage sont détectables, tandis que les erreurs du décodage ne le sont pas. Ainsi, si, par exemple, plusieurs algorithmes du décodage sont disponibles, le décodage pourrait être retenté par un autre algorithme lorsqu'une défaillance du décodage se produit. Les systèmes de codage sont analysées en détail dans n'importe quel ouvrage sur la théorie du codage, par exemple [28] [29].

1.6.3 Notion du code

Définition 1 *Un code C sur A est une partie non vide de l'espace de Hamming A^n, n est appelé longueur du code, les éléments de C sont appelés mots de code. Lorsque $A = \{0.1\}$ on parlera de code binaire.*

Définition 2 *On appelle distance minimale d'un code C sur A, l'entier :*

$$d = min\{d_H(x, y)/x, y \in C, x \neq y\} \tag{1.13}$$

Définition 3 *On appelle poids minimal d'un code C, l'entier :*

$$d = min\{\omega_H(x)/x \in C, x \neq 0\} \tag{1.14}$$

Distance de Hamming

Définition 4. *Soient un alphabet fini A de cardinal q contenant des symboles, et A^n l'ensemble des mots de longueur n sur A, $x = x(x_1, ..., x_n) \in A_n$ et $y = y(y_1, ..., y_n) \in A^n$. La distance de Hamming entre x et y, que l'on notera $d_H(x, y)$, est le nombre d'indices i tels que $x_i \neq y_i$.* Autrement écrit :

$$d_H(x, y) = \{|i|/0 \leq i \leq n, x \neq y\} \tag{1.15}$$

On vérifie que d_H est bien une métrique et on appelle alors espace de Hamming sur A l'ensemble A^n muni de la métrique d_H. *le poids de Hamming* $\omega_H(x)$ d'un mot $x \in A^n$ est le nombre de ses coordonnées non nulles. Donc

$$\omega_H(x) = d_H(x, 0) \tag{1.16}$$

Où 0 est le mot de A^n ayant toutes ses coordonnées égales à l'élément neutre de A.

Exemple 1 Soient $A = \{0, 1\}$, $n = 8$, $x = (0, 1, 0, 1, 0, 0, 0, 1)$, $y(1, 1, 1, 1, 1, 1, 1, 1)$ et $z = (0, 0, 0, 0, 1, 1, 1, 1)$
$d_H(x, y) = 5$, $d_H(x, z) = 5$, $d_H(y, z) = 4$.

1.6.4 Les codes en blocs les plus courants

Les codes BCH

Les codes BCH sont des codes cycliques. Ils ont été découvert à la fin des années 50 et l'acronyme est composé des lettres des noms de ses inventeurs, Raj Bose, D. K. Ray-Chaudhuri et Alexis Hocquenghem. (BCH) [30], [31]. Ces codes sont construits à partir d'un polynôme qui est défini en spécifiant ses zéros (ses racines). Ces codes définissent une méthode systématique pour construire des codes cycliques capables de corriger un nombre T d'erreurs arbitrairement fixé dans un bloc de N éléments binaires. Les codes BCH font appel à la théorie des corps de Galois. Le lecteur pourra se référer à Cohen et al. (1992) [32] pour plus de détails. Les performances de décodage de ces codes sont directement liées à la distance de Hamming du code en bloc utilisé. Plus cette distance est grande, meilleures sont les performances de décodage. Les codes BCH sont utilisés dans des applications telles que pour les communications satellitaires [33], *Solid-State Drive*

(SSD) [34].

Les codes Reed-Solomon

Les codes *Reed-Solomon* (RS) ont été inventés par Irving S. Reed et Gustave Solomon en 1960 [35]. Ces codes appartiennent à la classe des codes correcteurs d'erreurs cycliques non-binaires. Ils sont basés sur des polynômes générateurs qui font appel encore une fois à la théorie des corps de Galois [32]. Des symboles sont utilisés à la place d'éléments binaires. Ces codes sont donc capables de détecter et corriger plusieurs erreurs symboles. En ajoutant T symboles de vérification, un code RS peut détecter jusqu'à T symboles erronés et peut corriger jusqu'à $[T/2]$ symboles. Le choix de T est à la discrétion du créateur du code. Les codes RS sont utilisés dans beaucoup d'applications courantes telles que les *Compact Disc* (CD) [36], *Digital Versatile Disc* (DVD), disques Blu-ray, codes barres à 2 dimensions [37], etc.

Les Codes Polaires

Les Codes Polaires sont des codes en blocs linéaires. Leur invention est récente et proposée par Arıkan en 2008 dans [38]. Ce sont les seuls codes pour lesquels on peut démontrer mathématiquement qu'ils atteignent la limite de Shannon pour un code de taille infinie. De plus, ils ont une faible complexité de codage et de décodage en utilisant un algorithme particulier appelé *Successive Cancellation* - Annulation Successive (SC) [39] [40].

Les codes LDPC

Les codes *Low Density Parity Check* (LDPC) ont été découvert dans les années 60 par Gallager en 1962 [6]. Ces codes demandent une grande complexité calculatoire qui n'était pas disponible au moment de leur découverte. Les codes LDPC ont été négligés jusqu'aux années 90. Ils présentent un certain nombre d'avantages et leurs nombreux degrés de liberté rendent facile leur optimisation et adaptation à des contextes applicatifs très différents. La construction du code ainsi que le codage et le décodage sont bien détaillé dans le chapitre suivant.

1.7 Conclusion

Dans ce premier chapitre nous avons présenté le modèle classique d'une chaine de communications numériques. Cette dernière est composée de différents blocs. Un message passe tout d'abord par le codage source, puis le codage canal, ensuite la modulation numérique avant d'être transmis à travers le canal de transmission. Lors de la réception ce sont les fonctions inverses qui sont appliquées afin de retrouver le message original.

Des détails sur les trois canaux de transmission et les types de modulation ainsi qu'une description du théorème de Shannon ont été introduit.

Dans la dernière partie de ce chapitre une introduction général et des définitions sur les codes correcteur d'erreurs ont été définie, plus particulièrement la famille des codes en blocs linéaire dont les codes LDPC font partie, avec des exemples de codes en blocs existants.

Le prochain chapitre exposera un état de l'art détaillé sur la construction du code, le codage et le décodage du code LDPC.

Chapitre 2

Les Codes LDPC et algorithmes de décodage itératif

2.1 Introduction

Ce chapitre introduit des notions de base sur les codes LDPC. Tout d'abord une introduction détaillée sur les codes LDPC sera présentée. Une deuxième partie de ce chapitre présentera des méthodes de construction du code LDPC. Dans la troisième partie, l'opération de codage sera décrite. La quatrième partie de ce chapitre a pour rôle de décrire des algorithmes de décodage du code LDPC dont nous avons besoin dans les prochains chapitres.

2.2 Codes LDPC

2.2.1 Bref historique

Les codes LDPC, introduits par Gallager en 1962 [27] [6] sont des codes correcteurs d'erreur pseudo-aléatoires basées sur des matrices à vérification de parité de faible densité. Ces codes sont également appelés codes de Gallager. Hormis quelques exceptions comme les travaux de Zyablov et Pinsker [41] en 1975 et de Tanner [42] en 1981, ces codes n'ont eu qu'un faible impact dans la communauté de la théorie du codage et ce, à cause de leur complexité de codage et des moyens matériels nécessaires pour les décodeurs. Cette parenthèse durera jusqu'à l'introduction des turbo codes par Berrou, Glavieux et

Thitimajshima [5] en 1993.

Du fait de la dynamique engendrée par ces travaux, Mackay et Neal [7] en 1995 et Spielman et Sipser [8] en 1996 ont redécouvert les codes LDPC qui ont fait l'objet par la suite de nombreux travaux de recherches. Avant cela, Tanner dans [42] a étendu les codes de Gallager en remplaçant les vérifications de parité par des contraintes plus générales ce qui a permis d'obtenir une classe de codes couramment appelés codes de Tanner. Il proposa, alors, une extension de l'algorithme du décodage itératif introduit par Gallager en utilisant des décodeurs indépendants pour chaque contrainte. Wiberg [43], s'appuyant sur les travaux de Tanner, développa des algorithmes du décodage générique pour les codes de Tanner : *Min Sum* (MS) et *Sum-Product* (SP). Il montra, en particulier, que l'algorithme de Viterbi et l'algorithme de Tanner B sont des cas particuliers de l'algorithme MS. De même, il montra que l'algorithme Forward-backward et l'algorithme de Gallager ne sont que des cas particuliers de l'algorithme SP.

Mackay dans [7] et [16] a tout d'abord montré que l'algorithme du décodage développé par Gallager s'apparentait à l'algorithme de propagation de croyance de Pearl [15]. Davey et Mackay ont ensuite introduit une version non binaire des codes LDPC avec lesquels ils obtiennent de meilleures performances qu'avec les codes binaires [16]. Luby et al. [19] ont quant à eux introduit les matrices de vérification de parité avec une distribution du nombre de "1" sur les lignes et/ou sur les colonnes de la matrice de parité non-uniforme. Ces codes, appelés codes LDPC irréguliers, se sont montrés sous certaines conditions que nous développerons dans ce chapitre, plus performants que les codes réguliers. Davey et al. [12] ont, alors, présenté des codes de Gallager irréguliers et non binaires, avec lesquels ils ont obtenu de meilleurs performances que les meilleurs turbo codes connus.

Sous certaines conditions que nous discuterons également dans ce chapitre, les codes LDPC exhibent un phénomène de seuil. Ce seuil, obtenu pour une taille de bloc infinie, représente le plus faible rapport signal à bruit E_b/N_0 au-dessus duquel la probabilité d'erreur bloc est strictement nulle. Par conséquent, le seuil d'un code LDPC pourra représenter une mesure pertinente de qualité en considérant son écart par rapport à la capacité de Shannon. Ce phénomène a tout d'abord été observé par Gallager dans [6] pour un canal binaire symétrique lorsqu'il introduisit les codes réguliers et a ensuite été constaté par Luby et al. dans [19] pour des codes irréguliers. Cette observation a ensuite été généralisée pour un grand nombre de canaux à entrée binaire, incluant le canal binaire a effacement,

le canal binaire symétrique, le canal de Laplace, le canal à bruit additif blanc gaussien (BABG) [20].

La contribution théorique la plus significative depuis que Mackay prouva que les codes LDPC peuvent approcher de la capacité de Shannon avec un décodage optimal, vient de Richardson, Urbanke et al. [20] [11]. En effet, ces auteurs ont démontré que le comportement asymptotique moyen du décodage des codes LDPC était fonction des paramètres définissant le code. En utilisant cette méthode, appelée Évolution de Densité (*Density Evolution*), il est possible d'évaluer le seuil d'un code LDPC, et ainsi d'optimiser les valeurs des paramètres de manière à minimiser ce seuil. Notons que cette approche a permis d'obtenir sur le canal BABG, des codes présentant des performances très proches de la limite de Shannon (pour un rendement $R = 1/2$, le meilleur code LDPC exhibe un seuil à $0.0045dB$ de cette limite). Cette capacité de Shannon a d'ailleurs été atteinte pour le canal binaire à effacement [21].

2.2.2 Définitions

Un code LDPC de paramètres (N, d_v, d_c) est un code bloc linéaire binaire de longueur N défini par une matrice de vérification de parité H, creuse, de dimension $(M \times N)$

avec

$$M = \frac{d_v}{d_c} N \tag{2.1}$$

.

Les mots de code x consistent en une séquence de N bits satisfaisant un ensemble de M équations de contrôles de parité. Ceci se traduit par la relation suivante appelée équation de contrôle de parité :

$$H \times x^T = 0, \tag{2.2}$$

Le nombre de bits d'information est $K = N - M$ et le rendement R du code est $R \geq 1 - d_v/d_c$ L'inégalité venant du fait que le rang de la matrice H n'est pas nécessairement plein. La matrice H est composée de d_v élément "1" par colonne, d_c élément "1" par ligne et de zéros pour tous les autres éléments. Par conséquent, chaque élément du mot de code participe à d_v équations de contrôle de parité et chacune de ses équations est composée

de d_c bits.

Ci-dessous, nous avons représenté la matrice de contrôle de parité d'un code LDPC $(N = 9, d_v = 2, d_c = 3)$. Son rendement est $R = 1/3$ et son nombre de bits d'information est $K = 3$.

$$\text{H}= \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Les équations de parité associées à cette matrice et à un mot de code $x = [x_0, ..., x_7]$ sont :

$$X_1 + X_5 + X_7 = 0,$$

$$X_2 + X_3 + X_9 = 0,$$

$$X_2 + X_4 + X_7 = 0,$$

$$X_5 + X_6 + X_9 = 0,$$

$$X_3 + X_6 + X_8 = 0,$$

$$X_1 + X_4 + X_8 = 0,$$

Un code LDPC peut être également représenté graphiquement, cette représentation est appelée soit graphe de Tanner [42] ou soit, plus récemment, graphe factoriel [17]. Un graphe factoriel est un graphe bipartite contenant deux types de nœuds : les nœuds de donnée et les nœuds contrôle. Ces deux types de nœuds sont reliés entre eux par des branches. Dans le cas d'un code LDPC, les nœuds de données représentent le mot de code et les nœuds de contrôle, représentent les vérifications de parité. Dans la suite, les nœuds de contrôles sont appelés aussi nœuds fonctionnel ou nœuds de parités. Le $i^{\text{ème}}$ nœud de donnée et le $j^{\text{ème}}$ nœud de contrôle sont connectés par une branche si et seulement si $h_{i,j}$ est non nul. La figure 2.1 représente le graphe factoriel du code LDPC défini par la matrice H définie en dessus.

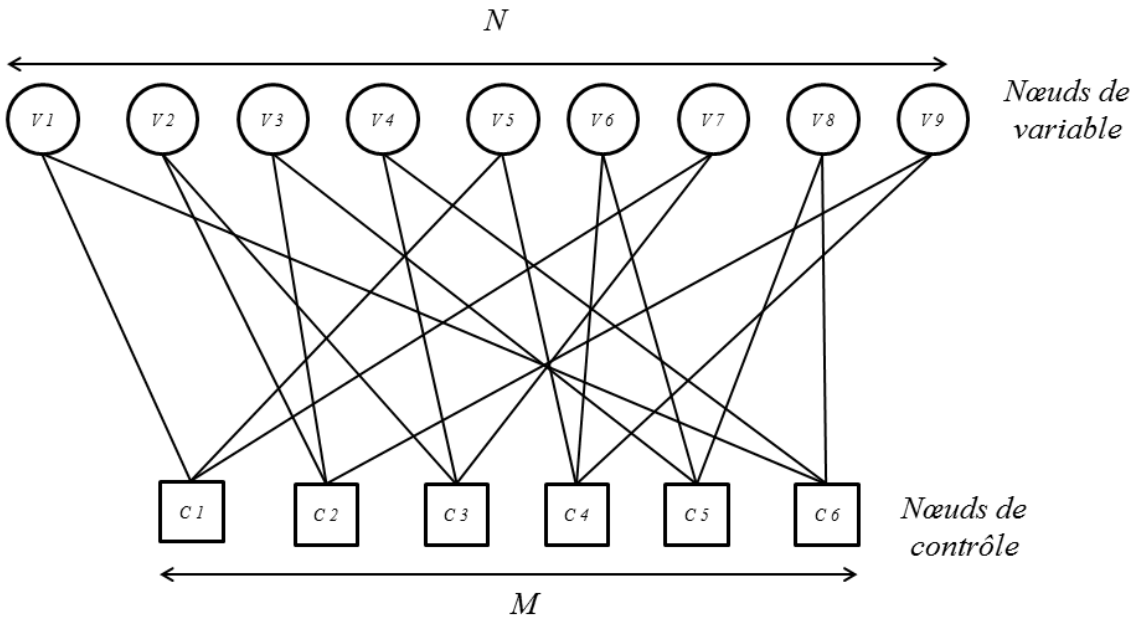


FIGURE 2.1 – Graphe factoriel d'un code LDPC régulier ($N = 9, d_v = 2, d_c = 3$) de rendement $R = 1/3$.

Le graphe factoriel d'un code LDPC est donc une représentation graphique très didactique de la structure de ces codes. Il est également important de noter que les représentations graphiques ont permis le développement et la mise en œuvre des algorithmes du décodage [17]. Nous détaillerons ce point dans la prochaine section.

2.2.3 Codes LDPC systématiques

L'introduction des codes LDPC systématiques par Mackay [44] a été principalement motivée par la réduction de la complexité de codage. Nous verrons dans cette section qu'en plus de rendre concrète l'implantation des codes LDPC, cette structure permet de coder en conservant de bout en bout la structure du code LDPC définie par la matrice de parité H.

Si G est la matrice génératrice du code LDPC à coder, On procède de la manière suivante pour la déterminer :

Transformation de la matrice H sous la forme systématique

$$H' = \begin{bmatrix} I & A \end{bmatrix} \tag{2.3}$$

où I représente la matrice identité

Calcul de

$$G = \begin{bmatrix} A^T & I \end{bmatrix} \tag{2.4}$$

Cette transformation sous la forme systématique se fait à l'aide d'éliminations suivant le pivot de Gauss. A la fin de la transformation, la matrice H d'origine doit être soumise au réarrangement des colonnes suivant les opérations effectuées durant l'élimination de Gauss.

Deux cas possibles peuvent se présenter à la fin de la transformation :

— H' ne comporte pas de lignes dépendantes : dans ce cas la transformation aboutit à une matrice identité de taille $m \times m$. Par conséquent A est de taille $(m \times N - m)$ et G de taille $((N - m) \times N)$,

— H' comporte au moins 2 lignes dépendantes : dans ce cas, la transformation fait apparaître des lignes tout à 0. Ces lignes sont supprimées et la matrice transformée comporte moins de lignes. Soit L le nombre de lignes dépendantes de H', la transformation aboutit à I de taille $((m - L) \times (m - L))$. A est alors de taille $((m - L) \times N - (m - L))$, et G de taille $(N - (m - L) \times N)$.

Dans tous les cas, considérons que G est de dimension $K \times N$, en posant $K = N - (m - L)$. Cela signifie que pour un mot à coder U de K bits, on obtient un mot codé $C_{[1\times N]} = U_{[1\times K]} \cdot G_{[K\times N]}$ de taille N. La taille des paquets de bits à émettre va donc dépendre de la matrice H utilisée et de la dépendance de ses lignes. Une fois que $H(m\times N)$ et $G(K \times N)$ sont construites, on peut vérifier que, par construction :

$$G \cdot H^T = 0 \tag{2.5}$$

L'équation (2.5) est l'équation de parité sous forme matricielle. Elle est équivalente à l'ensemble des K équations de parité qui caractérisent le code LDPC de taille $(N \times K)$. Dans ce cas, si un mot codé $C = U \cdot G$ est reçu sans erreur, on a $C \cdot H^T = 0$, puisque $C \cdot H^T = (U \cdot G) \cdot H^T = U \cdot (G \cdot H^T)$ et $G \cdot H^T = 0$.

Ainsi, la matrice H est utilisée pour vérifier la parité $v.H^T = 0$ où v est le mot reçu. Si le résultat est non nul, une erreur est détectée et doit être corrigée. Comme le décodage est basé sur la structure de la matrice de parité, la construction de cette dernière est une étape très importante dans la génération d'un code.

Exemple

Soit le code linéaire définit par la matrice génératrice suivante :

$$\text{G}= \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Un code systématique équivalent à ce code, peut-être obtenu de la manière suivante :

1. Remplacer ligne 2 par la somme de la ligne 1 et ligne 2 ($l_2 \leftarrow l_1 + l_2$)
2. Remplacer ligne 1 par la somme de la ligne 1 et ligne 2 ($l_2 \leftarrow l_1 + l_2$)
3. Remplacer ligne 3 par la somme de la ligne 3 et ligne 2 ($l_3 \leftarrow l_3 + l_2$)
4. Remplacer ligne 1 par la somme de la ligne 1 et ligne 3 ($l_1 \leftarrow l_1 + l_3$)

On obtient la matrice génératrice suivante :

$$G' = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Et la matrice de contrôle :

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

2.2.4 Les codes LDPC réguliers

En utilisant la représentation de Tanner, on définit les nœuds de variable, les nœuds qui représentent les bits (information et redondance) du mot de code. On définit également les nœuds de parité, ceux représentant la contrainte placée sur les nœuds de variable auxquels il est connecté. Les premiers codes LDPC proposés par Gallager dans [6] ont une structure régulière. Les nœuds de variable et les nœuds de parité ont des degrés de connexion d_v (respectivement d_c) constants (voir figure 2.2). Toutes les colonnes ont alors le même nombre de positions non nulles. Cette condition est valable aussi pour les lignes. Le nombre total de positions non nulles dans la matrice est égal au nombre d'arêtes du graphe. On en dérive :

$$N.d_v = (N-K).d_c \qquad \Rightarrow \qquad \frac{K}{N} = R = 1 - \frac{d_v}{d_c} \tag{2.6}$$

Avec le même couple (d_v, d_c), plusieurs codes réguliers peuvent être définis selon le choix des positions non nulles dans la matrice H.

2.2.5 Les codes LDPC irréguliers

Au lieu d'avoir des degrés de connexion fixes, les nœuds du graphe d'un code LDPC peuvent avoir des degrés de connexion différents, d'où l'appellation de "codes irréguliers". Dans l'article [19], Luby et al. , donnent une extension de l'étude de Gallager sur des graphes irréguliers.

Ils montrent que les performances des codes irréguliers sont meilleures et donnent une première approche de construction de codes irréguliers. Cette approche a été développée plus tard pour obtenir des performances proches de la limite de capacité de Shannon. La structure du code est définie à l'aide des deux polynômes $\lambda(x)$ et $\rho(x)$:

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1} \qquad 0 \leq \lambda_i \leq 1 \qquad \sum_{i=2}^{d_v} \lambda_i = 1 \tag{2.7}$$

$$\rho(x) = \sum_{j=2}^{d_c} \rho_j x^{j-1} \qquad 0 \leq \rho_i \leq 1 \qquad \sum_{i=2}^{d_c} \rho_i = 1 \tag{2.8}$$

λ_i et ρ_i sont les proportions des branches du graphe connectées à des nœuds de variable (respectivement de parité) dont le degré de connexion est égal à i. Comme décrit dans la référence [45] la figure 2.2 montre deux graphes de codes LDPC régulier et irrégulier.

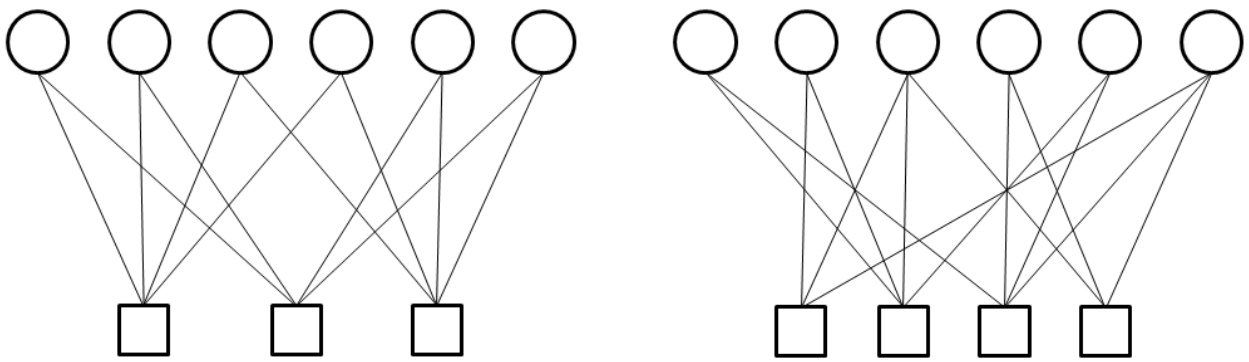


FIGURE 2.2 – Graphes de Tanner de codes LDPC régulier $(d_v = 2; d_c = 4)$ et irrégulier $(\lambda(x) = (8/14) \cdot x + (6/14) \cdot x^2)$; $(\rho(x) = (6/14) \cdot x^2 + (8/14) \cdot x^3)$.

2.2.6 Mesure des performances d'un code

Les performances de décodage permettent de caractériser la capacité d'un code correcteur d'erreurs à corriger des erreurs. Cette capacité est mesurée par la probabilité d'erreur binaire notée P_{eb}. Elle peut être représentée en pratique par le *Taux d'Erreur Binaire* (TEB) en anglais (*Bit Error Rat* (BER)). Soit N_e le nombre de bits erronés pour N bits transmis, alors le TEB est défini par le rapport N_e/N. Si la transmission consiste à transmettre des trames de longueurs finies, alors il est possible de définir le *Taux d'Erreur Trame* (TET)en anglais (*Frame Error Rat* (FER)). Soit T_e le nombre de trames contenant au moins 1 bit erroné parmi T trame transmises, alors le TET est défini par le rapport T_e/T. L'ajout de redondance dans le message à transmettre entraîne une perte d'efficacité du système. En effet les bits de redondance introduits ne véhiculent pas de l'information utile. Différents systèmes sont comparés en analysant la probabilité d'erreur bit ou paquet en fonction de l'énergie transmise par bit utile. On appelle gain de codage l'écart d'énergie par bit utile entre deux systèmes pour un taux d'erreur donné. Dans le cas de l'utilisation de techniques de codage avancées, l'évolution de la performance du code peut se diviser en trois régions comme illustré sur la figure 2.3. La première région correspond à un comportement où le décodage ne converge pas.

Dans certain cas, le décodage dégrade les performances par rapport à un système non codé : on parle de région de non convergence. A partir d'un certain rapport signal à bruit, appelé seuil de convergence, le décodage rentre dans une phase où la probabilité d'erreur diminue très rapidement avec le rapport signal sur bruit : on parle de la région du "*Waterfall*". Enfin, il existe une région où la probabilité d'erreur diminue de manière moins rapide que la région du waterfall. Ce comportement est spécifique de la région du "plancher d'erreur" ou "*error floor*".

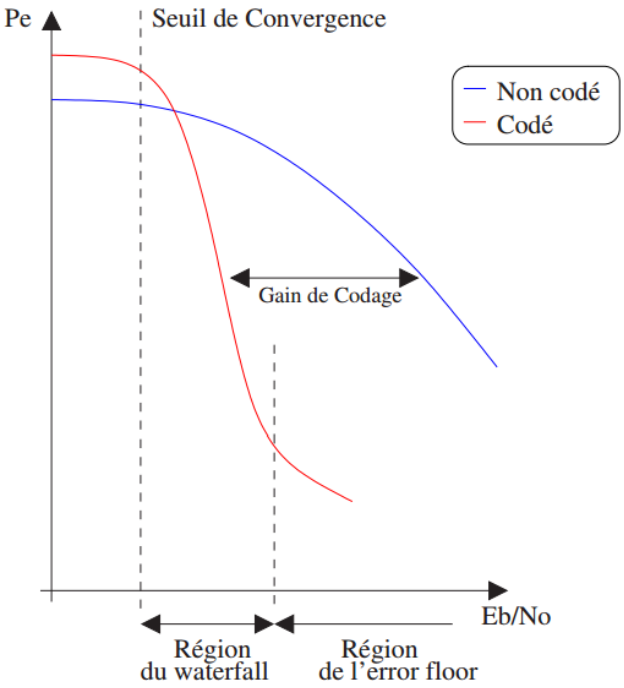


FIGURE 2.3 – Illustration des trois régions caractérisant les performances d'un système de codage.

2.3 Construction des codes LDPC

Construire un code LDPC revient à définir la position de tous les éléments non nuls dans la matrice de contrôle de parité. La principale difficulté dans la construction des codes LDPC réside dans le choix des paramètres définissant le code. Ainsi, pour une taille et un rendement donné, il faut choisir dans un premier temps le profil d'irrégularité des nœuds de données et des nœuds de contrôle. Des outils puissants, qui sont détaillés dans [46], permettent sous certaines hypothèses (souvent des cas asymptotiques) de fixer un profil d'irrégularité minimisant le seuil de convergence. Ces outils sont de la famille des algorithmes d'évolution de densité (*Density Evolution*) dont la base sont les travaux de Ten Brink développés initialement dans le cadre de l'analyse des Turbo-codes parallèles [47] (EXIT chart). Dans le cas des codes LDPC, l'évolution de densité avec approximation gaussienne a été réalisée par Chung, Richardson et Urbanke [48]. Cependant, ces résultats obtenus dans le cas asymptotique sont à utiliser avec précautions dans le cas de codes de longueur finie. Ces méthodes constituent néanmoins un bon point de départ pour la détermination des profils d'irrégularité.

Une fois le profil d'irrégularité fixé, la position de chaque élément non nul doit être déterminée selon la structure de code choisie. Un grand nombre d'algorithmes existe dans la littérature. La plupart des algorithmes essayent de construire un code ayant une bonne

distribution des cycles. Un algorithme particulièrement intéressant est l'algorithme *Progressive Edge Growing* (PEG) [49] et ses versions dérivées. Cet algorithme travaille sur la matrice de contrôle de parité colonne par colonne. A chaque nouvelle colonne, l'algorithme place les éléments non nuls, suivant la distribution des nœuds de données spécifiée, de manière à maximiser localement le cycle du nœud de données considéré, sous la contrainte d'une distribution des nœuds de contrôle fixée.

Alors que dans le cas asymptotique les codes LDPC construits aléatoirement sont très bons, dans le cas de taille finie, les constructions déterministes semblent fournir les meilleurs candidats. Ainsi les constructions de type *Repeat Accumulate* ou par expansion offrent de très bonnes performances. Une brève description de ces deux familles est proposée à titre d'exemple. De nombreuses autres familles existent dans la littérature ne sont pas adressées dans ce manuscrit.

2.3.1 Construction géométrique des codes LDPC

Les codes LDPC peuvent être construits de manière algébrique à partir de points et de lignes de la géométrie finie, comme la géométrie Euclidienne et la géométrie projective définies sur des champs finis [50]. Une géométrie finie est formée de points et lignes, qui ont les propriétés suivantes :

— chaque ligne a ρ points.

— chaque point appartient à γ lignes.

— deux points sont connectés par juste une ligne.

— deux lignes sont soit disjointes, soit leur intersection est un seul point.

Pour chaque type de géométrie, on peut construire des codes LDPC de type 1 et de type 2, qui sont vraiment connexes et leurs graphes sont conjugués entre eux : les nœuds des bits codés d'un graphe sont les nœuds de contrôle pour l'autre graphe. Pour le type 1, on peut former une matrice de parité H dont les lignes sont les vecteurs d'incidence des lignes existantes dans la géométrie finie et les colonnes sont les points. Le graphe correspondant à cette matrice n'a aucun cycle de longueur 4. Pour un code LDPC de type 2, la matrice de parité H est la transposée de la matrice correspondante de type 1 : les lignes sont les vecteurs d'incidence des points et les colonnes sont les lignes de la géométrie finie. Les graphes correspondants à ces matrices n'ont aucun cycle de longueur 4.

Les codes EG-LDPC de type 1 et 2 (basés sur la géométrie euclidienne) et les codes PG-LDPC de type 1 et 2 (basés sur la géométrie projective) ont des performances d'erreur presque identiques [50]. Dans la construction des codes EG-LDPC, on peut éliminer le point d'origine de la géométrie et toutes les lignes qui passent par l'origine. Ainsi, on met le code EG-LDPC de type 1 dans une forme cyclique et le code EG-LDPC de type 2 dans une forme quasi-cyclique. Ceci simplifie le codage. On doit garder en mémoire juste la première ligne de H et les autres lignes seront ajoutées à l'aide d'un registre à décalage linéaire. La différence entre les 4 types de codes (EG-LDPC de type 1 et 2, et PG-LDPC de type 1 et 2) est comment trouver les positions des valeurs de 1 dans la première ligne de la matrice de parité et les différents paramètres des codes obtenus. Les formules pour calculer ces paramètres sont données en [50].

Ces codes peuvent être raccourcis par élimination des colonnes de la matrice de parité [50]. Pour les codes EG-LDPC de type 1, les colonnes éliminées correspondent aux points d'un ensemble de paquets parallèles qui ne passent pas par l'origine de la géométrie finie. On obtient une nouvelle matrice irrégulière dans laquelle les colonnes ont le même poids, mais les lignes ont des poids inférieurs. Son noyau donne un code LDPC irrégulier plus court avec la même distance minimale que le code initial. Pour les codes EG-LDPC de type 2, il faut mettre la matrice de parité dans une forme circulaire et on élimine les colonnes correspondant à un ensemble de lignes. On obtient un code quasi-cyclique plus court. On doit mentionner que plus on réduit la matrice de parité, plus on obtient une meilleure performance d'erreur du code.

2.3.2 Construction des codes LDPC de Gallager

Pour construire la matrice de parité H d'un code LDPC de Gallager [27], il faut d'abord construire une sous-matrice H_i ayant un poids des colonnes égal à 1 et un poids de lignes ρ. Ensuite on doit trouver de permutations des colonnes de cette sous-matrice pour former les autres sous-matrices avec lesquelles on forme la matrice de Gallager de manière suivante :

$$H = \begin{pmatrix} H_1 \\ H_2 \\ . \\ . \\ . \\ H_\gamma \end{pmatrix}$$

Lorsqu'on choisit les permutations des colonnes des sous-matrices, on doit garder une bonne distance minimale de la matrice de parité H et il faut éviter les cycles courts dans son graphe de Tanner. Gallager ne donne aucune méthode de construction pour ce type de codes. Dans [50] [51] [52] on retrouve une telle méthode de construction basée sur la structure parallèle des lignes dans la géométrie Euclidienne $EG(m, 2^s)$ sur le corps de Galois $CG(2^s)$. Il y a beaucoup de manières possibles pour choisir les paquets des lignes parallèles et selon ce choix on obtient des codes LDPC EG-Gallager de dimensions différentes. Pour chacune des valeurs de m et s, on obtient un code LDPC de Gallager de longueur 2^{ms} pour différents valeurs de γ. Ces codes ont différents rendements et distances minimales. Si on compare la performance d'erreurs d'un tel code LDPC avec d'autres codes LDPC obtenus par ordinateur, qui ont le même rendement et presque la même longueur, on peut observer que le code EG-LDPC est meilleur.

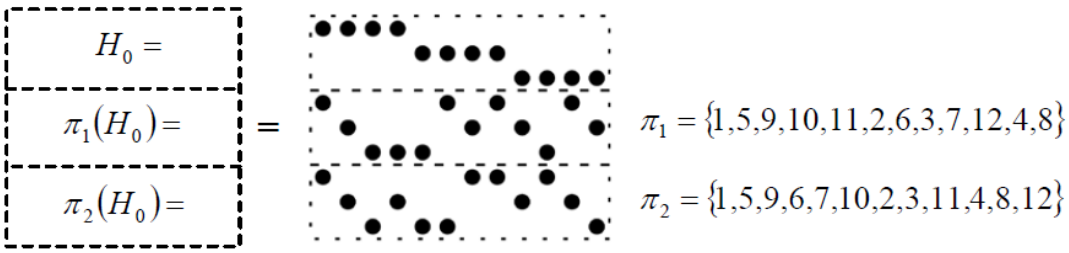


FIGURE 2.4 – Exemple de construction de la matrice de parité des codes LDPC de Gallager.

La figure 2.4 montre un exemple de construction de la matrice de parité H de Gallager, en se basant sur la sous-matrice H_0. Puis, il y a i matrices $\pi_i(H_0)$ sont empilés verticalement sur H_0, où $\pi_i(H_0)$ désigne la permutation des colonnes de H_0.

On ne peut pas construire de codes LDPC dans la forme de Gallager basés sur la géométrie projective, parce que les lignes dans une géométrie projective n'ont pas la même structure parallèle simple que les lignes de la géométrie Euclidienne.

2.3.3 Construction par expansion

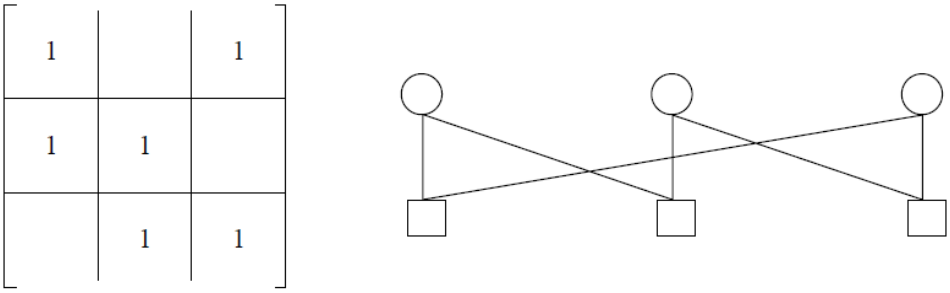
La construction de code LDPC par expansion consiste à définir un graphe de base associé à une matrice de contrôle de parité de base. L'expansion de cette matrice est réalisée en remplaçant chaque élément non nul de la matrice par une matrice d'expansion. Ce concept a initialement été proposé par Gallager [6]. Plus particulièrement, de nombreux travaux ont porté sur le cas où les matrices d'expansion sont des matrices identité permutées [53] [54] [55]. A titre d'exemple, la structure de matrice de contrôle de parité proposée dans [54] est de la forme suivante :

$$\mathrm{H} = \begin{pmatrix} I_0 & I_0 & \cdots & I_0 \\ I_0 & I_{p_{1,1}} & \cdots & I_{p_{1,L-1}} \\ \vdots & \vdots & \ddots & \vdots \\ I_0 & I_{p_{J-1,1}} & \cdots & I_{p_{J-1,L-1}} \end{pmatrix}$$

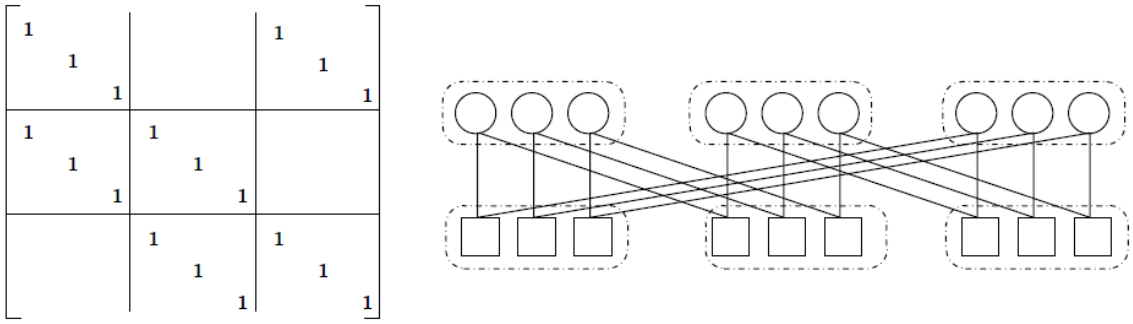
Où $I_x$représente une matrice identité de taille $z \times z$ circulairement permutée de x positions vers la droite. Le code ainsi défini est un code régulier quasi-cyclique de taille $N = L \times z$ et de rendement $R = 1 - J/L$. Les coefficients de permutation $p_{i,j}$,de chaque matrice identité peuvent être déterminés aléatoirement ou suivant des lois [54]. En particulier pour cette structure, il a été démontré que la longueur du plus petit cycle est bornée par 12. Des conditions nécessaires sont proposées dans [54] pour atteindre cette borne. Précédemment, Tanner avait proposé, dans une forme quasi-cyclique, des codes *Repeat Accumulate* [56]. A partir d'une structure bi-diagonale, chaque 1 d'une matrice de base est remplacé par une matrice identité permutée. Ce concept a été par la suite repris et généralisé : on parle alors de protographe [57]. Un code LDPC dérivé d'un protographe est un code dont le graphe est une copie permutée d'un graphe de base figure 2.5. Ce type de code peut être particulièrement intéressant pour une réalisation matérielle. Ce type de représentation permet de généraliser un bon nombre de constructions de code existant dans la littérature et notamment les constructions quasi-cycliques.

2.3.4 Construction des codes LDPC quasi-cycliques

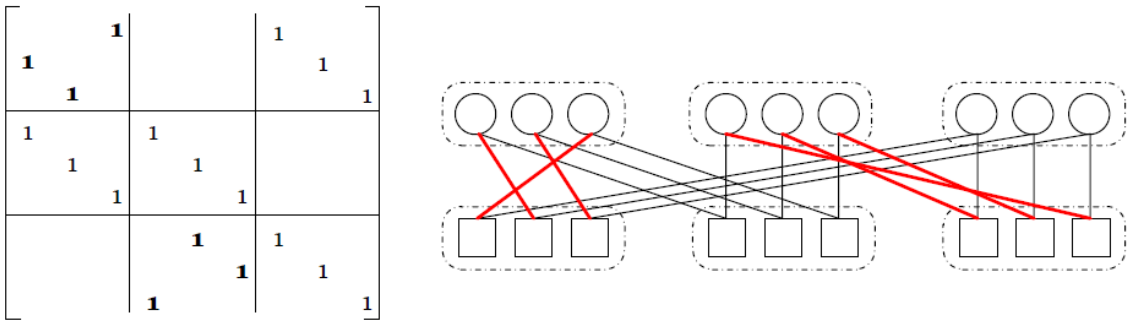
Cette méthode consiste en la décomposition d'une matrice carrée, régulière et circulaire en plusieurs matrices circulaires de mêmes dimensions, mais avec des poids différents [50]. On obtient ces nouvelles matrices à partir de chaque colonne de la matrice de parité



(a) Exemple d'un protographe.



(b) Le protographe a été répété trois fois.



(c) Le graphe dérivé est obtenu par permutation de certaines branches.

FIGURE 2.5 – Exemple de construction d'un code à partir d'un protographe et de matrices d'expansion de type identité permutées.

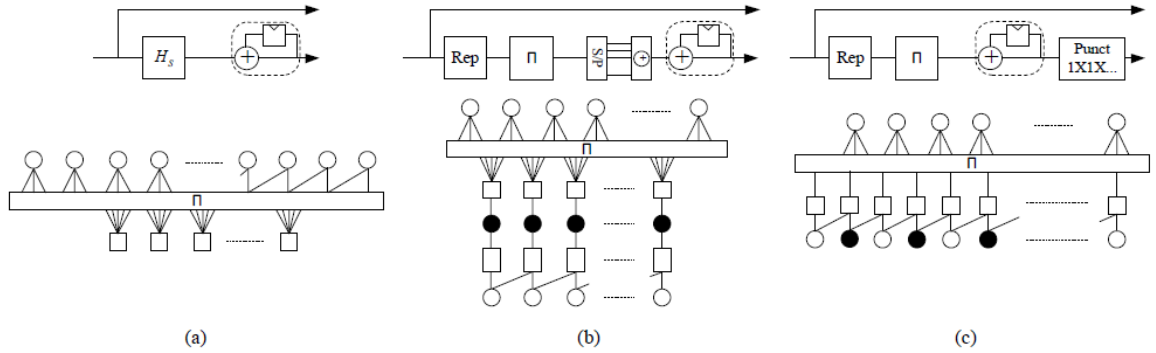


FIGURE 2.6 – Illustrations des différentes représentations d'un code de type *Repeat Accumulate*. Ces codes peuvent être vus comme des codes LDPC (a). Ils peuvent aussi être représentés par une concaténation d'un code de parité, alimenté par un organe de répétition (Rep) dont les sorties sont entrelacées (π), et d'un accumulateur (b). Enfin, ces codes peuvent être vus comme la concaténation série d'un répéteur, d'un accumulateur. Les sorties sont alors poinçonnées (Punct) pour obtenir le rendement codage choisi (c)

initiale, qui est décomposée en plusieurs colonnes de même longueur. Le poids de la colonne initiale est partagé parmi les différentes colonnes. A partir de chaque nouvelle colonne ainsi formée, on forme une matrice circulaire par permutations circulaires successives de la colonne en bas. La méthode présentée s'appelle la décomposition des colonnes d'une matrice de parité. De même, on peut décomposer la matrice initiale en descendants, en décomposant sa première ligne en plusieurs lignes et ensuite en faisant des permutations circulaires à la droite de chaque nouvelle ligne. Cette méthode s'appelle la décomposition des lignes. Si la matrice initiale est une matrice creuse, la matrice obtenue est aussi une matrice creuse de densité plus faible, qui donne un code LDPC quasi-cyclique dont le graphe de Tanner n'a pas de cycles de longueur 4.

Des matrices creuses circulaires peuvent être construites à partir des vecteurs d'incidence des lignes dans une géométrie Euclidienne ou projective. De plus, il n'existe pas deux lignes (ou deux colonnes) dans la même matrice ou dans différentes matrices circulaires ayant plus d'une valeur de 1 en commun. En conséquence, les codes LDPC quasi-cycliques peuvent être construits en décomposant une ou un groupe de ces matrices circulaires géométriques [58] [59].

2.3.5 Construction de type *Repeat Accumlate*

Cette famille de codes décrite brièvement dans la section précédente, consiste en une matrice de contrôle de parité divisée en deux, dont une partie est de type bi-diagonale. On rappelle que la matrice de contrôle de parité peut s'écrire :

$$H = \begin{bmatrix} H_s & H_p \end{bmatrix} \tag{2.9}$$

Dans le cas où la matrice H_p est strictement bi-diagonale et que la matrice H_s est régulière, on parle de codes de la famille *Repeat Accumulate* [60]. Ces codes sont des codes LDPC irréguliers avec une proportion de nœuds de degré 2 égale au nombre de bit de parité. Dans le cas où la matrice H_s est irrégulière on parle alors d'*Irregular Repeat Accumulate Code* [61]. De nombreux travaux ont porté sur l'étude de ces codes, et des performances comparables à celles des meilleurs codes LDPC irréguliers ont été obtenues [62]. Ils sont d'autant plus intéressants que le rendement de codage est fort [63]. Les premiers codes LDPC standardisés dans le cadre de la norme DVB-S2 appartiennent à cette catégorie de codes. Nous reviendrons par la suite plus précisément sur cette structure et ses particularités.

De nombreuses constructions ont été proposées pour la construction de la matrice H_s. Nous pouvons citer par exemple les structures de type $\pi - rotation$ [64] et dual $\pi - rotation$ [65] qui permettent une analyse, une construction et un encodage très simple. D'une manière générale, les codes de type Repeat Accumulate peuvent être représentés sous la forme illustrée dans la figure 2.6. Comme décrit précédemment, des constructions hybrides telles que représentées sur la figure 2.5 sont envisageables. Nous pouvons citer par exemple les codes de type *Accumulate* : *Repeat Accumulate codes* [66] [67] qui peuvent être vus comme des codes *Repeat Accumulate* précodés.

2.3.6 Construction déterministe

Les constructions aléatoires des codes LDPC n'ont pas trop de contraintes : ils peuvent s'adapter assez bien aux paramètres de la classe désirée. Le problème, c'est qu'ils ne garantissent pas que le *girth* soit assez petit. Alors, soit un post-traitement ou plus de contraintes sont ajoutées pour la conception aléatoire, ce qui donne parfois beaucoup de complexité. Afin de contourner le problème de *girth*, les constructions de types déterministes ont été développées. En outre les constructions explicites peuvent faciliter le codage, et peuvent être aussi plus facile à les implémenter. Ils semblent être plus efficaces que les précédentes constructions algébriques qui ont été basé sur les graphes d'expansion [68] [69] [70]. Dans [71], les auteurs ont conçu des codes LDPC de taux plus élevés

basé sur les systèmes de Steiner. Leur conclusion était que la distance minimale n'était pas assez élevée. L'inconvénient majeur pour les constructions déterministes des codes LDPC est qu'ils existent avec quelques combinaisons de paramètres. Ainsi, il peut être difficile de trouver celui qui correspond aux spécifications d'un système donné.

2.4 Codage des codes LDPC

Les codes LDPC ont la particularité d'être définis par leur matrice de contrôle de parité. Comme nous l'avons mentionné auparavant, du fait de leur complexité d'encodage et du décodage et des moyens matériels de l'époque, ces codes ont eu peu d'impact sur la communauté de la théorie du codage au moment de leur découverte. En effet, la manière triviale de déterminer le mot de code est d'utiliser la matrice génératrice G, facilement calculable à partir de la matrice de contrôle de parité H. Dans la majorité des cas, la matrice génératrice associée à un code dont la matrice de contrôle de parité est de faible densité est dense. Dans le cas où le code n'a pas de structure, la complexité de codage associé est alors importante. Pour réduire cette complexité, des approches que l'on peut classer dans deux grandes familles ont été développées. La première consiste à post-traiter la matrice de contrôle de parité de façon à introduire une forme facilement codable. La seconde est basée sur la construction d'une matrice de contrôle de parité contrainte, construite à l'origine pour faciliter l'encodage.

En ce qui concerne la première approche, les auteurs de [72] préconisent une transformation de la matrice de contrôle de parité par combinaisons linéaires de lignes et de colonnes en une autre matrice de contrôle de parité de forme semi-triangulaire.

La complexité d'encodage dépend alors d'un paramètre caractérisant l'écart entre la matrice semi-triangulaire et la matrice triangulaire. Une fois les bits intervenant dans la forme semi-triangulaire obtenue, les autres bits de redondance sont obtenus par substitution.

La seconde méthode consiste en la définition d'une structure de code contrainte. Une première construction très largement répandue, consiste à construire une matrice de contrôle de parité définie par :

$$H = \begin{bmatrix} H_s & H_p \end{bmatrix} \tag{2.10}$$

Le mot de code x est alors divisé en un mot d'information b et un mot de redondance r. la relation de parité définie par :

$$x \times H^t = 0 \tag{2.11}$$

$$\begin{bmatrix} H_s & H_p \end{bmatrix} \times \begin{bmatrix} b^t \\ r^t \end{bmatrix} = 0 \tag{2.12}$$

$$H_p r^t = H_s b^t \tag{2.13}$$

Pour simplifier les notations, nous appellerons vecteur de projection le vecteur v défini par :

$$\nu^t = H_s b^t \tag{2.14}$$

Il en résulte donc que l'ensemble des bits de parité peut se déduire de la façon suivante :

$$r^t = H_p^{-1} v^t \tag{2.15}$$

Cette relation montre que la première contrainte sur le code est l'existence de la matrice inverse H_p^{-1}. La matrice H_p peut-être de forme triangulaire permettant par simple substitution le calcul des bits de redondance. Plus particulièrement, les matrices H_p de type bi-diagonale sont intéressantes pour obtenir un codage simple. Dans le cas où la matrice est strictement bi-diagonale (cf. figure 2.7 (a)) le code LDPC (figure 2.6 (a)) peut-être vu indifféremment comme :

— un code de la famille des codes *Repeat Accumulate* [60] [61] (Figure 2.6 (c)).

— une concaténation série d'un code de parité et d'un code convolutif récursif (accumulateur) (figure 2.6 (b)).

— un code de la famille *self concatenated code* [73], dont le code de base est un accumulateur (figure 2.6 (c)).

La détermination des bits de redondance peut se faire, par la méthode de Gauss Jordan, en calculant dans un premier temps le vecteur de projection puis par une accumulation de ce vecteur : $r_k = r_{k-1} + v_k$ Une autre forme très intéressante de la matrice H_p de type

bi-diagonale est celle illustrée figure 2.7 (b). Ce type de structure a été retenu dans le cadre de la normalisation IEEE 802.16e (WIMAX Mobile) [74] et IEEE 802.11n (WiFi). Le premier bit de redondance est obtenu en sommant l'ensemble des bits du vecteur de projection v :

$$r^0 = \sum_i vi \tag{2.16}$$

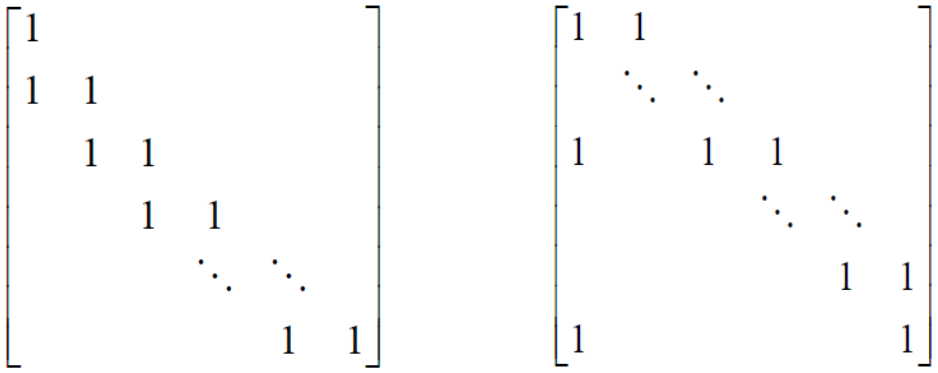


FIGURE 2.7 – Forme particulière de la matrice H_p normalisée dans des standards

L'ensemble des autres bits de redondance est alors calculé par une simple substitution. Cette forme de matrice peut être interprétée comme une forme circulaire de l'accumulateur. La présence de deux éléments non nuls dans une colonne provient du fait que, en pratique, l'accumulateur ne peut pas être circulaire sans une petite modification du code. Ce principe a été illustré par Berrou et al. dans le but d'améliorer la distance minimale d'un Turbo-code [75].

Une autre famille de codes permet un codage simple : ces codes font partie de la famille des codes Quasi-Cyclic (QC) [53]. Il a été démontré que pour certaines familles de codes QC-LDPC, le codage peut se faire de manière très simple avec des registres à décalages. De nombreux exemples sont décrits dans [76].

Nous avons vu que les méthodes d'encodage des codes LDPC sont fortement liées à la construction des codes. Ainsi chaque famille de codes aura une structure de codage particulière. Dans le paragraphe suivant, nous proposons d'établir un tour d'horizon sur les méthodes du décodage des codes LDPC.

2.5 Décodage du code LDPC

Par rapport aux autres types de codes, le décodage des codes LDPC ne pose pas autant de problèmes pour les chercheurs que leur construction. Le travail le plus difficile est de trouver les meilleures méthodes pour construire des codes LDPC efficaces. Un code LDPC peut être décodé par plusieurs méthodes, telles que :

1. décodage avec des décisions fermes
   - — décodage avec la logique majoritaire (MLG)
   - — décodage avec basculement de bit (BF)
2. décodage avec des décisions pondérées
   - — décodage basé sur la probabilité a posteriori (APP)
   - — décodage itératif basé sur la propagation de croyance en anglais "*Belief Propagation* (BP)" appelé aussi "*Sum Product* (SP)"
3. décodage avec décodage mixte (ferme et pondéré)
   - — décodage BF pondéré

La méthode MLG est la plus simple du point de vue de la complexité du circuit. La méthode BF demande un peu plus de complexité du circuit, mais elle donne des meilleures performances d'erreur que la méthode MLG. Les méthodes APP et SP donnent des meilleures performances d'erreur, mais nécessitent aussi une plus grande complexité du circuit. Le décodage BF pondéré représente un bon compromis entre les deux caractéristiques. La méthode SPA donne la meilleure performance d'erreur entre les 5 types de décodage.

2.5.1 Décodage MLG des codes LDPC

La méthode MLG à une seule étape [50] peut être appliquée au décodage des 4 types de codes LDPC.
On calcule les syndromes

$$s_j^{(l)} = e \times h_j^{(l)} = \sum_{i=0}^{n-1} e_i h_{i,j}^{(l)} \tag{2.17}$$

où $h_j^{(l)} (1 \leq j \leq \gamma)$ sont les γ lignes de H qui sont orthogonales sur le bit de la $l-$ème position. L'ensemble des sommes de contrôle $s_j^{(l)}$ sont orthogonales sur le bit d'erreur e_l

et on peut les utiliser pour l'estimation de e_l. Le bit d'erreur est bien corrigé si dans le vecteur d'erreur il y a moins de $\gamma/2$ erreurs.

2.5.2 Algorithme de décodage à basculement de bit (BF) des codes LDPC

Cette méthode est basée sur l'échange du nombre d'échecs de parité quand un bit de la séquence reçue est basculé. Il s'agit d'une méthode itérative de décodage [27] [6], [50]. Le décodeur calcule toutes les sommes de parité et ensuite il change chaque bit dans la séquence reçue s'il fait partie de plus de δ équations de parité échouées. La valeur de δ est un seuil fixé et dépend des paramètres du code (ρ, γ, d_{min}) et du SNR. À partir de la séquence modifiée, le décodeur recalcule les sommes de parité et le processus est répété jusqu'à ce que toutes les sommes de parité soient nulles.

Le nombre d'itérations du décodage est une variable aléatoire et dépend du rapport signal à bruit du canal. Pour des meilleures performances on peut utiliser des seuils adaptatifs δ et utiliser aussi une méthode hybride entre BF et MLG. Le décodage BF corrige beaucoup de séquences d'erreur qui possèdent plus de bits erronés que la capacité du code pour corriger les erreurs.

2.5.3 Décodages MLG et BF pondérés

Une meilleure performance du décodage est obtenue si on ajoute de l'information de fiabilité (qui demande une complexité du circuit additionnelle). Pour un canal AWGN, une mesure de fiabilité est l'amplitude du symbole reçu $|y_i|$ (plus l'amplitude est grande, plus la fiabilité de la décision ferme est grande). Pour un code LDPC caractérisé par sa matrice de parité H de dimension $J \times n$, on définit :

$$|y_j|^l_{min} = min|y_j| : 0 \leq i \leq n-1, h_{j,i} = 1 \tag{2.18}$$

et

$$E_l = \sum_{s_j^{(l)} \in S_l} (2s_j^l - 1)|y_j|^l_{min} \tag{2.19}$$

E_l est une somme orthogonale pondérée sur la $l - ème$ position du bit codé ;S_j est

l'ensemble de syndromes. La décision est : $e_l = 1$ si $E_l > 0$ et $e_l = 0$ si $E_l \leq 0$ qui modifie le décodage MLG à une étape en un décodage MLG pondéré. Cette relation de décision peut être utilisée aussi pour le décodage BF, donnant l'algorithme de décodage BF pondéré [50].

2.5.4 Décodage itératif par propagation de croyance

L'algorithme de propagation de croyances (*Belief Propagation*, BP) ou algorithme Somme-Produit (*Sum-Product* SP). Comme son nom lindique, cet algorithme propage le long des branches du graphe associé au code des messages, qui sont des probabilités ou des logarithmes des rapports de vraisemblance [77], [11], [16], [78], [12], le BP est une méthode itérative de décodage qui est très efficace pour les codes LDPC. Il converge relativement vite pour les codes EG-LDPC et PG-LDPC. La performance d'erreur dépend de quelques paramètres importants du code, tels que :

1. le périmètre du graphe de Tanner doit être suffisamment grand pour qu'il n'y ait pas des cycles courts, en particulier de longueur 4 ;
2. la distance minimale varie de manière inversement proportionnelle avec le périmètre. Ceci nous amène à un compromis car elle doit être suffisamment grande pour une bonne identification ;
3. les poids des lignes et des colonnes ;
4. le coefficient d'erreur - le nombre de mots-code de poids minimal ; pour des valeurs faibles, il donne une meilleure performance d'erreur si le rapport signal sur bruit est faible et les erreurs sont grandes.

On suppose que le message binaire $(x_1, x_2, \ldots, x_N)$ est transmis en utilisant la modulation BPSK. Puis la séquence est transmise sur un canal de bruit blanc additif gaussien (AWGN), et le symbole reçu est $(y_1, y_2, \ldots, y_N) \in \mathcal{Y}^N$. On définie :

- $\mathcal{H}$, graphe de tanner du code LDPC.
- N, nombre des nœuds de variable.
- M, nombre des nœuds de parité.
- $n \in 1, 2, ..., N$, les nœuds de variable de H.
- $m \in 1, 2, ..., M$, les nœuds de parité de H.

- $\mathcal{H}_C(n)$, Représente l'ensemble des nœuds de contrôle connecté au nœud de variable n.
- $\mathcal{H}_V(m)$, Représente l'ensemble des nœuds de variable connecté au nœud de contrôle m.
- $\mathcal{H}_C(n)\backslash m$, Représente tous les nœuds de contrôle connecté au nœud de variable n à l'exception le nœud m.
- $\mathcal{H}_V(m)\backslash n$, Représente tous les nœuds de variable connecté au nœud de contrôle m à l'exception le nœud n.

Par ailleurs, nous définissons les variables suivantes qui sont utilisées dans le présent document.

- λ_n : Représente l'information résultante du rapport de vraisemblance de symbole reçu y_n, pour un canal AWGN λ_n est définie par l'équation en dessous.

$$\lambda_n = log(\frac{P(x_i = 0|y_n)}{P(x_i = 1|y_n)}) = 2\frac{y_n}{\sigma^2} \tag{2.20}$$

  avec σ est la variance du bruit

- $\beta_{m,n}$: est définie comme le message provenant du nœud de contrôle vers le nœud de variable.
- $\alpha_{m,n}$: est le message provenant du nœud de variable vers le nœud de contrôle.

Initialisation

la première étape consiste à calculer la valeur du *log-likelihood ratio* pour les symboles reçu y_n

$$\lambda_n = log(\frac{P(x_n = 0|y_n)}{P(x_n = 1|y_n)}) = 2\frac{y_n}{\sigma^2} \tag{2.21}$$

Le processus itératif de l'algorithme Somme-Produits peut se décomposer en deux étapes, la mise à jour de l'ensemble des nœuds de parité et la mise à jour de l'ensemble des nœuds de variable.

Mise à jour des nœuds de contrôle CN

A la i-ème itération figure 2.8, chaque nœud de parité C_m utilise les messages $\alpha_{m,n'}$ reçus de son voisinage V_n pour calculer des messages $\beta_{m,n}$ adressés aux nœuds de variable l'équation 3.16. Notons que le calcul d'un message $\beta_{m,n}$ ne prend pas en compte le message fourni préalablement par n, $n' \in \mathcal{H}_V(m) \backslash n$.

$$\beta_{m,n} = \left(\prod_{n' \in \mathcal{H}_V(m) \backslash n} sgn\left(\alpha_{m,n'}\right) \right) \times \phi \left(\sum_{n' \in \mathcal{H}_V(m) \backslash n} \phi\left(|\alpha_{m,n'}|\right) \right) \tag{2.22}$$

Avec la fonction non linéaire

$$\phi(x) = -log\left(tanh\left(\frac{|x|}{2}\right)\right) \tag{2.23}$$

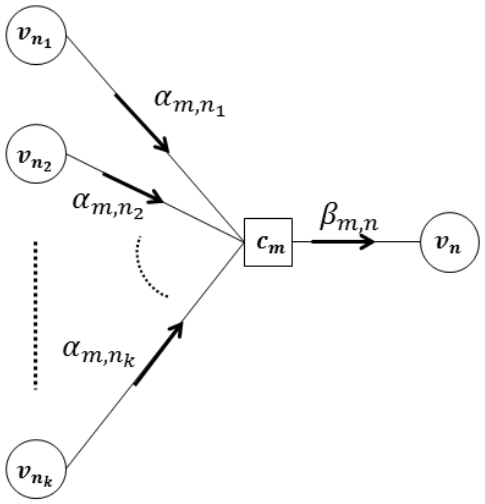


FIGURE 2.8 – Illustration de la mise à jour des messages $\beta_{m,n}$

Le premier et le deuxième terme de produit dans l'équation sont nommés respectivement la parité (signe) et la fiabilité (magnitude).

Mise à jour des nœuds de variable VN

Ce stade se caractérise par le calcul des messages $\alpha_{m,n}$ (figure 2.9) en utilisant les informations de canal λ_n et les messages entrants $\beta_{m',n}$ de tous les autres nœuds de contrôle reliés au nœuds de variable V_n, à l'exclusion des nœuds de contrôle C_m, $m' \in \mathcal{H}_C(n) \backslash m$ équation 2.24.

$$\alpha_{m,n} = \lambda_n + \sum_{m' \in \mathcal{H}_C(n) \backslash m} \beta_{m',n} \tag{2.24}$$

Après chaque itération, une décision peut être prise sur l'information a posteriori associée aux N nœuds de données :

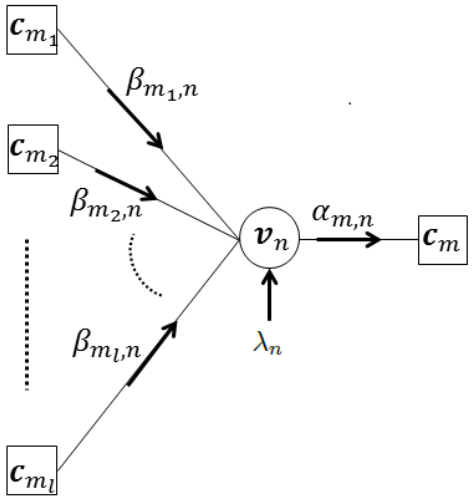


FIGURE 2.9 – Illustration de la mise à jour des messages $\alpha_{m,n}$

Calcul de l'information a posteriori

Quand la mise à jour des nœuds de variables est terminée, une information a posteriori doit être mis à jour sur chaque nœud de variable n figure 2.10, afin de pouvoir estimer le mot de code, le calcul de cette information $\tilde{\lambda}_n$ se fait en ajoutant les informations de canal λ_n et les messages $\beta_{m,n}$ provenant des nœuds de contrôle équation 2.25.

$$\tilde{\lambda}_n = \lambda_n + \sum_{m \in \mathcal{H}_C(n)} \beta_{m,n} \tag{2.25}$$

Prise de décision

A partir du vecteur $\tilde{\lambda}_n$ calculé, le vecteur de code $\hat{x} = \hat{x_1}, \hat{x_2}, \ldots, \hat{x_N}$ est estimé par :

$$\hat{x}_i = \begin{cases} 1 & \text{si et seulement si } \tilde{\lambda}_n \leq 0 \\ 0 & \text{si et seulement si } \tilde{\lambda}_n > 1 \end{cases} \tag{2.26}$$

Si $H \times \hat{x}^t = 0$ alors $\hat{x}$ est un mot de code valide donc le processus itératif converge et le décodage s'arrête. Sinon, le processus itératif continue jusqu'à ce qu'un mot de code

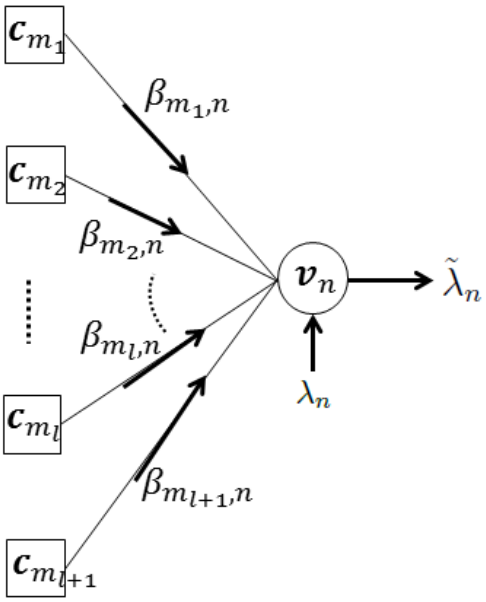


FIGURE 2.10 – Illustration de l'étape de calcul de l'information a posteriori

valide soit obtenu ou le nombre d'itérations atteint le nombre maximal I_{max}, ce qui met fin au processus de décodage.

2.5.5 Algorithme de décodage *Min Sum*

En pratique, des algorithmes dérivés de l'algorithme BP peuvent être considérés. Ces algorithmes peuvent être vus comme une simplification de l'algorithme BP, et donc sous-optimaux toujours sous l'hypothèse d'un graphe sans cycle. La diminution de performance engendrée par la sous-optimalité est à pondérer par le gain obtenu sur la complexité de l'organe du décodage. La majorité des algorithmes issus du BP reposent sur des opérations simplifiées de mise à jour des fiabilités en sortie des nœuds de contrôle.

La simplification la plus couramment utilisée est celle de l'algorithme *BP-Based* proposé par Fossorier [14] [79], connu aussi sous le nom *Min Sum*. Cette approximation repose sur le fait que le message calculé en sortie du nœud de contrôle est fortement dépendant du plus petit message entrant, en valeur absolue. La fonction $\phi(x)$ étant une fonction décroissante positive, on peut écrire :

$$\sum_i \phi(|x_i|) \geq \phi(\min_i |x_i|) \tag{2.27}$$

Et donc

$$\phi(\sum_i \phi(|x_i|)) \leq \min_i |x_i| \tag{2.28}$$

La fiabilité du message en sortie du nœud de contrôle peut donc être approximée par celle du message le moins sûr. Cette simplification algorithmique permet une réduction de la complexité du décodage au niveau des nœuds de contrôle. En effet la fonction non linéaire $\phi(.)$ est remplacée par une simple fonction minimum. De plus, le nombre de messages différents, en valeur absolue, en sortie d'un nœud de contrôle est alors réduit à deux : la plus petite et la deuxième plus petite contribution entrantes.

Dans le cas du BP, le nombre de messages en sortie du nœud de contrôle est égal au degré de connexion du nœud. Il en résulte donc une importante réduction des ressources mémoires nécessaires et de la complexité de l'entité de mise à jour. Une seconde propriété très intéressante de cet algorithme réside dans sa robustesse face à une mauvaise estimation des log-rapports de vraisemblances associées aux observations en sortie du canal. En effet, du fait de l'utilisation de l'opérateur minimum, tous les calculs se font à un coefficient multiplicatif près.

Par exemple, dans le cas d'une modulation *Binary Phase Shift Keying* (BPSK) et d'un canal de propagation à bruit blanc additif gaussien de variance σ^2 (AWGN), le log-rapport de vraisemblance du symbole reçu y, à présenter à l'entrée du décodeur, est égal à $2y/\sigma^2$. Dans le cas d'un décodage par l'algorithme *Min-Sum*, on peut présenter à l'entrée du décodeur un log-rapport de vraisemblance égal à αy. Ceci permet donc de s'affranchir de l'estimation de la variance du bruit.

Algorithm 1 Algorithme Min Sum

Entrée : $\underline{y} = (y_1, ..., y_N) \in Y^N$
Sortie : $\underline{\hat{x}} = (\hat{x}_1, ..., \hat{x}_N) \in \{0,1\}^N$

Initialisation

pour tout $n = 1, ..., N$ **faire**

$$\lambda_n = log\left(\frac{P(x_n = 0|y_n)}{P(x_n = 1|y_n)}\right);$$

pour tout $n = 1, ..., N$ et $m \in \mathcal{H}_C(n)$ **faire** $\alpha_{m,n} = \lambda_n$;

Processus itératif

▷ Mise à jour des nœuds CN.

pour tout $m = 1, ..., M$ et $n \in \mathcal{H}_V(m)$ **faire**

$$\beta_{m,n} = \left(\prod_{n' \in \mathcal{H}_V(m)\setminus n} sgn\left(\alpha_{m,n'}\right)\right)\left(\min_{n' \in \mathcal{H}_V(m)\setminus n} |\alpha_{m,n'}|\right);$$

▷ Mise à jour des nœuds VN.

pour tout $n = 1, ..., N$ et $m \in \mathcal{H}_C(n)$ **faire**

$$\alpha_{m,n} = \lambda_n + \sum_{m' \in \mathcal{H}_C(n)\setminus m} \beta_{m',n};$$

▷ Calcul de l'information à posteriori.

pour tout $n = 1, ..., N$ **faire**

$$\tilde{\lambda}_n = \lambda_n + \sum_{m \in \mathcal{H}_C(n)} \beta_{m,n};$$

▷ Prise de décision.

pour tout $n = 1, ..., N$ **faire**

$$\hat{x}_n = \frac{1 - sgn(\tilde{\lambda}_n)}{2};$$

si $\hat{x}$ est correcte, arrêter le processus itératif.

fin du processus itératif

2.6 Conclusion

Dans ce chapitre, nous avons décrit les notions de base sur les codes LDPC, quelques méthodes de construction et différentes méthodes de décodage. On a présenté les graphes de Tanner, qui sont une représentation utile des codes blocs linéaires et en particulier des codes LDPC. On a expliqué comment on peut construire différents types de codes LDPC, comme les codes LDPC aléatoires, les codes LDPC pseudo-aléatoires et les codes LDPC structurés (i.e. les codes basés sur les géométries finies et les codes quasi-cycliques).

Nous avons présenté dans ce chapitre une méthode de codage du code LDPC, ensuite concernant le décodage des codes LDPC, nous avons plusieurs méthodes : le décodage avec la logique majoritaire, le décodage avec basculement de bit, le décodage itératif basé sur la propagation de croyance BP et son algorithme dérivé le MS standard.

Cette thèse se focalise sur l'amélioration des algorithmes de décodage du code LDPC sur la base de la connaissance de la structure cyclique du code, sachant que l'existence des cycles courts influe négativement sur les performances de décodage, le prochain chapitre traite d'une façon détaillé le problème des cycles courts avec des solutions comment on peut lutter contre ce problème.

Chapitre 3

Amélioration des performances de l'algorithme *Belief Propagation*

3.1 Introduction

Dans un système de communication à canal bruité, plusieurs études ont été élaborées durant les 50 dernières années, afin de protéger l'information contre le bruit de canal et corriger les erreurs transmises. Le code LDPC est considéré parmi les codes les plus populaire, découvris dans le début des années 60 par Robert Gallager [27].

Un code LDPC peut être représenté graphiquement par un graphe de Tanner [42] ou sous forme de matrice creuse comme décrit dans le chapitre précédent. Dans une chaîne de transmission, l'information subit l'opération de codage canal et à la réception l'opération de décodage de canal, dans ce chapitre on s'intéresse plus particulièrement aux décodeurs de canal du code LDPC.

L'algorithme de décodage *Belief Propagation* décrit dans le chapitre 2 peut être vue comme un algorithme de propagation des messages sur le graphe factorielle associé, les messages transitant par les branches peuvent être soit des probabilités soit des logarithmiques des rapports de vraisemblance, le principe de la propagation est l'application de la règle de Bayes (soit sur chaque bit) et itérativement afin d'estimer les probabilités a posteriori de chaque bit. Dans le cas d'un décodage sur un graphe sans cycle (le graphe est alors un arbre) c'est-à-dire dans le cas où tous les messages sont indépendants, la factorisation locale des règles de Bayes conduit au calcul exact des probabilités a posteriori des nœuds de données [17]. Dans certain cas incluant celui des codes LDPC, le graphe

factoriel contient des cycles. Dans ces conditions, du fait de la dépendance cyclique des différents messages [16] [18], le calcul exact des probabilités a posteriori n'est plus assuré. Cependant, plus un graphe est creux plus la dépendance entre les différents messages sera faible. De plus, pour minimiser cette dépendance il est nécessaire de calculer plusieurs fois les messages sur les branches ce qui nécessite plusieurs itérations lors du décodage avec l'algorithme de propagation de croyance.

Récemment plusieurs études ont été élaborées dans le but de résoudre le problème de la dépendance des messages durant le décodage avec un code cyclique et réduire le temps de traitement (latence). Comme exemple l'algorithme *Uniformly Reweighted Belief Propagation* (URW-BP) [80] [81] est une version modifié de l'algorithme BP proposé par Wymeersch et al, peut arriver à des performances mieux que l'algorithme BP standard plus particulièrement pour les codes LDPC régulier en pondérant le graphe avec un facteur constant. Liu et de Lamare ont développé l'algorithme *Variable Factor Appearance Probability Belief Propagation* (VFAP-BP) [23] [82], cet algorithme rassemble la stratégie de pondération proposée par henk wymeersh et la connaissance de la structure cyclique du code pour la détermination du coefficient de pondération. Les résultats de simulation ont montré que l'algorithme VFAP-BP est meilleur en correction des erreurs et en convergence par rapport à l'algorithme URW-BP et le BP standard.

Dans ce chapitre on propose un nouvel algorithme qui se base sur une nouvelle stratégie de pondération des codes cycliques et une nouvelle méthode pour la détermination du facteur de pondération basée sur la connaissance de la structure cyclique du code avec utilisation de l'algorithme [83]. L'algorithme développé est appelé *Exponential Factor Appearance Probability Belief Propagation* (EFAP-BP) [84].

L'algorithme proposé peut être appliqué pour les codes réguliers ainsi que les codes irréguliers. Et moins compliqué par rapport à la méthode VFAP-BP, pour évaluation de cet algorithme, une étude du comportement de convergence et des performances BER/SNR a été effectué, les résultats de simulation ont montré que le nouvel algorithme est meilleur par rapport à l'algorithme VFAP-BP et le BP standard en performances BER/SNR et en rapidité de convergence.

Ce chapitre est organisé comme suit : la section II introduit le problème des cycles courts, ensuite une définition d'un cycle court et des méthodes comment on peut l'identifier dans le code est présenté dans la section III, l'algorithme VFAP-BP est décrit dans la

section IV, le nouvel algorithme EFAP-BP est détaillé dans la section VI, ensuite la sixième section de ce chapitre est consacré pour les résultats et simulations enfin conclusion.

3.2 Problème des cycles courts

L'existence des cycles courts dans le code crée une dépendance statistique chez les messages échangées par les nœuds de variable et les nœuds de contrôle, de sorte que les messages sortants ont une haute fiabilité ou de manière équivalente une faible qualité. Ce problème est souvent appelée "excès de confiance en anglais overconfident" ou "surestimation en anglais overestimation" [85] [81].

le problème de surestimation empêche la convergence de l'algorithme du décodage BP, cependant pour converger il faut un nombre important d'itérations, cela augmente le temps de traitement du décodeur (temps de latence).

Le calcul des cycles courts exactement dans un graphe arbitraire semble un calcul impossible. Cependant, l'algorithme de calcul des cycles court proposé dans la référence [83] transforme le problème de calcul des cycles courts en un calcul des chemins "*lollipop*" à travers des multiplications matricielles, notez que l'algorithme de calcul des cycles courts proposé dans [86] est plus performant et peut aussi être utilisé surtout dans le cas des codes de large dimension.

3.3 Algorithme pour le calcul des cycles courts

3.3.1 Introduction

On dit qu'un graphe de Tanner du code LDPC contient un cycle lorsqu'il existe un chemin pour aller d'un nœud et revenir au même nœud sans passer par les mêmes arêtes. Dans le cas où le graphe contient des cycles. On appelle le nombre de branches constituant un cycle, la longueur de ce cycle. On appelle "*girth*" du graphe, la longueur minimale des cycles sur ce graphe. Ce dernier a un impact sur les performances d'un code LDPC. Un exemple de cycles de longueurs 4, 6 et 8 est illustré à la figure 3.1. Un graphe sans cycles est appelé un arbre. On note que l'existence des cycles influe négativement sur les performances d'un code LDPC par un phénomène de surestimation comme décrit dans le paragraphe précédant.

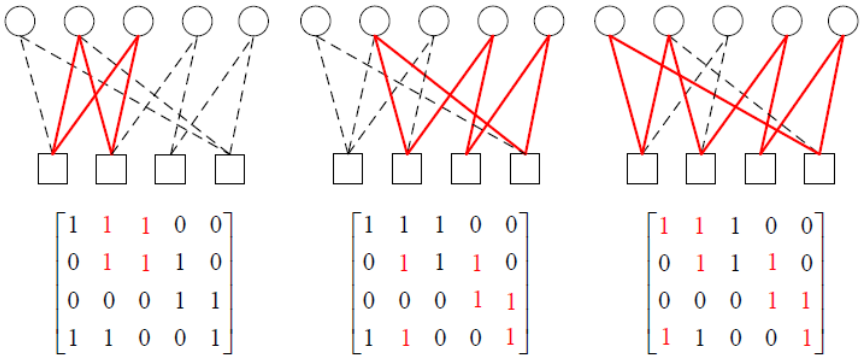


FIGURE 3.1 – Exemple de cycles de longueur 4, 6 et 8

Avec l'apparition des codes LDPC irrégulier [11], la structure cyclique et la régularité du code sont devenu des critères pour avoir un bon modèle graphique pour le code. Pour avoir des informations sur la structure cyclique du graphe "nombre de cycles courts, et la longueur du cycle le plus court", il est nécessaire de trouver un algorithme efficace qui permet de compter des longueurs de cycle distinct dans un graphe.

Le calcul des cycles courts exactement dans un graphe arbitraire semble être un problème difficile. Dans le papier [87] les auteurs ont proposé une méthode pour le calcul des cycles courts, mais la complexité augmente quand la dimension du cycle court dépasse la valeur 8, le même problème se pose avec la méthode proposé dans l'article [88].

Dans ce chapitre on va décrire deux algorithmes de calcul des cycles court :

3.3.2 Algorithme de Fan & Xiao

L'algorithme de Fan et Xiao consiste à étudier les différentes formes que peut prendre un cycle court dans le code, et les chercher.

Calcul du Cycle 4

Considérant une matrice de parité H de dimension $(m \times n)$. La figure 3.2 montre qu'un cycle 4 est composé de 4 élément non nul et appartient à deux lignes et deux colonnes. Un cycle 4 doit être en deux lignes, le nombre de combinaisons possible de deux lignes est donnée par la formule w en dessous :

$$w = C_m^2 = \frac{m(m-1)}{2} \tag{3.1}$$

En calculant la somme de deux lignes, les éléments du vecteur somme résultant sont

(0, 1, ou 2). si le nombre d'élément 2 dans le vecteur somme calculé est t_i, avec $0 \leq i \leq w$ et $t_i \geq 2$, alors il y a au moins un cycle court de dimension 4 dans cette matrice de parité, N_4 le nombre des cycles courts de dimension 4 est calculé par la formule :

$$N_4 = \sum_{i=1}^{w} C_{t_i}^2 = \sum_{i=1}^{w} \frac{t_i(t_i - 1)}{2} \tag{3.2}$$

L'algorithme suivant permet de calculer les cycles 4 dans un code :

Algorithm 2 Algorithme de calcul du cycle 4 dans le code.

for $i_1 = 1 : m$
  for $i_2 = (i_1 + 1) : m$
    $h = H(i_1, :) + H(i_2, :)$
    Si le nombre d'éléments 2 dans h est supérieur
    où égale à 2, alors il existe des cycles 4.
  end
end

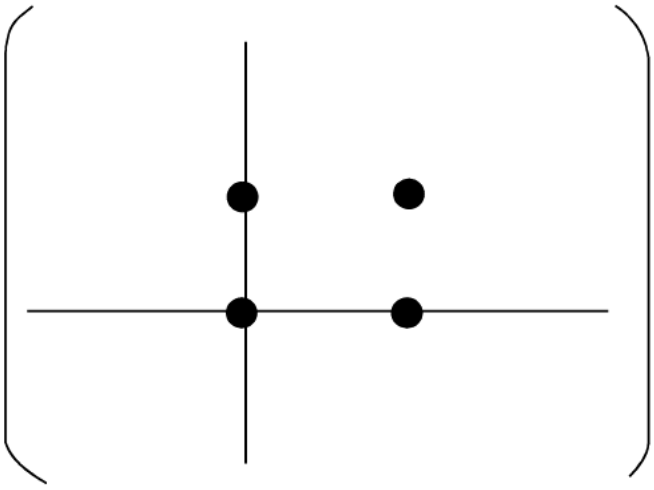


FIGURE 3.2 – Représentation d'un cycle 4

En dessous un exemple pour le calcul des cycles 4 dans le code.

Exemple 1

Considérant la matrice de contrôle de parité H suivante :

$$\text{H=} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

l'addition de la première ligne et la deuxième donne $h_1 = [2\ 1\ 1\ 2\ 1]$, la somme des

lignes 1 et 3 est $h_2 = [1\ 2\ 1\ 2\ 0]$, $h_3 = [1\ 1\ 2\ 2\ 1]$ est l'addition des lignes 2 et 3, le résultat prouve que le code H comporte $N_4 = 3$ cycles 4.

Calcul du cycle 6

Nous savons qu'un cycle 6 est composé de six éléments 1 et appartient à 3 lignes et 3 colonnes, donc la probabilité d'avoir un cycle 6 dans le code est $n = P_3^3 = 6$ comme présenté dans la figure 3.3.

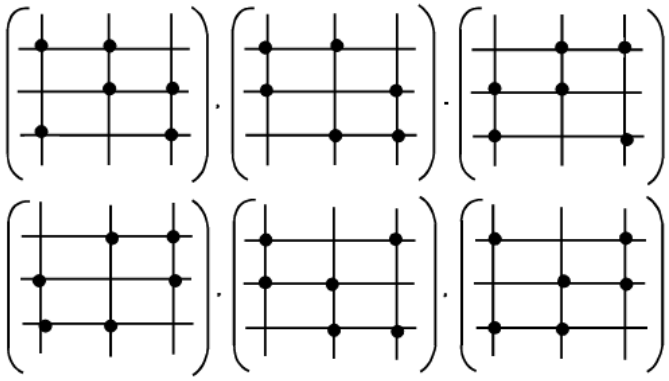


FIGURE 3.3 – Les différentes forme d'un cycle 6 dans le code

Pour calculer le nombre des cycles 6 dans toute la matrice H de dimension $(m \times n)$, on commence par chercher les combinaisons possible de 3 lignes dans le code et on calcule la somme comme nous avons fait précédemment pour le cycle 4, le nombre des combinaisons de trois lignes possible dans le code H est calculé par :

$$w = C_m^3 = \frac{m(m-1)(m-2)}{6} \tag{3.3}$$

En calculant la somme des trois ligne le vecteur résultant est composé des élément (0, 1, 2, ou 3), il existe un cycle 6 lorsqu'il y a trois éléments 2 dans le vecteur somme, la somme des cycles six est calculé par $N_6 = \sum_{i=1}^{w} t_i$ avec t_i égale à 1 si le nombre d'élément 2 dans la combinaison i est trois.

l'algorithme présenté en dessous permet de calculer le nombre des cycles 6 dans un code :

Exemple 2

Considérant la matrice de contrôle de parité H suivante :

Algorithm 3 Algorithme de calcul du cycle 6 dans le code.

for $i_1 = 1 : m$
  for $i_2 = (i_1 + 1) : m$
    for $i_3 = (i_2 + 1) : m$
      $h = H(i_1, :) + H(i_2, :) + H(i_3, :)$
      Si le nombre d'éléments 2 dans h est supérieur
      où égale à 3, alors il existe des cycles 6.
    end
  end
end

$$\text{H}= \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Par exemple, en ajoutant la première, deuxième et quatrième lignes, nous avons le vecteur [2 2 2 0 1 1 1], Il y a trois éléments 2 dans ce vecteur. Donc, il y a un cycle 6 dans la matrice H. En vérifiant toutes les combinaisons possibles de trois lignes dans cette matrice, le nombre des cycles 6 est $N_6 = 28$.

Les auteurs dans le papier [89] ont donné une formule général, donc pour un cycle de taille g appartient à $g/2$ lignes également $g/2$ colonnes, on peut calculer le nombre des cycles courts g en cherchant les formes de g dans tous les combinaisons possible de $g/2$ lignes, et on fait la somme pour w combinaisons possible de $g/2$.

$$w = C_m^{g/2} = \frac{m(m-1)...(m-(\frac{g}{2}-1))}{\frac{g}{2}!} \tag{3.4}$$

Cette méthode a deux inconvénients :

— pour chaque g, il faut concevoir un programme de calcul particulier.

— plus g augmente plus la complexité de calcule augmente.

Les auteurs Halford et Chugg ont proposés dans la référence [83] un algorithme plus efficace et plus rapide pour la détermination de la structure cyclique d'un code LDPC, l'algorithme est décrit en dessous,

3.3.3 Algorithme de Halford & Chugg

L'algorithme proposé par Halford et Chugg permet de calculer les cycles de dimension g, $g+2$ et $g+4$ dans un graphe bipartite dont g est le cycle le plus court. Un chemin appelé "*lollipop*" $(m, n-m)$ se réfère à un chemin de dimension n, c'est-à-dire une séquence de sommets $a_1, ..., a_{n+1}$, où tous les sommets sont distincts sauf $a_{n+1} = a_{m+1}$. Ainsi les cycles de longueur $2m$ sont des chemins *lollipop* $(0, 2m)$, exemple figure 3.4 est un chemin *lollipop* $(2, 4)$.

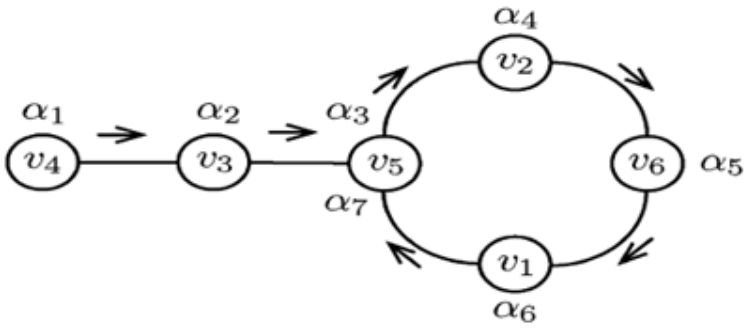


FIGURE 3.4 – Un chemin *lollipop* $(2, 4)$ où $\alpha_3 = \alpha_7$ alors que α_1,α_2,...,α_6 sont distinct

On considère un graphe $G(V, E)$ définie par un ensemble V constitué par deux sous ensembles des nœuds V_c et V_s et par l'ensemble E dont les éléments sont appelés arêtes (*Edges* en anglais).

Donné que $|.|$ représente la cardinalité de l'ensemble, nous définissons la matrice $P_{2k}^{(v_c)}$ comme $|V_c| \times |V_c|$ dans laquelle les (i, j) éléments sont le nombre de chemin de dimension 2k de $v_{c_i} \in V_c$ au $v_{c_j} \in V_c$.

De la même façon on définie la matrice $P_{2k+1}^{v_c}$ comme $|V_c| \times |V_s|$ dans laquelle les (i, j) éléments sont le nombre de chemin de dimension $2k+1$ de $v_{c_i} \in V_c$ au $v_{s_j} \in V_s$ On définie la matrice $L_{2k,2k-2k'}^{v_c}$ comme $|V_c| \times |V_c|$ dans laquelle les (i, j) éléments sont le nombre de chemins $(2k^{'}, 2k-2k')$ *lollipop* de $v_{c_i} \in V_c$ au $v_{c_j} \in V_c$. De même nous définissons la matrice $L_{2k'+1,2k-2k'}^{v_c}$ comme $|V_c| \times |V_s|$ dans laquelle les (i, j) éléments sont le nombre de chemins $(2k'+1, 2k-2k')$ *lollipop* de $v_{c_i} \in V_c$ au $v_{s_j} \in V_s$. Les matrices précitées satisfont les relations suivantes :

$$P_{2k+1}^{v_c} = P_{2k}^{v_c} E - \sum_{i=0}^{k-1} L_{(2i+1,2k-2i)}^{v_c} \tag{3.5}$$

$$P_{2k}^{v_c} = P_{2k-1}^{v_c} E^T - \sum_{i=0}^{k-1} L_{(2i,2k-2i)}^{v_c} \tag{3.6}$$

$$P_{2k+1}^{v_s} = P_{2k}^{v_s} E^T - \sum_{i=0}^{k-1} L_{(2i+1,2k-2i)}^{v_s} \tag{3.7}$$

$$P_{2k}^{v_s} = P_{2k-1}^{v_s} E - \sum_{i=0}^{k-1} L_{(2i,2k-2i)}^{v_c} \tag{3.8}$$

$$L_{(0,2k)}^{v_c} = (P_{2k-1}^{v_c} E^T) \circ I \tag{3.9}$$

$$L_{(0,2k)}^{v_s} = (P_{2k-1}^{v_s} E^T) \circ I \tag{3.10}$$

Avec le terme $\circ$ signifie le produit matricielle élément par élément (element-wise en anglais), avec E la matrice des arêtes dont l'élément (i, j) prend la valeur 1 s'il y a une connexion entre le iéme et le $j^{éme}$ nœud, et I est la matrice identité. Le nombre totale des cycles de dimension $2k$ est donnée par :

$$N_{2k} = \frac{1}{2k} Tr(L_{(0,2k)}^{v_c}) = \frac{1}{2k} Tr(L_{(0,2k)}^{v_s}) \tag{3.11}$$

Le terme $Tr(.)$ (Trace en anglais) désigne la somme des éléments du diagonal de la matrice, pour chercher le cycle g et les autres cycles dont la taille $g + 2$ et $g + 4$ dans le Graphe de Tanner, les équations de 1 à 6 sont développés et mis à jour de telle sorte que le calcul des cycles courts est égale au calcul des *lollipop*.

3.4 Algorithme VFAP-BP

l'algorithme VFAP-BP lutte contre le problème cité au dessus "surestimation" en multipliant chaque nœud de parité par un facteur de probabilité d'apparence FAP (*Factor Appearence Probability* en anglais) dépendant des cycles courts qui traversent ce nœud figure 3.5.

En attribuant différentes valeurs FAP $\rho_m(m = 0, 1, ..., M - 1)$ au nœuds de parité la Figure 3.5, les simulations ont montrés que l'algorithme VFAP-BP est meilleur en correction des erreurs et en convergence par rapport au URW-BP et au BP standard.

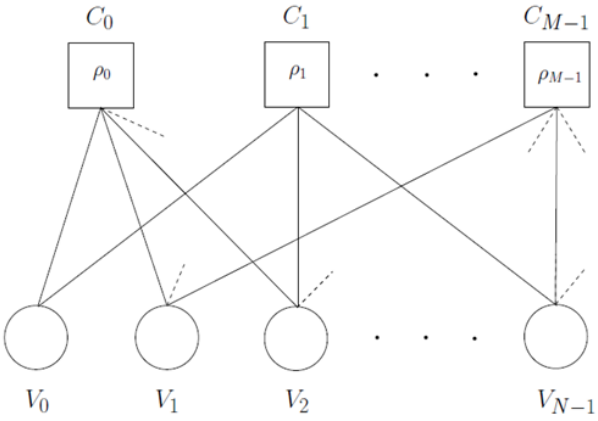


FIGURE 3.5 – Le modèle graphique représente l'attribution du facteur ρ au nœuds de parité dans un code LDPC lors du décodage VFAP-BP, $\rho_m(m = 0, 1, ..., M-1) = \{\rho_v, 1\}$

L'utilisation de l'algorithme consiste à définir les paramètres suivants :

— Le cycle court g dans le code.

— Le nombre de cycle g passant à travers chaque nœud de parité $C_m(m = 0, 1, ..., M-1)$ appelé $s_m(m = 0, 1, ..., M-1)$

— μ_g la moyenne des cycles g passant à travers tous les nœuds de parités.

— Le vecteur de pondération $\rho_m = [\rho_0, \rho_1, ..., \rho_{M-1}]$ se compose des facteurs de probabilité d'apparence (FAP) qui vont être attribué à chaque nœud de parité comme décrit dans la figure au dessus.

Le message sortant du nœud de contrôle peut être soit inchangées ou partiellement pondéré. Cela dépend si les massages sortants provenant d'un nœud de contrôle contribuent le passage des messages extrinsèque ou non. Un nœud de parité contrôle la convergence ou conduit à des croyances de faible qualité en raison de la création de la dépendance au sein des groupes des cycles court. Par conséquence, les deux cas peuvent être distingués par des critères simples :

si $s_m < \mu_g$ le nœud de contrôle C_m est considérée comme constructif puis $\rho_m = 1$; Sinon ce nœud de contrôle est déterminée en tant que nœud destructif et nous avons

$$\rho_m = \rho_v \tag{3.12}$$

où

$$\rho_v = \frac{2}{n_D} \tag{3.13}$$

n_D est la connectivité moyenne des N nœuds de variables, elle se calcule par la formule

suivante :

$$\bar{n_D} = \frac{1}{\oint_0^1 \lambda(x)dx} = \frac{M}{N \oint_0^1 \rho(x)dx} \tag{3.14}$$

$\lambda(x)$ et $\rho(x)$ sont les proportions des branches du graphe connectées à des nœuds de variable (respectivement nœuds de contrôle).

Ensuite on calcul la valeur du log-likelihood ratio pour les symboles reçu y_n

$$\lambda_n = log(\frac{P(x_n = 0|y_n)}{P(x_n = 1|y_n)}) = 2\frac{y_n}{\sigma^2} \tag{3.15}$$

Mise à jour des nœuds de parité CN

cette étape reste la même par rapport à l'algorithme BP, La mise à jours des messages $\beta_{m,n}$ se fait avec utilisation des messages α issu des nœuds de donnée connecté au nœuds de parité C_m, avec exclusion de la valeur α qui vient du nœud V_n.

$$\beta_{m,n} = \alpha_m \left(\prod_{n' \in \mathcal{H}_V(m) \backslash n} sgn\left(\alpha_{m,n'}\right) \right) \times \phi \left(\sum_{n' \in \mathcal{H}_V(m) \backslash n} \phi\left(|\alpha_{m,n'}|\right) \right) \tag{3.16}$$

Avec la fonction non linéaire

$$\phi(x) = -log\left(tanh\left(\frac{|x|}{2}\right)\right) \tag{3.17}$$

Mise à jour des nœuds de variable VN

comme décrit dans l'article [23] le passage des nœuds de variable vers les nœuds de parité se fait d'une façon extrinsèque $m' \in \mathcal{H}_C\left(n\right) \backslash m$ l'équation 3.18 :

$$\alpha_{m,n} = \lambda_n + \sum_{m' \in \mathcal{H}_C(n) \backslash m} \rho_{m'}\beta_{m',n} - (1 - \rho_m)\beta_{m,n} \tag{3.18}$$

avec ρ_m le coefficient de pondération déjà calculé.

Terminal

Quand la mise à jour des nœuds de variables est terminée, chaque bit d'indice n est mis à jour en ajoutant les informations de canal λ_n et les messages de contrôle β provenant des nœuds de contrôle.

$$\tilde{\lambda}_n = \lambda_n + \sum_{m \in \mathcal{H}_C(n)} \rho_m \beta_{m,n} \tag{3.19}$$

A partir du vecteur $\tilde{\lambda}_n$ mis à jour, le vecteur de code $\hat{x} = \hat{x_1}, \hat{x_2}, \ldots, \hat{x_N}$ est estimé par :

$$\hat{x}_n = \begin{cases} 1 & \text{si et seulement si } \tilde{\lambda}_n \leq 0 \\ 0 & \text{si et seulement si } \tilde{\lambda}_n > 1 \end{cases} \tag{3.20}$$

Si $H \times \hat{x}^t = 0$ alors $\hat{x}$ est un mot de code valide donc le processus itératif converge et le décodage s'arrête. Sinon, le processus itératif continue jusqu'à ce qu'un mot de code valide soit obtenu ou le nombre d'itérations atteint le nombre maximal I_{max}, ce qui met fin au processus de décodage.

Algorithm 4 Algorithme VFAP-BP

Initialisation :

1 : Trouver le cycle court g et le nombre des cycles court de longueur g s_m passant à travers chaque nœud de contrôle C_m ;

2 : Déterminer le facteur FAPs pour chaque nœud de contrôle : si $s_m < \mu_g$ $\rho_m = 1$, sinon $\rho_m = \rho_v$ avec $\rho_v = \frac{2}{n_D}$;

Décodage VFAP-BP :

Étape 1 : Saisir le nombre maximum des itérations I_{max} et initialiser $\lambda_n = 2\frac{y_n}{\sigma^2}$;

Étape 2 : Mettre à jour les messages passant des nœuds de contrôle C_m vers les nœuds de variable V_n ;

Étape 3 : Mettre à jour les messages passant des nœuds de variable V_n au nœuds de contrôle en utilisant l'équation 3.18, avec $\beta_{m',n}$ et $\beta_{m,n}$ sont nulles dans la première itération ;

Étape 4 : Mettre à jour la valeur $\tilde{\lambda}_n$ avec utilisation de l'équation 3.18 et estimer le vecteur $\hat{x}$;

Étape 5 : Arrêter le décodage si $H \times \hat{x}^t = 0$ ou I_{max} est atteint, sinon retour à l'étape 2.

3.5 l'algorithme EFAP-BP proposé

L'algorithme VFAP-BP [23] [82] a permit de résoudre le problème de surestimation provoqué par les cycles courts et donner des résultats meilleurs par rapport à l'algorithme BP avec un nombre minimum d'itérations. Mais cet algorithme a encore certains inconvénients tels que :

— La détermination du vecteur de pondération ρ est une opération très compliqué.

— Le vecteur ρ calculé n'est pas optimale.

— Une grande complexité ajouté au décodage par rapport au BP standard

Afin de réduire la complexité, automatiser et simplifier le calcul de la valeur FAP, et également améliorer la convergence. Nous proposons l'algorithme EFAP-BP. L'algorithme EFAP-BP est une version améliorée de l'Algorithme VFAP-BP pour les codes LDPC réguliers et irréguliers, cet algorithme introduit d'une part une nouvelle stratégie pour la pondération des nœuds qui consiste à faire une seule multiplication sur les nœuds de contrôle avec le facteur de pondération déjà calculé, cela a permit réduire la complexité de calcul et améliorer le convergence par rapport à l'algorithme VFAP-BP comme décrit dans le papier [90], d'une autre part la proposition d'une nouvelle fonction "exponentiel" pour la détermination du facteur de probabilité d'apparence FAP, cela permet d'attribuer à chaque nœud de contrôle un facteur dépendant du nombre des cycles court qui le traverse. l'utilisation de la nouvelle fonction a permis de se débarrasser du calcul de la moyenne de connectivité pour N nœuds de variables n_D, et rendre l'algorithme facile à automatiser. Dans la suite, nous décrivons les différentes étapes de l'algorithme EFAP-BP.

Initialisation

La première étape dans cet algorithme consiste à calculer :

1. Le cycle le plus court g dans le code.
2. $s_m(m = 0, 1, \ldots, M-1)$ le nombre des cycles court g existant dans chaque nœud de contrôle $C_m(m = 0, 1, \ldots, M-1)$
3. μ_g la moyenne des cycles court passant à travers tous les nœuds de contrôle.

Ensuite le calcul du facteur ρ_m avec la fonction suivante :

$$\rho_m = exp\left(-\frac{s_m}{k \times \mu_g}\right) \tag{3.21}$$

la valeur k est une constante déterminé empiriquement.
Le choix de la fonction exponentiel est inspiré par le fait que les nœuds de contrôle qui sont traversé par plusieurs cycles court favorisent un facteur de pondération inférieur à 1.

Ensuite on calcule la valeur du *log-likelihood ratio* pour les symboles reçu y_n

$$\lambda_n = log(\frac{P(x_n = 0|y_n)}{P(x_n = 1|y_n)}) = 2\frac{y_n}{\sigma^2} \tag{3.22}$$

Mise à jour des nœuds de parité CN

cette étape reste la même par rapport à l'algorithme BP, La mise à jours des messages $\beta_{m,n}$ se fait avec utilisation des messages α issu des nœuds de données connecté au nœuds de contrôle C_m, avec exclusion de la valeur α qui vient du nœud V_n.

$$\beta_{m,n} = \alpha_m \left(\prod_{n' \in \mathcal{H}_V(m) \backslash n} sgn\left(\alpha_{m,n'}\right) \right) \times \phi \left(\sum_{n' \in \mathcal{H}_V(m) \backslash n} \phi\left(|\alpha_{m,n'}|\right) \right) \tag{3.23}$$

Avec la fonction non linéaire

$$\phi(x) = -log\left(tanh\left(\frac{|x|}{2}\right)\right) \tag{3.24}$$

Mise à jour des nœuds de variable VN

Dans le but de détruire la dépendance crée entre les messages par l'effet des cycles courts, nous avons proposé dans cette partie de multiplier chaque nœud de contrôle par le facteur ρ_m correspondant (équation 3.25) comme décrit dans le l'article [90].

$$\alpha_{m,n} = \lambda_n + \sum_{m' \in \mathcal{H}_C(n) \backslash m} \rho_{m'} \beta_{m',n} \tag{3.25}$$

Avec ρ_m le coefficient de pondération déjà calculé.

Terminal

Quand la mise à jour des nœuds de variables est terminée, chaque bit d'indice n est mis à jour en ajoutant les informations de canal λ_n et les messages de contrôle β provenant des nœuds de contrôle.

$$\tilde{\lambda}_n = \lambda_n + \sum_{m \in \mathcal{H}_C(n)} \beta_{m,n} \tag{3.26}$$

A partir du vecteur $\tilde{\lambda}_n$ mis à jour, le vecteur de code $\hat{x} = \hat{x_1}, \hat{x_2}, \ldots, \hat{x_N}$ est estimé par :

$$\hat{x}_n = \begin{cases} 1 & \text{si et seulement si } \tilde{\lambda}_n \leq 0 \\ 0 & \text{si et seulement si } \tilde{\lambda}_n > 1 \end{cases} \tag{3.27}$$

Si $H \times \hat{x}^t = 0$ alors $\hat{x}$ est un mot de code valide donc le processus itératif converge et le décodage s'arrête. Sinon, le processus itératif continue jusqu'à ce qu'un mot de code valide soit obtenu ou le nombre d'itérations atteint le nombre maximal I_{max}, ce qui met fin au processus de décodage.

Algorithm 5 Algorithme EFAP-BP

Initialisation :

1 : Trouver le cycle court g et s_m le nombre des cycles court de longueur g passant à travers chaque nœud de contrôle C_m ;

2 : Déterminer le facteur ρ_m en utilisant l'équation 3.21 ;

Décodage EFAP-BP :

Étape 1 : Saisir le nombre maximum des itérations I_{max} et initialiser $\lambda_n = 2\frac{y_n}{\sigma^2}$ et $\alpha_{m,n} = \lambda_n$;

Étape 2 : Mettre à jour les messages passant des nœuds de contrôle C_m vers les nœuds de variable V_n ;

Étape 3 : Mettre à jour les messages passant des nœuds de variable V_n au nœuds de contrôle en utilisant l'équation 3.25 ;

Étape 4 : Mettre à jour la valeur $\tilde{\lambda}_n$ et estimer le vecteur $\hat{x}$;

Étape 5 : Arrêter le décodage si $H \times \hat{x^t} = 0$ ou I_{max} est atteint, sinon retour à l'étape 2.

3.6 Résultats & Simulations

Dans cette section les trois algorithmes décrits au dessus le EFAP-BP, le VFAP-BP, et le BP standard sont comparés. Cette comparaison consiste à évaluer les performances de décodage en taux d'erreur et en comportement de convergence. Les codes LDPC utilisé dans cette étude comparative sont, code régulier généré par la méthode PEG de dimension $(252, 504)$ de la bibliothèque de Mackay [91], et code irrégulier généré par la méthode PEG de dimension $(252, 504)$ aussi de la bibliothèque de Mackay [91].

Après un codage qui vérifie que $H \times x^t = 0$, le message codé x de dimension N est envoyé à travers un canal à bruit blanc additif gaussien en utilisant une modulation BPSK. Le récepteur reçoit le mot du code y reçu du canal. Le décodeur essaye de décoder le mot de code y_N afin de trouver le mot de code transmis estimé $\hat{x}$. En comparant les bits de $\hat{x}$ et x on aura le taux d'erreur binaire (BER).

3.6.1 Structure du code régulier

Le code régulier utilisé dans cette étude est un code généré par la méthode PEG de Mackay de longueur $N = 504$ et de rendement $R = 1/2$ avec $(d_v = 3, d_c = 6)$, le code est présenté dans la figure 3.6 en dessous, les points en bleu représentent les éléments non nuls.

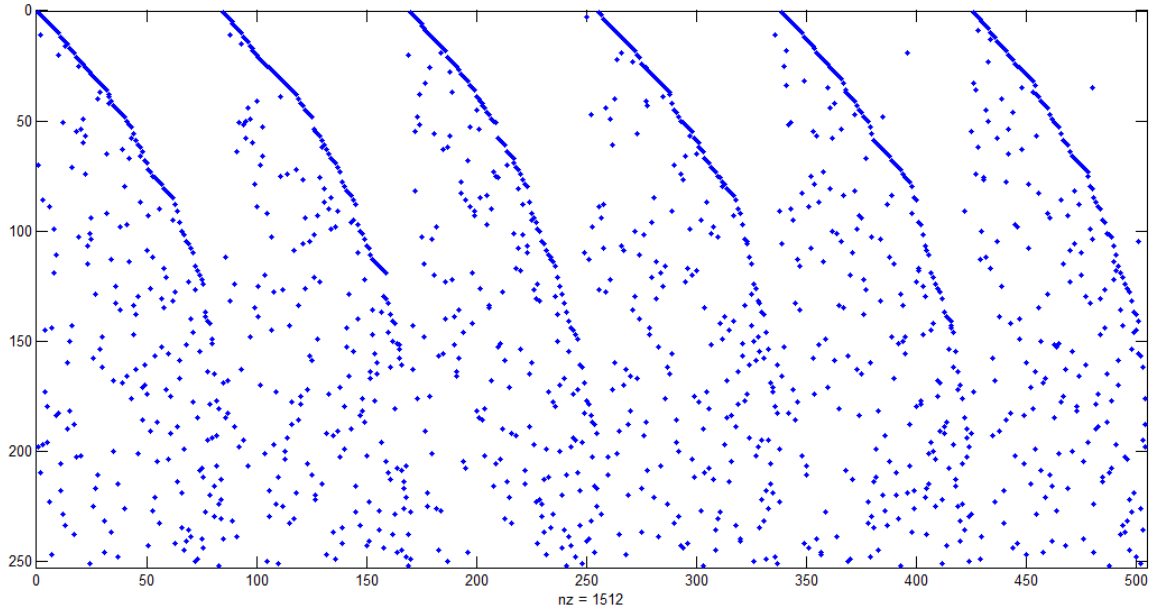


FIGURE 3.6 – Structure de la matrice de parité régulière généré par la méthode PEG avec les paramètres $(N = 504, R = 1/2, d_v = 3, d_c = 6)$

Les proportions des branches du graphe connectées au nœuds de variable (respectivement nœuds de contrôle) pour le code présenté au dessus sont décrites par les équations $\lambda(x) = x^2$ et $\rho(x) = x^5$ et la moyenne de connectivité pour les N nœuds de donnés $n_{D,reg} = 3$.

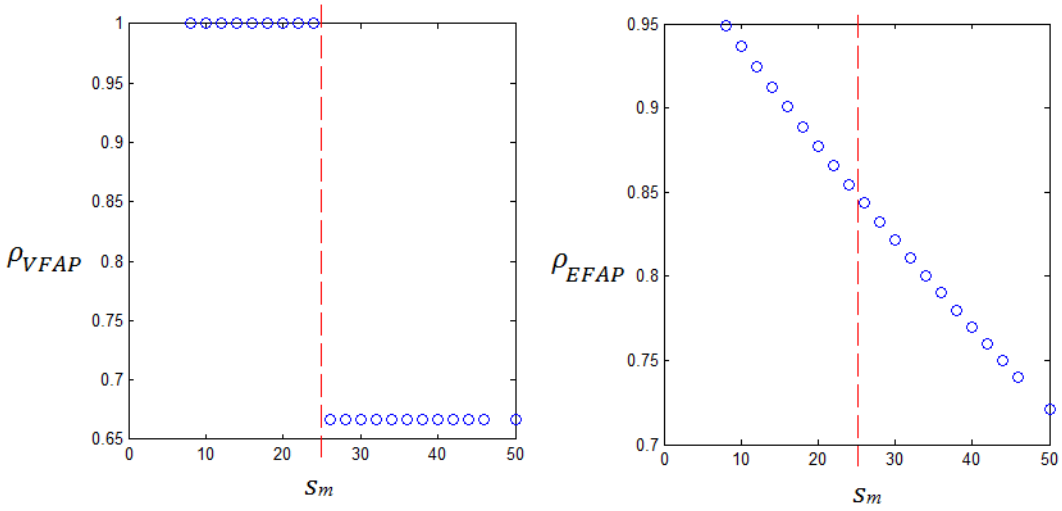


FIGURE 3.7 – Valeurs du facteur ρ en fonction du nombre des cycles court traversant chaque nœud de contrôle s_m pour le code PEG régulier.

Avec utilisation de l'algorithme de calcul des cycles court on trouve que le cycle le plus court dans ce code est $g = 8$, le nombre des cycles courts 8 est $N_g = 802$ et la moyenne des cycles court dans tous les nœuds de contrôles $\mu_g = 25.46$.

On appliquant les équations 3.13 et 3.14 cité dans la section 3.4, on trouve $\rho_v = 0.667$, concernant l'algorithme VFAP-BP, un nœud de contrôle sera pondéré par le facteur $\rho_v = 0.667$ si ce nœud est traversé par plus de 25 cycles 8, sinon il sera pondéré par la valeur 1 (voir figure 3.7 gauche).

Concernant l'algorithme EFAP-BP, le facteur de pondération varie suivant la fonction exponentiel (équation 3.21) déjà présenté dans le section 3.3, la (figure 3.7 droite) montre que le facteur ρ_m varie de 0.95 à 0.73 selon le nombre des cycles courts traversant chaque nœud s_m.

3.6.2 Convergence du code régulier

Les trois algorithmes de décodage EFAP-BP, VFAP-BP et le BP standard sont évalué en convergence et en correction des erreurs en utilisant le code régulier PEG, les résultats de simulation sont présenté en dessous :

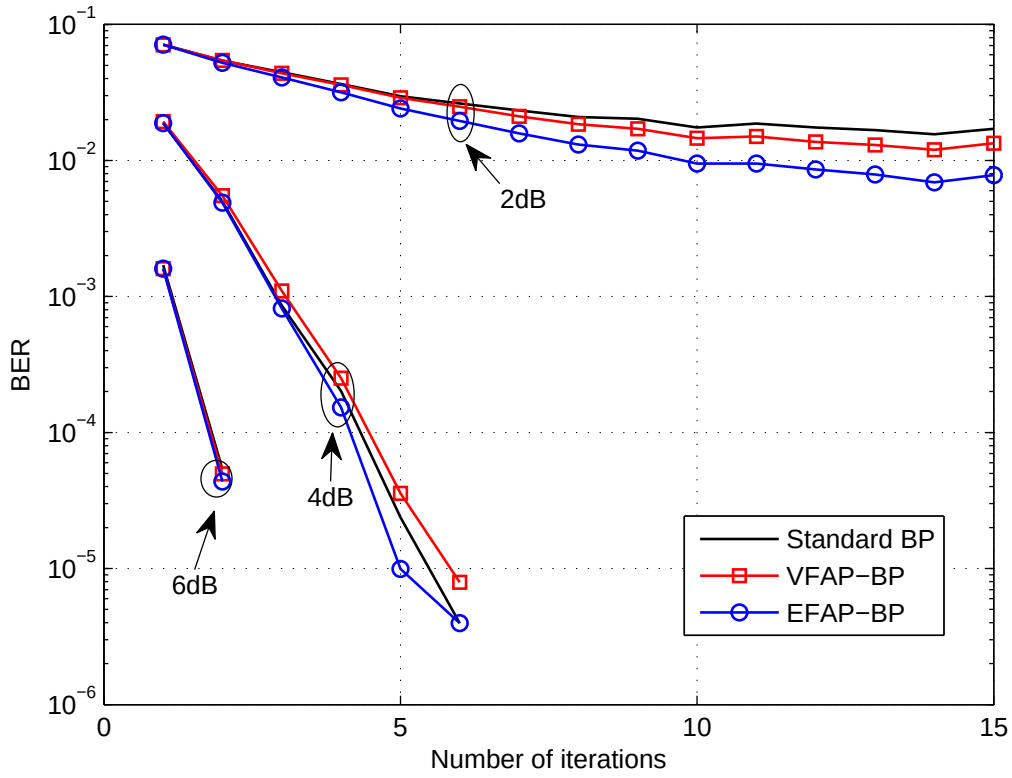


FIGURE 3.8 – Comparaison du comportement de convergence pour les algorithmes EFAP-BP, VFAP-BP et BP pour le code LDPC régulier de dimension $(252, 504)$, quand le SNR prend les valeurs SNR=[2,4,6]dB avec $k = 4$.

Dans la figure 3.8 on compare le comportement de convergence des trois algorithmes EFAP-BP, VFAP-BP et le BP standard sur les trois valeurs du SNR=[2, 4, 6]dB. cette simulation consiste à fixer le SNR sur une valeur constante et faire varier les itérations entre 1 et 15, Le résultat montre que l'algorithme proposé EFAP-BP converge plus rapidement par rapport aux algorithmes VFAP-BP et BP standard, plus particulièrement dans la région trop bruité SNR=2dB.

3.6.3 Performances du code régulier

La figure 3.9 illustre une comparaison des performances du code LDPC, en terme du BER/SNR, en utilisant les trois algorithmes avec deux valeurs maximal d'itérations $I_{max1} = 10$ et $I_{max2} = 60$, le résultat prouve que l'algorithme proposé EFAP-BP est plus performant avec un gain de 0.25 dB par rapport au VFAP-BP à un BER= 10^{-3} et avec un gain de 0.35 dB à un BER=10^{-3} par rapport au BP standard quand le nombre maximum des itérations est $I_{max1} = 10$.

Dans le cas ou le nombre des itérations est $I_{max2} = 60$, la simulation a montré que les

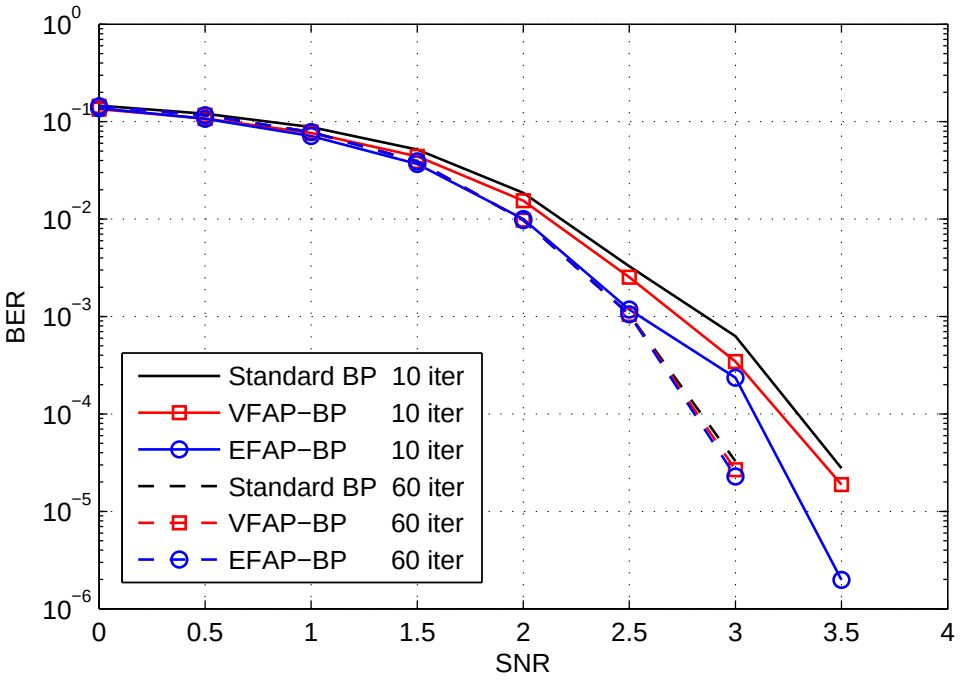


FIGURE 3.9 – Comparaison des performances d'erreurs de l'algorithme EFAP-BP, l'algorithme VFAP-BP et BP standard pour le code LDPC régulier $(d_v = 3, d_c = 6)$ de dimension $(252, 504)$ avec le maximum d'itération $I_{max} = (10, 60)$.

trois algorithmes ont un résultat très proche de sorte que le EFAP-BP est meilleur par rapport au VFAP-BP avec un gain cette fois de 0.05 dB à un BER=10^{-4} et de 0.1 dB à un BER=10^{-4} par rapport au BP standard.

3.6.4 Structure du code irrégulier

Concernant l'évaluation de l'algorithme de décodage EFAP-BP sur les codes irréguliers, nous avons choisi un code irrégulier de la famille PEG de Mackay de longueur $N = 504$ et de rendement $R = 1/2$, la figure 3.10 montre la distribution des éléments non nuls dans le code.

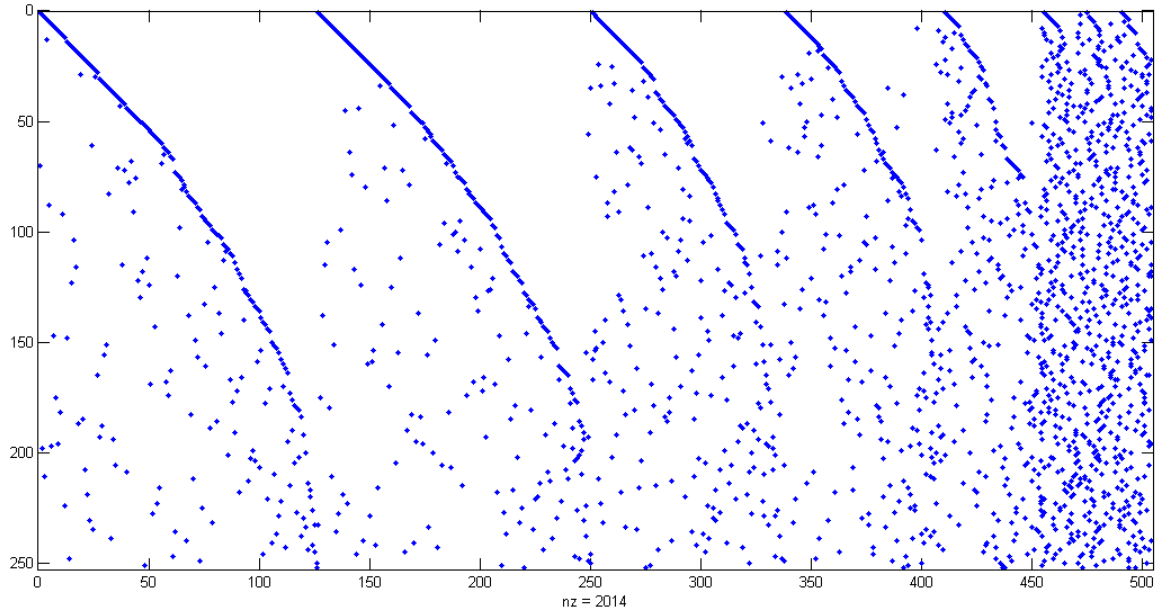


FIGURE 3.10 – illustration de la matrice de contrôle de parité PEG_{irreg} $(N = 504, R = 1/2)$, les points bleus correspondent aux positions de la valeur 1 dans la matrice de contrôle de parité.

Les proportions des branches du graphe connectées au nœuds de variable (respectivement nœuds de contrôle) pour le code présenté au dessus sont décrites par les équations suivantes :

$$\lambda(x) = 0.2393x + 0.21x^2 + 0.035x^3 + 0.1216x^4 + 0.0139x^6 + 0.0007x^{13} + 0.3724x^{14} \quad (3.28)$$

$$\rho(x) = 0.83x^7 + 0.079x^8 + 0.0873x^6 \quad (3.29)$$

Avec utilisation de l'équation 3.14 de la section 3.4 on calcule la moyenne de connectivité pour les N nœuds de variables $n_{D,irreg} = 4$.

Avec utilisation de l'algorithme de calcul des cycles courts [83] on trouve que le cycle le plus court $g = 6$, le nombre des cycles courts de taille $g = 6$ est $N_g = 13244$ et la moyenne des nœuds de contrôle est $\mu_g = 315.33$. Après le test des trois algorithmes on trouve le résultat :

Concernant l'algorithme VFAP-BP, le facteur de pondération ρ_v est déterminé par application de l'équation 3.13 $\rho_v = 2/n_{D,irreg} = 0.5$, pour un nœud de contrôle traversé par plus de $\mu_g = 315.33$ cycle 6, il sera pondéré par la valeur 0.5 sinon il sera pondéré par la valeur 1, (voir la figure 3.11 gauche).

Le facteur de pondération pour l'algorithme EFAP-BP, est calculé avec la fonction (équation 3.21) déjà présentée dans le section 3.5. La figure 3.11 droite montre que le facteur ρ_m varie de 0.91 à 0.6 selon le nombre des cycles courts traversant chaque nœud de contrôle s_m.

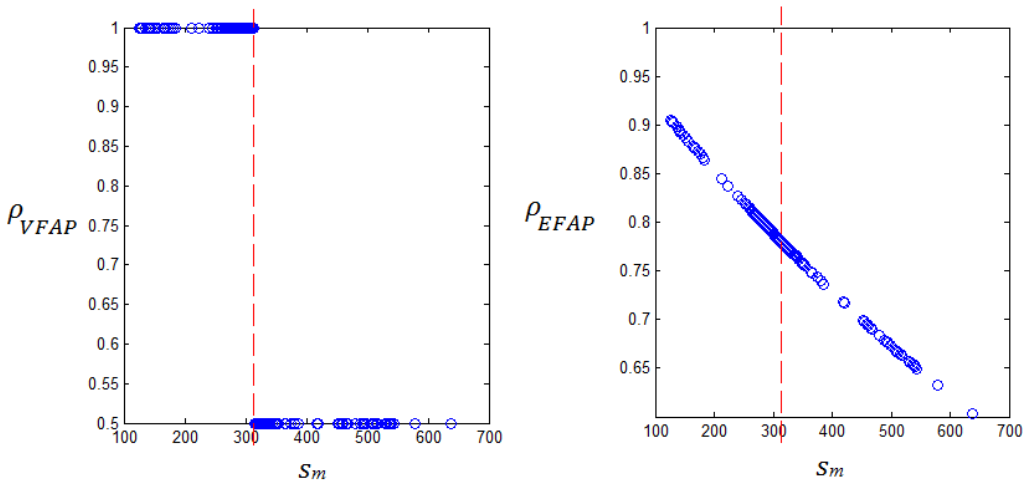


FIGURE 3.11 – Valeurs du facteur ρ en fonction du nombre des cycles courts traversant chaque nœud de contrôle s_m pour le code PEG irrégulier.

3.6.5 Convergence du code irrégulier

L'évaluation de rapidité de convergence des trois algorithmes EFAP-BP, VFAP-BP, et le BP standard pour le code irrégulier PEG est étudié en fixant le SNR sur trois valeurs $[2, 4, 6]dB$ et en variant les itérations de 1 à 15, les résultats de simulations présenté dans la figure 3.12 ont montré que l'algorithme EFAP-BP est meilleur en convergence par rapport à l'algorithme VFAP-BP et le BP standard. On remarque aussi d'après les figures 3.8 et 3.12 que l'algorithme EFAP-BP converge plus rapidement pour le cas de code irrégulier de $g = 6$ que pour le cas de code régulier de $g = 8$.

3.6.6 Performances du code irrégulier

Concernant l'évaluation des performances BER/SNR des décodeurs EFAP-BP, VFAP-BP et le BP standard en utilisant le code LDPC irrégulier pour les deux limites d'itérations

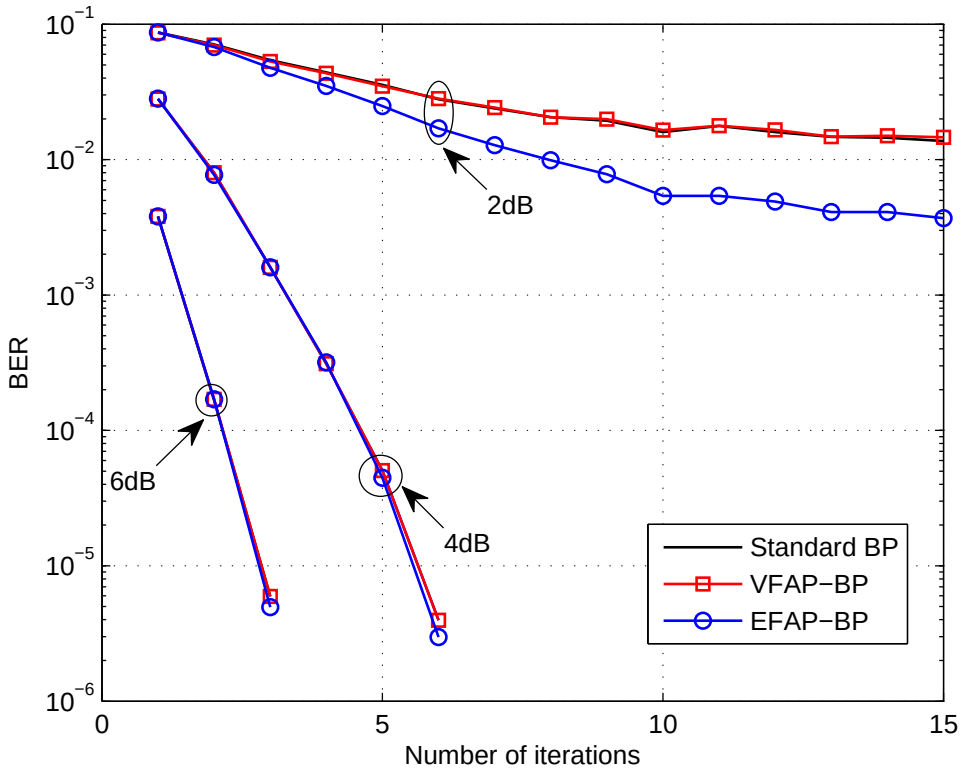


FIGURE 3.12 – comparaison du comportement de convergence pour les algorithmes EFAP-BP, VFAP-BP et BP pour le code LDPC irrégulier de dimension $(252, 504)$, quand le SNR prend les valeurs $SNR = [2, 4, 6]$ avec $k = 4$.

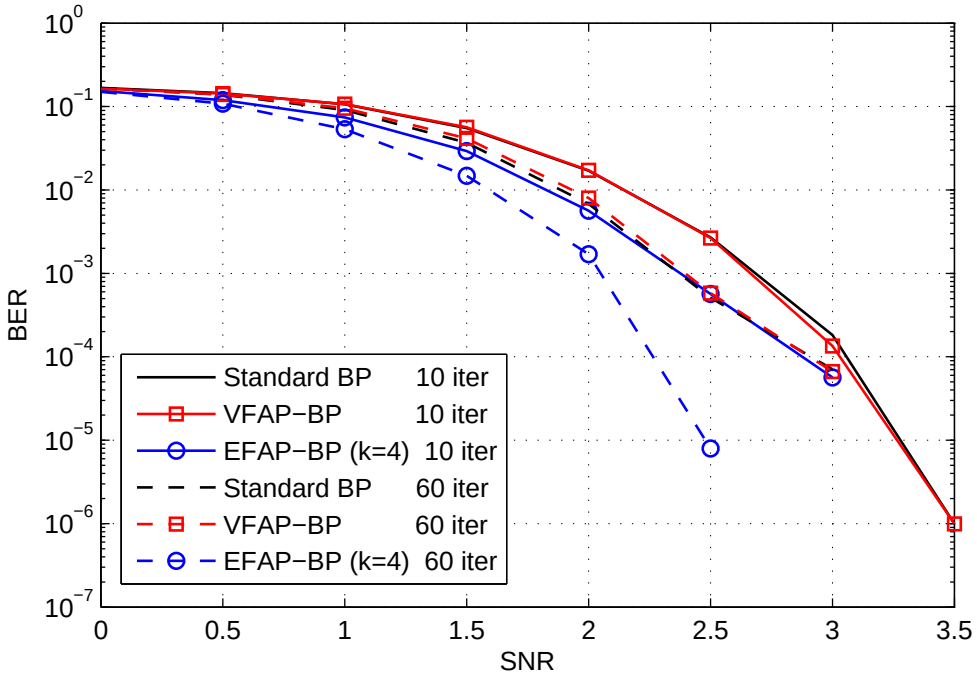


FIGURE 3.13 – Comparaison des performances d'erreurs de l'algorithme EFAP-BP, l'algorithme VFAP-BP et BP standard pour le code LDPC irrégulier de paramètres $(252, 504)$ avec le maximum d'itération $I_{max} = (10, 60)$.

$I_{max1} = 10$ et $I_{max2} = 60$, la figure 3.13 a montré que l'algorithme EFAP-BP est meilleur en correction des erreurs avec un gain 0.35 dB par rapport à l'algorithme VFAP-BP et au BP standard à un BER= $10^{-}3$ et lorsque le nombre des itérations maximum est limité sur $I_{max1} = 10$, on remarque aussi que le BP et le VFAP-BP ont donné le même résultat.

Quand la limité des itérations est fixé sur $I_{max2} = 60$, la simulation a montré que l'algorithme VFAP-BP et le BP standard ont donné presque le même résultat, le EFAP-BP est meilleur par rapport au VFAP-BP et le BP standard avec un gain de 1dB quand le BER=10^{-4}.

L'évaluation des performances BER/SNR a montré que l'algorithme EFAP-BP est plus efficace pour le code irrégulier de $g = 6$ par rapport au code régulier de $g = 8$.

3.7 Conclusion

Dans ce chapitre nous avons développé un nouvel algorithme de décodage du code LDPC qui exploite une nouvelle stratégie de pondération et la connaissance de structure cyclique du code dans le but de réduire l'effet négatif des cycles courts en respectant la complexité du décodage.

L'étude comparative a montré que l'algorithme proposé EFAP-BP est meilleur en performance de correction des erreurs et en rapidité de convergence par rapport aux algorithmes VFAP-BP et le BP standard pour les deux codes étudié régulier et irrégulier à un nombre limité des itérations, ce qui est souhaitable dans les systèmes de communication qui exigent une faible latence.

L'algorithme BP est une solution efficace pour résoudre le problème d'inférence, mais son inconvénient majeur est la complexité du traitement durant le processus itératif que les concepteurs le trouve toujours un obstacle qui empêche l'implémentation, pour cela nous avons proposé une nouvelle approche basé sur la connaissance des cycles courts pour améliorer les performances de l'algorithme *Min Sum* qui est très utilisé dans les applications réels, les détails de cette approche sont présenté dans le prochain chapitre.

Chapitre 4

Nouvelle méthode pour améliorer les performances du décodeur MS normalisé en utilisant un code LDPC de la norme WIMAX

4.1 Introduction

L'algorithme *Sum-Product* déjà présenté dans le chapitre 2 est une méthode optimale et puissante pour résoudre le problème d'inférence [15], l'inconvénient majeur de cet algorithme est la complexité de décodage et le temps d'exécution.

Plusieurs études ont été proposées dans la littérature pour améliorer les performances et réduire la complexité de cet algorithme. L'algorithme *Min-Sum* [14] [79] déjà décrit dans le chapitre 2 est une solution sous-optimale permet de réduire la complexité au prix de performance. Cette pénalité de performance est généralement considérée comme étant causée par l'approximation utilisée dans l'étape de traitement de nœud de contrôle, qui se traduit par une surestimation des messages allant des nœud de contrôle vers les nœuds de variable. Afin d'atténuer ce problème, plusieurs méthodes de correction ont été proposées dans la littérature [79] [92] [93] [94] [95] [96].

Les décodeurs *Normalized Min Sum* (NMS) et *Offset Min-Sum* (OMS) [92] [93] sont des versions légèrement modifiées du décodeur MS qui reposent sur l'utilisation soit d'une normalisation ou d'une correction pour compenser la surestimation des messages allant

des nœuds de contrôle vers les nœuds de variable. Récemment, plusieurs approches sont développées afin de concevoir des codes LDPC sans cycles courts [97], ou pondérer le graphe de Tanner, afin de lutter contre le problème du cycle court [23] [84] [90] [98]. Car pendant le processus de décodage itératif, l'existence des cycles courts dans le code affecte l'indépendance des messages extrinsèques échangées [16] [18], ce qui détruit globalement les performances et empêche la convergence des messages [85].

Dans ce chapitre on propose une nouvelle version de l'algorithme NMS qui permet de réduire l'effet des cycles courts. la nouvelle version est appelé *Girth Aware Normalized Min Sum* (GA-NMS), après une étude de la structure cyclique du code, la nouvelle méthode proposé GA-NMS consiste à attribuer à chaque nœud de parité un facteur dépendant du cycles courts qui le traverse dans le graphe de tanner. La raison derrière le choix des algorithmes MS et NMS dans cette étude revient au fait que l'algorithme de décodage MS est très utilisé dans les systèmes de communication réels, et l'algorithme NMS est version pondéré de l'algorithme MS par un facteur d'optimisation constant, on suppose dans cette étude que ce facteur d'optimisation peut prendre en considération la structure cyclique du code.

L'évaluation du nouveau algorithme proposé est faite en utilisant un code LDPC de la norme WIMAX du standard IEEE P802.16e [74], les résultats de comparaison ont montré que l'algorithme proposé est meilleur par rapport aux algorithmes de décodage NMS et MS en taux d'erreurs et rapidité de convergence.

La deuxième section de ce chapitre est consacré pour définir une méthode simple pour le calcul du nombre du cycle court qui traverse chaque nœud ensuite la troisième partie comporte une description le l'algorithme NMS dont nous avons besoin dans la suite de ce chapitre, la quatrième section comporte une description détaillé de l'algorithme proposé GA-NMS, les résultats d'évaluation des trois algorithmes GA-NMS, NMS et le MS sont présentés dans la cinquième partie, enfin conclusion.

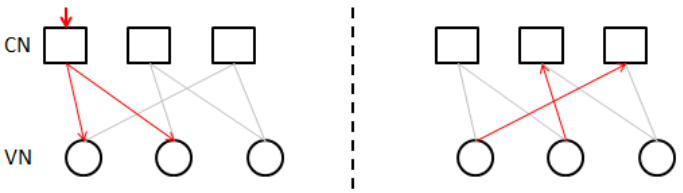
4.2 Détermination du cycle court traversant chaque nœud de contrôle

L'amélioration proposé dans le présent chapitre, nécessite la connaissance du cycle le plus court qui passe à travers chaque nœud de contrôle dans le graphe de tanner,

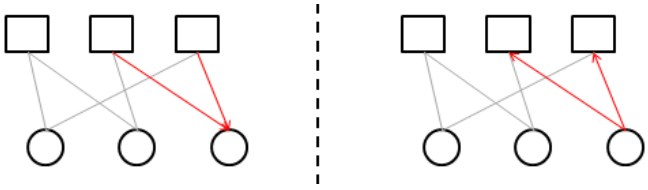
l'algorithme proposé par Thomas Halford [83] déjà présenté dans le chapitre 3 est efficace pour faire ce dernier calcul, en dessous nous proposons une méthode simple qui permet de déterminer le périmètre du cycle le plus court qui passe à travers un nœud de contrôle $G(m)$ dans le graphe de tanner.

Le principe est basé sur le fait d'envoyer la valeur 1 à travers un nœud de contrôle C_m du graphe et attendre son retour après un certain nombre d'itérations, le passage d'un nœud de contrôle vers un nœud de variable $b_{m,n}$ est calculé en prenant le maximum des valeurs provenant des nœuds de variable d'une façon extrinsèque, de la même façon on procède pour le calcul du passage d'un nœud de variable vers un nœud de contrôle $a_{m,n}$.

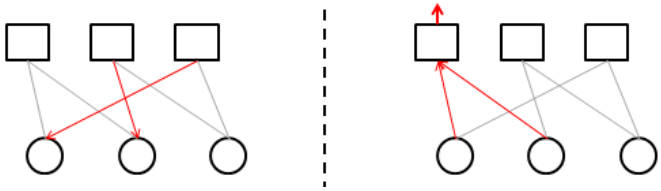
L'exemple présenté sur la Figure 4.1 consiste à mettre tout le système à zéros ensuite envoyer la valeur 1 à travers le nœud 1, après 3 itérations on obtient la valeur 1 à la sortie du nœud 1, le taille du cycle court est deux fois le nombre d'itérations, on peut déduire donc que le cycle court qui traverse le nœud 1 est de dimension $G(1) = 6$.



(a) Itération 1



(b) Itération 2



(c) Itération 3

FIGURE 4.1 – Modèle descriptif de la méthode de calcul du cycle court passant à travers un nœud.

La procédure de calcul est détaillé sur l'algorithme en dessous :

Algorithm 6 Algorithme de calcul des cycles courts traversant chaque nœud de contrôle

pour $m = 1$ to M **faire**
  Initialization :

  pour $m' = 1$ to M, $m' \neq m$ **faire**
    $\forall n \in \mathcal{H}_V(m')$ $\quad b_{m',n} = 0$
  fin pour
  $\forall n \in \mathcal{H}_V(m)$ $\quad b_{m,n} = 1$
  $it = 1$
  Procedure :
  tant que $(\forall n \in \mathcal{H}_V(m),\ a_{m,n} = 0)$ **faire**
    pour $n = 1$ to N **faire**
      $\forall m \in \mathcal{H}_C(n)$ $\quad a_{m,n} = \max_{m' \in \mathcal{H}_C(n) \setminus m} b_{m',n}$
    fin pour
    pour $m = 1$ to M **faire**
      $\forall n \in \mathcal{H}_V(m)$ $\quad b_{m,n} = \max_{n' \in \mathcal{H}_V(m) \setminus n} a_{m,n'}$
    fin pour
    $it = it + 1$
  fin tant que
  $G(m) = it \times 2$
fin pour

4.3 Algorithme NMS

L'algorithme sous optimale *Min Sum* peut être amélioré par traitement des messages en sortie du nœud de contrôle. L'approximation du minimum surestime les messages en sortie du nœud de contrôle. Ainsi une pondération, ou une correction, peut être apportée sur chaque message afin d'atténuer l'effet de la surestimation. L'algorithme est appelé *Normalized Min Sum* dans le cas d'une pondération [92] et *offset Min-Sum* dans le cas d'une correction [93]. Une multitude de techniques sont proposées dans la littérature pour le choix des facteurs de correction.

L'algorithme NMS [92] est une version quasi optimale de l'algorithme sous optimale *Min Sum* basé sur la multiplication de tous les nœuds de contrôle avec un facteur constant calculé empiriquement $0 < \gamma < 1$ pour atteindre des performances proche à l'algorithme *Sum Product*, les détails du décodage NMS sont présenté dans l'algorithme 7 en dessous.

4.3.1 Facteur d'optimisation γ

La valeur du facteur d'optimisation varie selon la valeur du *Signal to Noise Ratio* (SNR). Pour une valeur particulière du SNR on cherche le facteur γ qui cause le minimum

de FER ou BER. La figure 4.2 montre la variation du FER suivant le facteur γ, avec la valeur du SNR choisie SNR=2.5 dB, et un nombre des itérations maximum fixé sur $I_{max} = 50$, la valeur du FER minimale obtenu est $\gamma = 0.85$, dans la suite de ce chapitre, le facteur d'optimisation de l'algorithme de décodage NMS pour le code LDPC de la norme WIMAX du standard IEEE P802.16e est $\gamma = 0.85$.

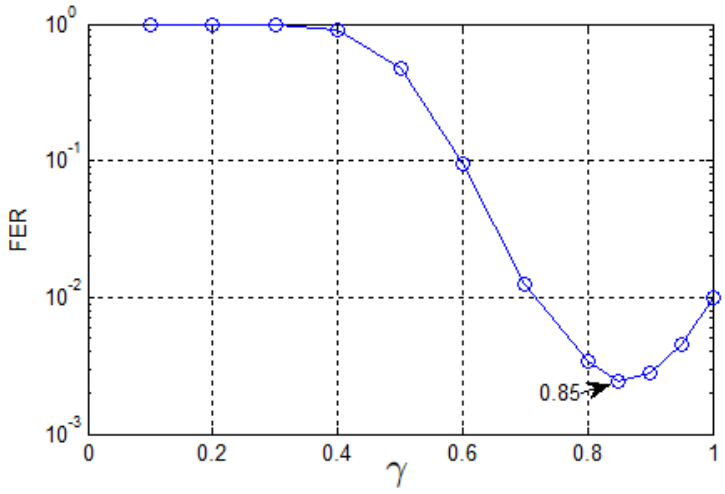


FIGURE 4.2 – Variation du FER en fonction du facteur d'optimisation γ quand le SNR=2.5 dB, en utilisant un code LDPC de la norme WIMAX du standard IEEE P802.16e de longueur $N = 576$ et de rendement $R = 1/2$

Algorithm 7 Algorithme Normalized Min Sum

Entrée : $\underline{y} = (y_1, ..., y_N) \in Y^N$
Sortie : $\underline{\hat{x}} = (\hat{x}_1, ..., \hat{x}_N) \in \{0,1\}^N$

Initialisation

pour tout $n = 1, ..., N$ **faire**

$$\lambda_n = log\left(\frac{P(x_n = 0|y_n)}{P(x_n = 1|y_n)}\right);$$

pour tout $n = 1, ..., N$ et $m \in \mathcal{H}_C(n)$ **faire** $\alpha_{m,n} = \lambda_n$;

Processus itératif

▷ Mise à jour des nœuds CN.

pour tout $m = 1, ..., M$ et $n \in \mathcal{H}_V(m)$ **faire**

$$\beta_{m,n} = \gamma \left(\prod_{n' \in \mathcal{H}_V(m)\backslash n} sgn\left(\alpha_{m,n'}\right)\right)\left(\min_{n' \in \mathcal{H}_V(m)\backslash n} |\alpha_{m,n'}|\right);$$

▷ Mise à jour des nœuds VN.

pour tout $n = 1, ..., N$ et $m \in \mathcal{H}_C(n)$ **faire**

$$\alpha_{m,n} = \lambda_n + \sum_{m' \in \mathcal{H}_C(n)\backslash m} \beta_{m',n};$$

▷ Calcul de l'information à posteriori.

pour tout $n = 1, ..., N$ **faire**

$$\tilde{\lambda}_n = \lambda_n + \sum_{m \in \mathcal{H}_C(n)} \beta_{m,n};$$

▷ Prise de décision.

pour tout $n = 1, ..., N$ **faire**

$$\hat{x}_n = \frac{1 - sgn(\tilde{\lambda}_n)}{2};$$

si $\hat{x}$ est correcte, arrêter le processus itératif.

fin du processus itératif

4.4 L'algorithme GA-NMS

Comme déjà décrit dans le chapitre précédant l'existence d'un cycle court crée une dépendance entre les messages extrinsèque, cela empêche la correction de l'information durant le processus itératif, nous avons appliqué la méthode VFAP décrite sur le chapitre précédant sur l'algorithme de décodage *Min Sum* [98], les résultats ont montrés que la méthode VFAP a permis de résoudre un peu le problème provoqué par les cycles courts et amélioré les performances par rapport au MS, sur la base de ce résultat nous avons

développé dans ce chapitre une nouvelle version de l'algorithme NMS pour un code LDPC de la norme WIMAX appelé *Girth Normalized Min Sum* [99], ce dernier consiste à mettre un facteur d'optimisation dépendant du cycle court passant à travers chaque nœud de contrôle dans le graphe de tanner figure 4.3, les nouvelles valeurs du facteur d'optimisation γ=$\{\gamma_1,\gamma_2,\ldots,\gamma_m\}$ sont déterminé d'une façon empirique.

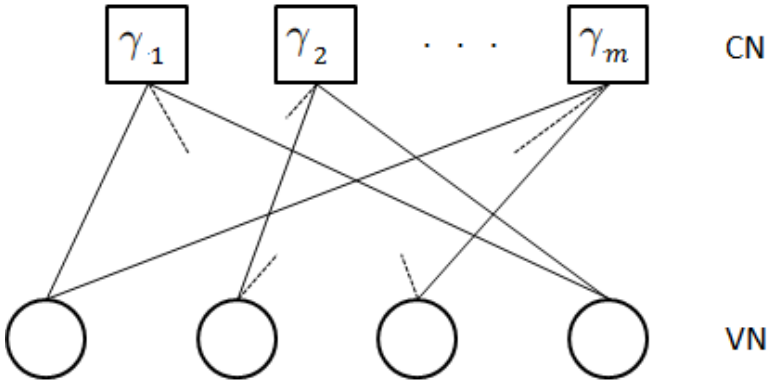


FIGURE 4.3 – Modèle graphique.

4.4.1 Structure du code LDPC de la norme WIMAX

Le code utilisé dans ce chapitre est un code LDPC de la norme WIMAX du standard IEEE P802.16e [74] avec un rendement $R = 1/2$, et de longueur $N = 576$, le code est constitué de (12×24) sous bloc de dimension $z \times z$ comme présenté dans la figure 4.4, les blocs vide représentent des matrices nulles, les blocs avec la valeur zéro sont des matrices identité, pour ceux qui comportent une valeur vont subir des rotations circulaires selon la valeur mentionné dans le bloc. Dans notre cas la valeur z choisie est de taille $z = 24$, la matrice obtenu est de dimension (288,576).

La distribution des éléments non nuls dans le code est présenté dans la figure 4.5, les points en bleu représentent les éléments non nuls.

	94	73						55	83			7	0									
	27				22	79	9				12		0	0								
			24	22	81		33				0			0	0							
61		47						65	25						0	0						
		39				84			41	72						0	0					
				46	40		82				79	0					0	0				
		95	53						14	18								0	0			
	11	73				2			47										0	0		
12				83	24		43				51									0	0	
					94		59			70	72										0	0
		7	65					39	49												0	0
43					66		41				26	7										0

FIGURE 4.4 – Structure en bloc de la matrice de parité du code LDPC avec un rendement 1/2 dans le standard IEEE P802.16e.

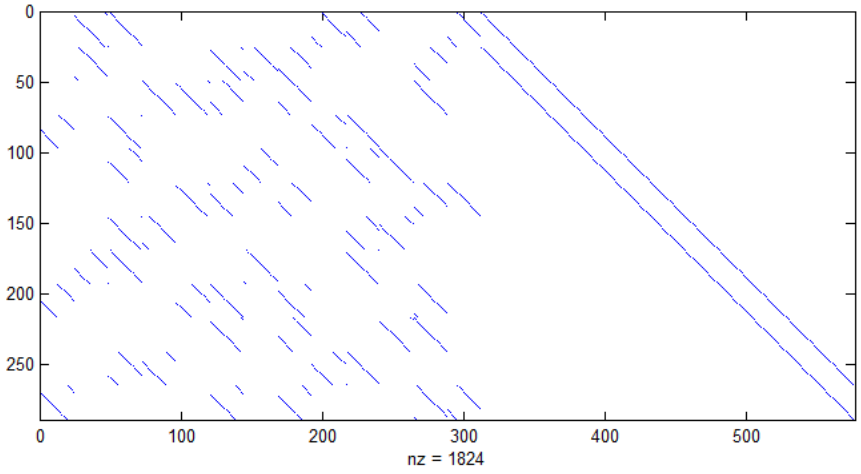


FIGURE 4.5 – Structure de la matrice H du standard IEEE P802.16e, avec les paramètres $(N = 576, R = 1/2)$

4.4.2 Structure cyclique du code

La première étape de l'algorithme GA-NMS est la connaissance de la structure cyclique du code, l'étude d'un code LDPC de la norme WIMAX du standard IEEE P802.16e a montré qu'on peut classer les nœuds de contrôle selon trois catégories, 144 nœuds de contrôle traversé par des cycles 4, 120 nœuds de contrôle dont le cycle le plus court est 6 et 24 nœuds de contrôle dont le cycle le plus court est 8 comme indiqué dans le tableau en dessous.

Sur la base de ce résultat, nous supposons que les nœuds de contrôle reliés à un cycle court 4 ont besoin d'un facteur d'optimisation γ_1 inférieur aux facteurs qui seront attribué au nœud traversé par les cycles courts 6 et 8 $\{\gamma_2, \gamma_3\}$, les nœuds de contrôle dont le cycle le plus court est 6 devrait être multiplié par un facteur d'optimisation $\gamma_1 < \gamma_2 < \gamma_3$,

TABLEAU 4.1 – Structure cyclique de la matrice H

Taille du cycle court	Nombre de cycle court	Nombre de CN	Nombre de VN
cycle 4	96	144	168
cycle 6	528	120	144
cycle 8	7344	24	264

finalement les nœuds de contrôle avec le cycle le plus court est 8 devrait se situer entre γ_2 et 1, Dans le paragraphe suivant nous faisons la détermination des facteurs d'optimisation $\{\gamma_1, \gamma_2, \gamma_3\}$ pour chaque classe de nœuds de contrôle.

4.4.3 Facteurs d'optimisation pour l'algorithme GA-NMS

Le facteur d'optimisation γ_1 qu'il sera attribué aux nœuds de contrôle qui sont traversé par le cycle court 4 est calculé en fixant γ_2 et γ_3 sur la constante 0.85 déjà calculé pour le NMS et en variant γ_1 de 0.1 vers 1, le même principe est utilisé pour la détermination de γ_2 et γ_3, la simulation est effectué en utilisant le code LDPC de la norme WIMAX, avec $I_{max} = 50$ et SNR=2.5 dB, les résultats (figure en dessous) ont montré que les facteurs $\{\gamma_1, \gamma_2, \gamma_3\}$ prennent les valeurs $\{0.8, 0.9, 0.95\}$.

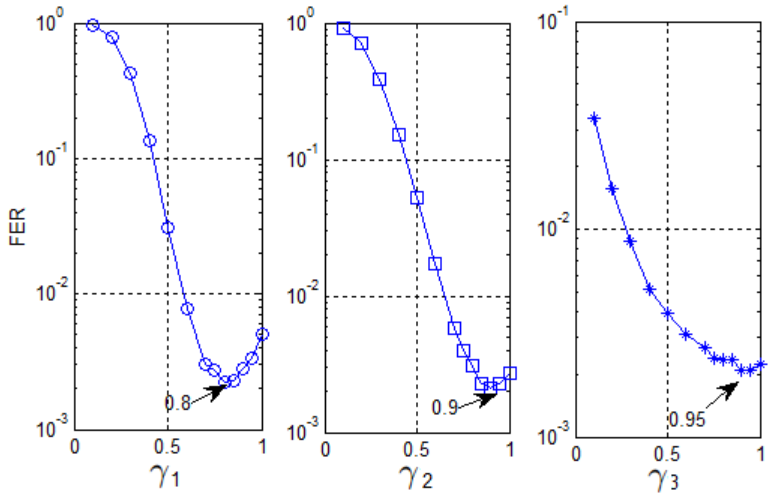


FIGURE 4.6 – Les valeurs $\gamma_1, \gamma_2, \gamma_3$ trouvé empiriquement. quand SNR=2.5 dB et $I_{max} = 50$.

Le reste de l'algorithme GA-NMS est similaire à l'algorithme NMS sauf ici le facteur d'optimisation est variable comme indiqué dans l'algorithme 8 en dessous.

Algorithm 8 Algorithme Girth Aware Normalized Min Sum

. . .

Processus itératif
pour tout $m = 1, ..., M$ est $n \in \mathcal{H}_V(m)$ **faire**

$$\alpha_{m,n} = \gamma_m \times \left(\prod_{n' \in \mathcal{H}_V(m) \backslash n} sgn\left(\beta_{m,n'}\right) \right) \min_{n' \in \mathcal{H}_V(m) \backslash n} |\beta_{m,n'}|$$

. . .

Fin du processus itératif

4.5 Résultats & Simulations

4.5.1 Comportement de convergence

Dans la région moins bruité SNR=[3, 4, 5, 6] on compare le comportement de convergence des trois algorithmes GA-NMS, NMS, MS en utilisant un code LDPC de la norme WIMAX avec un nombre des itérations variant entre 1 et 18 figure 4.7. la simulation a montré que le GA-NMS converge rapidement par rapport aux algorithmes NMS et le MS, ce résultat est du au fait que les facteurs obtenus empiriquement permettent de réduire l'effet des cycles courts.

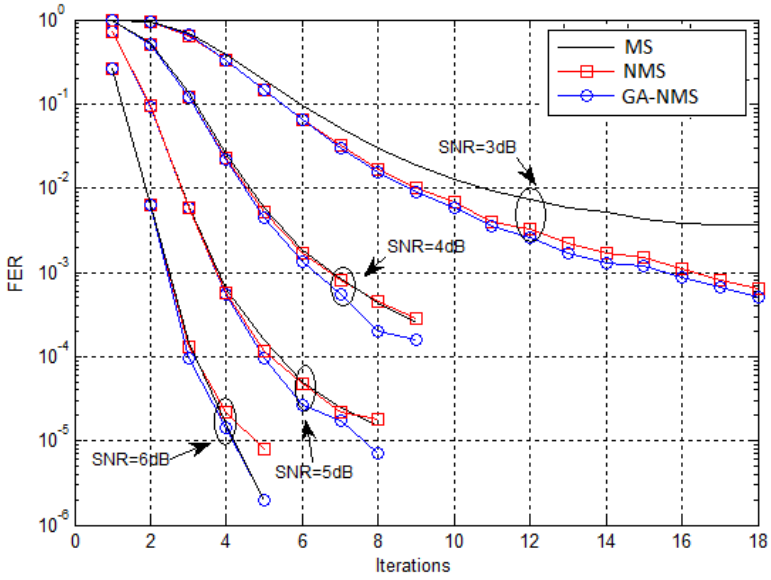


FIGURE 4.7 – Comparaison des comportements de convergence des algorithmes GA-NMS, NMS et MS pour le décodage du code LDPC dans le standard IEEE P802.16e, où le SNR est égale à 3 dB, 4 dB, 5 dB et 6 dB

4.5.2 Performances de correction d'erreurs

Cette étude a été réalisé en considérant un canal AWGN, avec le nombre d'itérations maximum est fixé sur 50 itérations, le BER et le FER des décodeurs GA-NMS, NMS, et MS ont été comparé, le résultat de simulation figure 4.8 a montré que le GA-NMS est meilleur en correction des erreurs par bit, par rapport au NMS avec un gain de 0.1 dB et au MS avec un gain de 0.65 dB quand le BER=10^{-6}, le GA-NMS est meilleur aussi en correction des erreurs par message par rapport au NMS avec un gain de 0.15 dB et au MS avec un gain de 0.25 dB quand le FER=10^{-4}.

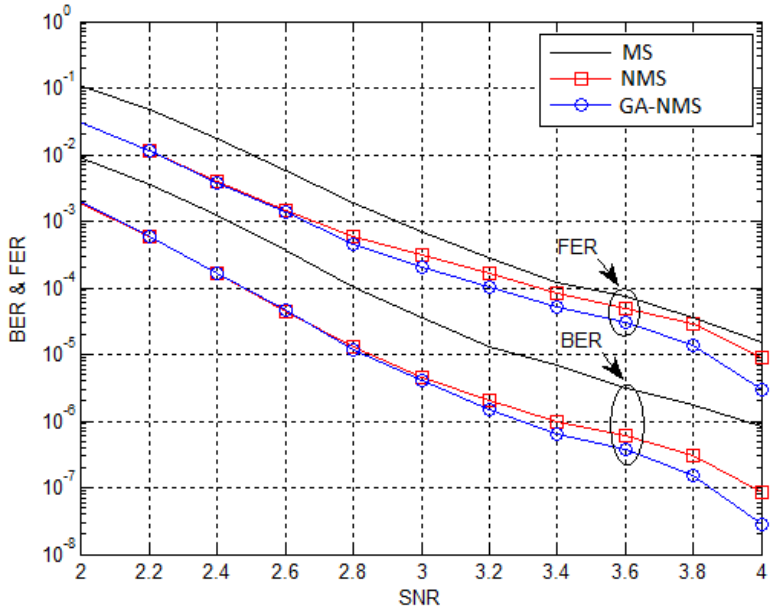


FIGURE 4.8 – Comparaison des performances BER&FER entre les algorithmes GA-NMS, NMS et le MS pour le décodage du code LDPC de la norme WIMAX, avec 50 itérations max

4.6 Conclusion

Dans ce chapitre nous avons développé une nouvelle version de l'algorithme NMS qui tient en compte la structure cyclique du code pour améliorer la qualité de décodage avec un minimum de complexité, l'idée derrière le choix du *Min Sum* et ces dérivés revient au fait que ce dernier est très utilisé dans les systèmes de communication. Le nouveau algorithme résultant est appelé GA-NMS, ce dernier a été évalué en utilisant un code LDPC de la norme WIMAX du standard IEEE P802.16e de taille N=576 et de rendement R=1/2, les résultats de simulation ont montré que l'algorithme proposé GA-NMS est meilleur en correction des erreurs et en rapidité de convergence, Ce nouveau algorithme est compatible à tous types de codes qui ont une structure cyclique similaire au code de la norme WIMAX étudié dans ce chapitre.

Conclusion & Perspectives

Conclusion Générale

Cette thèse est consacrée à l'étude et l'optimisation des décodeurs pour les codes LDPC. Par leurs bonnes performances, leurs facilité d'optimisation de leurs paramètres et leurs simplicité de codage, nous nous sommes naturellement intéressés aux codes LDPC.

Ce travail constitue donc une approche originale d'etude de structures de codes et d'algorithmes de décodage pour les codes LDPC.

Au début de ce document, nous avons présentés la théorie de l'information et les éléments constituant une chaîne de transmission, nous avons défini tous ce qui est en relation avec la communication numérique afin de mieux positionner les travaux de cette thèse. Par la suite, nous avons donné une vision complète à propos de la théorie de codage de canal, nous avons présenté d'une manière détaillée tous les côtés des codes LDPC et ses algorithmes de décodage.

Les apports de nos travaux s'articulent autour de deux types de décodage : un décodage basé sur l'algorithme BP et un décodage basé sur l'algorithme MS, qui sont présenté dans les deux derniers chapitres.

Le chapitre 3 constitue la première contribution dans cette thèse puisqu'il introduit l'algorithme proposé EFAP-BP. Après avoir présenter un état de l'art sur les cycles court, et comment on peut les identifier dans le code, ainsi une description de l'algorithme VFAP-BP, nous avons proposé un nouvel algorithme de décodage des codes LDPC qui tient en compte la structure cyclique du code pour l'amélioration des performances. Ce nouvel algorithme est compatible aux codes réguliers et irréguliers. Les résultats obtenus et présentés dans ce mémoire en les comparants avec les algorithmes existant dans la littérature nommé VFAP-BP et BP standard, montrent qu'on a pu réduire le taux d'erreur binaire, et améliorer la convergence avec un minimum de complexité.

Toujours dans le cadre de résoudre le problème de surestimation crée par les cycles court lors du décodage itératif et d'améliorer les performances de décodage nous avons développé l'algorithme GA-NMS présenté dans le chapitre 4, ce dernier est basé sur l'algorithme NMS et la connaissance de la structure cyclique du code, l'idée derrière le choix de l'algorithme NMS revient au fait que ce dernier est très sollicité dans les systèmes de communication réels, et qu'il dispose d'un facteur d'optimisation qui peut être encore optimisé, l'évaluation du nouvel algorithme a été effectué en utilisant un code LDPC dans le standard IEEE P802.16e, les résultats de simulation ont montrés que l'algorithme GA-NMS est meilleur en correction des erreurs par bit avec un gain de 0.1 dB par rapport au NMS et au MS avec un gain de 0.65 dB quand le $BER = 10^{-6}$, ainsi en correction des erreurs par mot de code avec un gain de 0.15 dB par rapport au NMS et de 0.25 dB par rapport au MS quand le $FER = 10^{-4}$ avec le nombre des itérations maximum est fixé sur 50 tout avec un minimum de complexité.

Perspectives et travaux futur

Le travail réalisé dans cette thèse consiste en un point de départ à toute une série de travaux possibles. Plusieurs perspectives sont envisageables.

— Étude des différentes versions modifié de l'algorithme *Min Sum* qui existe dans la littérature, et voir laquelle est plus fiable pour prendre en considération la structure cyclique du code.

— Trouver d'autre codes qui ont une structure cyclique similaire au code présenté dans le chapitre 4, les étudier et améliorer leurs performances en tenant compte leurs structures cycliques.

— Etude et implémentation des codes LDPC non binaire.

Bibliographie

[1] C. de la MAE. Des peintures rupestres d'afrique australe. [Online]. Available : http://mae.hypotheses.org/3239

[2] S. E. B. MORSE, "Improvement in the mode of communicating information by signals by the application of electromagnetism," U.S. Patent US 1647 A, 20, 1820.

[3] W. H. BRATTAIN, "Laboratory notebook, case 38139-7," *Bell Laboratories archives*, 15 December 1947.

[4] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding : turbo-codes." Geneva : IEEE ICC '93, May 1993, p. 1064 – 1070.

[6] R. Gallager, "Low density parity check codes," Ph.D. dissertation, Cambridge, MASS, July 1963.

[7] D. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, p. 1645, aug 1996.

[8] M. Sipser and D. A. Spielman, "Expender codes," *IEEE Trans. Information Theory*, vol. 42, pp. 1710–1722, 1996.

[9] D. J. C. MacKay, "Gallager codes that are better than turbo codes." Monticello : 36th Allerton Conf. Communication, Control, and Computing, Sept. 1998.

[10] Y. Kou, S. Lin, and M. Fossorier, "Construction of low density parity check codes : A geometric approach." Brest, France : 2nd Int. Symp. Turbo Codes and Related Topics, Sept. 2000.

[11] T. J. Richardson, M. A. Shokrollahi, and R. Urbanke, "Design of capacity approaching irregular low-density parity-chek codes," *IEEE Trans. Information Theory*, vol. 47, pp. 619–637, Feb 2001.

[12] M. Davey and D. MacKay, "Low density parity check codes over GF(q)," *IEEE Communications Letters*, vol. 2, pp. 1089–7798, Jun 1998.

[13] S. Lin, Y. Kou, and M. Fossorier, "Low density parity check codes construction based on finite geometries." San Francisco, Calif : GLOBECOM 2000, 2000.

[14] M. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagations," *IEEE Transaction Communications*, vol. 47, p. 673–680, May. 1999.

[15] J. Pearl, *Probabilistic Reasoning in Intelligent Systems : Networks of plausible inference*. Morgan Kaufmann, 1988.

[16] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Information Theory*, vol. 45, pp. 399–432, 1999.

[17] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum–product algorithm," *IEEE Trans. Information Theory*, vol. 47, pp. 498–519, Feb 2001.

[18] R. Lucas, M. Fossorier, Y. Kou, and S. Lin, "Iterative decoding of one-step majority logic decodable codes based on belief propagation," *IEEE Transactions on Communications*, vol. 48, pp. 931–937, June 2000.

[19] M. G. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs." New York, NY, USA : chez Proc. of the 30th ACM, STOC, 1998, pp. 249–258.

[20] T. J. Richardson and R. Urbanke, "The capacity of low-density-parity-check codes under message-passing decoding," *IEEE Trans. Information Theory*, vol. 47, pp. 599–618, Feb 2001.

[21] A. Shokrollahi and R. Storn, "Design of efficient erasure codes with differential evolution," in *2000 IEEE International Symposium on Information Theory (Cat. No.00CH37060)*, 2000, pp. 5–.

[22] S. Y. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *IEEE Communications Letters*, vol. 47, pp. 58–60, 2001.

[23] J. Liu and R. C. de Lamare, "Low-latency reweighted belief propagation decoding for LDPC codes," *IEEE COMMUNICATIONS LETTERS*, vol. 16, pp. 1660–1663, October 2012.

[24] J. A. FLEMING, "Improvements in instruments for detecting and measuring alternating electric currents," Britsh Patent 24/850, 16, 1904.

[25] C. Berrou, "Error-correction coding method with at least two systematic convolutional codings in parallel, corresponding iterative decoding method, decoding module and decoder," aug 29 1995, uS Patent 5,446,747. [Online]. Available : http://www.google.com/patents/US5446747

[26] E. PIRIOU, "Apport de la modélisation et de la synthèse haut niveau dans la conception d'architecture flexible dédiée aux turbocodes en blocs," Ph.D. dissertation, ELEC - Dépt. Electronique (Institut Mines-Télécom-Télécom Bretagne-UEB), UBS - Université de Bretagne Sud (UBS)., 2007.

[27] R. G. Gallager, "Low-density-parity-check," *IRE Trans. Information Theory*, vol. 2, pp. 21–28, Jan 1962.

[28] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1995.

[29] G. Kabatiansky, E. Krouk, and S. Semenov, *Error Correcting Coding and Security for Data Networks*. John Wiley & Sons Ltd, 2005.

[30] R. C. BOSE and D. K. RAY-CHAUDHURI, "On a class of error correcting binary group codes." *Information and Control*, vol. 3, pp. 68–79, March 1960.

[31] A. HOCQUENGHEM, "Codes correcteurs d'erreurs." *Chiffers*, vol. 2, p. 147–156, 1959.

[32] G. COHEN, J.-L. DORNSTETTER, P. GODLEWSKI, and J. C. BIC, *Codes correcteursd'erreurs : une introduction au codage algébrique. Collection technique et scientifique des télécommunications*. Masson, Paris Milan Barcelone, 1992.

[33] K. M. CHEUNG and F. POLLARA, *Phobos Lander Coding System : Software and Analysis, Rapport technique*. TDA progress report, 1988.

[34] *Sandforce sf2000 reference design and evaluation ssd, Rapport technique.* Seagate, 2014.

[35] I. S. REED and G. SOLOMON, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, p. 300–304, 1960.

[36] T. D. ROSSING, "The compact disc digital audio system," *The Physics Teacher*, vol. 25, p. 556–563, 1987.

[37] T. Onoda and K. Miwa, "Hierarchized two-dimensional code, creation method thereof, and read method thereof," Jun. 10 2009, eP Patent App. EP20,060,731,039. [Online]. Available : https://www.google.com.ar/patents/EP1916619A4?cl=en

[38] E. ARIKAN, "Channel polarization : A method for constructing capacity-achieving codes." Toronto, ON : IEEE International Symposium on Information Theory, 2008. ISIT 2008, 6-11 July 2008, p. 1173–1177.

[39] G. Berhault, "Exploration architecturale pour le décodage de codes polaires," Ph.D. dissertation, Université de Bordeaux, 09 October 2015.

[40] A. Balatsoukas-Stimming, P. Giard, and A. Burg, "Comparison of polar decoders with existing low-density parity-check and turbo decoders," in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, March 2017, pp. 1–6.

[41] V. Zyablov and M. Pinsker, "Estimation of the error-correction complexity of gallager low-density codes," *Peredachi Informatsii*, vol. 11, pp. 23–26, 1975.

[42] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions Information Theory*, vol. 27, pp. 533–547, Sept. 1981.

[43] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Likoping University, Sweden, 1996.

[44] D. Mackay, S. Wilson, and M. Davey, "Comparison of constructions of irregular codes," *IEEE Transactions Communications*, vol. 47, pp. 1449–1454, Oct. 1999.

[45] A. CHARAF, "Etude de récepteurs MIMO-LDPC itératifs," Ph.D. dissertation, TELECOM ParisTech, 2012.

[46] V. Mannoni, "Optimisation des codes LDPC pour communications multi-porteuses," Ph.D. dissertation, Université de Reims Champagne-Ardenne, 2004.

[47] S. t. Brink, "Convergence of iterative decoding," *Electronics Letters*, vol. 35, pp. 806–808, May 1999.

[48] S. Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation," *IEEE Transactions on Information Theory*, vol. 47, pp. 657–670, Feb. 2001.

[49] X.-Y. Hu, E. Elecftheriou, and D.-M. Arnold, "Progressive edge-growth tanner graphs." San Antonio, TX : IEEE Global Telecommunications Conference, Nov. 2002, pp. 995 – 1001.

[50] S. Lin and D. Costello, *Error Control Coding*. Prentice Hall, 2004.

[51] H. Tang, J. Xu, Y. Kou, S. Lin, and K. Abdel-Ghaffar, "On algebraic construction of gallager low density parity check codes," in *Proceedings IEEE International Symposium on Information Theory,*, 2002, pp. 482–.

[52] ——, "On algebraic construction of gallager and circulant low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1269–1279, June 2004.

[53] J. L. Fan, "Array codes as low-density parity-check." Brest, France : Second Iternational Symposium on Turbo Codes, 2000, p. 543.

[54] M. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrice," *IEEE Transactions on Information Theory*, vol. 50, pp. 1788–1793, 2004.

[55] R. M. Tanner, D. Srifhara, A. Sridharan, T. Fuja, and D. Costello, "LDPC block and convolutional codes based on circulant matrices," *IEEE Transactions on Information Theory*, vol. 50, p. 2966, Dec. 2004.

[56] R. M. Tanner, "On quasi-cyclic repeat-accumulate codes." Monticello : Proc. of the 37th Allerton Conference Communication, Control and Computing, 1999, pp. 543–546.

[57] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *Jet Propulsion Laboratory, INP Progress Report*, pp. 42–154, Aug. 2003.

[58] S. Lin, L. Chen, J. Xu, and I. Djurdjevic, "Near shannon limit quasi-cyclic low density parity-check codes." San Francisco, Calif : IEEE GLOBECOM 2003, 2003, pp. 2030–2035.

[59] L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near shannon limit quasi-cyclic low density parity-check codes," *IEEE Transactions Communications*, vol. 52, pp. 1038–1042, 2004.

[60] D. Divsalar, H. Jin, and R. McEliece, "Coding theorems for turbo-like codes." Proc. of the 36th Allerton Conference Communication, Control and Computing, 1998, p. 210.

[61] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes." Brest, France : Proc. 2nd Int. Symp. Turbo Codes and Related Topics, 2000, pp. 1–8.

[62] A. Roumy, S. Guemghar, G. Caire, and S. Verdu, "Design methods for irregular repeat accumulate codes," *IEEE Transaction on Information Theory*, vol. 50, pp. 1711–1727, Aug. 2004.

[63] M. Yang, W. Ryan, and Y. Li, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Transactions on Communications*, vol. 25, pp. 564–571, April. 2004.

[64] R. Echard, *On the Construction of Some Deterministic Low-Density Parity-Check Codes*. George Mason University, 2002.

[65] J. B. Doré, M. H. Hamon, and P. Pénard, "A structured LDPC code construction for efficient encoder design." Istanbul : Proc. IEEE International Conference on Communication, June 2006, pp. 1680–1685.

[66] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate repeat accumulate codes." Dallas, TX, USA : GLOBECOM '04. IEEE Global Telecommunications Conference, 2004, Dec. 2004, p. 509–513.

[67] K. M. Chugg, P. Thiennviboon, G. D. Dimou, P. Gray, and J. Melzer, "Accumulate repeat accumulate codes." Atlantic City, NJ : MILCOM 2005.IEEE Military Communications Conference, 2005., Oct. 2005, pp. 3117 – 3126.

[68] J. Lafferty and D. Rockmore, "Codes and iterative decoding on algebraic expander graphs." Honolulu, Hawaiin, U.S.A : Int. Symp. on Information Theory and its Applications, Nov. 2000.

[69] J. Rosenthal and P. O. Vontobel, "Constructions of regular and irregular LDPC codes using ramanujan graphs and ideas from margulis." Washington, DC : Proc. of the IEEE International Symposium on Information Theory, June. 2001.

[70] D. J. C. MacKay and M. S. Postol, "Weaknesses of margulis and ramanujan-margulis low-density parity-check codes," *Electronic Notes in Theoretical Computer Science*, vol. 74, 2003.

[71] D. J. C. MacKay and M. C. Davey, "Evaluation of gallager codes for short block length and high rate applications," *Codes, Systems and Graphical Models Springer*, vol. 123, pp. 113–130, 2000.

[72] T. J. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions Information Theory*, vol. 47, pp. 638–656, Feb. 2001.

[73] H. A. Loeliger, "New turbo-like codes." Ulm, Germany : Proc. of the IEEE Int. Symposium on Information Theory, July. 1997.

[74] *Air interface for fixed and mobile broadband wireless access systems, IEEE P802.16e/D12 Draft*, IEEE Std. 802.16e, oct 2005.

[75] C. Berrou, A. G. i Amat, Y. Ould-Cheikh-Mouhamedou, C. Douillard, and Y. Saouter, "Adding a rate-1 third dimension to turbo codes." Tahoe City, CA : IEEE Information Theory Workshop, 2007. ITW '07, Sep. 2007.

[76] L. Chen, "Construction of structured low-density-parity-check codes : Combinatorial and algebraic approaches," Ph.D. dissertation, UNIVERSITY OF CALIFORNIA, 2004.

[77] J. Y. Chouinard, *Théorie et pratique des codes correcteurs.* Université Laval, Janvier 2013.

[78] H. Wymeersch, H. Steendam, and M. Moeneclaey, "Log-domain decoding of LDPC codes over GF(q)," *IEEE Communications Society*, vol. 45, pp. 399–431, 2004.

[79] E. Eleftheriou, T. Mittelholzer, and A. Dholakia, "Reduced-complexity decoding algorithm for low-density parity-check codes," *IET Electronics Letters*, vol. 37, pp. 102–104, 2001.

[80] H. Wymeersch, F. Penna, and V. Savic, "Uniformly reweighted belief propagation : a factor graph approach." St. Petersburg : IEEE International Symposium on Information Theory, 2011, pp. 2000–2004.

[81] ——, "Uniformly reweighted belief propagation for estimation and detection in wireless networks," *IEEE Transaction on Wireless Communication*, vol. 11, pp. 1587–1595, Apr 2012.

[82] J. Liu and R. C. de Lamare, "Knowledge-aided reweighted belief propagation decoding for regular and irregular LDPC codes with short blocks." Paris, France : 9th IEEE International Symposium on Wireless Communications Systems, August 2012, pp. 984–988.

[83] T. R. Halford and K. M. Chugg, "An algorithm for counting short cycles in bipartite graphs," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 287–292, Jan 2006.

[84] A.KADI, S.NAJAH, and M.MRABTI, "An exponential factor appearance probability belief propagation algorithm for regular and irregular LDPC codes," *AEUE International Journal of Electronics and Communications Elsevier*, vol. 69, pp. 933–936, 2015.

[85] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "A new class of upper bounds on the log partition function," *IEEE Transaction on Information Theory*, vol. 51, pp. 2313–2335, July 2005.

[86] M. K. Dehkordi and A. H. Banihashemi, "A message-passing algorithm for counting short cycles in a graph." 1-5 : IEEE Information Theory Workshop, Jan 2010, pp. 984–988.

[87] N. Alon, R. Yuster, and U. Zwick, "Finding and counting given length cycles," *Algorithmica*, vol. 17, pp. 209–223, 1997.

[88] J. Fan and Y. Xiao, "A method of counting the number of cycles in LDPC codes," in *2006 8th international Conference on Signal Processing*, vol. 3, 2006.

[89] H. Dehghani, M. Ahmadi, S. Alikhani, and R. Hasni, "Calculation of girth of tanner graph in LDPC codes," *Trends in Applied Sciences Research*, vol. 7, pp. 929–934, 2004.

[90] A. KADI, S.NAJAH, and M.MRABTI, "Reduced latency and complexity of VFAP-BP algorithm for regular and irregular LDPC codes." Marrakech (Morocco) : The 7th edition of International Symposium on signal, Image, Video and Communications (ISIVC), November 19 - 21, 2014.

[91] D. J. MacKay. (2015) Encyclopedia of sparse graph codes [eb/ol]. [Online]. Available : http://www.inference.phy.cam.ac.uk/mackay/codes/data.html

[92] J. Chen and M. P. Fossorier, "Near optimum universal belief propagation based decoding of low density parity check codes," *IEEE Transaction Communications*, vol. 50, p. 406–414, 2002.

[93] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. on Communications*, vol. 53, pp. 1288–1299, 2005.

[94] J. Chen, R. M. Tanner, C. Jones, and Y. Li, "Improved min-sum decoding algorithms for irregular LDPC codes," in *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, Sept 2005, pp. 449–453.

[95] V. Savin, "Self-corrected min-sum decoding of LDPC codes," IEEE International Symposium on Information Theory. Toronto : IEEE, 2008.

[96] J. Zhang, M. Fossorier, and D. Gu, "Two-dimensional correction for min-sum decoding of irregular LDPC codes," *IEEE Communications Letters*, vol. 10, no. 3, pp. 180–182, Mar 2006.

[97] J. Fan, Y. Xiao, and K. Kim, "Design LDPC codes without cycles of length 4 and 6," *Research Letters in Communications*, vol. 2008, no. 5 page, 2008.

[98] A.KADI, S.NAJAH, M.MRABTI, and S.Belfkih, "Improving performances of the min sum algorithm for LDPC codes." Saidia, Maroc : in springer proceeding Medict, May 2015, pp. 381 – 388.

[99] A. Kadi, M. Mrabti, S. Najah, and E. Boutillon, "Novel method for improving performances of normalized MS decoder using WIMAX code," in *2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS)*, Oct 2016, pp. 1–5.