

Performances des codes LDPC

TP TS345

Romain Tajan

1 Performances de l'algorithme de propagation de croyances (BP)

Dans cette section, on étudie Les performances de l'algorithme de propagation de croyances (BP) pour différents codes.

1.1 Performances de BP pour le code (6,3)

Travail 1 *Après avoir extrait la matrice H contenue dans `DEBUG_6_3 alist`, faire un inventaire des propriétés de cette matrice pour le décodage BP*

- *cette matrice est-elle régulière/irrégulière ?*
- *donner les polynômes des degrés*
- *dessiner le graphe de Tanner associé à H .*

Travail 2 *Implémenter l'algorithme de propagation de croyances sur canal BEC. Tracer sur une même courbe, les taux d'effacements binaires pour le code décrit dans le fichier `DEBUG_6_3 alist` pour 1, 2, 3, 4 et 5 itérations. **Commenter votre résultat.** Pour ces tracés, considérer des probabilités d'effacements de $p = 0.1$ à $p = 1$ avec un pas 0.05 (on ajustera ces paramètres au besoin). **Coup de pouce :** se servir du travail précédent afin de justifier vos courbes.*

2 Étude des performances de l'algorithme BP et lien avec l'évolution de densité

Le but de cette section est d'étudier les performances d'ensemble des codes LDPC(n, λ, ρ). Pour le décodage, nous reprendrons le décodeur BP de la section précédente. Il s'agit donc ici de développer les fonctionnalités suivantes :

- écrire un premier algorithme permettant de tirer aléatoirement des codes LDPCs dans l'ensemble LDPC(n, λ, ρ);
- Comparer les performances de décodage des codes tirés aléatoirement

Afin de répondre au premier item, on cherche à générer une matrice H aléatoirement dans l'ensemble $LDPC(n, \lambda(X), \rho(X))$, où $\lambda(X)$ et $\rho(X)$ sont les polynômes des degrés.

Algorithme 1 Un algorithme pour générer des matrices H aléatoirement.

Require: $n \geq 0, m \geq 0, \lambda(X), \rho(X)$

1: Calculer \mathbf{L}, \mathbf{P}

▷ \mathbf{L} est un vecteur de taille n , L_j est le degré du nœud de variable x_j .

▷ \mathbf{P} est un vecteur de taille m , P_i est le degré du nœud de parité c_i .

Ensure: H ▷ Matrice de parité taille $m \times n$ de polynômes $\lambda(X), \rho(X)$

2: $H \leftarrow \text{zeros}(m, n)$

3: **for** $j \in [0, n - 1]$ **do**

4: **for** $i \in [0, L_j - 1]$ **do**

5: $\mathcal{P} \leftarrow \{i \in [0, m - 1] \mid H_{i,j} = 0\}$ ▷ Nœuds de parité n'ayant pas d'arrête avec x_j

6: $\mathcal{P} \leftarrow \mathcal{P} - \{i \in \mathcal{P} \mid \sum_j H_{i,j} = P_i\}$ ▷ Enlever les nœuds de parité de degré P_i

7: $m \leftarrow \min_{i \in \mathcal{P}} \sum_j H_{i,j}$ ▷ Trouver le plus petit degré des nœuds dans \mathcal{P}

8: $\mathcal{P} \leftarrow \{i \in \mathcal{P} \mid \sum_j H_{i,j} = m\}$ ▷ Ne garder que les nœuds de \mathcal{P} de plus petit degré

9: $i \leftarrow \text{Uniform}(\mathcal{P})$ ▷ Choisir au hasard une valeur dans \mathcal{P}

10: $H_{i,j} \leftarrow 1$ ▷ Ajouter une arrête entre x_j et c_i

11: **end for**

12: **end for**

Travail 3 On considère ici l'ensemble de codes LDPC(128, $\lambda = X^2, \rho = X^5$). Pour cette famille de code on aura alors $m = 64$ et $n = 128$. Tracer sur une même courbe 5 courbes de performances sur canal BEC pour 20 valeurs de $\epsilon \in [0.3, 0.5]$. Chaque courbe sera obtenue pour un tirage de matrice H particulier.

Commenter votre résultat au regard du cours.

3 Amélioration de l'algorithme de construction des matrices H

On remarque que la procédure décrite dans l'Algorithme 1 construit H nœud de variable par nœud de variable. On souhaite donc changer la sélection des nœuds de parité (lignes 5 à 9) afin que les cycles générés soient les plus grands possibles. Pour cela on va utiliser un arbre représentant le voisinage de la variable v_i (variable courante). Cet arbre est construit par étage ou chaque étage contient une liste de nœuds de variables et une liste de nœuds de parités. Chaque nœud de variable (respectivement de parité) ne peut apparaître qu'une fois dans l'arbre. Les étages sont construits comme suit :

- L'étage 0 contient v_i et tous les nœuds de parités liés à v_i .
- L'étage l contient les variables non présents dans l'arbres qui sont liés aux nœuds de parité
- L'étage l contient les parités liées aux variables de l'item précédent

On modifie alors l'algorithme 1 comme suit. Si $i = 0$, procéder comme dans 1. Initialiser un arbre tel que décrit plus haut. Pour $i > 0$, il y a deux possibilités :

1. tous les nœuds de variables sont présent dans l'arbre construit. Dans ce cas, créer une arrête entre v_i et un des nœuds de parité du dernier étage ayant le plus petit degré.
2. il existe des nœuds de variables absents l'arbre construit. Dans ce cas, créer une arrête entre v_i et un des nœuds de parité absent de l'arbre ayant le plus petit degré.

Travail 4 Toujours avec l'ensemble de codes LDPC(128, $\lambda = X^2, \rho = X^5$), $m = 64$ et $n = 128$ générer une matrice H avec la nouvelle procédure. Tracer sur la courbe précédente les performances du code LDPC pour cette matrice H .

Commenter votre résultat au regard du cours.

4 Calcul du seuil de décodage BP par évolution de densité

— écrire un algorithme réalisant l'évolution de densité et permettant de calculer p_{BP} , probabilité d'effacement du canal en dessous de laquelle un décodage parfait existe ;

Travail 5 *Toujours avec l'ensemble de codes LDPC(128, $\lambda = X^2$, $\rho = X^5$), $m = 64$ et $n = 128$ calculer le seuil de décodage et le faire figurer sur votre courbe.*

Commenter votre résultat au regard du cours.