Rapport de stage

Master SIC / STC

Laboratoire ETIS(UMR CNRS 8051 - UCP - ENSEA)

Généralisation du Belief Propagation

Jean-Christophe Sibel
Responsable : Sylvain Reynal

Avril 2009 - Septembre 2009

Remerciements

Je remercie Germain PHAM et Ludovic DANJEAN, stagiaires ETIS, pour leur soutien et leur aide aux différents points cruciaux de mon étude, je remercie Philippe HAMEAU, ingénieur Telecom-Paris pour son intérêt et son investissement, je remercie Sylvain REYNAL, mon responsable de stage, pour sa convivialité et sa patience.

Introduction

Les problèmes d'inférences statistiques sont présents dans de nombreux domaines tels que la physique statistique, la vision par ordinateur et la théorie de l'information. L'algorithme du *Belief Propagation* (BP), fondé sur des transports de messages locaux dans des structures de réseau, est un moyen efficace de résoudre de tels problèmes, grâce notamment à son interprétation graphique essentielle et accessible. Le parallèle entre la théorie de l'information et la physique statistique permet d'expliquer les bases de cet algorithme, et fait ressortir également ses défauts en terme de convergence.

La généralisation du Belief Propagation (GBP) provient d'une étude mené sur son aspect physique statistique, et réalise de meilleures performances que le Belief Propagation simple grâce à une nouvelle approche prenant en compte les défauts de convergence. Néanmoins, cette généralisation est moins rapide et demande une gestion de mémoire sur ordinateur bien plus ordonnée et économe.

Table des matières

Chapitre 1

Théorie de l'information, encodage et décodage

1.1 Principe

Les communications numériques sont établies sur un modèle de transmission composé de trois niveaux. Le premier niveau est l'émetteur, le deuxième niveau est le canal, et le troisième niveau est le récepteur (voir figure 1.1).

FIG. 1.1 – Modèle des communications numériques

L'information est représentée comme des vecteurs ou mots binaires $\mathbf{x}$ de taille N. Dans la cas idéal, le canal est non perturbatif donc l'information $\mathbf{x}$ envoyée par l'émetteur est exactement l'information $\mathbf{y}$ reçue par le récepteur :

$$\forall i \in \{1..N\} \quad y_i = x_i$$

Ce cas idéal est cependant peu fréquent. Un des modèles les plus rencontrés, et celui utilisé tout au long de cette étude, est le canal additif gaussien. Ce canal est modélisé comme un ajout de bruit blanc gaussien centré $\mathbf{b}$ sur l'entrée du canal :

$$\forall i \in \{1..N\} \quad y_i = x_i + b_i, \quad \mathbf{b} \sim \mathcal{N}(0, \sigma_b^2)$$

Les données obtenues en sortie de canal sont interprétées en terme de probabilité par des vraisemblances, soit des probabilités conditionnelles entre la sortie et l'entrée du canal. Dans le cas présent, la densité de probabilité donnant la vraisemblance est celle de $\mathbf{b}$:

$$p(y_i|x_i, \sigma_b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} e^{-\frac{(y_i - x_i)^2}{2\sigma_b^2}} \tag{1.1}$$

Etant donné que l'on travaille sur des mots binaires, pour chaque échantillon de sortie y_i on peut calculer deux vraisemblances :

$$P_{-\alpha,i} = p(y_i|x_i = -\alpha, \sigma_b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} e^{-\frac{(y_i + \alpha)^2}{2\sigma_b^2}}$$

$$P_{+\alpha,i} = p(y_i|x_i = +\alpha, \sigma_b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} e^{-\frac{(y_i - \alpha)^2}{2\sigma_b^2}}$$

où α est le coefficient de la modulation choisie ($\alpha = 1$ pour une modultation BPSK). Pour retrouver le bit d'information envoyée x_i, on cherche le maximum entre $P_{-\alpha,i}$ et $P_{+\alpha,i}$, on note $\hat{x}_i$ ce maximum :

$$\hat{x}_i = \arg\max_{\alpha}\{P_{-\alpha,i}, P_{+\alpha,i}\} \tag{1.2}$$

Si $\forall i \in \{1..N\}$, $\hat{x}_i = x_i$ alors le mot $\hat{\mathbf{x}}$ est un mot d'information, dit plus généralement mot de code.

Remarque : dans l'équation précédente, il convient d'effectuer une démodulation pour retrouver une estimation $\hat{\boldsymbol{x}} \in \{0,1\}^N$.

1.2 Encodage

Afin d'obtenir de bons résultats sur la maximisation de la vraisemblance, on effectue plusieurs opérations logiques sur les bits d'entrée, i.e. sur les composantes u_i d'un vecteur $\mathbf{u}$ qui créent des corrélations entre eux, et donnant le vecteur $\mathbf{x}$ utilisé ci-dessus.

1.2.1 Redondance

L'intérêt de ces opérations est de rendre l'information robuste au bruit, le principe fondamental étant la redondance de l'information. En effet, comme dans un cas réel, si une information est bruitée, il suffit de la répéter plusieurs fois pour en établir une moyenne qui en sera l'estimation la plus proche.
Par exemple, si on souhaite transmettre un simple bit u, on envoie en entrée de canal le mot $\mathbf{x} = [uuuuu]$, le bit estimé $\hat{u}$ en sortie est calculé par décision majoritaire.
Soient n_0, n_1 le nombre de 0 et de 1 dans le mot d'estimation :

$$\hat{u} = \arg\max_{i\in\{0,1\}}\{n_i\} \tag{1.3}$$

1.2.2 Vérification de parité

On crée une extension du principe de redondance en répétant les bits d'un mot dans plusieurs opérations de corrélation. On regroupe ces opérations dans une matrice dite *de vérification de parité* H, la règle d'utilisation et de construction de cette matrice étant la suivante : *pour tout mot de sortie $\hat{\boldsymbol{x}}$, si $\hat{\boldsymbol{x}} \in ker(H)$ alors $\hat{\boldsymbol{x}}$ est un mot de code.* On appelle *mot de code* un mot d'information, soit un vecteur solution au système d'équations contenu dans H.
Pour transmettre un vecteur d'information $\mathbf{u}$, il est donc nécessaire de l'encoder, i.e. de créer un vecteur associé $\mathbf{x}$ qui soit dans le noyau de H. Cette association est réalisee à l'aide d'une matrice dite *génératrice* G qui est telle que :

$$Im(G) = Ker(H)$$

Ainsi on a :

$$x = G.u \ , \ H.x = 0$$

En termes de dimensions, si on note k la longueur de l'information $\mathbf{u}$, et N la longueur du mot de code $\mathbf{x}$, alors $G \in \mathcal{M}_{N,k}(\{0,1\})$, $H \in \mathcal{M}_{N-k,N}(\{0,1\})$.

Exemple

On souhaite transmettre le mot suivant : $\mathbf{u} = [1101]^T$, $k = 4$. On utilise alors un encodeur G tel que :

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

On calcule le mot de code $\mathbf{x} = G.u$:

$$\mathbf{x} = \begin{pmatrix} u_1 \oplus u_2 \oplus u_3 \\ u_1 \oplus u_3 \oplus u_4 \\ u_1 \oplus u_3 \oplus u_4 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}$$

Pour le mot $\mathbf{u}$ défini ci-dessus, on a $\mathbf{x} = [0101101]^T$. La matrice de vérification de parité H est alors

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

Connaissant cette matrice à la réception, et recevant un mot v de taille $N = 7$, il suffit de vérifier que $v \in Ker(H)$ pour savoir si v est un mot de code. Si ce n'est pas le cas, les équations contenues dans H permettent de corriger les erreurs. Il existe cependant une limite au nombre d'erreurs qu'on peut corriger, ce nombre étant directement lié a la matrice H (nous n'en faisons pas le calcul ici car ce n'est pas le sujet de l'étude).

Supposons qu'on recoive $\mathbf{x} = [1101101]^T$, le premier bit v_1 est donc faux. Suppososns également qu'on connaisse le nombre d'erreurs (ici une seule). On calcule son image par H : $H.v = [111]^T \neq [000]^T$. Les trois équations de H sont donc faussées. Sachant qu'il y a un seul bit faux, on déduit que ce bit doit néssairement apparaître dans les trois équations : c'est le cas du bit v_1. On complémente alors v_1 et on obtient bien $H.v = [000]^T$. On peut en déduire alors l'information $\mathbf{u}$: on a effectué une transmission avec correction d'erreurs.

Ce cas est assez simple, car on a supposé connu le nombre d'erreurs, et ce nombre d'erreurs est faible. Dans la majorité des cas, le nombre d'erreurs est inconnu et peut être grand, la correction d'erreur devient alors fastidieuse voire impossible (limite de la correction).
Pour avoir une sortie estimée la plus proche de l'entrée, on utilise des systèmes d'équation, ou *codes*, plus sophistiqués, comprenant une grande redondance et dont on connaît les capacités de correction.

L'encodage est une méthode utile pour protéger l'information mais le principe de redondance nécessite d'augmenter la taille des mots de code donc de réduire le débit d'information. En utilisant M équations de vérification de parité pour un mot de code de taille N, la taille véritable de l'information est en réalité $k = N - M$, le reste étant de la redondance. On définit ainsi le rendement d'un code comme le rapport $R = \frac{k}{N}$ ou $R = 1 - \frac{M}{N}$, i.e. comme la part de véritable information dans le mot envoyé. Le débit encodé est toujours inférieur au débit sans encodage.

1.2.3 Codes Low-Density-Parity-Check

Les codes Low-Density-Parity-Check (LDPC) sont des codes qui respectent la définition ci-dessus, et dont la matrice de vérification de parité a la particularité d'être creuse, i.e. la densité de 1 dans la matrice est faible. Autrement dit, les corrélations établies entre les différents bits sont peu nombreuses. L'avantage principal est que le nombre d'opérations dans le décodage en est réduit (voir chapitre 2).
Parmi ces codes, on construit une catégorie de codes LDPC dits réguliers : le nombre d_c de 1 par ligne est constant, ainsi que le nombre d_v de 1 par colonne. La conséquence directe est l'égalité suivant :

$$Nd_v = Md_c$$

et donc

$$R = 1 - \frac{M}{N} \quad \text{soit} \quad R = 1 - \frac{d_v}{d_c}$$

Ces codes ont la réputation d'être de bons codes, c'est-à-dire que le nombre d'erreurs détectées ε par mot binaire est proportionnel à la taille N du mot lui-même :

$$\varepsilon = \alpha N - 1, \quad \alpha \in \Re$$

1.3 Décodage

L'encodage établit, comme vu précédemment, des relations logiques entre les bits envoyés, ces relations faisant également parti de la transmission. Ces relations sont utilisées afin de retrouver une information plus fidèle. Ainsi, au niveau de la réception, avant de calculer le maximum des vraisemblances (éq.1.2), on utilise ces relations, représentées sous-forme de liaisons entre noeuds dans un graphe.

1.3.1 Graphe de Tanner

Une matrice de vérification de parité H a un équivalent graphique, dit *graphe de Tanner*. On discerne deux types de noeuds au sein de ce graphe :
- N noeuds de variables ◯ $\{x_i\}_{1 \leq i \leq N}$ représentant chacun un bit,
- M noeuds de parité ■ $\{f_j\}_{1 \leq j \leq M}$ représentant chacun une équation de parité.

Le noeud f_j est relié au noeud x_i si et seulement si $H_{ij} = 1$.

Exemple

On utilise le code de Hamming dont la matrice de vérification de parité est :

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

Les coefficients H_{11}, H_{21} et H_{31} sont égaux à 1 donc le noeud x_1 est relié aux noeuds f_1, f_2 et f_3.
Les coefficients H_{21}, H_{22}, H_{24} et H_{26} sont égaux à 1 donc le noeud f_2 est relié aux noeuds x_1, x_2, x_4 et x_6.

Remarque : pour des codes LDPC, le nombre de branches connectées à un noeud de variable est constant égal à d_v, et le nombre de branches connectées à un noeud de parité est constant égal à d_c.

Le graphe de Tanner est utilisé comme support pour la plupart des algorithmes de décodage ou décodeurs. Le principe fondamental d'un algorithme de décodage utilsant cette représentation est de considérer chaque branche du graphe comme un message d'un noeud f_j vers un noeud x_i et réciproquement.

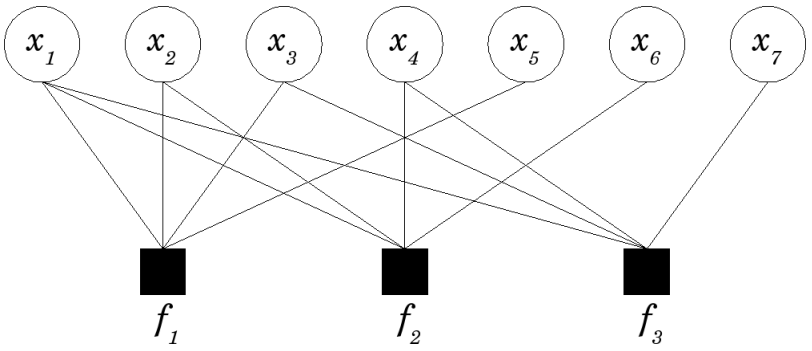

FIG. 1.2 – Graphe de Tanner du code de Hamming

1.3.2 Décodage hard

Pour des décodeurs de type hard, le message est la valeur que doit prendre le noeud destinataire selon le noeud émetteur. Par conséquent, pour décider de la valeur du noeud destinataire lorsqu'il y a plusieurs noeuds émetteurs, il convient d'effectuer une décision majoritaire : si plus de la moitié des noeuds émetteurs indiquent une valeur $\alpha \in \{0, 1\}$ pour le noeud destinataire, alors le noeud destinataire est décidé à α.

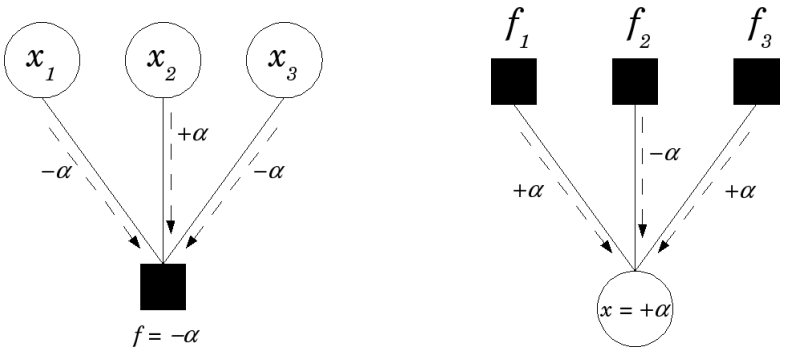

FIG. 1.3 – Décodage hard

1.3.3 Décodage soft

Pour des décodeurs de type soft, le message porte les probabilités sur le noeud destinataire conditionnellement au noeud émetteur. Si on note $m_{f_j \to x_i}$ le message du noeud émetteur f_j vers le noeud destinataire x_i, alors :

$$m_{f_j \to x_i}(x_i) = \frac{1}{Z}\, p(x_i|f_j) \tag{1.4}$$

de même pour le message du noeud émetteur x_i vers le noeud destinataire f_j :

$$m_{x_i \to f_j}(x_i) = \frac{1}{Z}\, p(f_j|x_i) \tag{1.5}$$

où Z est le coefficient de normalisation permettant au message de représenter une véritable probabilité.
La décision sur les noeuds x_i et f_j peut être faite de plusieurs façons. Le choix du décodeur est propre à l'utilisation, avec la contrainte suivante : un décodeur ne peut être le plus rapide et le plus précis à la fois, l'une des deux propriétés est toujours dominante.

Chapitre 2

Algorithme de décodage - Belief Propagation

L'algorithme du BP est un algorithme de décodage de type soft, son fonctionnement est donc fondé sur l'utilisation de probabilités. Peu rapide, cet algorithme est cependant réputé optimal au sens du maximum de vraisemblance, c'est-à-dire que les valeurs de décision sont les meilleures que l'on puisse trouver. Son fonctionnement peut être expliqué au travers d'une analogie à la physique statistique, grâce à l'utilisation de réseau de spins ainsi qu'au problème du calcul de l'énergie du résau.

2.1 Modèle de Ising

Soit un réseau de spins plongé dans un champ magnétique uniforme, $\overrightarrow{B}$. On représente ce réseau par un ensemble de N variables aléatoires $\mathbf{s} = \{s_i\}_{1 \leq i \leq N}$, où chaque variable aléatoire s_i représente les états du spin $\overrightarrow{S_i}$ associé, telles que : $\forall i \in \{1..N\}$, $s_i \in \{-1, +1\}$. La valeur $+1$ ou -1 de la variable s_i dépend physiquement de l'alignement ou de l'antialignement du spin $\overrightarrow{S_i}$ avec le champ magnétique.

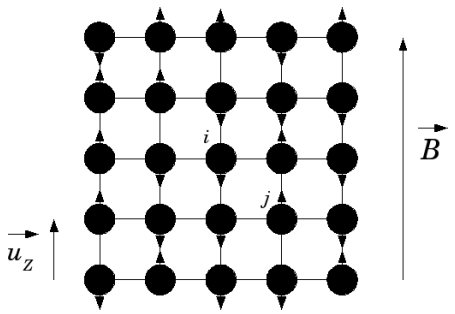

FIG. 2.1 – Réseau de spins - Modèle de Ising

On considère, selon [1], deux types d'énergie dans lesquelles la variable associée à un spin $\overrightarrow{S_i}$ intervient : une énergie $E_{ij}(s_i, s_j)$ entre deux spins voisins, et une énergie $E_i(s_i)$ entre le spin et le champ magnétique. On peut donc considérer que l'énergie globale du réseau s'écrit :

$$E(\mathbf{s}) = \sum_{i=1}^{N} \sum_{j \neq i} E_{ij}(s_i, s_j) + \sum_{i=1}^{N} E_i(s_i)$$

Le modèle de Ising est considéré comme à courte portée, i.e. les interactions sont négligeables entre spins non-voisins. On peut donc préciser l'équation précédente :

$$E(\mathbf{s}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}(i)} E_{ij}(s_i, s_j) + \sum_{i=1}^{N} E_i(s_i)$$

ou encore :

$$E(\mathbf{s}) = \sum_{i=1}^{N} \tilde{E}_i(s_i, \mathcal{N}(i)) + \sum_{i=1}^{N} E_i(s_i) \tag{2.1}$$

où $\mathcal{N}(i)$ est l'ensemble des voisins directs de s_i.

L'énergie d'interaction entre deux spins voisins s'écrit plus précisément sous la forme suivante :

$$E_{ij}(s_i, s_j) = -J_{ij} s_i s_j \tag{2.2}$$

où le terme $-J_{ij}$ est la constante de couplage entre les variables s_i et s_j, cette constante étant identique quelque soit la paire $\{s_i, s_j\}$ considéré, nous la noterons donc J. Ainsi, une interaction entre deux spins voisins vaut $+J$ ou $-J$ selon l'orientation des spins. Cette notation ne nous servira que pour établir le lien entre la physique statistique et la théorie de l'information. Nous conservons donc pour le moment l'équation (2.1).

La formulation de l'énergie du réseau permet d'expliciter alors la distribution de spins sur le réseau grâce à la loi de Boltzmann :

$$p(\mathbf{s}) = \frac{1}{Z}\, e^{-E(\mathbf{s})}$$

soit :

$$p(\mathbf{s}) = \frac{1}{Z}\, e^{-\sum_{i=1}^{N} \tilde{E}_i(s_i, \mathcal{N}(i)) - \sum_{i=1}^{N} E_i(s_i)}$$

On note $\tilde{E}_i(s_i, \mathcal{N}(i)) = -\ln \psi_i(s_i, \mathcal{N}(i))$ et $E_i(s_i) = -\ln \phi_i(s_i)$, donc :

$$p(\mathbf{s}) = \frac{1}{Z} \prod_{i=1}^{N} \psi_i(s_i, \mathcal{N}(i)) \prod_{i=1}^{N} \phi_i(s_i) \tag{2.3}$$

Le modèle de Ising est fondé sur le principe d'interaction entre voisins directs ou de premier ordre. Il est possible de faire un lien rapide avec la théorie des probabilités pour mettre en relation physique statistique et théorie de l'information. Pour un ensemble de N variables aléatoires $\mathbf{x} = \{x_i\}_{1 \leq i \leq N}$, la probabilité jointe est :

$$p(\mathbf{x}) = p(x_N) \prod_{i=1}^{N-1} p(x_i | x_{i+1}, ..., x_N)$$

Dans le cas d'un réseau de Markov, la probabilité jointe se réduit à :

$$p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i | \mathcal{N}(i))$$

où $\mathcal{N}(i)$ est l'ensemble des voisins directs de x_i.

Dans le cadre d'une transmission d'information, l'ensemble des variables $\mathbf{x} = \{x_i\}_{1 \leq i \leq N}$ représente l'ensemble des bits envoyés. Les bits reçus sont l'ensemble $\mathbf{y} = \{y_i\}_{1 \leq i \leq N}$. La probabilité jointe entre $\mathbf{x}$ et $\mathbf{y}$ est selon la relation de Bayes :

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}) p(\mathbf{y} | \mathbf{x})$$

soit :

$$p(\mathbf{x},\mathbf{y}) = \prod_{i=1}^{N} p(x_i|\mathcal{N}(i)) \prod_{i=1}^{N} p(y_i|x_i) \tag{2.4}$$

Les équations (2.3) et (2.4) sont équivalentes. Par identification, on peut préciser la nature des fonctions dans l'équation de Boltzmann. Selon l'équation (2.2), la fonction ψ_{ij} est une relation binaire entre les bits x_i et x_j telle que :

$$\psi_{ij}(x_i, x_j) = 1 \oplus x_i \oplus x_j \tag{2.5}$$

Cette fonction traduit la parité entre les variables x_i et x_j. On généralise pour des équations de parité comprenant un sous-ensemble $\mathcal{P}$ de $\{x_i\}_{1\leq i\leq N}$:

$$\psi_{\mathcal{P}}(\underline{x}_{\mathcal{P}}) = 1 \oplus \sum_{x_i\in\mathcal{P}} x_i \tag{2.6}$$

La fonction ϕ_i est la vraisemblance pour le bit x_i en sortie de canal (cf. éq.(1.1)) :

$$\phi_i(x_i) = p(y_i|x_i) \tag{2.7}$$

Exemple

On utilise le code de Hamming précédent. En lisant les relations données dans le graphe de Tanner, on obtient trois fonctions logiques : ψ_{1235}, ψ_{1246} et ψ_{1347}. Ainsi :

$$p(\mathbf{x},\mathbf{y}) = \frac{1}{Z} \prod_{a=1}^{M=3} \psi_a(\mathbf{x}_a) \prod_{i=1}^{N=7} p(y_i|x_i)$$

avec

$$\prod_{a=1}^{M=3} \psi_a(\mathbf{x}_a) = \psi_{1235}(x_1,x_2,x_3,x_5)\psi_{1246}(x_1,x_2,x_4,x_6)\psi_{1347}(x_1,x_3,x_4,x_7)$$

2.2 Belief Propagation

L'expression de la probabilité jointe :

$$p(\mathbf{x},\mathbf{y}) = \frac{1}{Z} \prod_{a=1}^{M} \psi_a(\mathbf{x}_a) \prod_{i=1}^{N} p(y_i|x_i)$$

se décline en utilisant la formule de Bayes :

$$p(\mathbf{x},\mathbf{y}) = p(\mathbf{y})p(\mathbf{x}|\mathbf{y})$$

Comme $\mathbf{y}$ est le vecteur d'observation, sa densité ne nous intéresse pas, on la considère comme une constante. On écrira alors, selon [2] :

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{Z} \prod_{i=1}^{N} \psi_i(x_i, \mathcal{N}(i)) \prod_{i=1}^{N} \phi_i(x_i) \tag{2.8}$$

qui est la probabilité a posteriori sur le vecteur $\mathbf{x}$. On rappelle que le but du décodage est de retrouver le mot de code $\mathbf{x}$, ce qui revient à retrouver chaque bit x_i de ce mot. L'algorithme doit donc être capable de calculer les probabilités marginales $p(x_i)$. En utilisant le graphe de Tanner d'un code et les équations (1.5) et (1.4), il est possible de calculer ces probabilités marginales.

2.2.1 Messages

L'utilisation d'un graphe de Tanner implique de discerner deux types de noeuds, on définit de même deux types de messages, selon [3].

Messages de type I

Considérons la situation où un message est transmis d'un noeud de variable vers un noeud de parité. Nous noterons P la sous-partie de taille q de l'ensemble des noeuds de parité telle que : $\forall f_{P_k} \in P$ où $k \in \{1..q\}$, $f_{P_k} \in \mathcal{N}(i)$, avec x_i le noeud de variable.

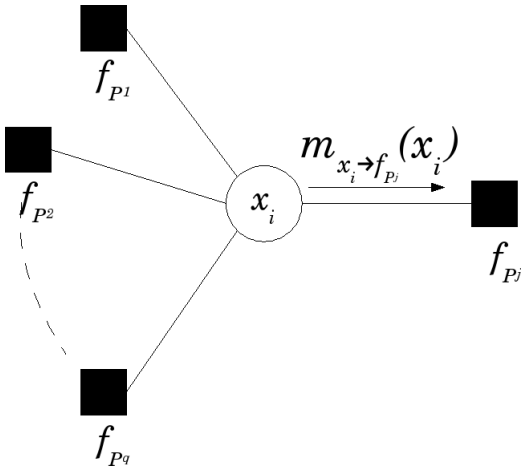

FIG. 2.2 – Message d'un noeud de variable vers un noeud de parité

Le message se calcule, d'après [3], selon l'expression suivante :

$$m_{x_i \to f_{P_j}}(x_i) = \frac{1}{Z} \prod_{k \in \mathcal{N}(i) \setminus P_j} m_{f_k \to x_i}(x_i) \tag{2.9}$$

Messages de type II

Considérons l'autre situation où un message est transmis d'un noeud de parité vers un noeud de variable. Nous noterons P la sous-partie de taille q de l'ensemble des noeuds de variable telle que : $\forall x_{P_i} \in P$ où $i \in \{1..q\}$, $x_{P_i} \in \mathcal{N}(j)$, avec f_j le noeud de parité.

Le message se calcule, d'après [3], selon l'expression suivante :

$$m_{f_j \to x_{P_i}}(x_{P_i}) = \frac{1}{Z} \sum_{x_{P_1}} ... \sum_{x_{P_{i-1}}} \sum_{x_{P_{i+1}}} ... \sum_{x_{P_q}} f_j(x_{P_1}, ..., x_{P_q}) \prod_{k \in \mathcal{N}(j) \setminus P_i} m_{x_k \to f_j}(x_k) \tag{2.10}$$

Croyances

On définit la croyance $b_i(x_i)$ comme la probabilité marginale calculée par le BP sur la variable x_i. On la calcule selon l'expression suivante :

$$b_i(x_i) = \frac{1}{Z} \prod_{k \in \mathcal{N}(i)} m_{f_k \to x_i}(x_i) \tag{2.11}$$

Le graphe de Tanner ne représente pas les données de sortie du canal, il est donc convenable dans les équations de les faire apparaître comme des messages incidents sur les noeuds de variable.

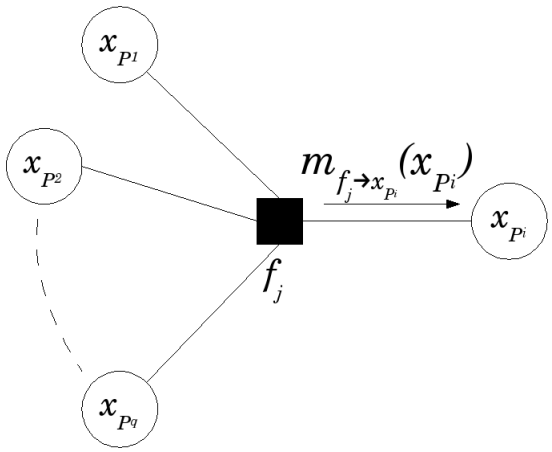

FIG. 2.3 – Message d'un noeud de parité vers un noeud de variable

2.2.2 Exemple trivial

Considérons un sous-graphe du graphe de Tanner, c'est-à-dire, un graphe composé d'un seul noeud de parité et de ses noeuds de variables voisins.

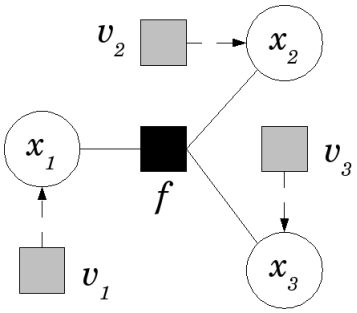

FIG. 2.4 – Sous-graphe de Tanner

On choisit de calculer la croyance du noeud x_1. Il faut donc considérer, selon l'équation (2.2.1) les messages incidents sur le noeud de parité f, ainsi que les messages provenant du canal. On a donc :

$$b_1(x_1) = \frac{1}{Z} m_{v_1 \to x_1}(x_1) m_{f \to x_1}(x_1)$$

avec $m_{v_1 \to x_1}(x_1) = p(y_1|x_1)$ la vraisemblance sur le noeud x_1, que l'on note ici $\phi_1(x_1)$. On décompose selon l'équation (2.2.1) :

$$m_{f \to x_1}(x_1) = \frac{1}{Z} \sum_{x_2} \sum_{x_3} f(x_1, x_2, x_3) m_{x_2 \to f}(x_2) m_{x_3 \to f}(x_3)$$

où $f(x_1, x_2, x_3) = 1 \oplus x_1 \oplus x_2 \oplus x_3$, l'interaction entre les noeuds de variable, que l'on note ici $\psi_{123}(x_1, x_2, x_3)$. On a également :

$$m_{x_2 \to f}(x_2) = m_{v_2 \to x_2}(x_2) = \phi_2(x_2)$$
$$m_{x_3 \to f}(x_3) = m_{v_3 \to x_3}(x_3) = \phi_3(x_3)$$

On obtient ainsi :

$$b_1(x_1) = \frac{1}{Z} \sum_{x_2} \sum_{x_3} \phi_1(x_1) \phi_2(x_2) \phi_3(x_3) \psi_{123}(x_1, x_2, x_3)$$

En comparant avec l'équation (2.8), on a finalement :

$$b_1(x_1) = \sum_{x_2} \sum_{x_3} p(x_1, x_2, x_3)$$

soit :

$$b_1(x_1) = p_1(x_1|\mathbf{y})$$

On obtient bien la marginale, a posteriori, de la variable x_1.

2.2.3 Exemple non-trivial

On utilise le même sous-graphe de Tanner, mais on y ajoute un deuxième noeud de parité.

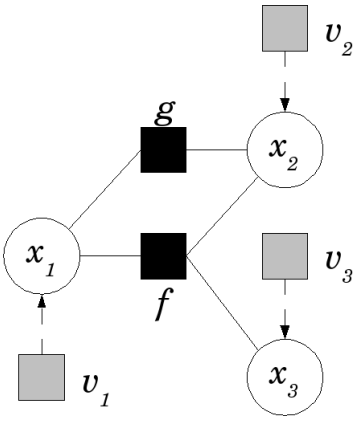

FIG. 2.5 – Sous-graphe de Tanner avec une boucle

Le calcul de la croyance sur la variable x_1 est basé sur les mêmes équations que précédemment :

$$b_1(x_1) = \frac{1}{Z}\phi_1(x_1) m_{f\to x_1}(x_1) m_{g\to x_1}(x_1)$$

On décompose selon l'équation (2.2.1) :

$$m_{f\to x_1}(x_1) = \frac{1}{Z} \sum_{x_2} \sum_{x_3} \psi_{123}(x_1, x_2, x_3) m_{x_2\to f}(x_2) \phi_3(x_3)$$

La différence avec l'exemple intervient dans la décomposition suivante :

$$m_{x_2\to f}(x_2) = \frac{1}{Z} m_{g\to x_2}(x_2) \phi_2(x_2)$$

où :

$$m_{g\to x_2}(x_2) = \frac{1}{Z} \sum_{x_1'} g(x_1', x_2) m_{x_1\to g}(x_1')$$

où $g(x_1', x_2) = 1 \oplus x_1' \oplus x_2$, l'interaction entre les noeuds de variable, que l'on note ici $\psi_{12}(x_1', x_2)$. Or, le message $m_{x_1\to g}$ n'est autre que le message $m_{f\to x_1}$ donc on obtient :

$$m_{f\to x_1}(x_1) = \frac{1}{Z} \sum_{x_1'} \sum_{x_2} \sum_{x_3} \phi_2(x_2)\phi_3(x_3)\psi_{123}(x_1, x_2, x_3)\psi_{12}(x_1', x_2) m_{f\to x_1}(x_1')$$

Le message $m_{f\to x_1}$ ne peut être exprimé sous la forme canonique donnée par l'équation (2.8), et boucle sur lui-même. Par conséquent, la seule expression juste de ce message serait celle qui comprend des développements à l'infini de ce message :

$$m_{f\to x_1}(x_1) = \frac{1}{Z}\sum_{x'_1}\sum_{x_2}\sum_{x_3}\phi_1(x'_1)\phi_2(x_2)\phi_3(x_3)\psi_{123}(x_1,x_2,x_3)\psi_{12}(x'_1,x_2)(\sum_{x''_1}\sum_{x'_2}\sum_{x'_3}...)$$

Ce genre de calcul n'est pas implémentable, et c'est pourquoi, en pratique, on considère une indépendance entre les noeuds de variables d'ordre supérieur à un.

2.3 Implémentation du Belief Propagation

L'algorithme du BP est, selon les équations (2.9) et (2.10), un algorithme itératif. Les messages se mettent à jour selon un ordre donné, c'est l'ordonnancement ou le *scheduling*.

Ordonnancement

Une itération du BP est définie comme la succession de deux étapes : le calcul des messages des noeuds de parité vers les noeuds de variables, et le calcul des messages des noeuds de variables vers les noeuds de parité. Le dernier calcul de l'algorithme donne les croyances sur les variables, qui ne sont autre que les messages sortant des noeuds de variable, on décide donc que le second calcul de l'itération est celui des messages des noeuds de parité vers les noeuds de variable. Il convient avant de rentrer dans le calcul des messages, d'initialiser ces messages, puisque la première itération a besoin des messages de l'itération précédente, on initialise ces messages aux vraisemblances.

Algorithme 1 : Une itération du Belief Propagation

Messages de type I

pour chaque *noeud de variable* v **faire**
  pour chaque *noeud de parité* $f_v \in \mathcal{N}_v$ **faire**
    Message de v *vers* f_v
    pour chaque *noeud de parité* $g_v \in \mathcal{N}_v \backslash f_v$ **faire**
      pour chaque *valeur de* v **faire**
        $m_{v\to f_v}(v) \times = m_{g_v\to v}(v)$

Messages de type II

pour chaque *noeud de parité* f **faire**
  pour chaque *noeud de variable* $v_f \in \mathcal{N}_f$ **faire**
    Message de f *vers* v_f
    pour chaque *état* $\bigcup_{v\in\mathcal{N}_f} v$ **faire**
      si $\bigoplus_{v\in\mathcal{N}_f} v = 0$ **alors**
        calcul de la valeur de v_f
        pour chaque *noeud de variable* $w_f \in \mathcal{N}_f \backslash v_f$ **faire**
          calcul de la valeur de w_f
          $m_{f\to v_f}(v_f) += \prod_{w_f\in\mathcal{N}_f\backslash v_f} m_{w_f\to f}(w_f)\phi_{w_f}(w_f)$

Le nombre d'itérations est décidé selon le taux de convergence des messages. En effet, on dit que le graphe *a convergé* si les messages sont identiques d'une itération à l'autre. Dans le cadre des boucles, le problème majeur est que la convergence n'est pas parfaite : les messages peuvent osciller entre quelques valeurs, ou alors peuvent converger vers un mauvais point fixe (mauvais mot de code).

Dans le cadre de l'implémentation, on s'intéresse surtout aux effets de boucles. On utilise trois codes dont la structure topologique du graphe de Tanner présente des boucles connues.

2.3.1 Code sans boucle

La matrice de vérification de parité est :

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

Le graphe de Tanner correspondant est un arbre, qui ne présente donc aucune structure de boucle.

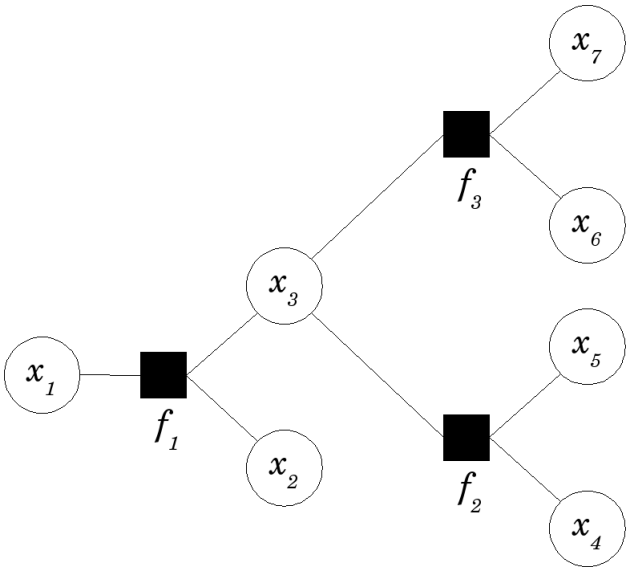

FIG. 2.6 – Graphe de Tanner du code sans boucle

2.3.2 Code à une boucle

La matrice de vérification de parité est :

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

Le graphe de Tanner correspondant n'est plus un arbre, puisque le noeud de variable x_7 crée une boucle $x_7 \rightarrow f_1 \rightarrow x_3 \rightarrow f_3 \rightarrow x_7$ en étant lié au noeud de parité f_1.

2.3.3 Code à plusieurs boucles

La matrice de vérification de parité est :

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

Le graphe de Tanner correspondant présente plusieurs boucles :

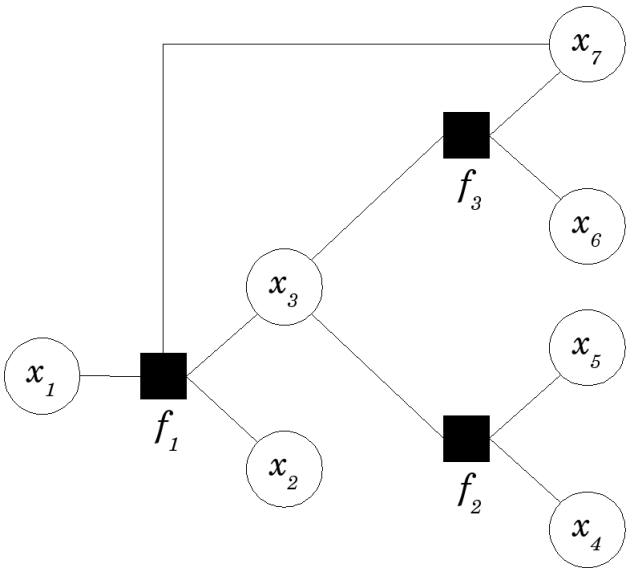

FIG. 2.7 – Graphe de Tanner du code à une boucle

- $x_7 \to f_1 \to x_3 \to f_3 \to x_7$
- $x_1 \to f_2 \to x_3 \to f_1 \to x_1$
- $x_1 \to f_2 \to x_3 \to f_3 \to x_6 \to f_1 \to x_1$

ce qui rend les calculs davantage approximatifs.

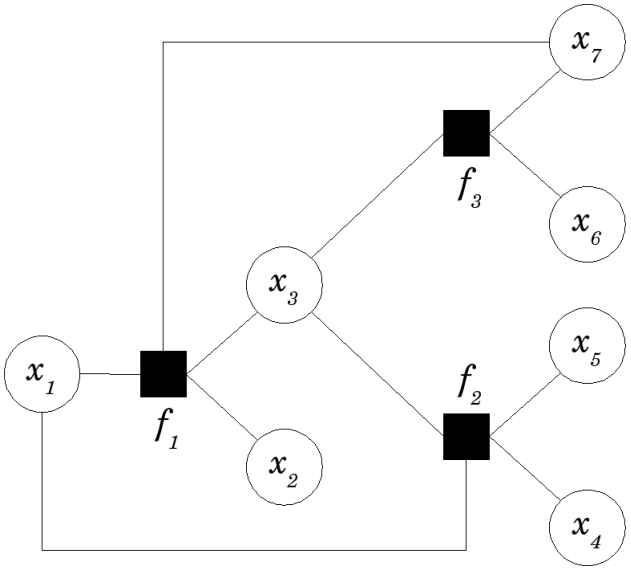

FIG. 2.8 – Graphe de Tanner du code à plusieurs boucles

2.3.4 Résultats

On représente sur un même graphique les taux d'erreurs binaires des trois codes, calculés selon le rapport signal à bruit.

Les trois courbes présentent un taux d'erreur binaire différent, le code sans boucle étant

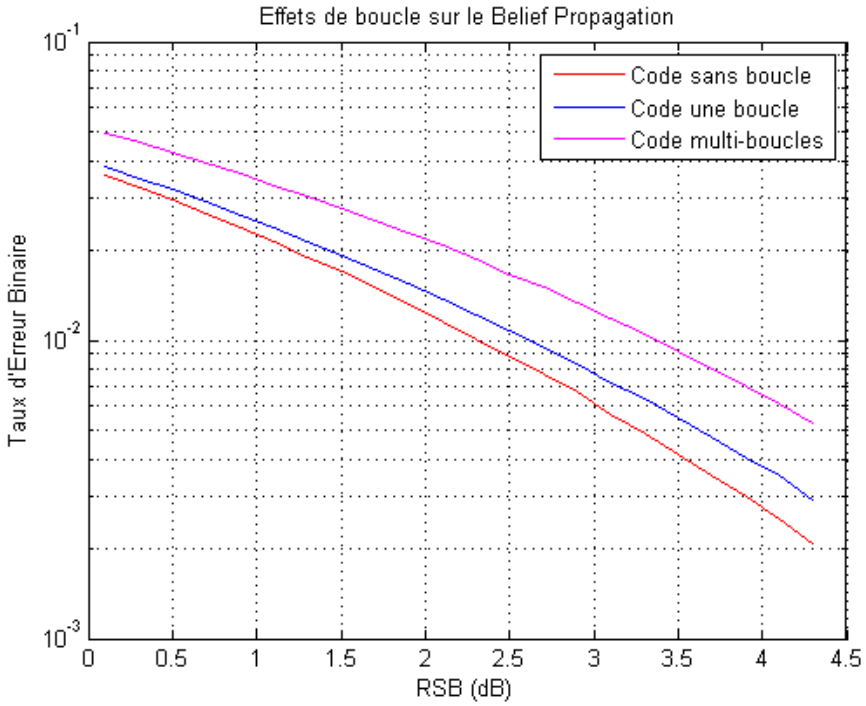

FIG. 2.9 – Effets de boucle sur le BP

le meilleur, et le code multi-boucles étant le moins bon. On pourra relever à titre d'exemple les valeurs suivantes pour un RSB de 4dB :

- $\text{TEB}_{\text{sans boucle}} = 2,8.10^{-3}$,
- $\text{TEB}_{\text{une boucle}} = 4,0.10^{-3}$,
- $\text{TEB}_{\text{multi-boucles}} = 6,5.10^{-3}$.

L'interprétation est simple : les hypothèses d'indépendances d'ordre un rendent le calcul approximatif en présence de boucles, et dégradent le taux d'erreur binaire. Il convient de considérer une approche du graphe comprenant ces effets de boucle, et évitant les hypothèses non vérifiées sur l'indépendance entre les noeuds.

Chapitre 3

Généralisation du Belief Propagation

Il est possible de corriger le BP en utilisant une vision statistique plus large que l'ordre un sur les graphes de Tanner.

3.1 Approximation de Kikuchi

Réutilisons le réseau de spin du chapitre précédent.

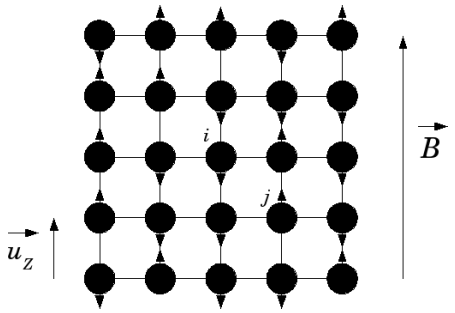

FIG. 3.1 – Réseau de spins - Modèle de Ising

Chaque noeud de ce réseau est relié à ses quatre voisins d'ordre un, exceptés les noeuds de bords sauf si le réseau est torique. Rappelons la situation dans laquelle le BP évolue : les messages incidents sur un noeud s_i sont calculés par hypothèse d'indépendance entre les voisins de s_i. Donc dans le cas de ce réseau, pour calculer la croyance sur le noeud s_i, l'hypothèse implique de supprimer toute branche créant une boucle dans le calcul (voir figure suivante). En supprimant les branches, on supprime les interactions ce qui équivaut à modifier le code. Cette solution n'est pas la meilleure, comme vue dans le chapitre précédent.

Pour éviter cette hypothèse, il est nécessaire de considérer un voisinage plus large. Le voisinage est créé selon les interactions entre les noeuds, pour que chaque voisinage ait un sens propre, au même titre qu'on rassemble des individus selon un caractère particulier. On obtient plusieurs voisinages interagissant par le biais de leurs noeuds communs. Les interactions échangées permettent de voir ces voisinages comme un nouveau réseau. Cette vision est l'*approximation de Kikuchi* (cf.[4]).

Dans le cadre de la physique statistique, l'objectif est de calculer l'énergie libre d'un réseau, étroitement liée aux marginales des variables représentées par chaque noeud du réseau. La vision par voisinage donne un algorithme de calcul sur les marginales, et donc

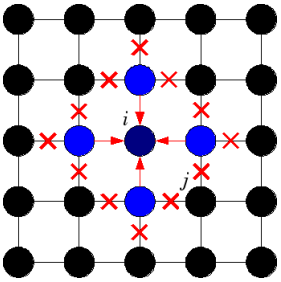

FIG. 3.2 – Réseau de spins - Approximation du BP

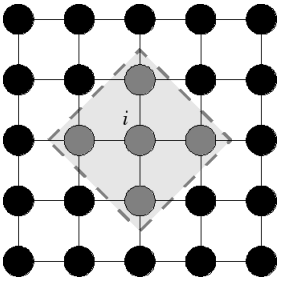

FIG. 3.3 – Voisinage sur un réseau

sur l'énergie plus précis.

3.1.1 Energie de Gibbs

Il existe un indicateur de précision sur les croyances calculées $b(\mathbf{x})$ par rapport aux marginales exactes $p(\mathbf{x})$, la *divergence de Kullback-Liebler*. Cette divergence n'est pas une distance au sens mathématique du terme puisqu'elle n'est pas symétrique, mais son expression crée un lien avec la notion d'énergie libre. Appliquée sur un réseau de distribution $p(\mathbf{x})$, son expression est :

$$D(b(\mathbf{x}), p(\mathbf{x})) = \sum_{\mathbf{x}} b(\mathbf{x}) \ln \frac{b(\mathbf{x})}{p(\mathbf{x})} \tag{3.1}$$

En développant avec la loi de Boltzmann $p(\mathbf{x}) = \frac{1}{Z}\ e^{-E(\mathbf{x})}$ il vient :

$$D(b(\mathbf{x}), p(\mathbf{x})) = \sum_{\mathbf{x}} b(\mathbf{x}) E(\mathbf{x}) + \sum_{\mathbf{x}} b(\mathbf{x}) \ln b(\mathbf{x}) + \ln Z$$

La divergence de Kullback-Liebler contient
- $\sum_{\mathbf{x}} b(\mathbf{x}) E(\mathbf{x})$, soit l'espérance mathématique $U(b(\mathbf{x}))$ de l'énergie selon la loi $b(\mathbf{x})$,
- $\sum_{\mathbf{x}} b(\mathbf{x}) \ln b(\mathbf{x})$, soit l'entropie $S(b(\mathbf{x}))$de la loi $b(\mathbf{x})$,
- $F_H = -\ln Z$ appelée *énergie libre de Helmholz*.

Le premier principe de la thermodynamique nous donne l'expression de l'énergie totale du système $G(b(\mathbf{x}))$:

$$G(b(\mathbf{x})) = U(b(\mathbf{x})) - S(b(\mathbf{x})) \tag{3.2}$$

soit :

$$G(b(\mathbf{x})) = D(b(\mathbf{x}), p(\mathbf{x})) - F_H \tag{3.3}$$

La divergence de Kullback-Liebler est strictement positive, et F_H est une constante du réseau, donc l'énergie $G(b(\mathbf{x}))$ possède un minimum dit *énergie libre de Gibbs* :

$$\min_{b(\mathbf{X})}\{G(b(\mathbf{x}))\} = F_H \tag{3.4}$$

atteint pour $b(\mathbf{x}) = p(\mathbf{x})$, soit lorsque le calcul des croyances est exact, sans boucle. On observe ainsi que calculer les marginales exactes des variables d'un graphe revient à rechercher l'énergie libre de Gibbs du réseau associé.
L'utilisation de voisinages permet d'avoir une expression de cette énergie dépendante de l'énergie de chaque voisinage :

$$F_H = \sum_{V \in \mathcal{V}} F_V - \sum_{I \in \mathcal{I}} c_I F_I \tag{3.5}$$

où $\mathcal{V}$ est l'ensemble des voisinages du réseau, $\mathcal{I}$ est l'ensemble des intersections de voisinage, c_I est un compteur donnant la redondance d'une intersection I. En effet, le fait d'avoir des intersections de voisinage dans le réseau implique que des noeuds interviennent dans plusieurs voisinages. Pour éviter de comptabiliser plusieurs fois la contribution d'une intersection au calcul de l'énergie F_H, on calcule un compteur par intersection qui pondère sa valeur dans F_H, selon [2].

Exemple

Dans le réseau des variables $\mathbf{x} = \{x_i | i \in \{1..6\}\}$ donné par la figure suivante, l'énergie libre vaut :

$$G(b(\mathbf{x})) = G_{1245}(x_1, x_2, x_4, x_5) + G_{2356}(x_2, x_3, x_5, x_6) - G_{25}(x_2, x_5)$$

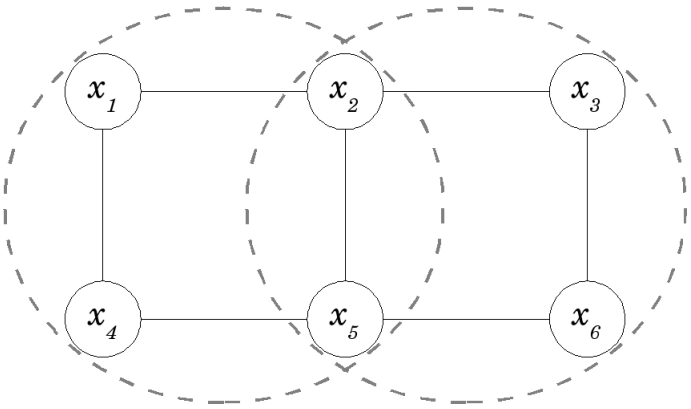

FIG. 3.4 – Energie libre - Redondance

L'ensemble $\{x_2, x_5\}$ contribue au calcul de l'énergie de chacun des voisinages, donc on soustrait son énergie une fois à l'énergie du réseau.

3.2 Graphe de Tanner - Graphe des régions

Soit le code suivant :

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

Le graphe de Tanner de ce code présente de multiples boucles, et comme vu dans le chapitre précédent, ce sont des sources de dégradation sur le décodage.

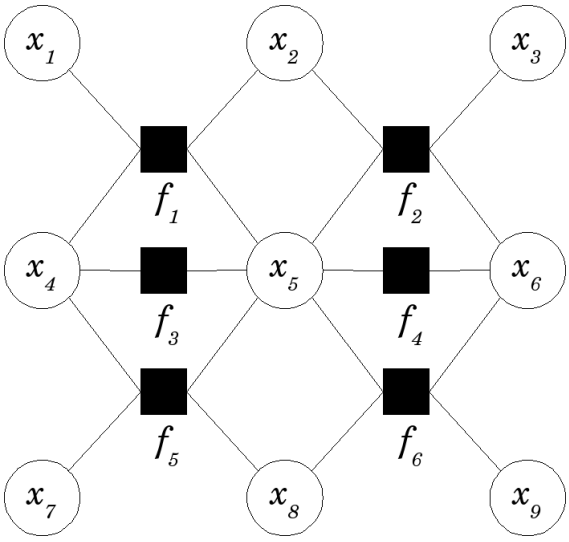

FIG. 3.5 – Graphe de Tanner à boucles

Le principe du GBP est d'*absorber* les cycles dans des ensembles de noeuds appelés *clusters* afin d'éviter les hypothèses d'indépendance. Cependant, il est difficile de répertorier simplement tous les cycles d'un code, on choisit donc comme cluster de taille maximale les clusters formés par les équations de parité et leurs variables. Il y a donc pour le code ci-dessus, quatre clusters globaux :

- $\{f_1, f_3, x_1, x_2, x_4, x_5\}$,
- $\{f_2, f_4, x_2, x_3, x_5, x_6\}$,
- $\{f_3, f_5, x_4, x_5, x_7, x_8\}$,
- $\{f_4, f_6, x_5, x_6, x_8, x_9\}$.

La définition de ces clusters permet de construire un deuxième graphe associé au code, fondé sur le *Cluster Variation Method* et sur les *diagrammes de Hasse*.

3.2.1 Diagramme de Hasse

Soit un ensemble de N éléments $(\mathcal{S}, \prec) = \{s_i | i \in \{1..N\}\}$, muni d'une relation d'ordre $\prec$.
Cette relation est dite *totale* si : $\forall (i,j) \in \{1..N\}^2$, $s_i \prec s_j$ ou $s_j \prec s_i$, autrement dit si tous les éléments sont comparables selon cette relation.
Cette relation est dite *partielle* si : $\exists i \in \{1..N\}$, $\exists j \in \{1..N\}$, $s_i \not\prec s_j$ et $s_j \not\prec s_i$ et $s_i \neq s_j$, autrement dit, il existe au moins deux éléments non comparables selon cette relation.
Dans le cas d'un ordre partiel, l'ensemble est appelé *poset* (pour *partially order set*), et on peut le représenter par un diagramme dit *diagramme de Hasse*.

Exemple

Soit l'ensemble $(\mathcal{S}, \subset) = \{\{x_1, x_2, x_4, x_5\}, \{x_2, x_5\}, \{x_4, x_5\}, \{x_5\}\, \{x_1\}\}$ où la relation d'ordre est l'inclusion. On a donc les relations suivantes :

- $\{x_2, x_5\} \subset \{x_1, x_2, x_4, x_5\}$,
- $\{x_4, x_5\} \subset \{x_1, x_2, x_4, x_5\}$,
- $\{x_5\} \subset \{x_2, x_5\}$,
- $\{x_5\} \subset \{x_4, x_5\}$.

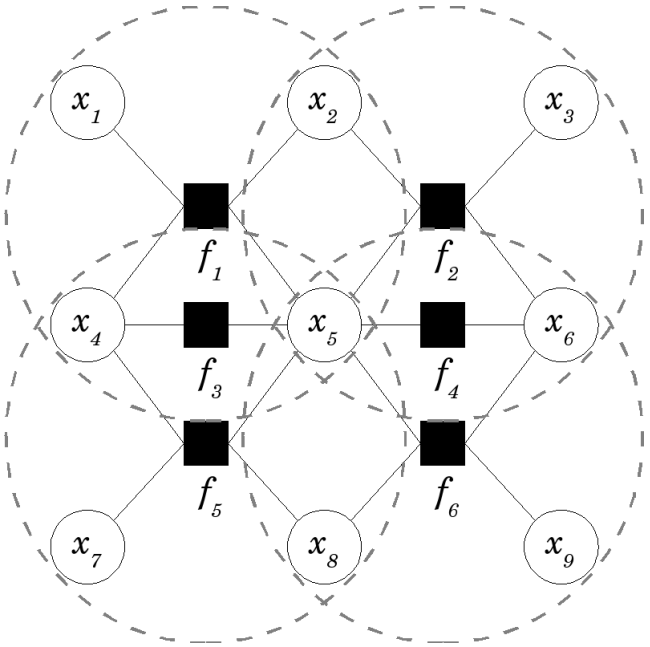

FIG. 3.6 – Clustering d'un graphe de Tanner

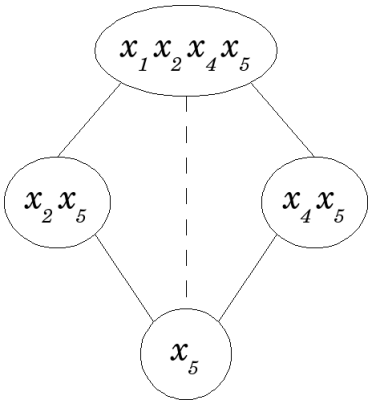

FIG. 3.7 – Diagramme de Hasse

Le diagramme de Hasse de cet ensemble est l'expression graphique de ces relations.

On ne représente pas les relations d'ordre indirect, comme $\{x_5\}$ avec $\{x_1, x_2, x_4, x_5\}$, s'il existe au moins un sous-ensemble intermédiaire selon la relation d'ordre, ici $\{x_2, x_5\}$ et $\{x_4, x_5\}$. Le diagramme de Hasse n'admet pas non plus de relation indirecte même s'il n'y a pas d'intermédiaires, c'est pourquoi $\{x_1\}$ n'est pas représenté, il est isolé.

On utilise cette représentation en arbre ordonné pour construire un diagramme du code, qu'on appelle *graphe des régions*, étant donné que l'ensemble des noeuds de variable est un poset avec la relation d'inclusion. Ce graphe est composé en différents niveaux, chaque niveau étant défini par la taille de ses clusters, ou *régions*. La construction du graphe des régions suit la même méthode que celle du diagramme de Hasse à la différence que les relations indirectes sont acceptées, sous-réserve qu'il n'y ait pas de noeuds intermédiaires. On appelle cette construction *cluster variation method*. Les deux niveaux extrêmes correspondent d'une part aux clusters globaux, construits selon les équations de parité, et d'autre part aux clusters

élémentaires ne contenant qu'une seule variable.

3.2.2 Cluster Variation Method

La construction du graphe des régions se fait, selon [2], comme suit :

1. Construction de l'ensemble des clusters, ou *régions*, globaux. Un cluster global contient la fonction de parité ou les fonctions de parité, si certaines sont incluses dans d'autres, et les variables de cette fonction,
2. Construction du niveau inférieur par recherche d'intersections entre les clusters globaux, ces intersections étant les échanges entre les clusters,
3. Construction de chaque niveau inférieur par recherche d'intersections entre les clusters du niveau supérieur.
   Remarque : les clusters d'un même niveau sont de même taille.

De même que pour la construction des clusters dans le cadre de la physique statistique et du calcul de l'énergie d'un réseau, on assigne à chaque cluster $\mathcal{R}$ un compteur $c_{\mathcal{R}}$ indiquant la redondance de ce cluster dans le graphe des régions. On calcule ce compteur comme suit :

$$c_{\mathcal{R}} = 1 - \sum_{S \subset \mathcal{S}(\mathcal{R})} c_S \tag{3.6}$$

où $\mathcal{S}(\mathcal{R})$ est l'ensemble des *super clusters* de $\mathcal{R}$, i.e. l'ensemble des clusters contenant $\mathcal{R}$. On définit les compteurs des clusters globaux à 1. La condition pour obtenir un graphe des régions sur un réseau de N noeuds de variable qui soit valide, c'est-à-dire qui respecte la non-redondance dans le calcul des énergies, est :

$$\forall i \in \{1..N\}, \quad \sum_{S \supset i} c_S = 1 \tag{3.7}$$

3.3 Belief Propagation Généralisé

En travaillant sur cette nouvelle représentation graphique d'un code, on déduit des informations d'inférences différentes de celles déduites d'un graphe de Tanner. Dans le graphe de Tanner, les données extraites sont individuelles, propres à un seul noeud de variable, ce qui est une autre manière de voir l'hypothèse dommageable d'indépendance.
Dans le graphe des régions, le principe fondamental est le même : on calcule des mises à jour de messages entre les clusters, et après un certain nombre d'itérations (définies plus loin), on extrait les estimations des marginales sur les clusters élémentaires. Dans la suite, on préfèrera appeler les clusters des régions.

3.3.1 Croyance-région

Dans le cas du graphe de Tanner, la croyance $b_i(x_i)$ sur une variable x_i est définie comme le produit de tous les messages incidents, voir l'équation (2.11). Dans le cas du graphe des régions, on étend cette définition à une région : la croyance $b_{\mathcal{R}}(\mathbf{x}_{\mathcal{R}})$ d'une région $\mathcal{R}$ est définie comme le produit de tous les messages incidents dans la région $\mathcal{R}$ et dans ses sous-régions (les régions incluses dans $\mathcal{R}$). On note $\mathcal{D}(\mathcal{R})$ l'ensemble de ses sous-régions et $\mathcal{R}$.

$$b_{\mathcal{R}}(\mathbf{x}_{\mathcal{R}}) = \frac{1}{Z} f_{\mathcal{R}}(\mathbf{x}_{\mathcal{R}}) v_{\mathcal{R}}(\mathbf{x}_{\mathcal{R}}) \prod_{\substack{X \not\subset \mathcal{D}(\mathcal{R}) \\ Y \subset \mathcal{D}(\mathcal{R})}} m_{X \to Y}(\mathbf{x}_Y) \tag{3.8}$$

où $f_{\mathcal{R}}(\mathbf{x}_{\mathcal{R}})$ est l'indicatrice sur l'équation de parité, si la région est une région globale. Dans le cas contraire, cette fonction est fixée à 1. $v_{\mathcal{R}}(\mathbf{x}_{\mathcal{R}})$ est le produit des vraisemblances de chacune des variables de la région, on notera le produit des vraisemblances par l'indicatrice de parité $L_{\mathcal{R}}(x_R)$, que nous appellerons le *coefficient local* de $\mathcal{R}$.

Exemple

Soit le graphe des régions déduit du graphe de Tanner précédent, donné par la figure suivante.

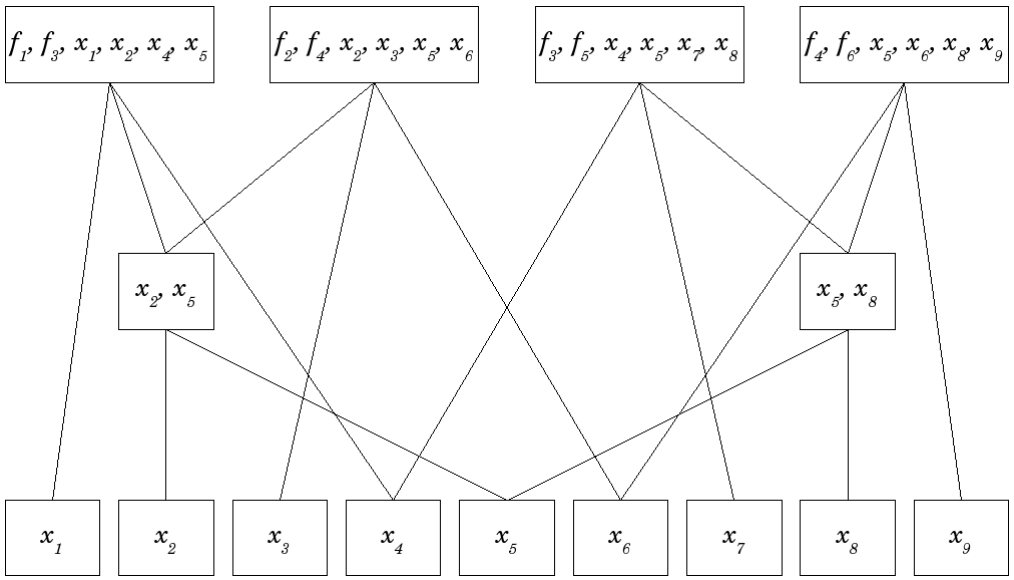

FIG. 3.8 – Graphe des régions

On définit la croyance $b_{1245}(\mathbf{x}_{1245})$ de la région grisée sur la figure suivante :

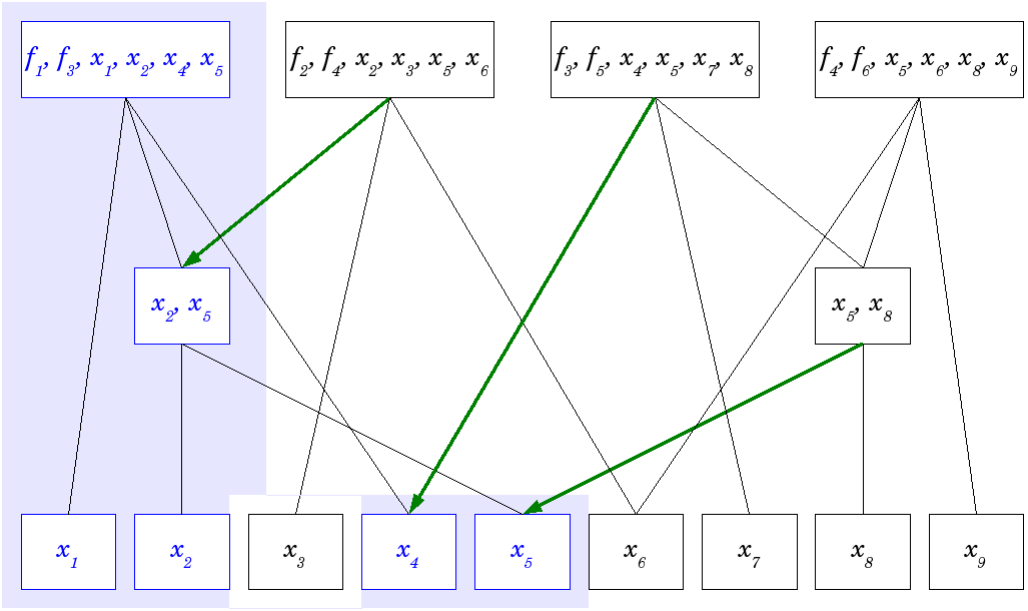

FIG. 3.9 – Croyance de la région $f_1, f_3, x_1, x_2, x_4, x_5$

en notant donc : $L_{1245}(\mathbf{x}_{1245}) = v_1(x_1)v_2(x_2)v_4(x_4)v_5(x_5) \times f_1(\mathbf{x}_{1245})f_3(\mathbf{x}_{45})$

$$b_{1245}(\mathbf{x}_{1245}) = \frac{1}{Z} L_{1245}(\mathbf{x}_{1245}) m_{2356\to 25}(\mathbf{x}_{25}) m_{4578\to 4}(x_4) m_{58\to 5}(x_5)$$

3.3.2 Messages

Les croyances sont des estimations de probabilités, elles respectent donc les mêmes règles de marginalisation. En effet, pour toute sous-région $\mathcal{S}$ d'une région $\mathcal{R}$, on a la marginalisation suivante :

$$b_{\mathcal{S}}(\mathbf{x}_{\mathcal{S}}) = \sum_{\mathbf{x}_R \setminus \mathbf{x}_S} b_{\mathcal{R}}(\mathbf{x}_{\mathcal{R}}) \tag{3.9}$$

En développant l'expression des croyances, on cherche à construire une équation de mise à jour du message de $\mathcal{R}$ vers $\mathcal{S}$. On a :

$$L_{\mathcal{S}}(\mathbf{x}_{\mathcal{S}}) \prod_{\substack{X \not\subset \mathcal{D}(\mathcal{S}) \\ Y \subset \mathcal{D}(\mathcal{S})}} m_{X \to Y}(\mathbf{x}_Y) = \sum_{\mathbf{x}_R \setminus \mathbf{x}_S} L_{\mathcal{R}}(\mathbf{x}_{\mathcal{R}}) \prod_{\substack{X \not\subset \mathcal{D}(\mathcal{R}) \\ Y \subset \mathcal{D}(\mathcal{R})}} m_{X \to Y}(x_Y)$$

Or $\mathcal{S} \subset \mathcal{D}(\mathcal{R})$ donc on peut factoriser les messages conformément au schéma suivant.

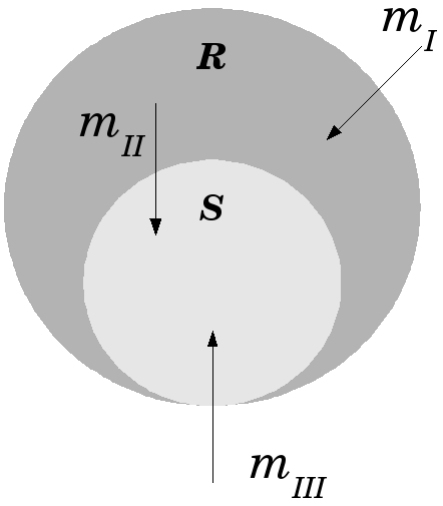

FIG. 3.10 – Factorisation des messages

$$L_{\mathcal{S}}(\mathbf{x}_{\mathcal{S}}) \overbrace{\prod_{\substack{X \subset \mathcal{D}(\mathcal{R}) \setminus \mathcal{D}(\mathcal{S}) \\ Y \subset \mathcal{D}(\mathcal{S})}} m_{X \to Y}(\mathbf{x}_Y)}^{\text{messages type II}} \overbrace{\prod_{\substack{X \not\subset \mathcal{D}(\mathcal{R}) \\ Y \subset \mathcal{D}(\mathcal{S})}} m_{X \to Y}(\mathbf{x}_Y)}^{\text{messages type III}}$$

$$= \sum_{\mathbf{x}_R \setminus \mathbf{x}_S} L_{\mathcal{R} \setminus \mathcal{S}}(\mathbf{x}_{\mathcal{R}} \setminus \mathbf{x}_{\mathcal{S}}) L_{\mathcal{S}}(\mathbf{x}_{\mathcal{S}}) \overbrace{\prod_{\substack{X \not\subset \mathcal{D}(\mathcal{R}) \\ Y \subset \mathcal{D}(\mathcal{R}) \setminus \mathcal{D}(\mathcal{S})}} m_{X \to Y}(x_Y)}^{\text{messages type I}} \overbrace{\prod_{\substack{X \not\subset \mathcal{D}(\mathcal{R}) \\ Y \subset \mathcal{D}(\mathcal{S}))}} m_{X \to Y}(x_Y)}^{\text{messages type III}}$$

On obtient par simplification sur les messages de type III :

$$\prod_{\substack{X \subset \mathcal{D}(\mathcal{R}) \setminus \mathcal{D}(\mathcal{S}) \\ Y \subset \mathcal{D}(\mathcal{S})}} m_{X \to Y}(\mathbf{x}_Y) = \sum_{\mathbf{x}_R \setminus \mathbf{x}_S} L_{\mathcal{R} \setminus \mathcal{S}}(\mathbf{x}_{\mathcal{R}} \setminus \mathbf{x}_{\mathcal{S}}) \prod_{\substack{X \not\subset \mathcal{D}(\mathcal{R}) \\ Y \subset \mathcal{D}(\mathcal{R}) \setminus \mathcal{D}(\mathcal{S})}} m_{X \to Y}(x_Y)$$

Les règles de mises à jour selon [2] indiquent que le membre de gauche est choisi comme étant celui mis à jour, le membre de droite étant calculé à l'itération précédente.

On obtient ainsi la mise à jour d'un produit de messages :

$$\prod_{\substack{X \subset \mathcal{D}(\mathcal{R}) \setminus \mathcal{D}(\mathcal{S}) \\ Y \subset \mathcal{D}(\mathcal{S})}} m^{update}_{X \to Y}(\mathbf{x}_Y) = \sum_{\mathbf{x}_R \setminus \mathbf{x}_S} L_{\mathcal{R} \setminus \mathcal{S}}(\mathbf{x}_{\mathcal{R}} \setminus \mathbf{x}_{\mathcal{S}}) \prod_{\substack{X \not\subset \mathcal{D}(\mathcal{R}) \\ Y \subset \mathcal{D}(\mathcal{R}) \setminus \mathcal{D}(\mathcal{S})}} m^{old}_{X \to Y}(x_Y)$$

En terme d'implémentation, on préfèrera une équation de mise à jour sur un seul message. Pour cela, il suffit de séparer le membre de gauche en un produit d'un message particuler par tous les autres, on choisira le message de la région $\mathcal{R}$ vers la région $\mathcal{S}$. On notera $\mathcal{E}(\mathcal{R})$ l'ensemble $\mathcal{D}(\mathcal{R})\backslash\mathcal{R}$.

$$m^{update}_{\mathcal{R}\to\mathcal{S}}(x_{\mathcal{S}}) \prod_{\substack{X\subset\mathcal{E}(\mathcal{R})\backslash\mathcal{D}(\mathcal{S})\\ Y\subset\mathcal{E}(\mathcal{S})}} m^{update}_{X\to Y}(\mathbf{x}_Y) = \sum_{\mathbf{x}_R\backslash\mathbf{x}_S} L_{\mathcal{R}\backslash\mathcal{S}}(\mathbf{x}_{\mathcal{R}}\backslash\mathbf{x}_{\mathcal{S}}) \prod_{\substack{X\not\subset\mathcal{D}(\mathcal{R})\\ Y\subset\mathcal{D}(\mathcal{R})\backslash\mathcal{D}(\mathcal{S})}} m^{old}_{X\to Y}(x_Y)$$

soit

$$m^{update}_{\mathcal{R}\to\mathcal{S}}(x_{\mathcal{S}}) = \frac{\sum_{\mathbf{x}_R\backslash\mathbf{x}_S} L_{\mathcal{R}\backslash\mathcal{S}}(\mathbf{x}_{\mathcal{R}}\backslash\mathbf{x}_{\mathcal{S}}) \prod_{\substack{X\not\subset\mathcal{D}(\mathcal{R})\\ Y\subset\mathcal{D}(\mathcal{R})\backslash\mathcal{D}(\mathcal{S})}} m^{old}_{X\to Y}(x_Y)}{\prod_{\substack{X\subset\mathcal{E}(\mathcal{R})\backslash\mathcal{D}(\mathcal{S})\\ Y\subset\mathcal{E}(\mathcal{S})}} m^{update}_{X\to Y}(\mathbf{x}_Y)} \tag{3.10}$$

Exemple

Soit le code suivant :

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

Le graphe des régions de ce code est sur la figure suivante. Les régions sont dénotées par les indices de leurs variables et on a retiré les fonctions de parité pour une meilleur lecture du graphe.

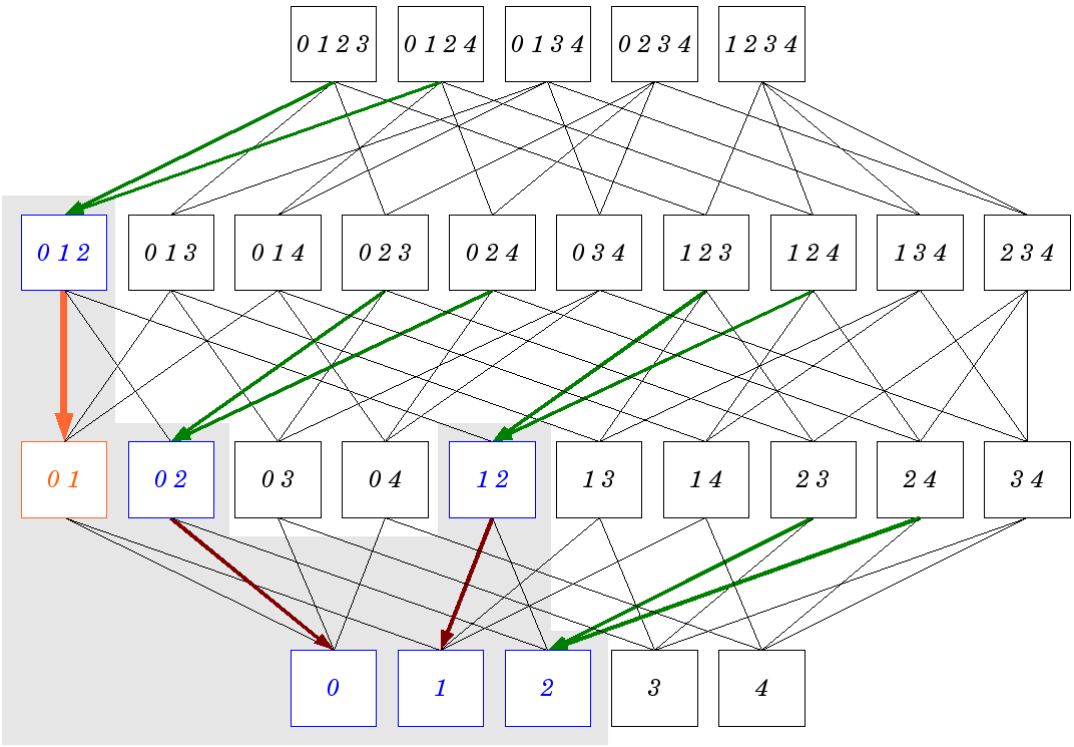

FIG. 3.11 – Messages $m_{012\to 01}(\mathbf{x}_{12})$

Bien que ce code n'ait pas grand intérêt, car de rendement nul, son graphe des régions est suffisamment plein pour donner un bon exemple de calcul. Le message de la région

$\{x_0, x_1, x_2\}$ vers la région $\{x_0, x_1\}$ se met à jour selon l'équation suivante :

$$\begin{aligned} m^{update}_{012\to 01}(\mathbf{x}_{01}) = \sum_{x_2} L_2(x_2) \quad & \times m^{old}_{0123\to 012}(\mathbf{x}_{012}) m^{old}_{0124\to 012}(\mathbf{x}_{012}) \\ & \times m^{old}_{023\to 02}(\mathbf{x}_{02}) m^{old}_{024\to 02}(\mathbf{x}_{02}) \\ & \times m^{old}_{123\to 12}(\mathbf{x}_{12}) m^{old}_{124\to 12}(\mathbf{x}_{12}) \\ & \times \frac{1}{m^{new}_{02\to 0}(x_0) m^{new}_{12\to 0}(x_1)} \end{aligned} \tag{3.11}$$

Conformément à la figure correspondante :
- les messages de couleurs verte sont les constituants du numérateur (les vraisemblances ne sont pas présentes dans le graphe mais elles sont considérées comme sources de messages),
- les messages de couleurs rouge sont les constituants du dénominateur.

3.3.3 Itération

Du fait de la séparation des messages dans la mise à jour, l'ordre de mise à jour ne doit pas être aléatoire. En effet, nous avons besoin pour chaque message des messages du niveau inférieur, on choisit donc de mettre à jour les messages par niveau en partant du niveau le plus bas. Un tel parcours porte le nom de *parcours en largeur d'abord*, d'après [5]. Une itération du GBP est définie comme la mise à jour de tous les messages selon cet ordre.

3.3.4 Initialisation

L'initialisation choisie est une distribution uniforme sur un message. Un message a autant d'états que son destinataire, chaque message portant la pondération de cet état. Ainsi, un message $m_{X\to Y}$ est en fait un vecteur de taille $2^{\text{nombre de variables dans } Y}$, ce qui permet d'établir l'initialisation :

$$\forall (X, Y), \text{ avec } \nu = \text{nombre de variables dans } Y, \quad m^{init}_{X\to Y}(x_Y) = \frac{1}{2^\nu}$$

3.3.5 Algorithme

(voir algorithme 2)

De même que dans le BP, on décide du nombre d'itérations selon le taux de convergence du graphe.

3.4 Avantages - Inconvénients

Nous présentons dans cette sous-section quelque codes particuliers qui illustrent l'avantage du GBP sur le BP, et également d'autres codes qui illustrent les limites du GBP.

3.4.1 Code à cycle-2

Voici un graphe de Tanner d'un code très simple de taille 4 présentant un unique cycle de taille 2 (la taille est donnée par le nombre de noeuds de variable).

On déduit le graphe des régions.

On calcule ensuite différentes croyances, conformément aux équations précédentes, pour les comparer avec les marginales exactes. Par symétrie, on ne calculera que les croyances sur les variables x_1 et x_2.

Algorithme 2 : Une itération du Belief Propagation Généralisé

pour chaque *niveau* n **faire**
  pour chaque *région* $\mathcal{R}_n$ **faire**
    pour chaque *sous-région directe* $\mathcal{S}_\mathcal{R}$ **faire**
      Message de $\mathcal{R}_n$ *vers* $\mathcal{S}_\mathcal{R}$
      pour chaque *état* $\boldsymbol{x}_\mathcal{S}$ *de* $\mathcal{S}_\mathcal{R}$ **faire**
        $produit = L_{\mathcal{R}_n \backslash \mathcal{S}_\mathcal{R}}(\boldsymbol{x}_{\mathcal{R}_n} \backslash \boldsymbol{x}_\mathcal{S})$
        pour chaque *région* $\mathcal{E}$ *du graphe* **faire**
          si $\mathcal{E} \not\subset \mathcal{R}_n$ **alors**
            pour chaque $\mathcal{K} \in \mathcal{D}(\mathcal{E})$ **faire**
              si $\mathcal{K} \in \mathcal{D}(\mathcal{R}_n) \backslash \mathcal{D}(\mathcal{S}_\mathcal{R})$ **alors**
                $produit = produit \times m_{\mathcal{E} \to \mathcal{K}}(x_\mathcal{K})$
        $m_{\mathcal{R}_n \to \mathcal{S}_\mathcal{R}}(\boldsymbol{x}_\mathcal{S}) = m_{\mathcal{R}_n \to \mathcal{S}_\mathcal{R}}(\boldsymbol{x}_\mathcal{S}) + produit$
        si *dernier passage sur la configuration* $\boldsymbol{x}_\mathcal{S}$ **alors**
          pour chaque *région* $\mathcal{E}$ *du graphe* **faire**
            si $\mathcal{E} \subset \mathcal{R}_n \backslash \mathcal{S}_\mathcal{R}$ **alors**
              pour chaque $\mathcal{K} \in \mathcal{D}(\mathcal{E})$ **faire**
                si $\mathcal{K} \in \mathcal{D}(\mathcal{S}_\mathcal{R})$ **alors**
                  $m_{\mathcal{R}_n \to \mathcal{S}_\mathcal{R}}(\boldsymbol{x}_\mathcal{S}) = \dfrac{m_{\mathcal{R}_n \to \mathcal{S}_\mathcal{R}}(\boldsymbol{x}_\mathcal{S})}{m_{\mathcal{E} \to \mathcal{K}}(x_\mathcal{K})}$

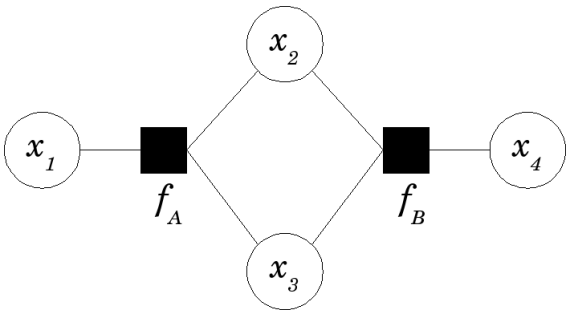

FIG. 3.12 – Code à cycle-2

Croyance sur x_1

$$
\begin{aligned}
b_1(x_1) \quad &= v_1(x_1) m_{123 \to 1}(x_1) \\
&= v_1(x_1) \sum_{x_2} \sum_{x_3} v_2(x_2) v_3(x_3) f_A(\mathbf{x}_{123}) m_{234 \to 23}(\mathbf{x}_{23}) \\
&= v_1(x_1) \sum_{x_2} \sum_{x_3} v_2(x_2) v_3(x_3) f_A(\mathbf{x}_{123}) \sum_{x_4} v_4(x_4) f_B(\mathbf{x}_{234})
\end{aligned}
$$

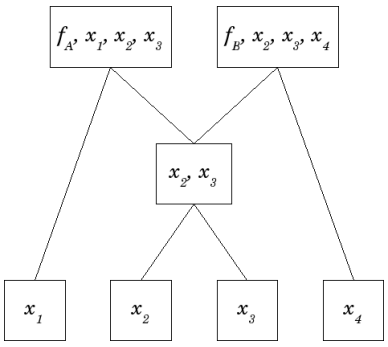

FIG. 3.13 – Graphe des régions - Code à cycle-2

$$= \sum_{x_2} \sum_{x_3} \sum_{x_4} v_1(x_1) v_2(x_2) v_3(x_3) v_4(x_4) f_A(\mathbf{x}_{123}) f_B(\mathbf{x}_{234}) \tag{3.12}$$

On obtient une écriture canonique, identique à la marginale exacte de x_1.

Croyance sur x_2

$$\begin{aligned} b_2(x_2) &= v_2(x_2) m_{23\to 2}(x_2) \\ &= v_2(x_2) \sum_{x_3} v_3(x_3) m_{123\to 23}(\mathbf{x}_{23}) m_{234\to 23}(\mathbf{x}_{23}) \\ &= v_2(x_2) \sum_{x_3} v_3(x_3) \sum_{x_1} v_1(x_1) f_A(\mathbf{x}_{123}) \sum_{x_4} v_4(x_4) f_B(\mathbf{x}_{234}) \\ &= \sum_{x_1} \sum_{x_3} \sum_{x_4} v_1(x_1) v_2(x_2) v_3(x_3) v_4(x_4) f_A(\mathbf{x}_{123}) f_B(\mathbf{x}_{234}) \end{aligned} \tag{3.13}$$

On obtient une écriture canonique, identique à la marginale exacte de x_2.
Les croyances étant exactes malgré le cycle du graphe, l'algorithme du GBP est meilleur que le BP.

3.4.2 Code à cycle-3

Reprenons le graphe de Tanner et ajoutons un noeud de variable dans le cycle.

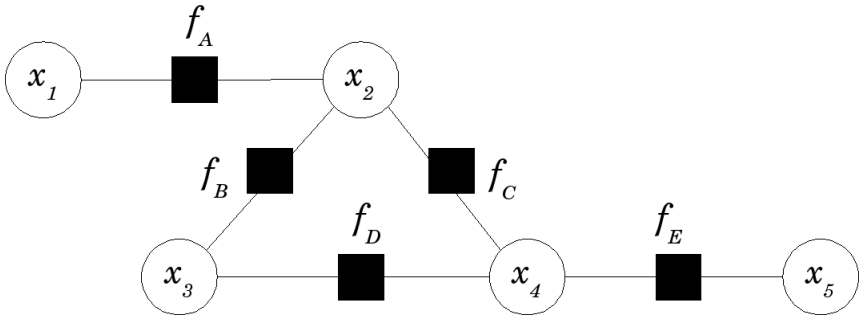

FIG. 3.14 – Code à cycle-3

On déduit son graphe des régions.

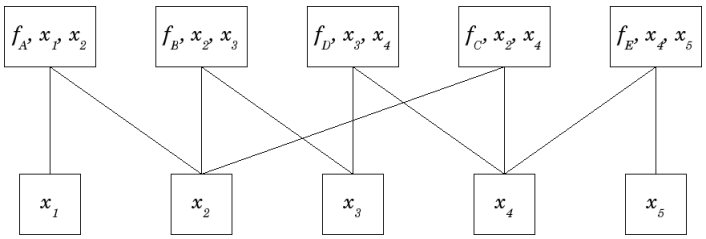

FIG. 3.15 – Graphe des régions - Code à cycle-3

Croyance sur x_1

$$\begin{aligned} b_1(x_1) &= v_1(x_1)m_{12\to 1}(x_1) \\ &= v_1(x_1)\sum_{x_2} v_2(x_2)f_1(\mathbf{x}_{12})m_{23\to 2}(x_2)m_{24\to 2}(x_2) \end{aligned}$$

On calcule le message $m_{23\to 2}(x_2)$:

$$\begin{aligned} m_{23\to 2}(x_2) &= \sum_{x_3} v_3(x_3)f_B(\mathbf{x}_{23})m_{34\to 3}(x_3) \\ &= \sum_{x_3} v_3(x_3)f_B(\mathbf{x}_{23})\sum_{x_4} v_4(x_4)f_C(\mathbf{x}_{24})m_{24\to 4}(x_4)m_{45\to 4}(x_4) \end{aligned}$$

or :

$$m_{24\to 4}(x_4) = \sum_{x_2} v_2(x_2)f_C(\mathbf{x}_{24})m_{23\to 2}(x_2)m_{12\to 2}(x_2)$$

Donc on obtient en résumé : $m_{23\to 2}(x_2) = g(m_{23\to 2}(x_2))$, où g est la fonction détaillant le calcul ci-dessus. Ceci correspond à une boucle dans le calcul, boucle non résolvable. La marginalisation de x_1 est donc approximative, le GBP n'a pas absorber le cycle.

3.4.3 Code à cycles imbriqués de taille 2

On choisit le graphe de Tanner suivant.

On a un cycle global de taille 4 et plusieurs cycles de taille 2 et 3 imbriqués dans le global.

Croyance sur x_1

$$\begin{aligned} b_1(x_1) &= v_1(x_1)m_{12\to 1}(x_1) \\ &= v_1(x_1)\sum_{x_2} v_2(x_2)f_A(\mathbf{x}_{12})m_{2345\to 2}(x_2) \\ &= v_1(x_1)\sum_{x_2} v_2(x_2)f_A(\mathbf{x}_{12})\sum_{x_3}\sum_{x_4}\sum_{x_5} v_3(x_3)v_4(x_4)v_5(x_5) \\ &\times f_B(\mathbf{x}_{23})f_C(\mathbf{x}_{24})f_D(\mathbf{x}_{2345})f_E(\mathbf{x}_{35})f_F(\mathbf{x}_{45})m_{56\to 5}(x_5) \end{aligned}$$

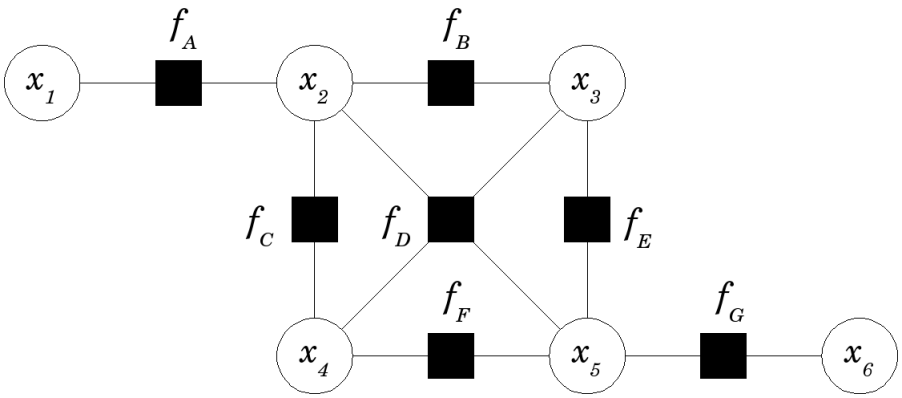

FIG. 3.16 – Code à cycles imbriqués

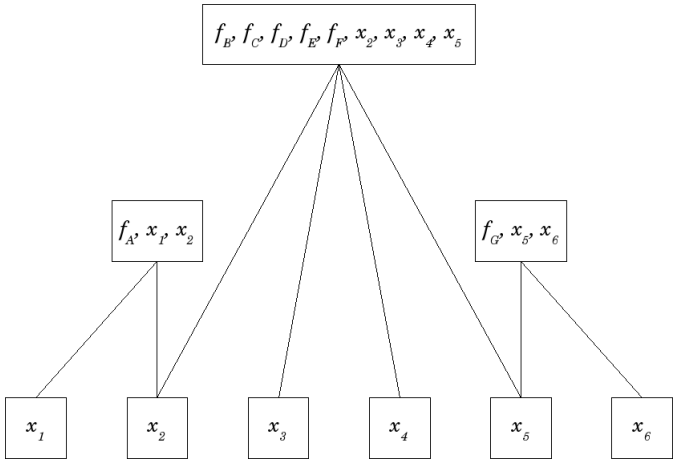

FIG. 3.17 – Graphe des régions - Code à cycles imbriqués

$$
\begin{aligned}
&= v_1(x_1) \sum_{x_2} v_2(x_2) f_A(\mathbf{x}_{12}) \sum_{x_3} \sum_{x_4} \sum_{x_5} v_3(x_3) v_4(x_4) v_5(x_5) \\
&\times f_B(\mathbf{x}_{23}) f_C(\mathbf{x}_{24}) f_D(\mathbf{x}_{2345}) f_E(\mathbf{x}_{35}) f_F(\mathbf{x}_{45}) \sum_{x_6} v_6(x_6) f_G(\mathbf{x}_{56}) \\
&= \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} \sum_{x_6} v_1(x_1) v_2(x_2) v_3(x_3) v_4(x_4) v_5(x_5) v_6(x_6) \\
&\times f_A(\mathbf{x}_{12}) f_B(\mathbf{x}_{23}) f_C(\mathbf{x}_{24}) f_D(\mathbf{x}_{2345}) f_E(\mathbf{x}_{35}) f_F(\mathbf{x}_{45}) f_G(\mathbf{x}_{56})
\end{aligned}
$$

On retrouve la marginalisation exacte. On peut montrer de la même manière que les autres croyances sont les marginales exactes.
De ces résultats empiriques, on peut tenter de déduire des propositions sur le rapport entre topologie de graphe et capacité de correction.

Proposition 1 : *Tout code dont les cycles sont élémentaires, i.e. de taille* 2*, est décodé par le GBP.*

Proposition 2 : *Pour tout code dont les cycles sont de taille au moins* 3*, les croyances sur les noeuds de variables ne sont pas exactes.*

3.4.4 Résultats

L'implémentation est terminée mais les simulations sont longues, étant donnée l'aspect très complexe de l'algorithme. Les premières données récupérées ne sont pour le moment pas suffisamment probantes pour illustrer les situations de la section précédente.
On considère les codes suivants : code à cycle de taille 2, code à cycle de taille 3. On compare les valeurs de taux d'erreurs binaires avec notamment la sortie sans décodage par GBP.

RSB	2-cycle GBP	2-cycle	3-cycle GBP	3-cycle
1.0	0.0546	0.1319	0.0644	0.2009
1.2	0.0532	0.1226	0.0626	0.1929
1.4	0.0473	0.1203	0.0575	0.1845
1.6	0.0441	0.1120	0.0524	0.1819
1.8	0.0397	0.1120	0.0503	0.1771
2.0	0.0371	0.1032	0.0444	0.1716
2.2	0.0326	0.0987	0.0411	0.1638
2.4	0.0313	0.0935	0.0377	0.1592
2.6	0.0256	0.0882	0.0312	0.1518
2.8	0.0235	0.0832	0.0352	0.1493
3.0	0.0240	0.0810	0.0316	0.1448
3.2	0.0195	0.0730	0.0265	0.1375
3.4	0.0174	0.0695	0.0251	0.1310
3.6	0.0145	0.0639	0.0219	0.1262
3.8	0.0144	0.0601	0.0184	0.1199
4.0	0.0122	0.0550	0.0185	0.1177
4.2	0.0120	0.0539	0.0165	0.1103
4.4	0.0084	0.0498	0.0158	0.1057
4.6	0.0077	0.0437	0.0130	0.0999
4.8	0.0069	0.0421	0.0110	0.0930
5.0	0.0064	0.0360	0.0090	0.0865
5.2	0.0061	0.0339	0.0087	0.0853
5.4	0.0056	0.0314	0.0075	0.0786
5.6	0.0045	0.0283	0.0070	0.0744
5.8	0.0031	0.0252	0.0055	0.0701

On a l'illustration qu'une cycle-3 (cycle de taille 3) dégrade la correction par rapport à un cycle-2 : le GBP ne converge pas exactement vers les marginales exactes. Cependant, les performances sont meilleurs qu'avec un décodage sans GBP.

Conclusion - Perspectives

D'après l'étude mené sur le GBP dans quelques cas simples, on comprend que cet algorithme ne peut pas rendre optimal n'importe quel décodage. Les bonnes conditions topologiques décrites ne sont malheureusement pas fréquentes dans les codes modernes type LDPC. La plupart de ces codes contiennent des cycles de taille 4 minimum, ce qui rend le GBP approximatif. Les futures simulations pourtont nous informer sur le meilleur choix à faire dans l'approximation entre BP et GBP. L'idéal pour bien décoder serait de pouvoir construire un graphe des régions dont les clusters globaux seraient circonscrits aux cycles de taille maximal, l'absorption serait alors totale. Cette construction nécessite cependant une connaissance parfaite de la topologie du graphe de Tanner des codes, ce qui est une étude exhaustive et fastidieuse.

Le GBP a l'avantage malgré tout de lier deux domaines : la physique statistique et la théorie de l'information. Les modèles d'approximation de l'énergie libre d'un réseau sont à exploiter encore plus en profondeur pour tenter de ressortir une approche encore plus globale afin de corriger les approximations de cycle.

Bibliographie

[1] Hidetoshi Nishimori. *Statistical Physics of Spin Glasses and Information Processing : An Introduction*. Clarendon Press, 2001.

[2] Jonathan S. Yedida, William T. Freeman, and Yair Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Décembre 2004.

[3] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. Février 2001.

[4] Jonathan S. Yedida, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. Janvier 2002.

[5] Arnaud Revel. Cours d'intelligence artificielle master sic. 2008/2009.

[6] David Mac Kay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[7] Claude Coulon and Stéphanie Moreau. *Physique statistique et thermodynamique : cours et exercices corrigés*, dunod edition, 2000.