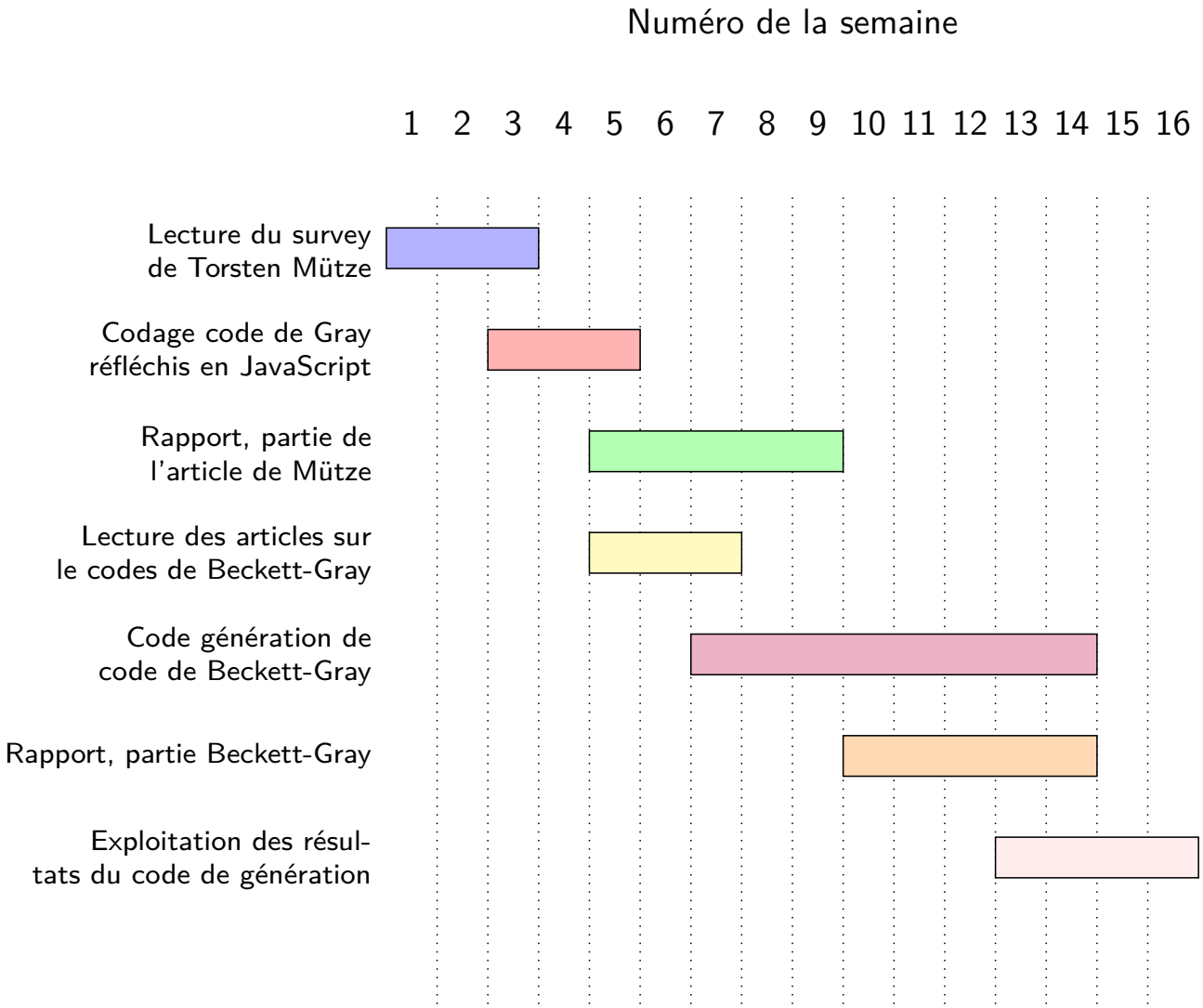
# Exploration des Codes de Gray et des Codes de Beckett-Gray

Lucas Noirot et Mattéo Mennesson

Université de Montpellier

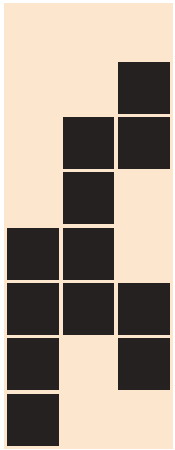
Année universitaire 2023-2024

# Organisation du Travail



# Les codes de Gray

$$000 \rightarrow 001 \rightarrow 011 \rightarrow 010 \rightarrow 110 \rightarrow 111 \rightarrow 101 \rightarrow 100$$



Code de Gray $n = 3$

Minimiser le nombre de bits qui changent.

## Définition

*On dit que deux cycles hamiltonien* $c_1 = (v_1, \cdots, v_n)$ *et* $c_2 = (w_1, \cdots, w_n)$ *sont isomorphes si il existe* $\sigma \in \mathcal{G}_n$ *tel que : Pour tout* $i \in \{1, \cdots, n\}$, $v_i = \sigma(w_i)$ *où* $\sigma(w_i)$ *désigne la permutation des coordonnées de* $w_i$.

## Exemple

*Soit* $\sigma = (12)$

$\sigma(100 \to 101 \to 111 \to 110 \to 010 \to 011 \to 001 \to 000) =$

$010 \to 011 \to 111 \to 110 \to 100 \to 101 \to 001 \to 000$

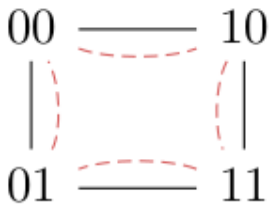
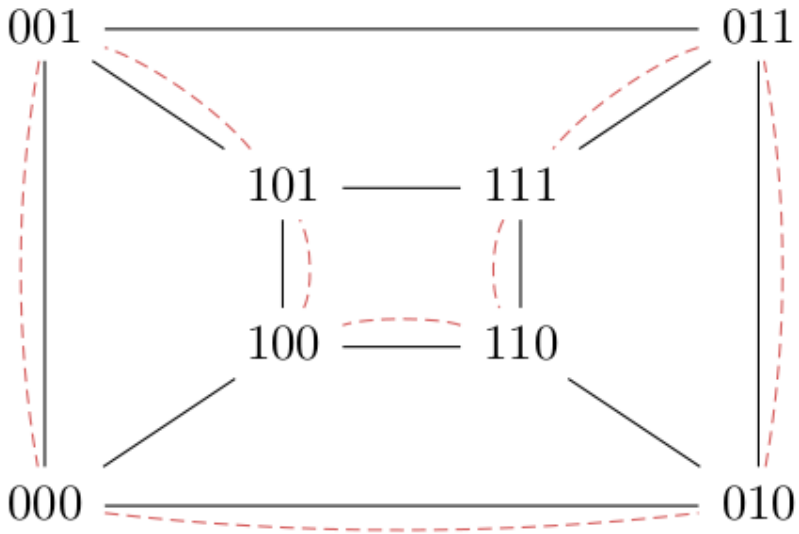
## Définition

*La réversion d'un cycle hamiltonien* $c := v_1, \cdots v_k$ *est* $rev(c) := v_k, \cdots, v_1$.

## Exemple

$rev(000 \to 001 \to 011 \to 010 \to 110 \to 111 \to 101 \to 100) = 100 \to 101 \to 111 \to 110 \to 010 \to 011 \to 001 \to 000$

# Graphe de retournement et hypercube

| Code de Gray associé | Hypercube |
|---|---|
| $0 \to 1$ |   |
| $00 \to 10 \to 11 \to 01$ |   |
| $000 \to 001 \to 101 \to 100 \to 110 \to 111 \to 011 \to 010$ |   |

Cycle hamiltonien en rouge

# Les codes de Beckett-Gray

Un bit à 1 peut passer à 0 seulement si c'est celui qui est passé à 1 il y a le plus longtemps.

Représentation d'un code de Beckett-Gray pour n=5, où le bit à 1 depuis le plus de temps est en rouge

On remarque que les intervalles ne peuvent jamais être emboîtés.

Code de Beckett-Gray partiel pour $n = 4$, correct et incorrect

**Proposition**

*La réversion d'un code de Beckett-Gray est un code de Beckett-Gray.*

**Proposition**

*Un isomorphisme d'un code de Beckett-Gray est un code de Beckett-Gray.*

# Heuristiques

Question ouverte : est-ce qu'il existe un code de Beckett-Gray pour n=9 ?

Pour $n = 5$, il faut 64 476 166 appels de fonctions pour générer tous les codes de Beckett-Gray et un temps d'exécution de 2.839 secondes en moyenne.

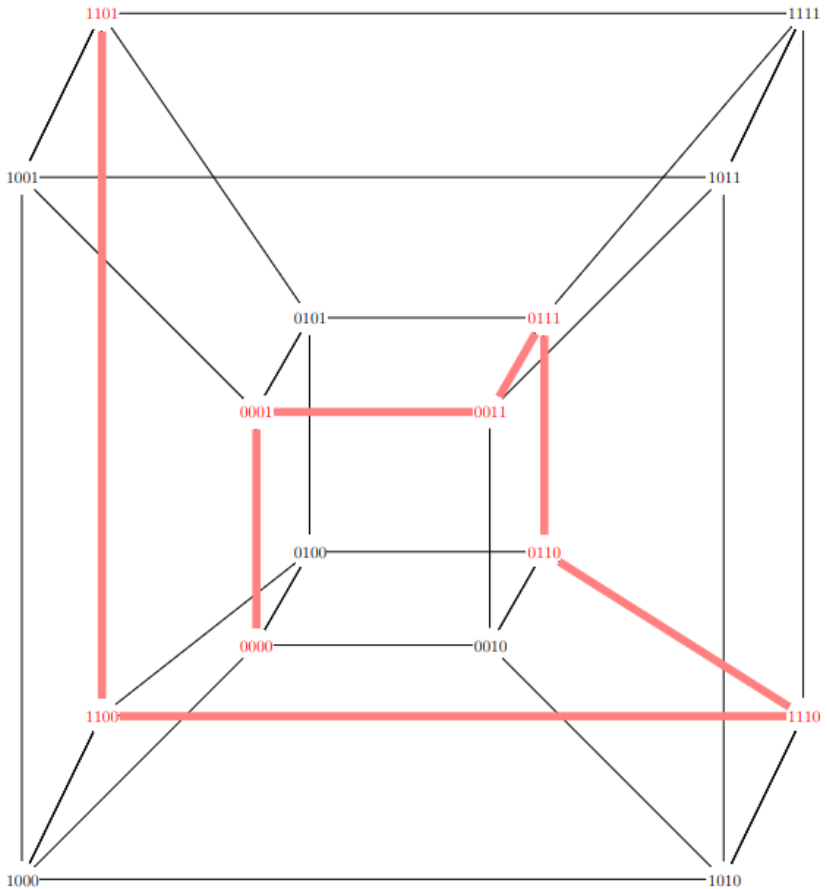
Comment réduire le temps d'exécution ?

$\Rightarrow$ Implémentation d'heuristiques pour réduire la taille de l'arbre de recherche.

# Heuristique : sommets pendants

## Définition

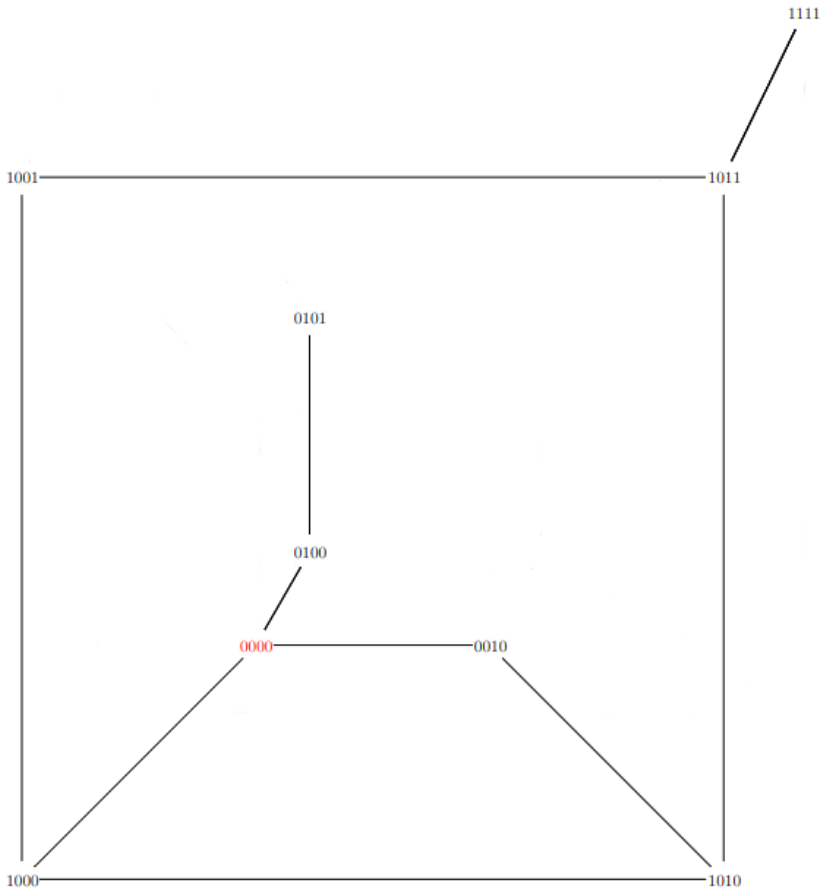
*Un sommet pendant est un sommet ayant un degré de* 1.




Chemin simple pour n=4

# Heuristique : sommets pendants

Il y a deux sommets pendant : pas possible

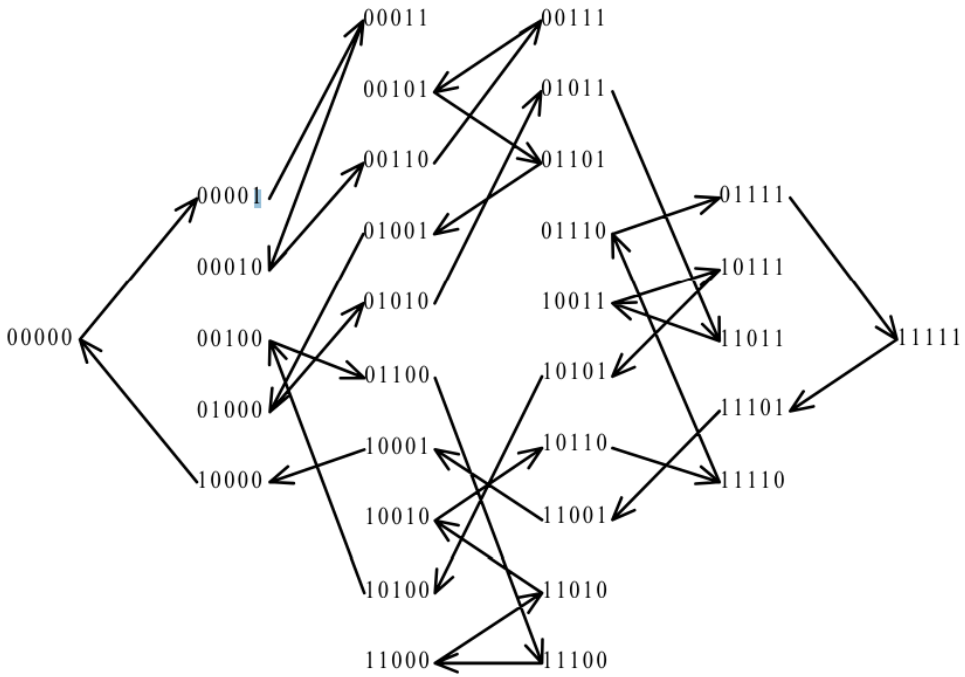



Graphe réduit avec les sommets restants

# Heuristique : propriété eulérienne

## Definition

Un pont d'un graphe connexe est une arête dont la suppression rend le graphe non connexe.




Cycle hamiltonien pour un code de Beckett-Gray n=5

# Heuristique : Nombre d'itération pour changer un 1 en 0

| Nombre de 1 consécutifs | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Nombre d'occurrences | 0 | 720 | 1320 | 4920 | 11040 | 8400 | 2400 | 0 |

**Exemple**

*100 →101 →111→ 110→ 010→ 011→ 001→ 000*

*en rouge 4 bit à 1 consécutifs*

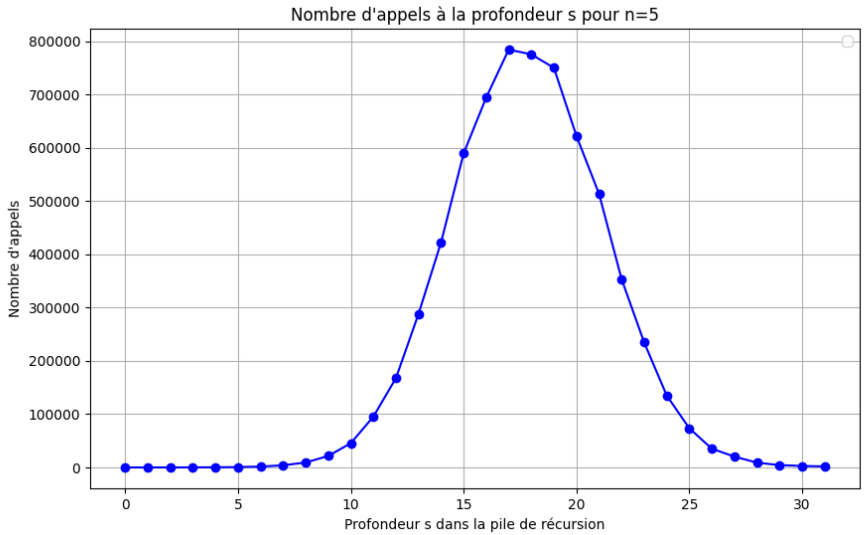
*en vert 4 bit 1 consécutifs*

Problème → on ne peut pas savoir à l'avance le nombre maximum de 1 consécutifs.

# Résultats sur la génération des codes de Beckett-Gray

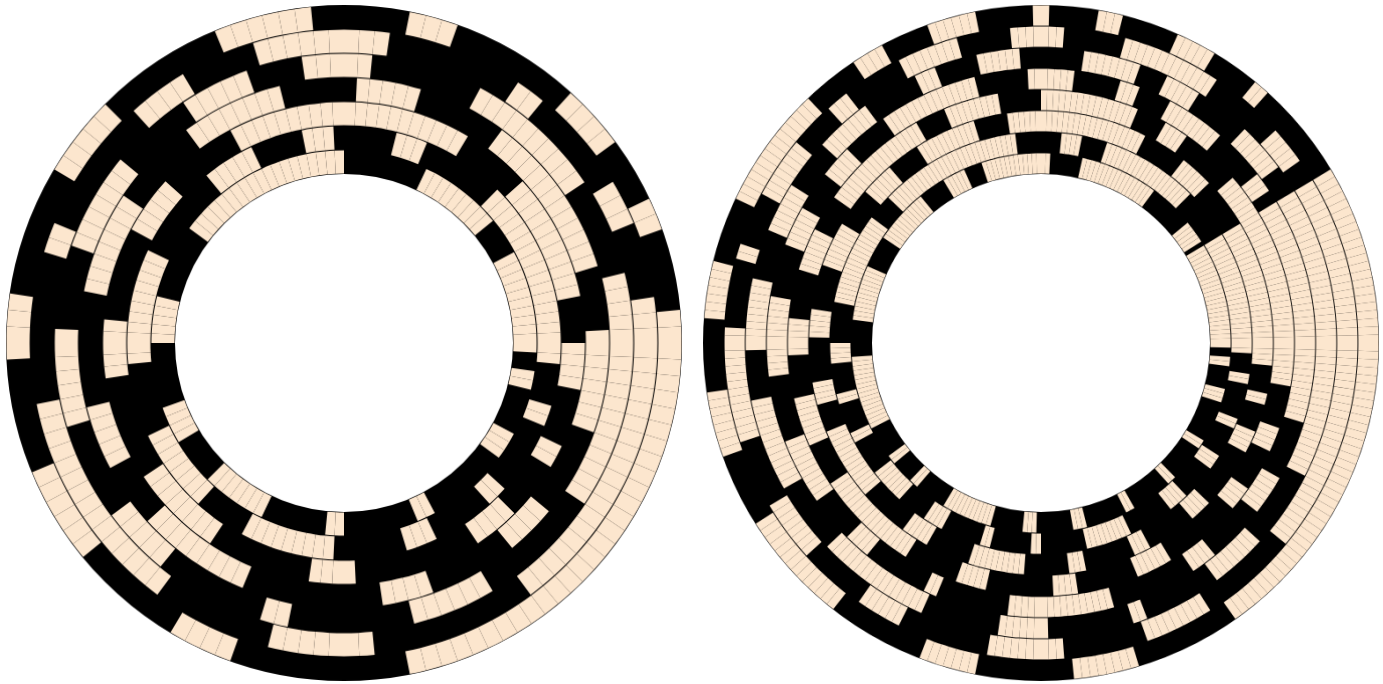
Génération des codes de Beckett Gray n = 5 en 14.5 secondes en Python.

Génération des codes de Beckett Gray n = 5 en 0.428 secondes et 6 654 406 appels de fonction en C++, 6.6 fois plus rapide que sans heuristique.



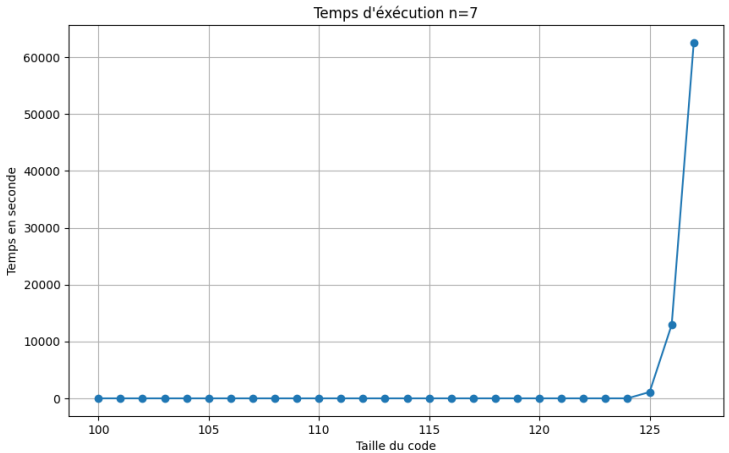

Nombre d'appels de fonction pour chaque profondeur à n = 5

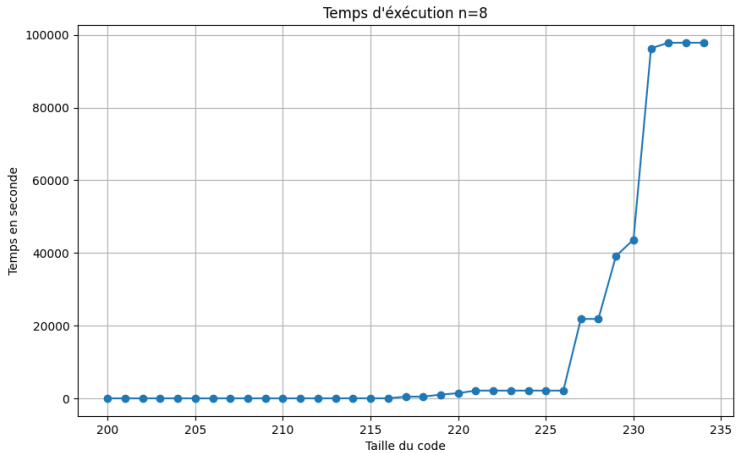
# Résultats sur la génération des codes de Beckett-Gray



Code de Beckett-Gray $n = 7$ et Code de Beckett-Gray partiel pour $n = 8$

# Résultats sur la génération des codes de Beckett-Gray




Temps d'exécution pour générer un code de Beckett-Gray partiel pour $n = 7$ et $n = 8$

# Études statistiques

But : déduire des propriétés plus globales qui nous permettraient d'améliorer les temps de génération de listes exhaustives.

**Proposition**

*On trouve* 1920 *codes de Beckett-Gray cycliques pour* $n = 5$*, on les numérote par ordre de génération :* $c_1, c_2, c_3, \cdots$

**Proposition**

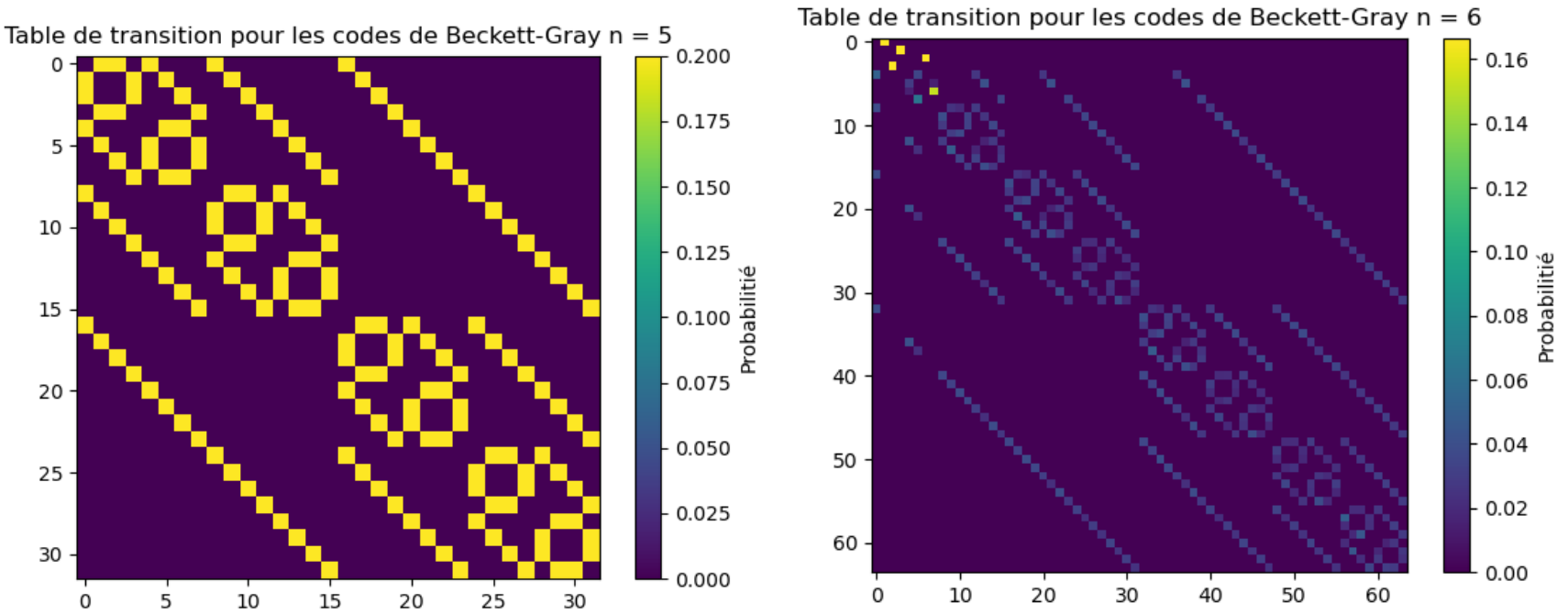
*On dénombre 8 codes de Beckett-Gray non-isomophes.*

# Tables de transition

On va travailler avec la représentation décimales des codes, par exemple :
$100 \to 101 \to 111 \to 110 \to 010 \to 011 \to 001 \to 000$
devient $4 \to 5 \to 7 \to 6 \to 2 \to 3 \to 1 \to 0$

**Définition**

*Une table de transition est une matrice $n \times n$ où $n = 2^k$ est le nombre de valeurs du code de Beckett-Gray sur $k$ bits.*
*Chaque cellule $(i, j)$ de la matrice représente la probabilité que l'entier $i$ soit suivi par l'entier $j$ dans l'ensemble des codes de Beckett-Gray pour $k$ fixé.*

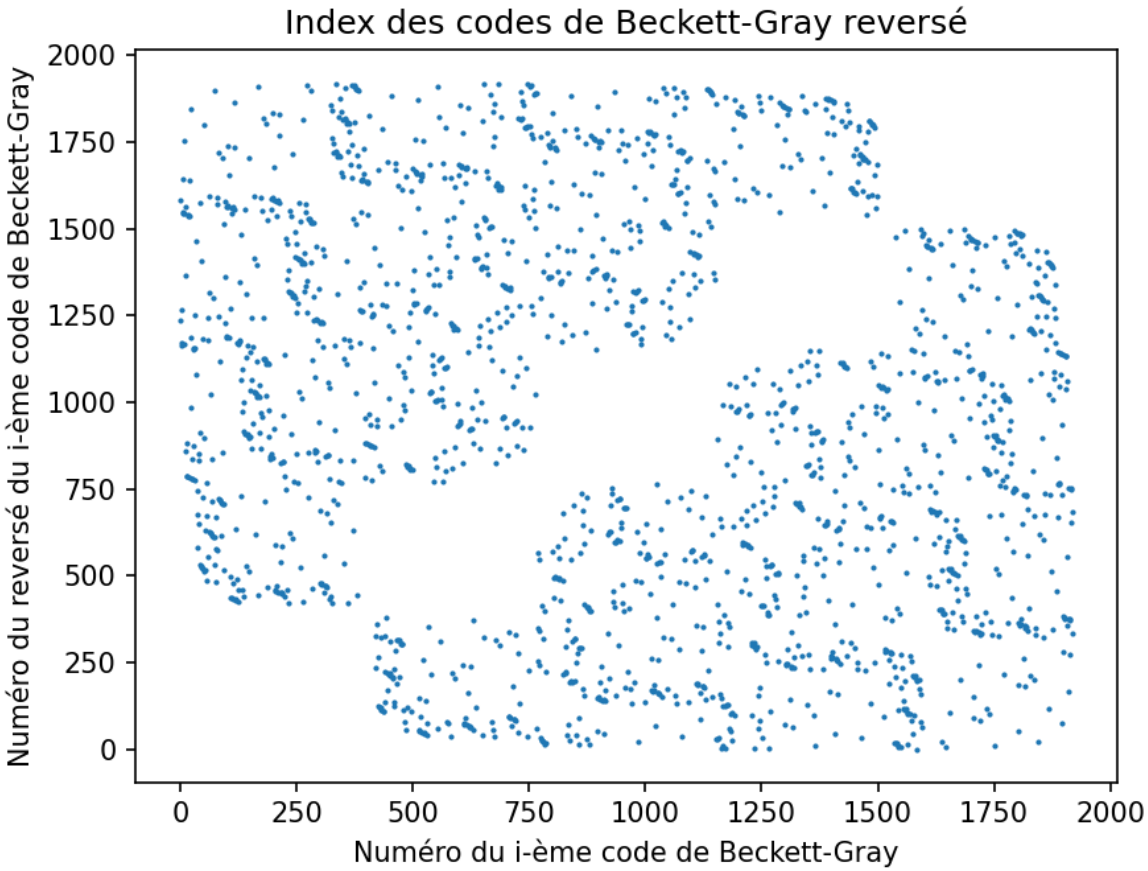



Tables de transition pour $k = 5$ et $k = 6$

# Indices des réversions

## Définition

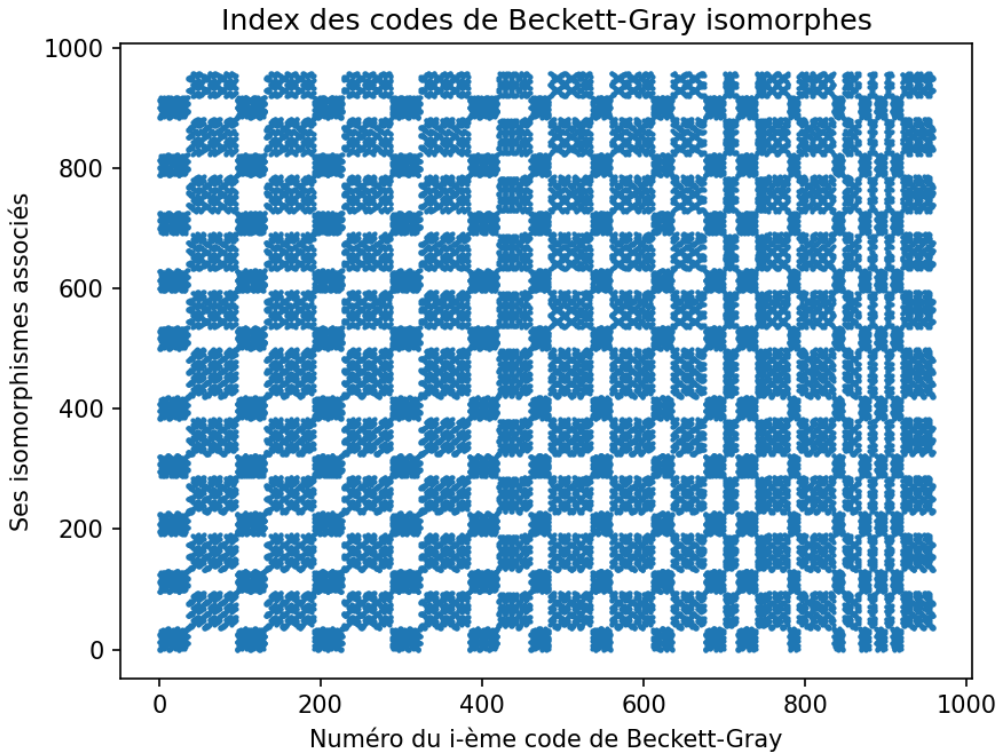
*Soit $x, y \in [\![1, 1920]\!]$. Un point $(x, y)$ est bleu si le code $c_x$ est la réversion du code $c_y$.*



# Études des indices des isomorphismes

## Définition

*Soit $x, y \in [\![1, 960]\!]$. Un point $(x, y)$ est bleu si les codes $c_x$ et $c_y$ sont isomomorphes.*




Code de Beckett-Gray $n = 5$ associés à leurs isomorphismes

# Conclusion et perspectives

$\rightarrow$ S'intéresser à d'autres problèmes ouverts

$\rightarrow$ Ce qu'on a mis en oeuvre

$\rightarrow$ Utiliser plus de puissance de calcul et plus de temps.

$\rightarrow$ Utiliser les statistiques pour développer de nouvelles heuristiques

Merci pour votre attention