Chaînes de Markov, Simulations & Mélange d'un jeu de cartes :

Cyprien MANGEOT - Fantine KOLLAR

M1 IMSD 2021-2022

Table des matières

1 Introduction

Combien de fois doit-on mélanger un jeu de cartes avant que l'on puisse considérer les cartes comme étant dans un ordre aléatoire ?
La réponse à cette question est loin d'être évidente, elle dépend d'énormément de facteurs. Il s'agit en effet d'une étude de processus aléatoire et nous allons, au cours de cette étude, formaliser plusieurs choses, notamment :

- Détailler la manière dont le paquet sera mélangé.
- Préciser le nombre de cartes dans le paquet.
- Définir clairement ce que l'on entend par ordre aléatoire, ou presque aléatoire.

Pour répondre à cette question, nous allons considérer deux méthodes de mélange très différentes et les analyser.
Le premier mélange auquel nous allons nous intéresser est le mélange "Top-to-Random". Nous verrons que ce processus met en lumière une Chaîne de Markov irréductible et apériodique, nous allons donc utiliser de nombreux résultats relatifs aux Chaînes de Markov.
Le deuxième mélange auquel nous allons nous intéresser est l'un des modèles de battage les plus naturels : le mélange à l'américaine. Il porte également le nom de GSR car il a été étudié par Gilbert et Shannon en 1955, puis repris par Reeds en 1981. Pour étudier ce processus aléatoire, nous allons introduire le mélange inverse associé au Riffle shuffle (nom anglais).

On aura recours à des concepts très utiles et nécessaires, notamment le critère uniforme fort et la distance en variation totale. Cette étude va reposer sur deux problèmes combinatoires très connus : le problème du collectionneur de coupons et le paradoxe des anniversaires.

Ainsi, dans une première partie, nous allons nous pencher sur ces deux problèmes. Nous ferons ensuite une analyse complète du mélange Top-to-Random pour enfin s'intéresser au mélange à l'américaine.
On constatera que le premier mélange, bien qu'intéressant à étudier mathématiquement, n'est pas très efficace dans la pratique. Au contraire, le Riffle Shuffle est un mélange à la fois naturel et bien plus efficace.

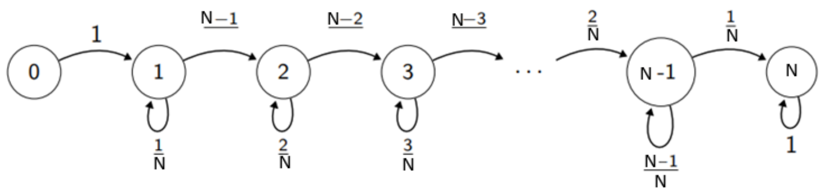
2 Préliminaires

2.1 Le collectionneur de coupons

Un collectionneur veut posséder toute une collection de coupons. Il y a exactement N coupons différents à trouver. On les numérote de 1 à N. $\forall k \in [1, N]$, la probabilité d'obtenir le k-ème coupon vaut $\frac{1}{N}$. Il y a donc équiprobabilité. Chaque achat est indépendant des autres.

On se demande quel est le nombre moyen de coupons qu'il faut acheter avant d'avoir la collection complète ? Pour répondre à cette question, nous allons étudier le processus aléatoire associé.

On note X_n l'état de la collection à l'instant n, à cet instant, le collectionneur possède donc n vignettes distinctes de la collection. On a $X_n \in \mathbb{N}$ et X=$(X_n)_{n\geq 0}$ est une chaîne de Markov qui part de $X_0 = \emptyset$.
Chaque état de la chaîne correspond au nombre de coupons distincts que le collectionneur possède.
Ci-dessous le graphe associé à cette chaîne de Markov.



Proposition 1. *Supposons que l'on possède déjà k coupons différents. Alors, la probabilité d'obtenir à l'achat suivant un coupon en double (un des k coupons que l'on possède déjà) vaut $\frac{k}{N}$.*
Ainsi, la probabilité de devoir recourir à exactement s achats avant d'obtenir un coupon distinct des k coupons déjà détenus vaut $(\frac{k}{N})^{s-1}(1 - \frac{k}{N})$.

Démonstration. Notons A l'évènement "Devoir recourir à exactement s achats avant d'obtenir un coupon distinct des k déjà rencontrés". On cherche $\mathbb{P}(A)$. Soit A_i l'évènement "Trouver un coupon distinct des k coupons déjà rencontrés au i-ème achat". Alors,

$$\mathbb{P}(A) = \mathbb{P}((\cap_{i=1}^{s-1}\overline{A_i}) \bigcap A_s) = (\prod_{i=1}^{s-1} \mathbb{P}(\overline{A_i})) \times \mathbb{P}(A_s)$$

En effet, on partitionne A, puis on utilise le fait que les A_i sont indépendants. On sait d'après l'énoncé que $\forall i \in \{1, ..., s-1\}$, $\mathbb{P}(\overline{A_i}) = \frac{k}{N}$.

On obtient donc,

$$\mathbb{P}(A) = (\frac{k}{N})^{s-1}(1 - \frac{k}{N})$$

$\square$

Nous allons utiliser les notations suivantes :

- On note τ le nombre total de coupons dans la collection lorsque le collectionneur obtient tous les N coupons pour la première fois.
- Soit $i \in \{0, ..., N\}$. On note τ_i le nombre de coupons présents dans la collection lorsque la vignette i est obtenue pour la première fois. (Trivialement $\tau_0 = 0$)

Remarque 1. *τ et les τ_i (pour $i \in \{0, ..., N\}$) sont des temps d'arrêt dans ce cas précis.*

Proposition 2. *Ainsi, le nombre moyen d'achats nécessaires pour obtenir les N coupons différents de la collection vaut :*

$$\mathbb{E}(\tau) = N \sum_{k=1}^{N} \frac{1}{k} \simeq N\, ln\, N$$

Démonstration. On décompose τ_n comme suit :

$$\tau_n = \tau_1 + (\tau_2 - \tau_1) + ... + (\tau_n - \tau_{n-1})$$

On se fixe un entier i $\geqslant 0$. La probabilité de succès pour la variable aléatoire $\tau_i - \tau_{i-1}$ (entre deux temps d'arrêts) est $\frac{N-i+1}{N}$ (car c'est la probabilité d'obtenir un i-ème coupon distinct des $i-1$ coupons déjà détenus).
Or, les $\tau_i - \tau_{i-1}$ suivent une loi géométrique de paramètre $\frac{N-i+1}{N}$.
Donc $\forall i \in \{0, ..., N\}$, $\mathbb{E}(\tau_i - \tau_{i-1}) = \frac{N}{N-i+1}$.
Par conséquent, on obtient :

$$\mathbb{E}(\tau) = \sum_{i=1}^{N} \mathbb{E}(\tau_i - \tau_{i-1}) = \sum_{i=1}^{N} \frac{N}{N-i+1}$$

. On fait le changement de variable $k = N - i + 1$ et on obtient :

$$\mathbb{E}(\tau) = N \sum_{k=1}^{N} \frac{1}{k} \simeq N\, ln\, N$$

$\square$

Nous allons maintenant étudier une proposition qui nous sera très utile par la suite,

Proposition 3. *Pour tout $c > 0$,*

$$\mathbb{P}(\tau > \lceil N \ln N + cN \rceil) \leqslant \mathrm{e}^{-c}$$

Démonstration. On rappelle que τ est le nombre total de coupons dans la collection lorsque le collectionneur obtient tous les N coupons pour la première fois. Notons $m := \lceil N \ln N + cN \rceil$.
$\mathbb{P}(\tau > m)$ représente la probabilité que la collection soit complète après m achats. C'est donc la probabilité que $\forall k \in \{1, ..., N\}$, le k-ème coupon soit obtenu avant qu'on atteigne les m achats.
Autrement dit, si on pose l'évènement : A_i : "le i-ème coupon n'a pas été tiré lors des $\lceil N \ln N + cN \rceil$ premiers achats". On obtient que

$$\mathbb{P}(\tau > \lceil N \ln N + cN \rceil) = \mathbb{P}(\bigcup_{i=1}^{N} A_i) \leqslant \sum_{i=1}^{N} \mathbb{P}(A_i)$$

On cherche la valeur de $\mathbb{P}(A_i)$. Soit $\forall k \in \{1, ..., N\}$.
La probabilité que le k-ème coupon ait été obtenu lors des m premiers achats vaut $\frac{1}{N}$.
Donc la probabilité que le k-ème coupon n'ait pas été obtenu lors des m premiers achats vaut $(1 - \frac{1}{N})$.
Ainsi, $\mathbb{P}(A_i) = (1 - \frac{1}{N})^m$
On a donc :

$$\sum_{i=1}^{N} \mathbb{P}(A_i) = \sum_{i=1}^{N} (1 - \frac{1}{N})^m = N\, (1 - \frac{1}{N})^m$$

Soit la fonction $f : x \mapsto (1 - \frac{1}{x})^x$. Cette fonction est croissante sur $]1; +\infty[$.
De plus, on sait que $\lim\limits_{x \to +\infty} f(x) = \frac{1}{e}$.
Il en découle que $\forall n \geq 1$, $(1 - \frac{1}{n})^n < \frac{1}{e}$. Par conséquent,

$$(1 - \frac{1}{N})^m = (1 - \frac{1}{N})^{N \times \frac{m}{N}} \leqslant \mathrm{e}^{-\frac{m}{N}}$$

Ainsi,

$$\sum_{i=1}^{N} \mathbb{P}(A_i) \leqslant N\, \mathrm{e}^{-\frac{m}{N}}$$

Et comme

$$N\, \mathrm{e}^{-\frac{m}{N}} = N\, \mathrm{e}^{-\frac{\lceil N \ln N + cN \rceil}{N}} = N\, \mathrm{e}^{\lceil -\ln N + c \rceil} = e^{-c}$$

On obtient comme demandé que

$$\mathbb{P}(\tau > \lceil N \ln N + cN \rceil) \leqslant \mathrm{e}^{-c}$$

□

2.2 Le paradoxe des anniversaires

On considère une assemblée de n personnes. On se demande quelle est la probabilité que toutes les dates d'anniversaire soient différentes (si $n \geq 365$, $p(n) = 0$ forcément).

Proposition 4. *Soit $p(n)$ la probabilité que les n personnes de l'assemblée aient une date d'anniversaire distincte. On a :*

$$p(n) = \prod_{k=1}^{n-1} (1 - \frac{k}{365})$$

Démonstration. Considérons une assemblée de n personnes. Cherchons la probabilité $p(n)$. Numérotons les n personnes de l'assemblée de 1 à n.
$p(n) = \prod_{k=1}^{n} \mathbb{P}(A_k)$ où A_k est l'évènement "La k-ème personne de l'assemblée a une date d'anniversaire différente de celle des $(k-1)$-ème premières personnes".
Il nous reste donc à déterminer les $\mathbb{P}(A_k)$. En effet, $\mathbb{P}(A_1) = \frac{365}{365}$ car on a 365 dates possibles et aucune n'est déjà prise.
Puis $\mathbb{P}(A_2) = \frac{365-1}{365}$ car on a 365 dates possibles mais l'une d'elle est déjà prise par la première personne de l'assemblée.
Et ainsi de suite jusqu'à $\mathbb{P}(A_n) = \frac{365-(n-1)}{365}$
Il suit que $p(n) = \prod_{k=1}^{n-1} \frac{365-k}{365} = \prod_{k=1}^{n-1} (1 - \frac{k}{365}) = \prod_{k=1}^{n-1} (1 - \frac{k}{365})$. □

Remarque 2. *On décide de tracer les valeurs que prend cette probabilité $p(n)$ en fonction de n. Pour ce faire, nous avons codé une fonction calculant la probabilité $p(n)$ en fonction de n. Pour $n = 140$, nous avons calculé $p(k), \forall k \in [1, 140]$.Puis on trace tous les points et on obtient le graphique suivant :*

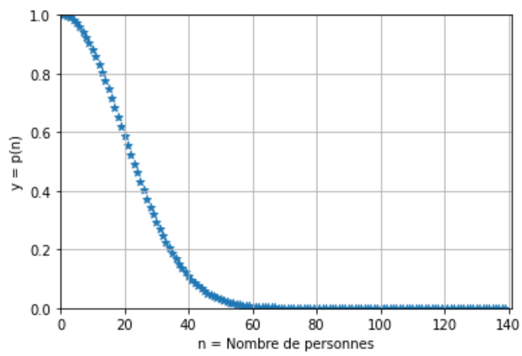


FIGURE 1 – Graphique de la probabilité $p(n)$

(Vous trouverez le code ayant permis de réaliser ce graphique en annexe.)
On remarque que $p(n)$ est inférieure ou égale à $1/2$ lorsque $n \geq 23$. Elle devient inférieure à $0,09$ lorsque $n \geq 42$. Puis $p(n) = 0$ lorsque $n \geq 365$.

Proposition 5. *De manière analogue, si n boules sont placées indépendamment et aléatoirement dans K boîtes, alors la probabilité qu'aucune boîte ne contienne plus d'une boule vaut :*

$$p(n,K) = \prod_{i=1}^{n-1}(1 - \frac{i}{K})$$

Démonstration. Considérons nos n boules, et nos K boîtes. Cherchons la probabilité $p(n,K)$. Numérotons les n boules de 1 à n.
$p(n,K) = \prod_{i=1}^{n} \mathbb{P}(A_i)$ où A_i est l'évènement "La i-ème boule ne se trouve pas dans une boîte contenant déjà une boule".

Il nous reste donc à déterminer les $\mathbb{P}(A_i)$.
En effet, $\mathbb{P}(A_1) = \frac{K}{K}$ car on a K boîtes possibles pour la première boule et aucune n'est déjà prise. Puis $\mathbb{P}(A_2) = \frac{K-1}{K}$ car on a K boîtes possibles mais l'une d'elles est déjà prise par la première boule.

Et ainsi de suite jusqu'à $\mathbb{P}(A_n) = \frac{K-(n-1)}{K}$
Il suit que

$$p(n,K) = \prod_{i=1}^{n-1} \frac{K-i}{K} = \prod_{i=1}^{n-1}(1 - \frac{i}{K}) = \prod_{i=1}^{n-1}(1 - \frac{i}{K})$$

□

3 Mélange Top-to-Random

3.1 Etude du processus aléatoire

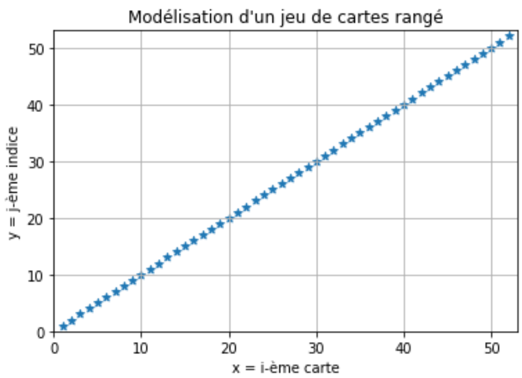
Dans cette partie, nous allons faire l'étude du mélange Top-to-Random.
On se donne un jeu de N cartes, ce mélange consiste à prendre la carte du haut du paquet et à l'insérer au hasard dans le reste du paquet.
A chaque mélange, on a N places pour notre carte. Ces places étant équiprobables de probabilité $\frac{1}{N}$. Le modèle Top-to-Random fait apparaître une chaîne de Markov dont l'espace d'état est Σ_N (ensemble des permutations à N éléments) et dont les probabilités de transition sont définies comme suit : Si on désigne par P la matrice de transition de la chaîne, on a que $\forall \sigma, \gamma \in \Sigma_N$:

$$P(\sigma,\gamma) = \begin{cases} \frac{1}{N} & \text{si } \gamma = \sigma \circ c_i^* \text{pour i} \in \{1, ..., N\} \\ 0 & \text{sinon.} \end{cases}$$

*ou c_i est le cycle (1 ... i)
Il s'agit ainsi d'une chaîne de Markov homogène. Soulignons également que la probabilité de transition est non nulle si et seulement si γ préserve l'ordre

des $N - i$ cartes du bas.

Dans la suite de ce document, nous allons afficher l'état d'un jeu de cartes à l'aide de graphiques. Voici l'état d'un jeu de cartes rangé.



Si une étoile se trouve à la position (i, j), c'est que la carte i se trouve à la position j dans le paquet. Il est donc logique que pour un jeu de cartes rangé, la modélisation corresponde à une droite.
(Vous trouverez le code ayant permis de produire ce graphique en annexe.)

Nous avons codé une fonction permettant de simuler le Mélange Top-to-Random. Vous trouverez en annexe le code.

Nous pouvons donc afficher l'état d'un paquet rangé une fois qu'on lui a appliqué un coup de mélange top.

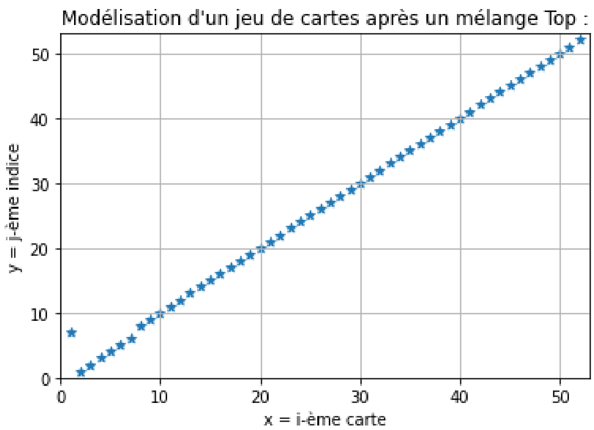


FIGURE 2 – Modélisation d'un mélange Top, on voit que la première carte se trouve désormais à la 7e position dans le paquet

Vous trouverez le code ayant permis de produire ce graphique en annexe.
Les graphes représentant l'état du paquet après 5, puis 50 mélanges Top successifs se trouvent en annexe.
Le code associé se trouve également en annexe.

Proposition 6. *Pour toute permutation π, il existe exactement N permutations σ telles que :*

$$P(\sigma, \pi) = \frac{1}{N}$$

Exemple 1. *Nous allons faire un exemple simple pour concrétiser cette modélisation. On se place dans Σ_3, analysons les probabilités de transition, partant de l'identité de Σ_3, d'arriver dans les états possibles avec le mélange Top.*

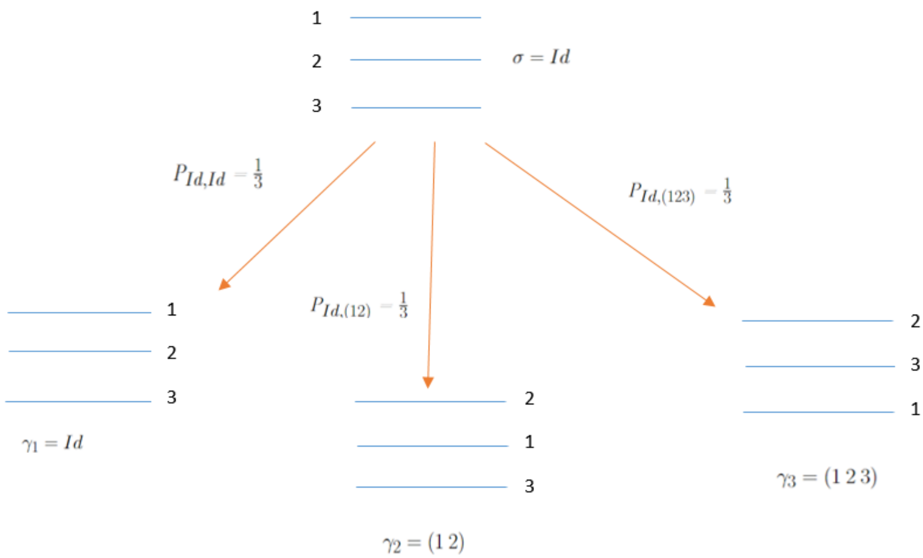


FIGURE 3 – Probabilités de transition du mélange Top pour 3 cartes.

On voit bien qu'il y a N (=3) permutations possibles et chacune de probabilité $\frac{1}{3}$

3.2 Convergence de la chaîne

Nous allons étudier la convergence de la chaîne de Markov associée au modèle de mélange Top-to-Random. Dans toute cette partie, on va noter μ_n la distribution de probabilité de la chaîne à l'instant n.
Nous allons d'abord définir la loi uniforme sur Σ_N.

Définition 1. *La distribution uniforme U est définie sur Σ_N comme suit :*
$\forall \sigma \in \Sigma_N$,

$$U(\sigma) = \frac{1}{N!}$$

Nous allons maintenant donner une définition très importante de cette partie, qui nous sera très utile pour toute la suite.

Définition 2. *Soient μ et ν deux distributions de probabilité sur Σ_N, on appelle distance de variation totale (d_V) la quantité suivante :*

$$\begin{aligned} d_V(\mu,\nu) &= max\,\{|\mu(\sigma)-\nu(\sigma)| : \sigma \in \Sigma_n\} \\ &= \frac{1}{2}\sum_{\pi\in\Sigma} |\mu(\pi)-\nu(\pi)| \end{aligned}$$

Théorème 1. *La chaîne de Markov associée au mélange Top-to-Random $(X_t)_{t\in\mathbb{N}}$ est irréductible et apériodique sur un ensemble fini. Elle possède de plus une unique probabilité invariante , qui est la mesure uniforme U vers laquelle elle converge en loi sur Σ_N.*

Nous allons démontrer point par point ce théorème.

Démonstration. Nous allons montrer les trois points clés du théorème. dans toute cette démonstration, on se donne σ un élément de Σ_N

- On montre d'abord que la chaîne est irréductible, on doit donc montrer que chaque état est accessible à partir de n'importe quel état. Voici un programme qui permet de se rendre compte que cela fonctionne :
  - On montre que chaque état est accessible depuis l'identité à travers ce code
  - Le principe est le même dans l'autre sens : on part d'une certaine permutation et on accède à l'identité après une répétition de mélanges Top.
- Montrons maintenant que la chaîne est apériodique. On utilise pour cela la Définition 6. La chaîne est irréductible, comme montré au point précédent, il y a donc une seule classe de communication. Celle-ci est de période 1, en effet : $p_{\sigma\sigma} > 0$ (on applique l'identité à l'état σ). La chaîne est donc bien apériodique.
- Montrons désormais que la chaîne admet pour unique probabilité invariante la loi uniforme U. En effet, on a :

  $$(UP)(\sigma) = \sum_{\tau\in\Sigma_N} U(\sigma)P(\tau,\sigma) = \frac{1}{N!}\sum_{\tau\in\Sigma_N} P(\tau,\sigma)$$

  Or, l'ensemble $\{\ \tau \in \Sigma_N : \mathrm{P}(\tau\ ,\ \sigma) \neq 0\}$ est de taille N et la probabilité d'obtenir chacun des éléments de cet ensemble est de $\frac{1}{N}$, ainsi :

  $$(UP)(\sigma) = \frac{1}{N!}(N\times\frac{1}{N}) = \frac{1}{N!} = U(\sigma)$$

On obtient donc que la mesure uniforme est stationnaire pour P. On vient de montrer que la chaîne était irréductible, apériodique et possède une loi stationnaire sur un ensemble fini, elle est donc récurrente positive. On peut donc appliquer le théorème ergodique ; la chaîne converge donc vers cette unique probabilité stationnaire.

□

Le théorème étant démontré, on a que la convergence de $(\mathrm{X}_n)_{n\in\mathbb{N}}$ vers sa mesure stationnaire U implique que :

$$\lim_{n\to\infty} d_V(\mu_n, U) = 0$$

Dans les prochains paragraphes, nous allons déterminer la vitesse de convergence de la chaîne vers sa mesure stationnaire.

3.3 Critère d'arrêt uniforme fort

3.3.1 Temps de remontée d'une carte

Dans toute cette partie, on s'intéresse au temps de remontée de la dernière carte du paquet rangé (associé à permutation identité). Dorénavant, on notera cette carte D.
On appelle T_i la variable qui compte le nombre de mélanges effectués avant que i cartes se trouvent sous D pour la première fois. On décompose donc T_N (D est remontée au sommet du paquet pour la première fois) comme suit :

$$T_N = T_1 + (T_2 - T_1) + ... + (T_{N-1} - T_N)$$

Théorème 2 (Temps d'arrêt uniforme fort pour le mélange Top)**.** *Soit T le temps auquel D est insérée pour la première fois dans le paquet. T est un critère d'arrêt uniforme fort pour le mélange Top-to-Random. On a donc :*

$$T = T_N + 1$$

On va démontrer ce théorème, et pour ce faire nous allons utiliser les deux résultats suivants :

Proposition 7. *On suppose que les k premières cartes du paquet sont ordonnées suivant une permutation uniforme. On insère la carte $(k+1)$ uniformément à l'une des $k+1$ positions possibles.*
Alors l'ordre des $(k+1)$ cartes est uniforme dans Σ_{k+1}.

Proposition 8. *Soit X une variable aléatoire suivant la loi uniforme sur un ensemble E fini. Soit A une partie non vide de E. Alors, X, conditionné à être dans A, reste de loi uniforme sur A.*

Démonstration. On va démontrer le Théorème 2, on doit donc vérifier que T est bien un critère d'arrêt uniforme fort pour le mélange Top-to-Random. On rappelle que l'on doit donc montrer que si le processus est interrompu après k étapes, alors les permutations associées à l'ordre des cartes ont une distribution uniforme. Procédons par récurrence :
On pose l'hypothèse $(\mathrm{H}_k)_{k\geqslant 1}$: A l'instant T_k, les k cartes sous D ont une distribution uniforme.
Initialisation : on prend k = 1, (H_1) : une seule carte sous D, donc une seule permutation, de manière immédiate, l'hypothèse est vérifiée.
On suppose désormais que (H_k) est vrai pour $1 \leq k \leq N-1$, montrons que dès lors, (H_{k+1}) est également vérifiée.
On sait qu'à l'instant T_k, les k cartes sous D sont ordonnées selon une permutation uniforme. A l'instant T_{k+1}, on insère une $(k+1)$-ème carte sous D et d'après les Propositions 7 :
Cette carte s'insère uniformément sur l'une des $(k+1)$ positions possibles.
Puis, d'après la proposition 8 :
La nouvelle permutation associée à l'ordre des cartes est uniforme sur les $(k+1)!$ permutations possibles.
Ainsi, T est bien un critère d'arrêt uniforme fort pour le mélange Top. □

Remarque 3. *Nous avons codé une fonction permettant de simuler le temps d'arrêt uniforme fort du mélange Top. Vous trouverez cette fonction en annexe.*

La proposition suivante permet d'établir une idée de l'écart entre μ_n et la distribution uniforme U sur Σ_N en fonction de T :

Proposition 9. *Pour tout entier $n \in \mathbb{N}$, on a :*

$$d_V(\mu_n, U) \leqslant \mathbb{P}(T > n)$$

Démonstration. Si X est une variable aléatoire prenant ses valeurs dans Σ_N, alors si on note Q sa distribution de probabilité, on a que, pour S une partie de Σ_N : Q (S) $= \mathbb{P}(X \in S)$: c'est la probabilité que X prenne une certaine valeur dans S.
On se place maintenant dans le cas que l'on souhaite étudier, c'est-à-dire le cas de la distribution uniforme, on prend donc $Q = U$:

$$U(S) = \mathbb{P}(X \in S) = \frac{|S|}{N!}$$

On va maintenant étudier la quantité μ_n

$$\begin{aligned}
\mu_n(S) = \mathbb{P}(X_n \in S) &= \mathbb{P}(X_n \in S, T \leqslant n) + \mathbb{P}(X_n \in S, T > n) \\
&= \sum_{i \leqslant n} \mathbb{P}(X_n \in S, T = i) + \mathbb{P}(X_n \in S, T > n) \\
&= \sum_{i \leqslant n} U(S)\mathbb{P}(T = i) + \mathbb{P}(X_n \in S \mid T > n) \times \mathbb{P}(T > n) \\
&= U(S)(1 - \mathbb{P}(T > n)) + \mathbb{P}(X_n \in S \mid T > n)\mathbb{P}(T > n) \\
&= U(S) + (\mathbb{P}(X_n \in S \mid T > n) - U(S))\mathbb{P}(T > n)
\end{aligned}$$

Le passage de la deuxième à la troisième ligne s'obtient grâce à la Définition 13. Or la quantité $\mathbb{P}(X_n \in$ S $\mid$ T $>$ n) - U(S) est une différence entre deux probabilités, donc : $\mathbb{P}(X_n \in S \mid T > n) - U(S) \leqslant 1$
Il en résulte que :

$$\begin{aligned}
&\mid \mu_n(S) - U(S) \mid \leqslant \mathbb{P}(T > n) \\
\implies &d_V(\mu_n, U) \leqslant \mathbb{P}(T > n)
\end{aligned}$$

□

On affiche l'état du paquet une fois que le Temps d'arrêt uniforme fort est atteint. Le code ayant permis de produire ce graphique se trouve en annexe.

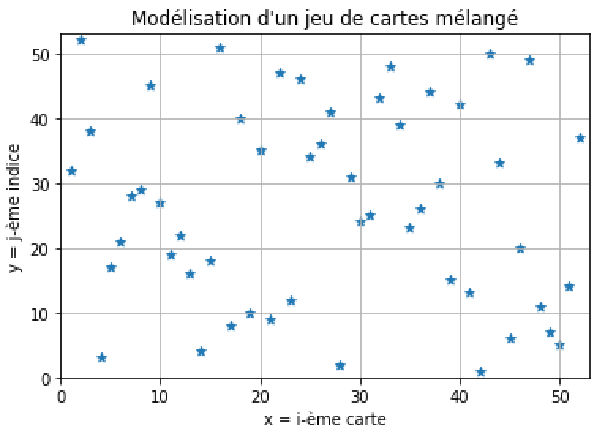


Figure 4 – état du paquet une fois T atteint.

On remarque que les points sont bien dispersés sur le graphique, ce qui laisse à penser que le jeu de cartes est bien mélangé.

Maintenant qu'on a toutes ces informations, on a désormais besoin de voir quelle est la loi de T, pour cela, il nous faut retourner au problème du collectionneur.

3.3.2 Retour au collectionneur de coupons

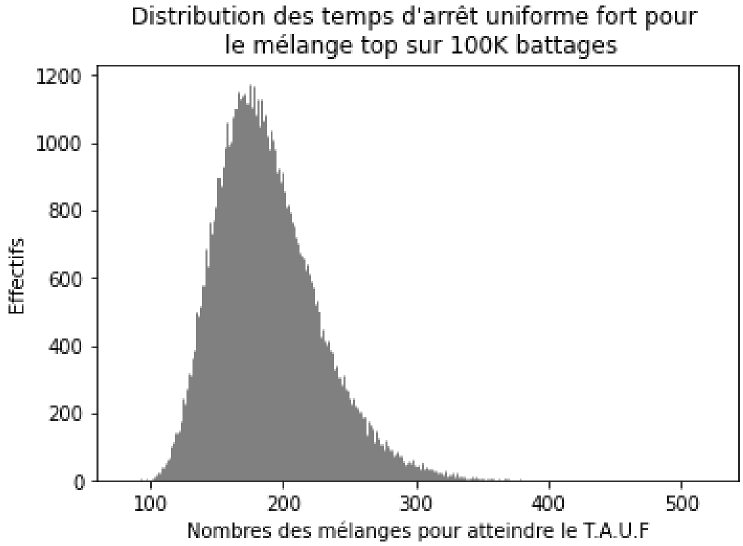
Dans cette section, nous allons faire le lien entre le problème du collectionneur et du mélange Top-to-Random. On se fixe un entier $i \geqslant 1$.
On rappelle que dans la section qui est dédiée à ce problème, nous avons désigné par τ_i le nombre de coupons présents dans la collection lorsque la vignette i est obtenue pour la première fois. Cela revient à dire que c'est le temps d'attente pour que le nombre de coupons distincts qu'il possède passe de $i-1$ à i.
Pour ce qui est du mélange, on a appelé $T_i - T_{i-1}$ le moment où D passe de la position $N-(i+1)$ à la position $N-i$.
Tous les temps $(T_i)_{i \in \mathbb{N}^*}$ sont indépendants entre eux et $\forall i \leqslant N$: T_i - T_{i-1} suit une loi géométrique de paramètre $\frac{i}{N}$. On rappelle que τ_i - $\tau_{i-1} \sim G(\frac{N-i+1}{N})$.
La quantité T_i - T_{i-1} représente le nombre d'opérations effectuées pour passer du moment où il y a $(i-1)$ cartes sous D au moment où il y a i cartes sous D.
Soit k $\in \mathbb{N}$ On remarque que :

$$\mathbb{P}(T_i - T_{i-1} = k) = \mathbb{P}(\tau_{N-i+1} - \tau_{N-i} = k)$$

Ceci montre que T et τ ont la même loi, donc la même espérance.

On décide de produire l'histogramme de distribution du temps d'arrêt uniforme fort pour le mélange Top. Pour cela, on a besoin d'une fonction qu'on appellera $compteur\_temps\_arret\_moyen(T, n)$.
Cette fonction prend en argument un jeu de cartes T et un entier n. Elle va calculer sur n battages, la moyenne de mélanges top nécessaires pour atteindre le temps d'arrêt uniforme fort. Elle va également nous donner la liste des n temps d'arrêt.
Vous trouverez cette fonction en annexe.

Voici donc le graphique de distribution du temps d'arrêt uniforme fort pour le mélange Top sur 100.000 battages :
(Le code ayant permis de produire ce graphique se trouve en annexe.)



On voit que les valeurs semblent se concentrer autour de N ln N $\simeq$ 205.
On peut désormais utiliser tous les résultats que nous avons déjà démontrés dans la première partie et les appliquer à T. On a notamment la remarque (très importante) suivante :

Remarque 4. *Pour tout $c > 0$,*

$$\mathbb{P}(T > \lceil N \ln N + cN \rceil) \leqslant \mathrm{e}^{-c}$$

En particulier, comme l'espérance de T vaut approximativement $N \ln N$, il est assez peu probable que T ne soit bien plus élevé que cette valeur attendue. Donc on peut se dire que $N \ln N$ est un minorant du nombre de mélanges nécessaires.

Dans la partie suivante nous allons utiliser de nombreux résultats de la partie sur le problème du collectionneur. Cela nous permettra de nous donner une idée du nombre de mélanges nécessaires pour que les cartes du paquet soient considérées comme rangées dans un ordre aléatoire.

3.4 Majoration du nombre de mélanges

Dans cette section, nous allons nous demander à partir de quel moment peut-on considérer que le paquet est suffisamment mélangé pour le modèle Top-to-Random. Pour ce faire, nous allons, comme dit précédemment, utiliser les résultats déjà démontrés dans la première partie.
Commençons par utiliser la Proposition 9, celle-ci nous indique que :

$$d_V(\mu_n, U) \leqslant \mathbb{P}(T > n) \quad \forall n \in \mathbb{N}$$

Or, nous avons montré que T et τ suivent la même loi. On a donc $\mathbb{P}(\tau > n) = \mathbb{P}(T > n)$. On applique donc l'inégalité précédente à $\lceil N \ln N + cN \rceil$ et on utilise la Proposition 3 pour enfin obtenir que :

$$d_V(\mu_{\lceil N \ln N + cN \rceil}, U) \leqslant e^{-c} \tag{1}$$

Cette inégalité nous permet de nous rendre compte qu'après N $\ln N$ + cN mélanges, la distance en variation totale entre μ_n et U est significativement proche de 0. Maintenant, la question est de savoir à partir de quelle valeur de d_V on peut considérer que le paquet est assez mélangé. Pour répondre à cette question, nous allons introduire le "temps de mélange". On peut déjà se donner une petite idée au vu du graphique ci-dessous :

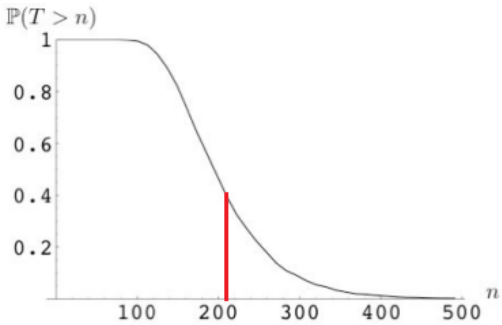


FIGURE 5 – évolution de $\mathbb{P}(T > n)$ pour N=52 en fonction de n.
Source : "Agrégation externe de mathématiques, session 2009, Épreuve de modélisation, option A : Probabilités et Statistiques"

Sur le graphique, on observe une décroissance très rapide de $\mathbb{P}(T > n)$ au voisinage de 205 $\simeq$ N ln N (l'espérance de T) ce qui est logique puisque T est un critère d'arrêt uniforme fort. On voit donc que l'on peut considérer que la distance en variation totale entre μ_n et U devient petite (donc que le paquet commence à être bien mélangé) à partir du moment où $n \simeq N \ln(N)$. On peut donc conjecturer en disant qu'au moins 205 mélanges seront nécessaires. Peaufinons cette observation.

Définition 3. *Le temps de mélange $t_{mix}(\epsilon)$ associé à une chaîne de Markov est le temps mis par la distance en variation totale pour atteindre une certaine valeur fixée ϵ >0.*

En d'autres termes, on se fixe un ϵ assez petit et une fois que t_{mix} atteint cette valeur, on peut considérer que la chaîne (ici le paquet) est bien mélangée. Il est d'usage de prendre $\frac{1}{4}$ pour valeur de ϵ. On peut, avec notre ϵ choisi, définir t_{mix} de la façon suivante :

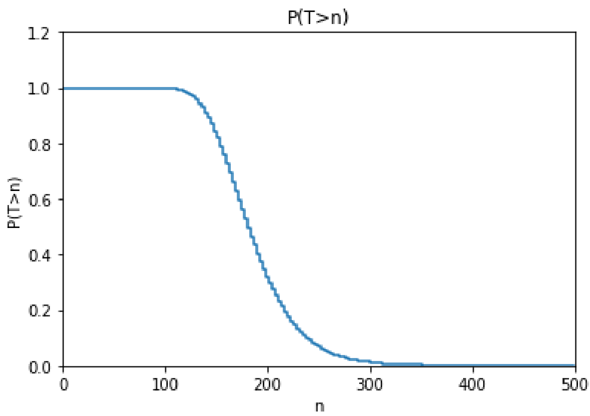
$$t_{mix} = t_{mix}(\frac{1}{4}) = min\{t : d_V(\mu_t, U) \leqslant \frac{1}{4}\}$$

Avec cette nouvelle contrainte et d'après (1), il suffit de choisir c = ln(4) et on obtient cette dernière inégalité :

$$t_{mix} \leqslant N\ ln\ N + N\ ln(4)$$

Donc pour N = 52, cela donne approximativement 278 mélanges. Cela signifie concrètement qu'il faut entre 205 (Espérance de T pour N = 52) et 278 mélanges pour que l'ordre des cartes du paquet soit dans un ordre aléatoire.

On décide de voir ce qu'il en est en pratique.
On va tracer le graphique de $\mathbb{P}(T > n)$. Pour cela, on simule un échantillon de 100.000 battages. On reprend l'histogramme de distribution de temps d'arrêt uniforme fort pour le mélange Top.
Puis, à l'aide des données de cet histogramme, on trace la fonction de répartition empirique F_k de notre échantillon.



Vous trouverez le code ayant permis de produire ces graphiques en annexe.
A l'aide du code en annexe, nous avons récupéré l'abscisse correspondante à $\epsilon = \frac{1}{4}$. Et le nombre de mélanges nécessaires était de 213.

Ce résultat est dans la fourchette qu'on avait trouvée. Dans tous les cas, le nombre de mélanges nécessaires est bien trop élevé. De plus, le trop gros écart entre le minorant et le majorant nous amène à nous intéresser à d'autres modèles pour mélanger un paquet.

Nous allons donc maintenant nous intéresser à un modèle bien plus réaliste et surtout plus efficace : le Riffle-Shuffle.

4 Mélange Riffle-Shuffle

4.1 Etude du processus aléatoire

Le mélange à l'américaine, aussi appelé *riffle shuffle* en anglais, consiste à séparer un paquet de cartes en deux suivant une loi binomiale. Ensuite, on entrelace les cartes des deux sous-paquets afin de reformer un seul paquet.
Soit un jeu de n cartes rangé, où les cartes sont numérotées de 1 à n, n étant la dernière carte du paquet, celle du bas de la pile.

Première étape :
Il suffit de couper le paquet en deux. Nous aurons alors un paquet de t cartes avec probabilité $\frac{1}{2^n}\binom{n}{t}$, où $0 \leq t \leq n$. Le deuxième paquet sera composé de $n-t$ cartes. Pour simuler la coupe du paquet, on va utiliser la loi binomiale.

Soit A_t l'évènement "Il y a t cartes dans le premier paquet".
Alors, $\mathbb{P}(A_t) = \frac{1}{2^n}\binom{n}{t}$. Si on trace le graphique de $\mathbb{P}(A_t)$, $\forall t \in [1, n]$, on remarquera qu'il coïncide exactement avec la densité de la loi binomiale de paramètres $n \in \mathbb{N}$ et $p = \frac{1}{2}$. En effet, si $X \sim Bin(n, p) = Bin(n, \frac{1}{2})$, alors $\mathbb{P}(X = t) = \binom{n}{t}p^t(1-p)^{n-t} = \binom{n}{t}\frac{1}{2^n}$.

Nous avons modélisé cette étape de mélange à l'aide de cette fonction.
Après avoir fait tourner 100 fois ce programme, nous avons représenté les différents indices de coupe sur un histogramme. Voici le code ayant permis de réaliser cet histogramme.

Deuxième étape :
Considérons à présent nos deux paquets $P1$ et $P2$, contenant respectivement r et l cartes. Soit P le paquet qu'on obtient en mélangeant $P1$ et $P2$. On va vider $P1$ et $P2$ par le bas en faisant tomber simultanément les cartes de leur pile sur la paquet P.

Pour mieux comprendre cette étape de mélange, on va le découper en n instants.
A l'instant k, on suppose que le paquet $P1$ contient r_k cartes, et que le paquet $P2$ en contient l_k. Le paquet P en contient donc $n - r_k - l_k$. On choisira de faire tomber sur P la dernière carte de $P1$ avec probabilité $\frac{r_k}{r_k+l_k}$.
La probabilité que la dernière carte de $P2$ tombe sur P vaudra donc $\frac{l_k}{r_k+l_k}$.
On recommence ce procédé jusqu'à ce que $P1$ et $P2$ soient vidés.

Pour modéliser cette deuxième étape, nous avons écrit une deuxième fonction *melange* qui prend en argument deux paquets de cartes. Cette fonction nous renvoie le paquet mélangé (en suivant le processus décrit plus haut).

Grâce à ces deux fonctions, on peut créer la fonction $riffle\_shuffle$ qui prend en argument un paquet de cartes. Cette fonction nous renvoie le paquet de carte après un coup de mélange à l'américaine.
Vous trouverez ces deux fonctions en annexe

Définition 4 (Sous-séquences croissantes)**.** *Soit un jeu de n cartes. Nous associons à l'ordre des cartes du paquet une permutation de Σ_n. Les mélanges à l'américaine correspondent exactement aux permutations $\pi \in \Sigma_n$ telles que la séquence $(\pi(1), \pi(2), ..., \pi(n))$ puisse se décomposer en deux sous-séquences croissantes entrelacées (sauf dans le cas où on retombe sur l'identité). On définira une telle permutation comme suit :*
$\exists t \in \{1, ...n\}$, $\pi(1) < ... < \pi(t)$ *et* $\pi(t+1) < ... < \pi(n)$
Soit A_t l'ensemble des permutations pouvant se décomposer en deux sous-séquences croissantes (une sous-séquences à t éléments, une autre à $n-t$ éléments.
$A_t = \{\pi \in \Sigma_n, \pi(1) < ... < \pi(t) \textit{ et } \pi(t+1) < ... < \pi(n)\}$

On décide d'appliquer au paquet initial (l'identité) deux coups de mélange à l'américaine successifs, puis d'afficher les états du paquet après chacun de ces mélanges :

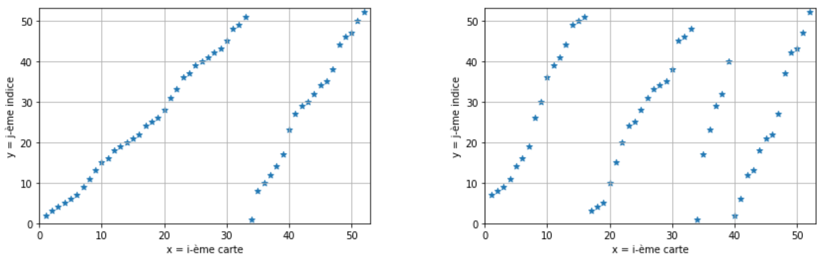
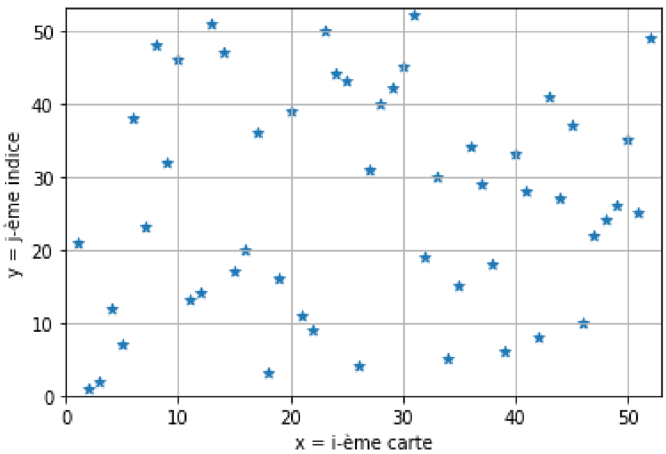


Figure 6 – Jeu de cartes après deux mélanges consécutifs

On voit bien les sous-séquences croissantes sur ces deux graphiques.
On affiche ensuite l'état du jeu de cartes après 7 mélanges consécutifs :



Après 7 mélanges on voit que le graphique ressemble à un nuage de points et on ne distingue plus les sous séquences croissantes. Vous trouverez les graphiques des états intermédiaires en annexe. (Voici le code.)

Lemme 1. *$\forall t, t' \in \{1, ..., n\}$, Supposons $t \neq t'$, alors $A_t \cap A_{t'} = \{Id\}$*

Démonstration. Supposons que $t < t'$. Soit $\pi \in A_t \cap A_{t'}$.
Alors, d'après la Définition 4

$$\pi(1) < ... < \pi(t) \ \ et \ \ \pi(t+1) < ... < \pi(n)$$

$$\pi(1) < ... < \pi(t') \ \ et \ \ \pi(t'+1) < ... < \pi(n)$$

Or $t < t'$ donc $t + 1 \leq t'$, ce qui implique que :

$$\pi(1) < ... < \pi(t) < \pi(t+1) < ... < \pi(n)$$

Donc $\forall i \in \{1, ...n\}$, $\pi(i) = i$, i.e $\pi = id$.
On démontre la même chose pour le cas où $t > t'$ à l'aide d'un raisonnement analogue. □

Proposition 10. *Notons Ω l'ensemble des permutations pouvant se décomposer en deux sous-séquences croissantes. Alors :*

$$Card(\Omega) = 2^n - n$$

Démonstration.

$$Card(\Omega) = Card(\ \{Id\} \ \bigcup \ (\bigcup_{t=1}^{n} A_t \backslash \{Id\} \ ))$$

Sur l'égalité ci-dessus, on a enlevé l'identité à chaque ensemble A_t pour ne pas la compter plusieurs fois. Puis on l'a rajoutée une seule fois devant l'union.

$$Card(\Omega) = Card(\{Id\}) + Card(\bigcup_{t=1}^{n} A_t \backslash \{Id\} \ )$$

On a l'égalité ci-dessus car l'union est disjointe.

$$Card(\Omega) = Card(\{Id\}) + \sum_{t=1}^{n} Card(A_t \backslash \{Id\})$$

On a l'égalité ci-dessus grâce à la Lemme 1.

$$Card(\Omega) = Card(\{Id\}) + \sum_{t=1}^{n} (Card(A_t) - Card(\{Id\}))$$

Et comme il y a $\binom{n}{k}$ permutations dans l'ensemble A_k, on obtient l'égalité suivante :

$$Card(\Omega) = 1 + \sum_{t=1}^{n} \left(\binom{n}{t} - 1 \right) = 1 - \left(\binom{n}{0} - 1 \right) + \sum_{t=0}^{n} \left(\binom{n}{t} - 1 \right)$$

Sur l'égalité ci-dessus, on a ajouté et enlevé le terme $t = 0$ pour pouvoir appliquer la formule du binôme de Newton. On obtient alors,

$$Card(\Omega) = 1 + 2^n - (n + 1) = 2^n - n$$

□

Proposition 11. *On note $Rif : \Sigma_n \longrightarrow \mathbb{R}$ la distribution de probabilité d'un jeu de n cartes après un coup de mélange à l'américaine sur le jeu rangé. Autrement dit, on applique à l'identité un coup de mélange à l'américaine. Soit $\pi \in \Sigma_n$.*

$$Rif(\pi) = \begin{cases} \frac{n+1}{2^n} & \textit{si } \pi = id, \\ \frac{1}{2^n} & \textit{si } \pi \textit{ consiste en deux séquences croissantes,} \\ 0 & \textit{sinon.} \end{cases}$$

$Rif(\pi)$ représente la probabilité d'obtenir π après avoir appliqué un coup de mélange à l'américaine sur l'identité.

Démonstration. Soit un jeu de n cartes.
Nous allons démontrer que $Rif(\pi) = \frac{n+1}{2^n}$ si $\pi = id$.
Soit A_k l'évènement "Couper le paquet de cartes en deux sous-paquets P_1 (contenant k cartes) et P_2 (contenant $n - k$ cartes)".
Soit B l'évènement "Tout le paquet P_2 tombe avant le paquet P_1".
Soit J_1 l'évènement "Le paquet P_1 tombe en entier d'un coup". Soit J_2 l'évènement "Le paquet P_2 tombe en entier d'un coup".
On précise que P_1 est le sous-paquet "du dessous", et P_2 est le sous-paquet "du dessus". Alors,

$$Rif(\pi) = \sum_{k=0}^{n} \mathbb{P}(A_k)\mathbb{P}(B) = \sum_{k=0}^{n} \mathbb{P}(A_k)\mathbb{P}(J_1)\mathbb{P}(J_2|J_1)$$

Or,

$$\mathbb{P}(A_k) = \binom{n}{k}\frac{1}{2^n}$$
$$\mathbb{P}(J_1) = \frac{k}{n}\frac{k-1}{n-1}\dots\frac{1}{n-k+1}$$
$$\mathbb{P}(J_2|J_1) = \frac{n-k}{n-k}\frac{n-k-1}{n-k-1}\dots\frac{1}{1}$$

Donc,

$$\begin{aligned} Rif(\pi) &= \sum_{k=0}^{n}\binom{n}{k}\frac{1}{2^n}\frac{k}{n}\frac{k-1}{n-1}\dots\frac{1}{n-k+1}\frac{n-k}{n-k}\frac{n-k-1}{n-k-1}\dots\frac{1}{1} \\ &= \frac{1}{2^n}\sum_{k=0}^{n}\binom{n}{k}\frac{k!(n-k)!}{n!} \\ &= \frac{1}{2^n}\sum_{k=0}^{n}\binom{n}{k}\frac{1}{\binom{n}{k}} = \frac{1}{2^n}\sum_{k=0}^{n}1 \\ &= \frac{n+1}{2^n} \end{aligned}$$

Par ailleurs, il est évident que $Rif(\pi) = 0$ lorsque π n'est ni l'identité, ni une permutation composée de deux séquences croissantes. $\square$

Exemple 2. *Nous allons maintenant donner un exemple pour illustrer le fait que $Rif(\pi) = \frac{1}{2^n}$ lorsque π consiste en deux séquences croissantes. On se place dans Σ_5 ; on part de l'identité, à chaque fois, la probabilité d'affecter un 0 ou un 1 à une carte vaut $\frac{1}{2}$. On obtient ainsi la permutation ci-dessous avec probabilité $\frac{1}{2}\times\frac{1}{2}\times\dots\times\frac{1}{2}$ et on obtient bien $\frac{1}{2^5}$*

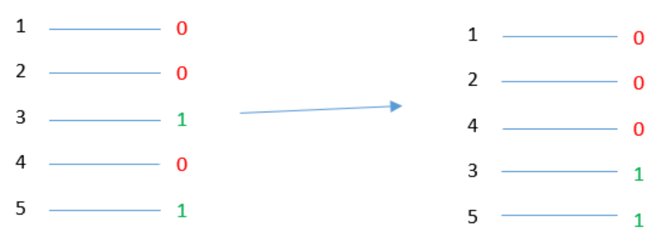


FIGURE 7 – Exemple de configuration pour le mélange inverse dans Σ_5

4.2 Mélange inverse

4.2.1 Description du mélange

On se place à l'instant t, et on décrit un battage du mélange inverse. On considère un paquet de n cartes. On affecte de manière aléatoire et indépendante à chaque carte la valeur 0 ou 1 avec probabilité $\frac{1}{2}$. On fait ensuite remonter les cartes affectées d'un 0 en haut du paquet tout en préservant l'ordre des cartes. Puis on recommence.

Exemple 3. *Voici un battage de mélange inverse sur un jeu de 7 cartes.*

0	1
1	2
1	3
0	4
1	5
0	6
0	7

0	1
0	4
0	6
0	7
1	2
1	3
1	5

Vous trouverez en annexe A4 le code permettant de simuler le mélange inverse.

4.2.2 Critère d'arrêt uniforme fort

Nous allons caractériser le critère d'arrêt du mélange inverse. A chaque étape du mélange inverse, on garde en mémoire la valeur affectée à chaque carte. Cela veut dire qu'à la k-ème étape de mélange, chaque carte sera affectée d'une suite aléatoire de 0 et de 1, i.e d'un élément de $\{0,1\}^k$.

Théorème 3 (Critère d'arrêt pour le mélange inverse)**.** *Nous avons vu précédemment comment stocker, au cours de plusieurs mélanges inverses successifs, le mot (composé de 0 et de 1) associé à chaque cartes.*
Le critère d'arrêt est le suivant :
"Dès le moment où les mots associés à chaque carte sont tous différents entre eux, alors on peut arrêter le mélange".

Nous allons traiter un exemple pour illustrer le stockage des bits sur trois mélanges successifs.

Remarque 5. *Sur l'exemple ci-dessous, on constate qu'il faut recourir à 3 coups de mélange inverse pour que le temps d'arrêt soit atteint. En effet les mots (composés de 0 et de 1) associés à chaque carte sont tous différents entre eux.*

Exemple 4. *Traitons un exemple de stockage des bits sur trois mélanges inverses successifs.*

0	1
1	2
0	3
0	4
1	5

On fait remonter les cartes codées d'un zéro.

0	1
0	3
0	4
1	2
1	5

10	1
00	3
00	4
11	2
11	5

On fait remonter les cartes codées d'un zéro.

00	3
00	4
10	1
11	2
11	5

000	3
100	4
010	1
011	2
111	5

On fait remonter les cartes codées d'un zéro.

000	3
010	1
011	2
100	4
111	5

En rouge, ce sont les nouveaux bit associés à chaque cartes. Ils se trouvent sur la colonne de droite. Puis on ré-organise le jeu en faisant remonter les cartes dont le mot associé commence par un zéro. Puis on recommence.

Proposition 12. *Comme chaque bit est choisi aléatoirement et indépendamment des autres, le critère d'arrêt est donc uniforme fort.*

4.3 Lemme d'inversion de Reed

Proposition 13. *Nous avons vu ce que représentait Rif à la Proposition 11. Posons maintenant $\overline{Rif} : \Sigma_n \longrightarrow \mathbb{R}$.*
Soit $\pi \in \Sigma_n$, $\overline{Rif}(\pi)$ représente la probabilité de tomber sur π après un coup de mélange inverse sur le paquet ordonné. Autrement dit, $\overline{Rif}$ est la fonction qui caractérise la distribution de probabilité pour le mélange inverse.
On a l'égalité suivante

$$\forall \pi \in \Sigma_n \, , \, \overline{Rif}(\pi) = Rif(\pi^{-1})$$

Démonstration. Démontrons l'égalité ci-dessus.

1. Supposons que $\pi = id$:
   Cela revient à démontrer que $Rif(id) = \overline{Rif}(id)$.
   On sait par la Proposition 11 que $Rif(id) = \frac{n+1}{2^n}$.
   Démontrons alors que $\overline{Rif}(id) = \frac{n+1}{2^n}$.

Pour retrouver l'identité après avoir appliqué une simulation de mélange inverse à l'identité : Il faudrait que toutes les cartes codées d'un zéro se trouvent déjà en haut du paquet.
Soit A_i l'évènement "Il y a i cartes affectées d'un zéro en haut du paquet". Alors,

$$\overline{Rif}(id) = \sum_{k=0}^{n} \mathbb{P}(A_i)$$

Et comme $\forall i \in \{0, ..., n\}$, $\mathbb{P}(A_i) = \frac{1}{2^n}$ (Les 0 et les 1 sont choisis avec probabilité $\frac{1}{2}$) :

$$\overline{Rif}(id) = \frac{1}{2^n} \sum_{k=0}^{n} 1 = \frac{n+1}{2^n}$$

2. Supposons que π consiste en deux séquences croissantes :
   On veut montrer que $\overline{Rif}(\pi) = Rif(\pi^{-1})$.
   En posant $\sigma = \pi^{-1}$, cela revient à démontrer que :

$$\overline{Rif}(\sigma^{-1}) = Rif(\sigma)$$

   On sait par la Proposition 11 que $Rif(\sigma) = \frac{1}{2^n}$.
   Démontrons alors que $\overline{Rif}(\sigma^{-1}) = \frac{1}{2^n}$.
   En fait $\forall \pi \in \Omega \backslash \{Id\}$, $\overline{Rif}(\pi) = \frac{1}{2^n}$
   C'est vrai car les 0 et les 1 sont choisis avec probabilité $\frac{1}{2}$.

3. Supposons que π ne se situe dans aucun des deux cas précédents :
   L'égalité se démontre encore par un raisonnement analogue.

□

Définition 5. *On définit Rif^{*k} comme étant l'enchaînement de k mélanges à l'américaine sur le paquet rangé. De même, $\overline{Rif}^{*k}$ sera défini comme l'enchaînement de k mélanges inverses sur le paquet rangé.*

Théorème 4 (Lemme d'inversion de Reed)**.** *Soit U la distribution de probabilité de la loi uniforme. Nous avons déjà défini Rif et $\overline{Rif}$. On a l'égalité suivante :*

$$d_V(Rif^{*k}, U) = d_V(\overline{Rif}^{*k}, U)$$

Démonstration. D'après la Définition 2, on a :

$$d_V(Rif^{*k}, U) = \sum_{\pi \in \Sigma_n} |Rif^{*k}(\pi) - U(\pi)|$$

D'après la Proposition 13, on obtient :

$$d_V(Rif^{*k}, U) = \sum_{\pi \in \Sigma_n} |\overline{Rif}^{*k}(\pi^{-1}) - U(\pi)|$$

De plus, $U(\pi) = U(\pi^{-1})$, $\forall \pi \in \Sigma_n$. Donc,

$$d_V(Rif^{*k}, U) = \sum_{\pi \in \Sigma_n} |\overline{Rif}^{*k}(\pi^{-1}) - U(\pi^{-1})|$$

Soit $\pi \in \Sigma_n$, comme π possède un unique inverse dans Σ_n : Parcourir les π^{-1} sur Σ_n revient exactement à parcourir les π sur Σ_n. Donc,

$$d_V(Rif^{*k}, U) = \sum_{\pi \in \Sigma_n} |\overline{Rif}^{*k}(\pi) - U(\pi)|$$

Par conséquent, on obtient bien :

$$d_V(Rif^{*k}, U) = d_V(\overline{Rif}^{*k}, U)$$

□

4.3.1 Retour au paradoxe des anniversaires

Théorème 5. *Après avoir effectué k mélanges à l'américaine sur un jeu de n cartes rangé, la distance en variation totale vérifie :*

$$d_V(Rif^{*k}, U) \leq 1 - \prod_{i=1}^{n-1}(1 - \frac{i}{2^k})$$

Démonstration. D'après le Lemme d'inversion de Reed, on sait que :

$$\forall \pi \in \Sigma_n \, , \, \overline{Rif}(\pi) = Rif(\pi^{-1})$$

Soit $\tilde{T}$ est le temps d'arrêt du mélange inverse.
D'après la proposition 9, on a que :

$$d_V(Rif^{*k}, U) \leqslant \mathbb{P}(\tilde{T} > k)$$

Or, d'après la Proposition 5 du paradoxe des anniversaires : Si n boules sont placées indépendamment et aléatoirement dans K boîtes, alors la probabilité qu'aucune boîte ne contienne plus d'une boule vaut :

$$p(n, K) = \prod_{i=1}^{n-1}(1 - \frac{i}{K})$$

A la k-ème étape de mélange inverse, on décide de placer les cartes ayant le même mot (de $\{0,1\}^k$) dans une même boîte. Il y a donc $K = 2^k$ boîtes possibles.
La probabilité qu'une boîte reçoive plus d'une carte est donc égale à la probabilité que le temps d'arrêt uniforme fort pour le mélange inverse ne soit pas atteint.

Par analogie avec le paradoxe des anniversaires, cette probabilité vaut donc :

$$\mathbb{P}(\tilde{T} > k) = 1 - \prod_{i=1}^{n-1}(1 - \frac{i}{2^k})$$

Par conséquent, on obtient bien que :

$$d_V(Rif^{*k}, U) \leq 1 - \prod_{i=1}^{n-1}(1 - \frac{i}{2^k}) = \mathbb{P}(\tilde{T} > k)$$

□

4.4 Majoration du nombre de mélanges

Proposition 14. *On reprend la notion de t_{mix} définie dans la partie 2. Pour un jeu de n cartes, dans le cas du Riffle Shuffle, on a l'inégalité suivante :*

$$t_{mix} \leqslant 2\ln_2(\frac{4n}{3})$$

pour n assez grand.

Démonstration. Prenons k = 2 $\ln_2$ ($\frac{n}{c}$) pour c $\geqslant$ 1 constante. On a :

$$\mathbb{P}(\tilde{T} \leqslant k) = \prod_{i=1}^{n-1}(1 - \frac{i}{2^k})$$

On va appliquer le log afin de transformer le produit en somme et nous simplifier la tâche. On a donc, en remplaçant k par la valeur choisie :

$$\begin{aligned}\ln(\prod_{i=1}^{n-1}(1 - \frac{c^2 i}{n^2})) &= \sum_{i=1}^{n-1}\ln(1 - \frac{c^2 i}{n^2}) \\ &= -\sum_{i=1}^{n-1}(\frac{c^2 i}{n^2}) + o(\frac{i^2}{n^4})\end{aligned}$$

La dernière égalité ayant été obtenue par les Séries de Taylor. On se retrouve donc avec :

$$\begin{aligned}\ln(\mathbb{P}(\tilde{T} \leqslant k)) &= -\frac{c^2 n(n-1)}{2n^2} + o(\frac{n^3}{n^4}) \\ &= -\frac{c^2}{2} + o(\frac{1}{n})\end{aligned}$$

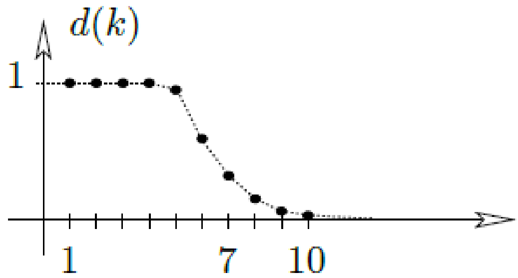
On va désormais passer à la limite lorsque n tend vers $+\infty$, on obtient ainsi :

$$\lim_{n\to\infty}\ln(\mathbb{P}(\tilde{T} \leqslant k)) = -\frac{c^2}{2} \Rightarrow \lim_{n\to\infty}\frac{\mathbb{P}(\tilde{T} \leqslant k)}{\exp(-\frac{c^2}{2})} = 1$$

On prend maintenant c $< \sqrt{2\ln(\frac{4}{3})} \simeq 0.759$, donc lorsque n $\to \infty$, on a que $\mathbb{P}(\tilde{T} \leqslant k) \to \frac{3}{4}$ et ainsi que $\mathbb{P}(\tilde{T} > k) \to \frac{1}{4}$
Pour n suffisamment grand, on prend donc c $= \frac{3}{4}$ pour avoir $t_{mix} \leqslant 2\ln_2(\frac{4n}{3})$ et la proposition est démontrée. $\square$

Ainsi, pour $n = 52$, cela donne approximativement 12 mélanges.
Mais en pratique, on ne procède jamais à 12 mélanges. Et on ne peut pas vraiment évaluer le nombre de mélanges à faire en utilisant la distance de variation totale. En effet le calcul serait bien trop conséquent pour un paquet de 52 cartes.

L'étude plus poussée de Persi Diaconis, le nombre de mélanges à l'américaine optimal qu'il faudrait faire est de 7. On pourrait représenter le graphique de la distance en variation totale :



Source : Article donné pour base du mémoire

Au vu des graphiques que nous avons produits pour le mélange à l'américaine, on pourrait penser que 3 ou 4 mélanges suffisent. Mais d'après ce graphique, on constate que le jeu n'est en fait pas du tout ordonné. Alors qu'à partir du 7 ou 8ème mélange, le jeu commence à s'approcher très sensiblement de la distribution uniforme.

5 Conclusion

On rappelle que l'objectif de cette recherche était de trouver le nombre optimal de mélanges à faire pour qu'un jeu de cartes soit bien mélangé. L'étude nous a permis de répondre à la question dans le cadre des deux mélanges traités. Nous avons analysé le mélange Top-to-Random ainsi que le mélange à l'américaine.

Nous avons décidé de nous placer dans un modèle réaliste. Soit donc t_{mix} le nombre de mélanges à faire pour que le jeu de $N = 52$ cartes soit bien mélangé. Les résultats des deux problèmes combinatoires nous ont permis de minorer et de majorer t_{mix}.

Pour le mélange Top-to-Random, on a établi qu'il fallait mélanger $N\ln(N) = 205$ fois un jeu de 52 cartes pour que la distance en variation totale avec la distribution uniforme soit significativement inférieure à 1. Aussi, en prenant $c = \ln(4)$, on a vu qu'il fallait mélanger $N\ln(N) + cN = 278$ fois pour que celle-ci tombe en dessous de $\frac{1}{4}$. En modélisant 100.000 mélanges, on trouvait qu'il fallait mélanger le jeu de cartes 213 fois pour s'approcher significativement de la distribution uniforme.

Quant au mélange à l'américaine, nous avons majoré t_{mix} par 12. De plus, il a été démontré par Persi Diaconis qu'il fallait mélanger au minimum 5 fois pour que la distance en variation totale (par rapport à la distribution uniforme) soit significativement inférieure à 1. A partir de 7 mélanges, la distance en variation totale s'approche de zéro.

Finalement, on constate que ce mélange est bien plus réaliste et efficace que le mélange Top-to-Random. Mais cette étude reste très subjective. En effet, les résultats peuvent varier d'un battage à l'autre. On atteindra parfois le temps d'arrêt uniforme fort très rapidement "par chance". Et inversement.

De plus, nous n'avons traité que deux mélanges différents, alors qu'il existe une infinité de façon de mélanger un jeu de cartes. On pourrait alors se demander que deviendraient les résultats si nous avions pris une autre méthode de mélange.

Si on changeait le mélange Top-to-Random en mélange Random-to-Random (au lieu de choisir la carte haut du paquet, on choisirait une carte uniformément au hasard), le modèle serait-il meilleur ?

6 Annexes

6.1 Rappels sur les Chaînes de Markov

Cette partie va être constituée de définitions et de résultats importants concernant les chaînes de Markov qui nous seront très utiles par la suite. Nous allons d'abord définir ce qu'est une chaîne de Markov.

Définition 6. *Soit $(X_n)_{n\in\mathbb{N}}$ une suite à valeurs dans E un espace fini ou dénombrable. Le processus stochastique $(X_n)_{n\in\mathbb{N}}$ est une chaîne de Markov si : $\forall n > 0$ et $\forall(i_0, i_1, ..., i_{n-1}, i, j) \in E^{n+2}$*

$$\begin{aligned} p_{i,j} &= \mathbb{P}(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) \\ &= \mathbb{P}(X_{n+1} = j | X_n = i) \end{aligned}$$

Dans ce cas,

- l'ensemble E est appelé espace d'état.
- $\mathbb{P}(X_{n+1} = j | X_n = i)$ est la probabilité de transition de l'état i à l'état j.
- l'égalité précédente est appelée propriété de Markov.

On dit que la Chaîne de Markov est homogène si les probabilités de transition sont indépendantes de n.

Définition 7. *On appelle Matrice de Transition P de la chaîne (X_n) , la matrice :*

$$P = \{p_{i,j}\}_{i,j\in E}$$

Une matrice de transition est stochastique, c'est-à-dire

1. $\forall i, j \in E$, $p_{i,j} \geq 0$, i.e tous ses éléments sont des réels positifs.
2. $\forall i \in E$, $\sum_{j\in E} p_{i,j} = 1$, i.e la somme des termes de n'importe quelle ligne de P donne toujours 1.

On se donne une chaîne de Markov $(\mathrm{X}_n)_{n\in\mathbb{N}}$, de matrice de transition P et on suppose désormais E fini.

Définition 8. *Soient i et j des éléments de E, on dit que j est accessible depuis i s'il existe un entier $M \geq 0$ tel que $p_{i,j}(M) > 0$. On dit que i et j communiquent si i est accessible depuis j (que l'on note $i \leftarrow j$) et que j est accessible depuis i (que l'on note $i \to j$). On note ceci $i \leftrightarrow j$.*

Remarque 6. *Si i et j communiquent, alors on dira qu'ils sont dans la même classe communicante. On remarque que la notion de classe communicante est une relation d'équivalence sur E, elle permet donc de partitionner E.*

Définition 9. *On dit que la chaîne* $(X_n)_{n\in\mathbb{N}}$ *est irréductible si elle ne possède qu'une seule classe de communication. Autrement dit, une chaîne est irréductible si tout état est accessible à partir de n'importe quel autre état.*

Définition 10. *1. Soit* $i \in E$. *La période de l'état* i *est la quantité :*

$$d = pgcd\{n \geq 1 | p_{i,i}^n > 0\}$$

où $p_{i,i}$ *représente le coefficient de* P^n *(matrice de transition en n étapes) à la position* (i, i).

2. On dit qu'un état est apériodique si sa période vaut 1

3. Si deux états communiquent, alors ils ont la même période.

4. Une chaîne est apériodique si tous ses états sont apériodiques.

Lemme 2. *Les états d'une chaîne de Markov irréductible ont tous la même période. De plus, une chaîne de Markov est apériodique si elle est de période 1.*

Définition 11. *Une probabilité* π *sur* E *est appelée probabilité invariante (ou stationnaire) d'une chaîne de Markov de matrice de transition* P *si elle vérifie*
$\pi P = \pi$

Les probabilités stationnaires ont un rôle très important quant à la convergence d'une chaîne de Markov.

Lemme 3. *Si* (X_n) *est irréductible, apériodique et à espace d'état fini, elle admet une unique probabilité stationnaire* π *et* $\forall i \in E$, $\pi(i) > 0$.
De plus, pour toute distribution initiale,

$$\lim_{n\to\infty} \mathbb{P}(X_n = i) = \pi(i)$$

Nous allons désormais nous intéresser aux temps d'arrêts et critères d'arrêts.

Définition 12. *Un temps d'arrêt est une variable aléatoire* $T : \Omega \to \mathbb{N}\cup\{\infty\}$ *telle que l'événement* $\{T=n\}$ *(où* $n \in \mathbb{N}$*) ne dépend que d'informations déjà connues :* X_0, X_1, ... , X_n

Définition 13. *Un critère d'arrêt fort pour une chaîne de Markov* (X_n), *de probabilité invariante* π *est un temps d'arrêt* τ, *qui dépend éventuellement de l'état initial* x *tel que :*

$$\mathbb{P}_x(t = \tau, X_\tau = y) = \mathbb{P}_x(t = \tau)\pi_y$$

Celui-ci est dit "uniforme fort" si la condition suivante est vérifiée :
Soit $k \in \mathbb{N}$. *Si le processus est interrompu après exactement k étapes, alors les permutations associées à l'ordre potentiel des cartes ont une distribution uniforme.*

En d'autres termes, c'est un temps d'arrêt pour lequel nous sommes sûrs d'avoir atteint la probabilité stationnaire.

6.2 Graphiques

Affichage de l'état du paquet après 5, puis 50 mélanges top successifs :

Retour au document.

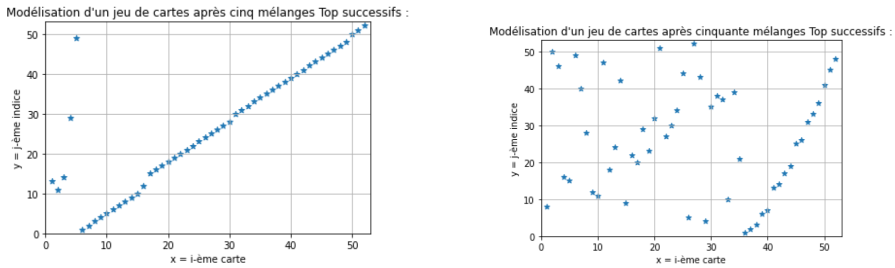
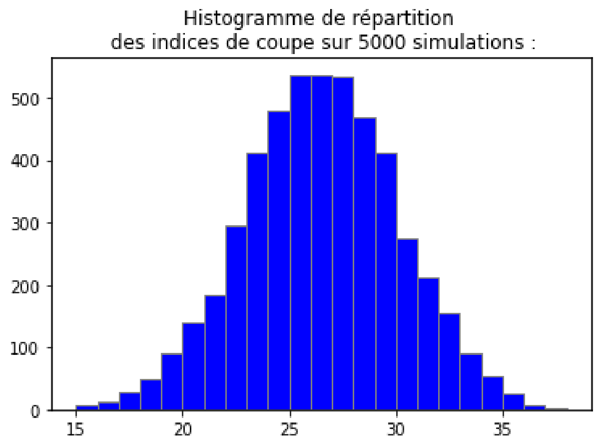


FIGURE 8 – Jeu de cartes après 5 mélanges Top consécutifs, puis 50

Histogramme de répartition des indices de coupe sur 5000 expériences :

Retour au document.



Affichage d'un jeu de cartes après 3, 4, 5 puis 6 mélanges à l'américaine successifs :

Retour au document.

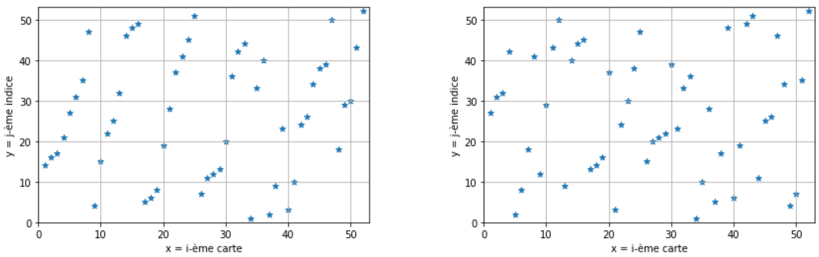


Figure 9 – Jeu de cartes après trois mélanges consécutifs, puis quatre

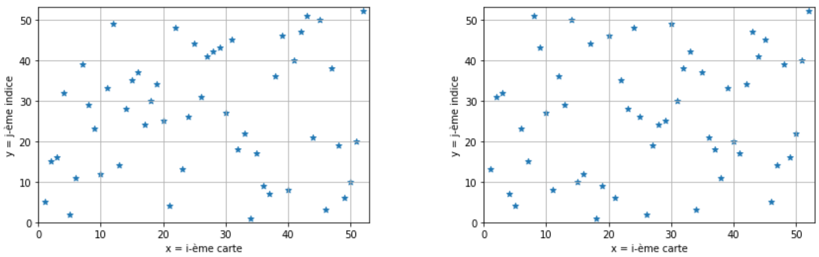


Figure 10 – Jeu de cartes après cinq mélanges consécutifs, puis six

6.3 Algorithmes

Code permettant de produire le graphique de $p(n)$:
Retour au document.

```
import matplotlib.pyplot as plt
from matplotlib import pyplot

def calcul_de_proba(n):
    resultat=1
    for i in range(n):
        resultat=resultat*(1 - (i/365))
        #print(resultat)
    return resultat

def vecteur(n):
    proba=[]
    indice_nb_personnes=[]
    for i in range(n):
        proba+=[calcul_de_proba(i)]
        indice_nb_personnes+=[i]
    return proba,indice_nb_personnes

#print(calcul_de_proba(140))
T=vecteur(140)[0]
indice=vecteur(140)[1]
#print(T)
plt.figure()
plt.scatter(indice,T,marker="*")
plt.axis([0,len(T)+1,0,1])
plt.xlabel('n = Nombre de personnes')
plt.ylabel('y = p(n)')
plt.grid()
plt.show()
```

Code permettant d'afficher l'état du paquet de carte rangé :
Retour au document.

```
'''ON AFFICHE LE JEUX DE CARTE RANGE '''
jeu_de_cartes=[i for i in range(1,53)]
indice_du_paquet=[i for i in range(1,53)]
T=jeu_de_cartes
plt.figure()
plt.scatter(T,indice_du_paquet,marker="*")
plt.axis([0,len(jeu_de_cartes)+1,0,len(indice_du_paquet)+1])
plt.title("Modélisation d'un jeu de cartes rangé")
plt.xlabel('x = i-ème carte')
plt.ylabel('y = j-ème indice')
plt.grid()
plt.show()
```

Code permettant de simuler le mélange Top-to-Random :
Retour au document.

```python
''' 1 MELANGE TOP '''
def melange(T):
    i=randint(1,len(T)+1)
    T_new=T[1:i]+[T[0]]+T[i:len(T)]
    return T_new

#####################################
'''  n MELANGES TOP  '''
def melange_n_fois(T,n):
    for k in range(n):
        T=melange(T)
    return(T)
```

Code permettant d'afficher l'état d'un paquet après un mélange top :
Retour au document.

```python
'''AFFICHAGE DU PAQUET APRES UN MELANGE TOP'''
T=melange(jeu_de_cartes)
plt.figure()
plt.scatter(T,indice_du_paquet,marker="*")
plt.axis([0,len(jeu_de_cartes)+1,0,len(indice_du_paquet)+1])
plt.title("Modélisation d'un jeu de cartes après un mélange Top :")
plt.xlabel('x = i-ème carte')
plt.ylabel('y = j-ème indice')
plt.grid()
plt.show()
```

Code permettant d'afficher l'état du paquet après 5, puis 50 mélanges top consécutifs :
Retour au document.

```python
'''AFFICHAGE DU PAQUET APRES CINQ MELANGES TOP SUCCESSIFS'''
T=melange_n_fois(jeu_de_cartes,5)
plt.figure()
plt.scatter(T,indice_du_paquet,marker="*")
plt.axis([0,len(jeu_de_cartes)+1,0,len(indice_du_paquet)+1])
plt.title("Modélisation d'un jeu de cartes après cinq mélanges Top successifs :")
plt.xlabel('x = i-ème carte')
plt.ylabel('y = j-ème indice')
plt.grid()
plt.show()
```

```
'''AFFICHAGE DU PAQUET APRES CINQUANTE MELANGE TOP SUCCESSIFS'''
T=melange_n_fois(jeu_de_cartes,50)
plt.figure()
plt.scatter(T,indice_du_paquet,marker="*")
plt.axis([0,len(jeu_de_cartes)+1,0,len(indice_du_paquet)+1])
plt.title("Modélisation d'un jeu de cartes après cinquante mélanges Top successifs :")
plt.xlabel('x = i-ème carte')
plt.ylabel('y = j-ème indice')
plt.grid()
plt.show()
```

Code permettant de montrer que la chaîne de Markov associé au mélange Top-to-Random est irréductible :

Retour au document.

```
import matplotlib.pyplot as plt
from matplotlib import pyplot
from random import*
import numpy.random as rd
from math import*
from Fonctions_annexes import*

#On insère e dans T à l'indice i
def inserer(T,i,e):
    n=len(T)
    B=[0]*(n+1)
    for k in range(i):
        B[k]=T[k]
    B[i]=e
    for k in range(i,n):
        B[k+1]=T[k]
    return(B)
```

```python
def Id_vers_random(T):
    #On définit l'identité
    Id=[i for i in range(1,len(T)+1)]
    print("On part de Id=",Id)
    print("On veut arriver à T=",T)
    print("")
    #On va transformer Id en mettant la carte du haut dans le paquet jusqu'à obtenir T
    while Id!=T:
        print("Nouvelle boucle")
        #Premièrement, on prend la première carte de Id
        #Et on cherche l'indice de la carte qui la précède dans T
        #(modulo len(T) pour les cas ou on est au bord du paquet)
        #Indice_1 représente l'indice de la carte qui précède Id[0] dans T
        compteur=0
        while T[compteur]!=Id[0]:
            compteur+=1
        Indice_1=(compteur-1)%len(T)

        #Deuxièmement, on récupère la valeur de la carte qui précède Id[0] dans T
        #valeur représente la valeur de la carte qui précède Id[0] dans T
        valeur=T[Indice_1]

        #Troisièmement, on récupère l'indice de 'valeur' dans Id
        compteur2=0
        while valeur!=Id[compteur2]:
            compteur2+=1
        #Indice_2 représente l'indice ou l'on doit insérer la carte Id[0]
        #pour qu'elle soit au bon endroit
        Indice_2=(compteur2+1)%(len(Id)+1)

        #Quatrièmement, on insère Id[0] après 'valeur' dans Id
        Id=inserer(Id,Indice_2,Id[0])
        #Puis on retire Id[0] puisqu'on vient de la mettre ailleurs dans le paquet
        Id=Id[1:len(T)+1]
        #Et ensuite on fait de l'affichage pour que ça soit plus clair
        print("On insère la carte du haut de la pile après la", Indice_2,"-ème carte")
        print("On obtient",Id)
        print("")
    return Id
```

```
On part de Id= [1, 2, 3, 4, 5]
On veut arriver à T= [2, 5, 1, 4, 3]

Nouvelle boucle
On insère la carte du haut de la pile après la 5 -ème carte
On obtient [2, 3, 4, 5, 1]

Nouvelle boucle
On insère la carte du haut de la pile après la 2 -ème carte
On obtient [3, 2, 4, 5, 1]

Nouvelle boucle
On insère la carte du haut de la pile après la 3 -ème carte
On obtient [2, 4, 3, 5, 1]

Nouvelle boucle
On insère la carte du haut de la pile après la 3 -ème carte
On obtient [4, 3, 2, 5, 1]

Nouvelle boucle
On insère la carte du haut de la pile après la 5 -ème carte
On obtient [3, 2, 5, 1, 4]

Nouvelle boucle
On insère la carte du haut de la pile après la 5 -ème carte
On obtient [2, 5, 1, 4, 3]

[2, 5, 1, 4, 3]
```

Code permettant de simuler le temps d'arrêt uniforme fort pour le mélange Top-to-Random :

Retour au document.

```python
'''  CODAGE DU TEMPS D'ARRET UNIFORME FORT POUR LE MELANGE TOP  '''
def temps_arret_unif_fort(T):
    compteur=0
    #reference est définie comme étant la dernière carte du paquet
    reference=T[len(T)-1]
    #Tant que la dernière carte du paquet n'est pas remontée à la surface, on continue
    while T[0]!=reference:
        compteur+=1
        T=melange(T)
    T=melange(T)
    compteur+=1
    #On retourne le jeux de carte ainsi que le nb de mélange effectué
    return T,compteur
```

Code permettant d'afficher l'état du paquet une fois que le temps d'arrêt uniforme fort est atteint :

Retour au document.

```
'''AFFICHAGE DU PAQUET UNE FOIS QU'ON A ATTEINT LE TEMPS D'ARRET UNIFORME FORT '''
plt.figure()
test=temps_arret_unif_fort(jeu_de_cartes)
plt.scatter(test[0],indice_du_paquet,marker="*")
plt.axis([0,len(jeu_de_cartes)+1,0,len(indice_du_paquet)+1])
plt.title("Modélisation d'un jeu de cartes mélangé")
plt.xlabel('x = i-ème carte')
plt.ylabel('y = j-ème indice')
plt.grid()
plt.show()
```

Code de la fonction compteur_temps_arret_moyen(T,n) :

Retour au document.

```
''' MOYENNE DU TEMPS D'ARRET UNIFORME FORT POUR LE MELANGE TOP SUR n BATTAGES
    ET LISTE DES TEMPS D'ARRET UNIFORME FORT SUR CES n BATTAGES   '''
def compteur_temps_arret_moyen(T,n):
    Indice_temps_arret=[]
    moyenne=0
    for i in range(n):
        test=temps_arret_unif_fort(T)
        moyenne=moyenne+test[1]
        Indice_temps_arret+=[test[1]]
    moyenne=moyenne/n
    return moyenne,Indice_temps_arret
```

Code permettant de produire l'histogramme de distribution du temps d'arrêt uniforme fort pour le mélange Top sur 100.000 battages :

Retour au document.

```
''' MOYENNE DU TEMPS D'ARRET UNIFORME FORT POUR LE MELANGE TOP SUR 400 BATTAGES '''
test1=compteur_temps_arret_moyen(jeu_de_cartes,100000)

###############################################################################
'''  AFFICHAGE DE L'HISTOGRAMME DE DISTRIBUTION DES TEMPS D'ARRETS UNIFORME
    FORT POUR LE MELANGE TOP SUR 400 BATTAGES '''

plt.figure()
pyplot.hist(test1[1], range = (min(test1[1]),max(test1[1])), bins = ceil(len(test1[1])/10),
            color = 'blue',edgecolor = 'grey')
pyplot.xlabel('Nombres des mélanges pour atteindre le T.A.U.F')
pyplot.ylabel('Effectifs')
pyplot.title('Distribution des temps d\'arrêt uniforme fort pour \n le mélange top sur 100K battages')
```

Code permettant de produire le graphique de la fonction de répartition empirique :
Retour au document.

```
'''FONCTION CALCULANT LE NOMBRE D'ELEMENTS DE T QUI SONT INFERIEURS OU EGAUX A d '''
def eff(T,d):
    if max(T)<d:
        return len(T)
    else:
        i=0
        while T[i]<=d:
            i+=1
    return i

################################################################################
'''FONCTION CALCULANT LA FREQUENCE CUMULEE '''
def calcul_prob(T,a):
    return 1-eff(T, a)/len(T)
```

```
'''AFFICHAGE DE LA FONCTION DE REPARTITION EMPIRIQUE '''
#var est le vecteur contenant nos temps d'arrêt uniforme
#fort rangés dans l'ordre croissants
var=np.sort(test1[1])
#x est une subdivision de l'ensemble [0,500] avec un pas de 3
x=np.arange(0, 500, 3)
#y1 est le vecteur contenant les fréquences cumulées à chaque pas
y1=[]
for i in range(len(x)):
    y1+=[calcul_prob(var,x[i])]
#On affiche le graphe
plt.figure()
plt.step(x,y1)
plt.axis([0,500,0,1.2])
pyplot.xlabel('n')
pyplot.ylabel('P(T>n)')
pyplot.title('P(T>n)')

######################################################################
'''ON RECUPERE LA VALEUR DE L'ABSCISSE '''
ind=0
while y1[ind]>0.25:
    ind+=1
print("VALEUR=",x[ind])
```

```python
'''AUTRE MOYEN D'AFFICHER LA FONCTION DE REPARTITION EMPIRIQUE'''
var_1=np.sort(test1[1])
y_var=[]
for i in range(len(var_1)):
    y_var+=[1-i/len(var_1)]
plt.figure()
plt.step(var_1,y_var)
plt.axis([0,500,0,1.2])
pyplot.xlabel('n')
pyplot.ylabel('P(T>n)')
pyplot.title('P(T>n)')
```

Code permettant de générer l'indice de coupe du paquet :
Retour au document.

```python
'''  COUPER LE JEU DE CARTES EN DEUX PAQUETS '''
def couper_paquet(n,p):
    return(rd.binomial(n,p))
```

Code permettant de générer l'histogramme de répartition des indices de coupe :
Retour au document.

```python
''' VOIR LA REPARTITION DES INDICES DE COUPE  '''

T=[]
for i in range(5000):
    T+=[couper_paquet(52,1/2)]
plt.figure()
pyplot.hist(T, range = (min(T),max(T)), bins = max(T)-min(T), color = 'blue',edgecolor = 'grey')
plt.title("Histogramme de répartition \n des indices de coupe sur 5000 simulations :")
```

Code permettant de modéliser un coup de mélange à l'américaine :
Retour au document.

```
''' DEUXIEME PARTIE DU MELANGE A L'AMERICAINE  '''
def melange(T1,T2):
    nouveau_paquet=[]
    while T1!=[] or T2!=[]:
        p=len(T1)/(len(T1)+len(T2))
        if rd.binomial(1,p)==1:
            nouveau_paquet=[T1[len(T1)-1]]+nouveau_paquet
            T1=T1[0:len(T1)-1]
        else:
            nouveau_paquet=[T2[len(T2)-1]]+nouveau_paquet
            T2=T2[0:len(T2)-1]
    return nouveau_paquet

############################################################
'''  MELANGE A L'AMERICAINE   '''
def riffle_shuffle(T):
    i=couper_paquet(len(T),1/2)
    T1=T[0:i]
    T2=T[i:len(T)+1]
    Tnew=melange(T1,T2)
    return Tnew
```

Code permettant d'afficher les états du paquet rangé après 1,...,7 mélanges à l'américaine successifs :
Retour au document.

```
'''ON AFFICHE LE JEUX DE CARTE APRES 7 MELANGES A L'AMERICAINE SUCCESSIFS  '''
T=jeu_de_cartes
for i in range(7):
    T=riffle_shuffle(T)
    plt.figure()
    plt.scatter(T,indice_du_paquet,marker="*")
    plt.axis([0,len(jeu_de_cartes)+1,0,len(indice_du_paquet)+1])
    plt.xlabel('x = i-ème carte')
    plt.ylabel('y = j-ème indice')
    plt.grid()
    plt.show()
```

7 Bibliographie

Références

[1] D. ALDOUS and P. DIACONIS. *Shuffling cards and stopping times*. 1986.

[2] *Agrégation externe de mathématiques - Épreuve de modélisation, option A : Probabilités et Statistiques* 2009.

[3] Dean KATSAROS and Yaping YUAN *Card-Shuffling Analysis* May 11, 2018.

[4] Christina LI, Yuxin XIE, William YUE *Markov Chains and Card Shuffling* November 29, 2020

[5] John SYLVESTER *Lecture 4 : Card Shuffling and Covertime* 2019.

[6] Philip LIANG *Finite Markov Chains and the Top-to-Random Shuffle* University of Chicago, 2013.

[7] Régine MARCHAND *Chaînes de Markov à espace d'état dénombrables* Université de Lorraine, 2022.

[8] Justin GUO *Card Shuffling* 2013.

[9] Camilla YANG & Benjamin ROUSSEAU *TPE : Mélanger un jeu de cartes* 2018.