# 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, $\lambda$, $\rho$). 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, $\lambda$, $\rho$) ;
— Comparer les performances de décodage des codes tirés aléatoirements

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:** $n \geq 0$, $m \geq 0$, $\lambda(X)$, $\rho(X)$
Calculer **L**, **P**
  ▷ **L** est un vecteur de taille $n$, $L_j$ est le degré du nœud de variable $x_j$.
  ▷ **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 $\lambda(X)$, $\rho(X)$
$H \leftarrow zeros(m, n)$
**for** $j \in [0, n-1]$ **do**
  **for** $i \in [0, L_j - 1]$ **do**
    $\mathcal{P} \leftarrow \{i \in [0, m-1] | H_{i,j} = 0\}$ ▷ Noeuds de parité n'ayant pas d'arrête avec $x_j$
    $\mathcal{P} \leftarrow \mathcal{P} - \left\{i \in \mathcal{P} | \sum_j H_{i,j} = P_i\right\}$ ▷ Enlever les noeuds de parité de degré $P_i$
    $m \leftarrow \min_{i \in \mathcal{P}} \sum_j H_{i,j}$ ▷ Trouver le plus petit degré des noeuds dans $\mathcal{P}$
    $\mathcal{P} \leftarrow \left\{i \in \mathcal{P} | \sum_j H_{i,j} = m\right\}$ ▷ Ne garder que les noeuds de $\mathcal{P}$ de plus petit degré
    $i \leftarrow Uniform(\mathcal{P})$ ▷ Choisir au hasard une valeur dans $\mathcal{P}$
    $H_{i,j} \leftarrow 1$ ▷ Ajouter une arrête entre $x_j$ et $c_i$
  **end for**
**end for**

# 3 Amélioration de l'algorithme de construction des matrices $H$

— écrire un algorithme permettant de calculer la maille du graphe de Tanner d'un code LDPC de matrice de parité H ;
— Adapter l'algorithme de la section précédente afin de construire un graphe de plus grande maille ;

# 4 Calcul du seuil de décodage BP par évolution de densité

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