

# Codage de sources avec information adjacente et connaissance imparfaite de la corrélation : le problème des cadrans

Elsa DUPRAZ<sup>1</sup>, Aline ROUMY<sup>2</sup>, Michel KIEFFER<sup>1,3,4</sup>

<sup>1</sup>L2S - CNRS - SUPELEC - Univ Paris-Sud, 3 rue Joliot Curie, 91190 Gif-sur-Yvette, France

<sup>2</sup>INRIA, Campus de Beaulieu, 35042 Rennes, France

<sup>3</sup>À temps partiel au LTCI – CNRS Télécom ParisTech, 75013 Paris, France

<sup>4</sup>Institut universitaire de France

elsa.dupraz@lss.supelec.fr, aline.roumy@inria.fr, michel.kieffer@lss.supelec.fr

**Résumé** – Cet article aborde le problème du codage d’une source  $X$  avec information adjacente  $Y$  disponible uniquement au décodeur dans le cas où  $P(Y|X)$  est mal connue au codeur et au décodeur. Ce type de problème apparaît lors de la compression de données prélevées par des capteurs en réseau. Nous considérons des sources  $q$ -aires et supposons qu’il existe  $Z$  tel que  $Y = X + Z$  et  $i \in GF(q)$ , inconnu, tel que  $P(Z = i) = 1 - (q - 1)p$  et  $P(Z = j) = p, j \neq i$ , avec  $p \in \mathbb{P} \subset [0, 1/q]$  également inconnu. Pour un paramètre  $p$  fixé, l’entropie de  $Z$  est indépendante de  $i$ . Cependant, l’incertitude sur  $p$  et la méconnaissance de  $i$  rendent inefficaces les schémas classiques de codage avec information adjacente au décodeur. Cet article propose un schéma de codage reposant sur des codes LDPC non binaires. Le décodeur exploite un algorithme EM pour estimer conjointement  $p, i$  et les symboles émis par la source. Un estimateur permettant d’initialiser efficacement l’algorithme EM est également proposé. Nous montrons qu’un estimateur au sens du maximum de vraisemblance naïf ne permet pas d’estimer  $i$  et  $p$  correctement pour l’initialisation et proposons une méthode d’estimation plus pertinente. L’efficacité du schéma proposé est comparée avec celle de techniques classiques, montrant sa supériorité tant en termes d’erreurs d’estimation de  $X$  qu’en termes de temps de calcul.

**Abstract** – This paper considers the problem of lossless source coding of a source  $X$  with the help of a side information  $Y$  available at the decoder only. Here, we assume that the conditionnal distribution  $P(Y|X)$  is not perfectly known, both at the encoder and the decoder. Indeed, this distribution can be difficult to obtain in practical situations such as distributed coding of measurements by sensors in a network. We consider  $q$ -ary sources and assume an additive model, *i.e.*, there exists a random variable  $Z$  such that  $Y = X + Z$  and  $i \in GF(q)$  such that  $P(Z = i) = 1 - (q - 1)p$  and  $P(Z = j) = p, j \neq i$ , with  $p \in \mathbb{P} \subset [0, 1/q]$  also unknown. For some fixed parameter  $p$ , the entropy of  $Z$  is independent of  $i$ . Nonetheless, in this setup, the classical coding schemes with side information at the decoder are inefficient due to the uncertainty on  $p$  and to the fact that  $i$  is unknown. This paper proposes a coding scheme based on non-binary LDPC codes. The decoder estimates jointly  $p$  and  $i$  with the help of an EM algorithm. We also propose a method to initialize the EM algorithm properly. We show that a naive Maximum Likelihood estimator does not give satisfying estimates of  $p$  and  $i$  for the initialization and we propose a more efficient estimation method. The coding scheme we propose is compared to classical solutions. It performs better in terms of error probability and complexity.

## 1 Introduction

Dans les schémas de codage d’une source  $X$  avec information adjacente  $Y$  disponible uniquement au décodeur (voir Figure 1), la distribution jointe  $P(X, Y)$  est en général supposée parfaitement connue. Ceci a permis de concevoir des solutions de codage reposant sur des codes de canal [11], comme les codes LDPC [7, 8]. Cependant, en pratique, la distribution jointe peut être difficile à déterminer. Ainsi, dans un réseau de capteurs,  $X$  représente la mesure d’un capteur, qui doit faire parvenir une version compressée de  $X$  en un point de collecte, et  $Y$  représente les mesures prélevées par les autres capteurs dont le point de collecte peut se servir pour retrouver  $X$  :  $P(X, Y)$  peut dépendre du temps, de la configuration des capteurs, *etc.* La distribution marginale  $P(X)$  peut être estimée au codeur et utilisée lors de la compression [5], mais il est en revanche plus difficile de déterminer  $P(Y|X)$ , puisque  $X$  et  $Y$  ne sont pas disponibles simultanément au même endroit [2].

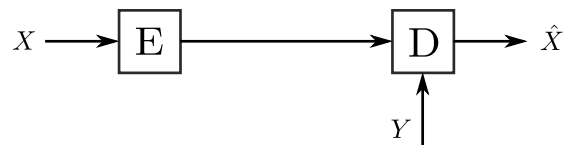


FIGURE 1 – Codage de Slepian-Wolf

Dans cet article, nous considérons le codage de sources  $q$ -aires dans le cas où  $P(Y|X)$  est mal connue. Plus spécifiquement, nous supposons qu’il existe  $Z$  tel que  $Y = X + Z$  et  $i \in GF(q)$ , inconnu, tel que  $P(Z = i) = 1 - (q - 1)p$  et  $P(Z = j) = p, j \neq i$ , avec  $p \in \mathbb{P} \subset [0, 1/q]$  également inconnu. Ce type de modèle permet de tenir compte d’une incertitude sur le niveau de corrélation entre  $X$  et  $Y$ , par l’intermédiaire de  $p \in \mathbb{P}$ , mais également sur le type de corrélation, par l’intermédiaire de  $i$ . Par exemple, dans le cas binaire, avec  $\mathbb{P} = [0, 0.1]$ , le fait de ne pas connaître  $i$  se traduit par une méconnaissance de la probabilité de transition  $p = P(Y = 1|X = 0) = P(Y =$

$0|X = 1$ ), qui peut être dans  $[0, 0.1]$  ou dans  $[0.9, 1]$ . Malgré cette incertitude, les cas  $p = 0.1$  et  $p = 0.9$ , par exemple, donnent la même entropie, et donc théoriquement le même débit de codage. De plus, on peut démontrer facilement à partir de [9], qu'un code LDPC a la même performance pour une probabilité de transition  $p$  ou  $1-p$  et les algorithmes de décodage proposés dans [7] sont tout à fait capables de prendre en compte le cas  $p > 0.5$ .

En revanche, quand on ne connaît pas  $p$  et qu'on sait pas non plus *a priori* si la probabilité de transition est donnée par  $p$  ou  $1-p$ , on se retrouve confrontés à deux difficultés. La première est de choisir un débit de codage adapté. La deuxième est de construire un décodeur efficace malgré le manque de connaissances sur la distribution de probabilités. Plusieurs solutions ont été proposées pour répondre à ces questions ; certaines proviennent du codage de canal et nécessiteraient un travail d'adaptation à notre problème. La solution la plus classique consiste en l'utilisation d'un canal de retour. Le codeur transmet une description de la source à un certain débit, le décodeur tente de reconstruire la source en utilisant les paramètres de la distribution de probabilité correspondant à ce débit. Si le décodage échoue, le décodeur demande plus de débit au codeur par l'intermédiaire du canal de retour [10]. Outre la difficulté de mise en place d'un canal de retour, cette solution implique de répéter le processus de décodage un certain nombre de fois, ce qui induit une complexité et des délais de transmission élevés. Une autre solution serait d'utiliser un algorithme de décodage LDPC de type *min-somme* [3], qui n'a pas besoin de la connaissance de  $P(Y|X)$  dans son exécution. En revanche, pour être véritablement efficace, l'algorithme doit être initialisé avec un coefficient donné, calculé, lui, à partir de cette distribution.

Cet article propose un schéma de codage sans canal de retour, et reposant sur des codes LDPC non binaires (voir Partie 3), directement inspiré de [4]. Le décodeur exploite un algorithme Expectation Maximization (EM) pour estimer conjointement  $p$ ,  $i$  et les symboles émis par la source. Nous montrons qu'un estimateur au sens du maximum de vraisemblance naïf ne permet pas de fournir des estimées de  $i$  et  $p$  satisfaisantes pour initialiser l'algorithme EM correctement (Partie 4). L'efficacité du schéma proposé est comparée avec des techniques classiques, montrant sa supériorité tant en termes d'erreurs d'estimation de  $X$  qu'en termes de temps de calcul (Partie 5).

## 2 Modèle considéré

Les variables aléatoires sont notées en lettres majuscules, leurs réalisations en minuscules, et les vecteurs en gras. Les sources  $X$  et  $Y$  génèrent des suites de variables aléatoires  $\{X_n\}_{n=1}^{+\infty}$  et  $\{Y_n\}_{n=1}^{+\infty}$  indépendantes, identiquement distribuées (iid) à valeurs dans  $\text{GF}(q)$ , où les quatre opérations élémentaires sont notées  $\oplus$ ,  $\ominus$ ,  $\otimes$ ,  $\oslash$ . Pour  $q = \alpha^m$ , avec  $\alpha$  premier, on note  $r = \exp\left(\frac{j2\pi}{\alpha}\right)$  la racine primitive associée à  $\text{GF}(q)$ . Le modèle de corrélation est décrit par  $Y = X \oplus Z$ , où  $Z$  est une source qui génère des variables aléatoires iid dans  $\text{GF}(q)$ .  $X$  est distribuée uniformément, tandis que la distribution de  $Z$  est décrite par un vecteur de paramètres  $\theta = [\theta_0, \dots, \theta_{q-1}]$ , avec  $\theta_k = P(Z = k)$ . La valeur prise par  $\theta$  n'est connue ni du codeur

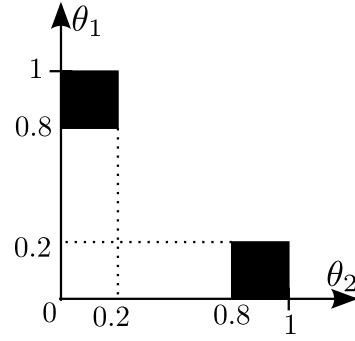


FIGURE 2 – Exemple dans le cas binaire : les zones noircies représentent l'ensemble des valeurs possibles pour  $\theta_1$  et  $\theta_2$

ni du décodeur et  $\theta$  peut varier d'une suite  $\{Z_n\}_{n=1}^{+\infty}$  à une autre. Nous faisons l'hypothèse que  $\theta$  peut prendre  $q$  formes différentes :  $\theta^{(0)}(p) = [1 - (q-1)p, \dots, p]$ ,  $\theta^{(1)}(p) = [p, 1 - (q-1)p, p, \dots, p]$ , etc., avec  $p \in \mathbb{P} \subset [0, 1/q]$ , et  $\mathbb{P}$  connu. On note  $\mathcal{P}_\theta$  l'ensemble des vecteurs  $\theta$  possibles. Cet ensemble n'est pas connexe. L'index  $i$  de  $\theta^{(i)}(p)$  est appelé le *cadran* de  $\theta$ , il permet de désigner la composante connexe de  $\mathcal{P}_\theta$  à laquelle  $\theta^{(i)}(p)$  appartient. La Figure 2 représente un exemple dans le cas binaire, avec  $\mathbb{P} = [0, 0.2]$ .

Pour ce modèle, le plus petit débit atteignable par un schéma de codage de source avec information adjacente au décodeur est donné par [1]

$$R = \sup_{\theta \in \mathcal{P}_\theta} H(X|Y, \theta). \quad (1)$$

le sup étant atteint pour  $q$  valeurs différentes de  $\theta$ , de cadrans différents, mais de même  $p$ .

Dans [4], un schéma de codage est proposé pour le type de source considéré ci-dessus, mais dans le cas où  $\mathcal{P}_\theta$  n'est constitué que d'une seule composante connexe (un seul cadran). Ce schéma ne peut être appliqué directement dans le cas où  $\mathcal{P}_\theta$  comporte plusieurs cadrans. En effet, le décodeur introduit dans [4] estime conjointement le vecteur de source  $\mathbf{x}$  et le vecteur de paramètres  $\theta$ , à l'aide d'un algorithme EM. Pour être efficace, ce dernier doit être correctement initialisé, en particulier dans le bon cadran, ce qui nécessite de proposer un premier estimateur grossier de  $\theta$ . Cependant, l'estimateur proposé dans [4] ne fonctionne pas lorsque  $\mathcal{P}_\theta$  comporte plusieurs cadrans, comme on le verra dans la partie 4.

## 3 Codage et décodage LDPC

Le système de codage auquel nous nous intéressons s'appuie sur des codes LDPC dans  $\text{GF}(q)$  que nous décrivons ici. Une autre possibilité serait de transformer les symboles non-binaires en plan de bits que l'on coderait séparément. Mais cela nécessiterait de mettre en œuvre des méthodes complexes pour prendre en compte les dépendances entre plans de bits lors du décodage [6].

Le codeur réalise le codage du vecteur de source  $\mathbf{x}$  de longueur  $n$  à l'aide d'une matrice creuse  $H$  de taille  $n \times m$ , de distributions de degrés  $(\lambda(x), \rho(x))$ , et dont les coefficients non nuls sont dans  $\text{GF}(q)$ . Le décodeur dispose du mot de code  $\mathbf{u} =$

$H^T \mathbf{x}$  et d'un vecteur d'information adjacente  $\mathbf{y}$  de longueur  $n$ , à partir desquels il réalise une estimation au sens du Maximum *a posteriori* de  $\mathbf{x}$  à l'aide d'un algorithme somme-produit que nous décrivons ici. On peut représenter par un graphe  $\mathcal{G}$  les dépendances entre les différentes variables aléatoires du problème. Dans ce graphe, on appelle nœuds variables (NV) les sommets représentant les variables  $X_1 \dots X_n$  et nœuds de parité (NP) les sommets représentant les composantes  $U_1 \dots U_m$  du mot de code. Une arête relie un NV  $i$  à un NP  $j$  si et seulement si  $H_{i,j} \neq 0$ . On note  $\mathcal{N}_p(i)$  l'ensemble des NP connectés à un NV  $i$  et  $\mathcal{N}_v(j)$  l'ensemble des NV connectés à un NP  $j$ .

L'algorithme somme-produit est un algorithme de passage de messages dans  $\mathcal{G}$ . Les messages sont des vecteurs de dimension  $q$  calculés itérativement et initialisés à chaque NV  $i$  avec

$$m_k^{(0)}(i) = \log \frac{P(X_i = 0 | Y_i = y_i)}{P(X_i = k | Y_i = y_i)}. \quad (2)$$

À l'itération  $\ell$ , on calcule les messages des NP  $j$  vers les NV  $i$

$$\mathbf{m}^{(\ell)}(j, i) = \mathcal{A}[\bar{u}_j] \mathcal{F}^{-1} \left( \mathbf{M}^{(\ell)}(j, i) \right) \quad (3)$$

avec

$$\mathbf{M}^{(\ell)}(j, i) = \prod_{i' \in \mathcal{N}_v(j) \setminus i} \mathcal{F} \left( W[\bar{H}_{i',j}] \mathbf{m}^{(\ell-1)}(i', j) \right) \quad (4)$$

avec  $\bar{s}_j = \ominus s_j \otimes H_{i,j}$ ,  $\bar{H}_{i',j} = \ominus H_{i',j} \otimes H_{i,j}$  et  $W[a]$  est une matrice  $q \times q$  telle que  $W[a]_{k,i} = \delta(a \otimes i \ominus k)$ ,  $\forall 0 \leq k, i \leq q-1$ .  $\mathcal{A}[k]$  est une matrice  $q \times q$  qui transforme un message  $\mathbf{m}$  en un message  $\mathbf{l} = \mathcal{A}[k]\mathbf{m}$  avec  $l_j = m_{j \oplus k} - m_k$ .  $\mathcal{F}$  est une transformée de Fourier particulière dont la  $k$ -ème composante a pour expression  $\mathcal{F}_k(\mathbf{m}) = \sum_{j=0}^{q-1} r^{k \otimes j} e^{-m_j} / \sum_{j=0}^{q-1} e^{-m_j}$ . Ensuite, chaque NV  $i$  envoie un message  $\mathbf{m}^{(\ell)}(i, j, y_i)$  aux NP  $j$  auquel il est relié, et calcule un message *a posteriori*  $\tilde{\mathbf{m}}^{(\ell)}(i, y_i)$ . Ces messages sont donnés par :

$$\mathbf{m}^{(\ell)}(i, j) = \sum_{j' \in \mathcal{N}_p(i) \setminus j} \mathbf{m}^{(\ell)}(j', i) + \mathbf{m}^{(0)}(i), \quad (5)$$

$$\tilde{\mathbf{m}}^{(\ell)}(i) = \sum_{j' \in \mathcal{N}_p(i)} \mathbf{m}^{(\ell)}(j', i) + \mathbf{m}^{(0)}(i, y_i). \quad (6)$$

A partir de (6), chaque NV  $i$  calcule une estimée  $x_i$  telle que

$$\hat{x}_i^{(\ell)} = \arg \max_{k \in \text{GF}(q)} \tilde{m}_k^{(\ell)}(i). \quad (7)$$

L'algorithme se termine si  $\mathbf{u} = H^T \hat{\mathbf{x}}^{(\ell)}$  ou si  $l = L_{\max}$ , le nombre maximum d'itérations.

## 4 Algorithme EM et initialisation

Dans le cas où la distribution conditionnelle  $P(Y|X)$  n'est pas parfaitement connue, la matrice  $H$  est tout d'abord dimensionnée en utilisant (1), c'est-à-dire pour le pire cas. Ensuite, si on ne dispose pas de la distribution conditionnelle, on ne peut initialiser l'algorithme somme-produit avec (2). On va donc chercher à estimer conjointement  $\mathbf{x}$  et  $\theta$  à l'aide d'un algorithme EM. Cet algorithme itératif produit à l'itération  $\ell$  des

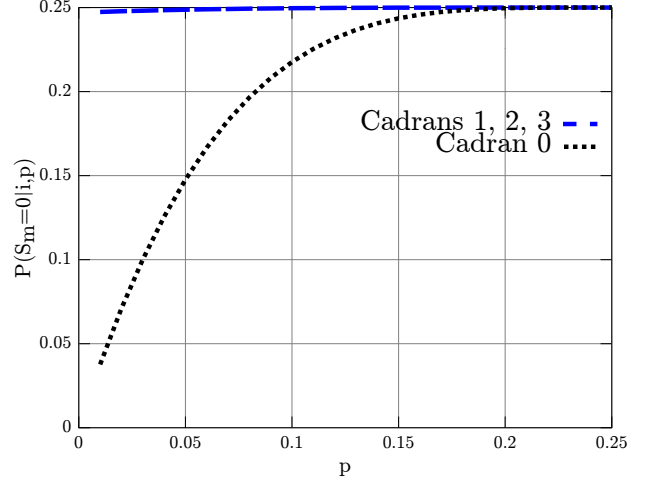


FIGURE 3 –  $P(S_m = 0|i, p)$  en fonction du cadran  $i$  et de  $p$ , pour  $q = 4$ ; les courbes pour  $i = 1, 2, 3$  sont superposées

estimées  $\hat{\theta}^{(\ell)}$  et  $\hat{\mathbf{x}}^{(\ell)}$  de  $\theta$  et  $\mathbf{x}$ . D'après [4],

$$\forall k \in \text{GF}(q), \theta_k^{(\ell+1)} = \frac{\sum_{i=1}^n P_{y_i \ominus k, i}^{(\ell)}}{\sum_{i=1}^n \sum_{k'=0}^{q-1} P_{y_i \ominus k', i}^{(\ell)}} \quad (8)$$

où  $P_{k,i}^{(\ell)} = P(X_i = k | y_i, \mathbf{u}, \theta^{(\ell)})$ . Ces quantités sont fournies par l'algorithme de décodage LDPC décrit dans la partie 3, initialisé avec  $\hat{\theta}^{(\ell-1)}$ .

Une estimée grossière  $\hat{\theta}^{(0)}$  pour initialiser l'algorithme EM est obtenue par le décodeur à l'aide de  $\mathbf{s} = H^T \mathbf{x} - H^T \mathbf{y} = H^T \mathbf{z}$ . On fait l'hypothèse<sup>1</sup> que les variables aléatoires intervenant dans le calcul des composantes  $s_j$  de  $\mathbf{s}$  sont distinctes et indépendantes de celles intervenant dans le calcul de  $s_k$ ,  $j \neq k$ . A partir de cette hypothèse, on effectue une estimation au sens du Maximum de Vraisemblance (MV) de  $\theta$  à partir de  $\mathbf{s}$ . La log-vraisemblance  $L_i(p)$  en fonction du cadran  $i$  et de  $p$  est

$$L_i(p) = \sum_{m=1}^M \log \mathcal{F}_{u_m}^{-1} \left( \prod_{j=1}^{d_c} \mathcal{F}(W[H_{j,m}] \theta^{(i)}(p)) \right), \quad (9)$$

Une estimée  $\hat{\theta}$  de  $\theta$  est obtenue en maximisant  $L_i(p)$  par rapport à  $i$  et à  $p$ . Pour cela, on peut produire une estimée  $\hat{p}_i = \arg \max_p L_i(p)$  par cadran  $i$  et conserver le couple  $(i, \hat{p}_i)$  maximisant  $L_i(\hat{p}_i)$ . Cependant, nous avons observé que cette méthode ne produit des estimées satisfaisantes de  $i$  et de  $p$  que lorsque  $i = 0$  est le vrai cadran.

Ce résultat peut être interprété de la manière suivante. Lorsque le cadran est  $i = 0$ ,  $\mathbf{z}$  contient une majorité de composantes nulles et il en est de même pour  $\mathbf{s}$ . En effet, la probabilité qu'une composante de  $\mathbf{s}$  soit nulle varie fortement avec la probabilité qu'une composante de  $\mathbf{z}$  soit nulle (*i.e.*,  $1 - (1 - q)p$ ),

1. Cette hypothèse est fautive, puisqu'une variable  $X_n$  peut intervenir dans plusieurs sommes, mais comme chaque  $X_n$  intervient seulement dans un petit nombre de sommes, elle semble raisonnable pour obtenir une initialisation grossière de  $\theta$ .

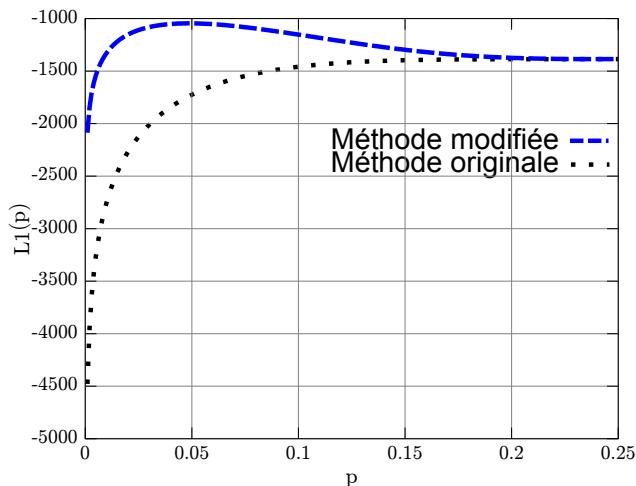


FIGURE 4 –  $L_1(p)$  et  $\tilde{L}_1(p)$  lorsque le vrai cadran est  $i = 1$ , calculées à l’aide d’un vecteur  $\mathbf{s}$  de 1000, pour  $p = 0.05$ ;  $L_1(p)$  ne permet pas d’estimer  $p$

| Initialisation   | $P_e$                | Temps (s) |
|------------------|----------------------|-----------|
| Aléatoire        | $7.2 \times 10^{-3}$ | 47.0      |
| Méthode de [4]   | $7.5 \times 10^{-1}$ | 20.0      |
| Méthode proposée | $< 10^{-5}$          | 9.1       |

TABLE 1 – Comparaison des initialisations possibles

et donc varie fortement avec  $p$ , voir la figure 3. Lorsque le cadran est  $i \neq 0$ ,  $\mathbf{z}$  contient beaucoup d’éléments non nuls. Ces éléments sont combinés aléatoirement via la matrice  $H$  pour donner des composantes de  $\mathbf{s}$  dont les valeurs sont distribuées plus ou moins uniformément sur  $\text{GF}(q)$ , avec une distribution dépendant très peu de  $p$ , voir la figure 3.

Pour résoudre cette difficulté, nous proposons, pour chaque cadran  $i$ , de calculer  $\tilde{\mathbf{s}}^{(i)} = \mathbf{s} \ominus H^T \mathbf{i} = H^T(\mathbf{z} \ominus \mathbf{i})$  où  $\mathbf{i}$  est un vecteur de taille  $n$  contenant uniquement des  $i$ . Lorsque le vrai cadran est  $i$ ,  $\mathbf{z}$  contient une majorité de  $i$  et  $\mathbf{z} - \mathbf{i}$  contient une majorité de 0. Avec cette modification, la probabilité  $P(\check{S}_m^{(i)} = 0 | p, i)$  que la  $i$ -ème composante de  $\mathbf{s}$  soit nulle exprimée en fonction de  $p$  pour le cadran  $i$  aura la même allure que  $P(S_m = 0 | p, i = 0)$ , voir la figure 3. Ainsi,  $p$  et  $i$  peuvent alors être estimés efficacement, en calculant  $\hat{p}_i = \arg \max \tilde{L}_i(p)$ , la log-vraisemblance (9) calculée pour  $\tilde{\mathbf{s}}^{(i)}$ . Puis on conserve le couple  $(i, \hat{p}_i)$  maximisant  $\tilde{L}_i(\hat{p}_i)$ . La figure 4 montre sur un exemple où  $i = 1$  que, contrairement à  $L_1(p)$ ,  $\tilde{L}_1(p)$  permet d’estimer efficacement  $p$ .

## 5 Simulations

Considérons le modèle décrit au paragraphe 2, avec  $q = 4$  et  $\mathbb{P} = [0, 0.08]$ . Quelle que soit la méthode de décodage considérée, 100 vecteurs de 1000 symboles sont générés, 20 itérations de décodeur LDPC et 3 itérations de l’algorithme EM sont réalisées. Pour chaque vecteur, le cadran  $i$  et le paramètre  $p$  sont générés aléatoirement et uniformément. Un code LDPC

de distributions de degrés  $\lambda(x) = x$  et  $\rho(x) = 0.411x + 0.559x^2 + 0.030x^4$  est utilisé, ce qui donne un débit de  $R = 1.6$  bits/symbole.

Pour un premier jeu d’expériences, l’algorithme EM est initialisé aléatoirement. On suppose que le vrai cadran est  $i = 0$ , on tire  $p$  aléatoirement et uniformément dans  $\mathbb{P}$  et on applique l’algorithme EM. S’il ne converge pas, on teste le cadran  $i = 1$  et on effectue la même opération jusqu’à convergence. Dans le second jeu d’expériences, l’algorithme EM est initialisé à l’aide de l’argument du maximum de  $L_i(p)$ . Dans le troisième, il est initialisé avec l’argument du maximum de  $\tilde{L}_i(p)$ .

Le tableau 1 décrit la probabilité d’erreur de décodage de  $\mathbf{x}$  ainsi que le temps nécessaire au décodage pour les trois jeux d’expériences, montrant l’importance d’une initialisation correcte de l’algorithme EM.

## Références

- [1] I. Csiszar. Linear codes for sources and source networks : Error exponents, universal coding. *IEEE Trans. on Inf. Th.*, 28(4) :585–592, 1982.
- [2] L. Cui, S. Wang, S. Cheng, and M. Yeary. Adaptive binary Slepian-Wolf decoding using particle based belief propagation. *IEEE Trans. on Comm.*, 59(9) :2337–2342, 2011.
- [3] D. Declercq and M. Fossorier. Extended minsum algorithm for decoding LDPC codes over  $\text{GF}(q)$ . In *Int. Symp. on Inf. Th.*, pages 464–468. IEEE, 2005.
- [4] E. Dupraz, A. Roumy, and M. Kieffer. Practical coding scheme for universal source coding with side information at the decoder. In *Accepted to the Data Comp. Conf., Snowbird*, March 2013.
- [5] S. Jalali, S. Verdú, and T. Weissman. A universal scheme for Wyner-Ziv coding of discrete sources. *IEEE Trans. on Inf. Th.*, 56(4) :1737–1750, 2010.
- [6] G. Lechner and C. Weidmann. Optimization of binary LDPC codes for the  $q$ -ary symmetric channel with moderate  $q$ . In *Int. Symp. on Turbo Codes and Rel. Topics*, pages 221–224. IEEE, 2008.
- [7] A. Liveris, Z. Xiong, and C. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Comm. Lett.*, 6 :440–442, 2002.
- [8] T. Matsuta, T. Uyematsu, and R. Matsumoto. Universal Slepian-Wolf source codes using Low-Density Parity-Check matrices. In *IEEE Int. Symp. on Inf. Th., Proceedings.*, pages 186–190, June 2010.
- [9] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke. Design of capacity-approaching irregular Low-Density Parity-Check codes. *IEEE Trans. on Inf. Th.*, 47(2) :619–637, 2001.
- [10] D. Varodayan, A. Aaron, and B. Girod. Rate-adaptive codes for distributed source coding. *EURASIP Signal Processing*, 86(11) :3123–3130, 2006.
- [11] Z. Xiong, A.D. Liveris, and S. Cheng. Distributed source coding for sensor networks. *IEEE Sig. Proc. Mag.*, 21(5) :80–94, Sep 2004.