Université de Montréal

Échantillonnage des distributions continues non uniformes en précision arbitraire et protocole pour l'échantillonnage exact distribué des distributions discrètes quantiques

par
Claude Gravel

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

Mars, 2015

RÉSUMÉ

La thèse est divisée principalement en deux parties. La première partie regroupe les chapitres 2 et 3. La deuxième partie regroupe les chapitres 4 et 5. La première partie concerne l'échantillonnage de distributions continues non uniformes garantissant un niveau fixe de précision. Knuth et Yao démontrèrent en 1976 comment échantillonner exactement n'importe quelle distribution discrète en n'ayant recours qu'à une source de bits non biaisés indépendants et identiquement distribués. La première partie de cette thèse généralise en quelque sorte la théorie de Knuth et Yao aux distributions continues non uniformes, une fois la précision fixée. Une borne inférieure ainsi que des bornes supérieures pour des algorithmes génériques comme l'inversion et la discrétisation figurent parmi les résultats de cette première partie. De plus, une nouvelle preuve simple du résultat principal de l'article original de Knuth et Yao figure parmi les résultats de cette thèse. La deuxième partie concerne la résolution d'un problème en théorie de la complexité de la communication, un problème qui naquit avec l'avènement de l'informatique quantique. Étant donné une distribution discrète paramétrée par un vecteur réel de dimension N et un réseau de N ordinateurs ayant accès à une source de bits non biaisés indépendants et identiquement distribués où chaque ordinateur possède un et un seul des N paramètres, un protocole distribué est établi afin d'échantillonner exactement ladite distribution.

Mots clef : Échantillonnage, distribution, inversion, partitionnement, acceptation et rejet, entropie, complexité de communication.

ABSTRACT

The thesis is divided mainly into two parts. Chapters 2 and 3 contain the first part. Chapters 4 and 5 contain the second part. The first part is about sampling non uniform continuous distributions with a given level of precision. Knuth and Yao showed in 1976 how to sample exactly any discrete distribution using a source of unbiased identically and independently distributed bits. The first part of this thesis extends the theory of Knuth and Yao to non uniform continuous distributions once the precision is fixed. A lower bound and upper bounds for generic algorithms based on discretization or inversion are given as well. In addition, a new simple proof of the original result of Knuth and Yao is given here. The second part is about the solution of a problem in communication complexity that originally appeared within the field of quantum information science. Given a network of N computers with a server capable of generating random unbiased bits and a parametric discrete distribution with a vector of N real parameters where each computer owns one and only one parameter, a protocol to sample exactly the distribution in a distributed manner is given here.

Keywords: Sampling, distribution, inversion, discretization, von Neumann rejection, entropy, communication complexity.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ANNEXES

NOTATION

$\emptyset$	ensemble vide
$\mathbb{N}$	ensemble des nombres naturels
$\mathbb{Z}$	ensemble des nombres entiers
$\mathbb{Q}$	ensemble des nombres rationnels
$\mathbb{R}$	ensemble des nombres réels
$\mathbb{C}$	ensemble des nombres complexes
$\imath$	$\imath = \sqrt{-1}$
$\cap$	intersection
$\cup$	union
$\sum$	somme
$\prod$	produit
$*$	convolution (de deux distributions)
$\int$	intégrale
$(a_1, a_2, \ldots)$	suite de nombres
$(a_i)_{i \in I}$	suite de nombres indicés dans l'ensemble I
$(A)_{ij}$	entrée à la i^{e} rangée et à la j^{e} colonne d'une matrice A
$\otimes$	produit de Kronecker (de deux matrices)
$\odot$	produit d'Hadamard (de deux matrices de même taille)
F, G	généralement des distributions (fonctions cumulatives)
f, g	généralement des densités
X, Y	généralement des variables (ou vecteurs) aléatoires
U	généralement une variable aléatoire continue et uniforme sur l'intervalle $[0, 1]$ ou une transformation unitaire
$\mathbb{1}$	fonction indicatrice
p	un paramètre (loi de Bernoulli), fonction de masse pour la distribution de GHZ, etc selon le contexte
$\mathbf{P}\{E\}$	probabilité d'un événement E, $\mathbf{P}$ est une lettre générique pour une mesure de probabilité

$\mathbf{E}(X)$	espérance de X
$\mathcal{E}$	entropie
$\lfloor a \rfloor$	partie entière de a (plancher)
$\lceil a \rceil$	partie entière par excès de a (plafond)
$\{a\}$	singleton ou la partie fractionnaire de a selon le contexte
$\|v\|_p$	norme ℓ_p du vecteur v
$x = 0.\overline{x_1 \cdots x_k}$	si $x \in [0,1] \cap \mathbb{Q}$, alors $0.\overline{x_1 \cdots x_k} = \sum_{j=1}^{k} \frac{x_j}{2^j} \sum_{\ell=0}^{\infty} \frac{1}{2^{k\ell}}$

REMERCIEMENTS

Mes remerciements vont à beaucoup de personnes, mais je ne mentionne que ceux qui sont directement impliqués : Gilles Brassard (directeur), Luc Devroye (codirecteur), Luc Gravel et Guy Bertrand. Sans le financement de Gilles Brassard, je n'aurais pas pu compléter mon doctorat. Tout ce que j'ai appris de l'informatique quantique lui est essentiellement dû et c'est avec lui que j'ai pu approfondir et découvrir le sujet. J'ai grandement apprécié sa flexibilité et la confiance qu'il m'a accordée à me laisser travailler dans ce que j'aime. Sans la générosité en temps et en idées ainsi que la patience de Luc Devroye, je n'aurais pas pu écrire cette thèse. Il y a beaucoup de méthodes probabilistes dans ma thèse et il m'a inspiré dans ce domaine. Il a aussi mis de l'ordre dans mes idées et il m'a permis de les rendre à maturité, sans quoi je n'aurais pas pu compléter mon doctorat. Je remercie particulièrement Pierre McKenzie, de l'Université de Montréal, pour ses suggestions, commentaires et corrections. Je remercie mon frère Luc pour sa lecture attentive et ses suggestions. Je remercie mon ami Guy Bertrand, premier conseiller linguistique à Radio-Canada, pour ses conseils, qui m'ont facilité l'écriture et la correction de cette thèse.

INTRODUCTION

Dans cette thèse, il est question d'échantillonner des distributions de probabilité ou, en d'autres termes, de générer des variables aléatoires. Que voulons-nous dire par générer une variable aléatoire ? Par exemple, lançons une pièce de monnaie et supposons que pile apparaît avec probabilité p. Nous pouvons réexprimer le problème en associant à pile la valeur 1 et à face la valeur 0. Ainsi la probabilité d'obtenir la valeur 1 est p. Si nous disposons d'une variable aléatoire U continue et uniforme sur l'intervalle $[0, 1]$, alors nous n'avons qu'à comparer U et p pour générer aléatoirement une pile ou une face. En effet, si $U < p$, alors nous retournons 1, sinon 0. L'exemple de pile ou face suppose que U nous est donnée par un moyen quelconque. L'hypothèse qu'il est possible physiquement d'obtenir U est irréaliste malgré le fait que cette hypothèse soit commune dans la littérature. Nous pouvons voir U comme une suite de bits non biaisés i.i.d. (*i*ndépendants et *i*dentiquement *d*istribués) en examinant le développement binaire de $U = 0.U_1U_2\ldots$ De nos jours, il est possible physiquement de générer des bits non biaisés i.i.d. et, par conséquent, il est intéressant de se demander combien de bits nous devons espérer lire de U ou générer afin de décider d'une pile ou d'une face. Pour générer une pile ou une face en n'utilisant que les bits $U_1, U_2, \ldots$, à partir de l'expansion binaire de $p = 0.p_1p_2\ldots$, nous comparons successivement U_i et p_i et nous nous arrêtons dès que $U_i \neq p_i$. Appelons T le plus petit nombre naturel tel que $U_T \neq p_T$. Si $U_T < p_T$, alors nous retournons 1 (pile) sinon 0 (face). Notons que T est une variable aléatoire, qui T représente le nombre de bits qu'il faut lire de U ou tout simplement le nombre de bits aléatoires non biaisés i.i.d. à générer par un moyen quelconque qui nous est donné. Dorénavant, nous allons utiliser l'expression « bits aléatoires » pour désigner des bits aléatoires non biaisées i.i.d. sauf mention du contraire. Dans ce cas précis qu'est notre exemple de pile ou face, T suit une loi géométrique de paramètre $1/2$ et il faut donc en espérer 2 bits. Nous nous intéressons à l'espérance $\mathbf{E}(T)$ tout au long de cette thèse dans des situations autres que le jeu de pile ou face.

Cette thèse se divise principalement en deux grandes parties. La première partie

regroupe les chapitres 2 et 3 et la deuxième partie regroupe les chapitres 4 et 5. Le chapitre 1 est essentiel à tous les chapitres subséquents. Les chapitres 2 et 3 concernent l'échantillonnage de distributions discrètes et continues. Les chapitres 4 et 5 concernent également l'échantillonnage de distributions discrètes, celles-là issues de la mécanique quantique, dans un contexte relatif à la complexité de la communication et utiles en informatique quantique.

Le chapitre 1 aborde différentes notions utilisées dans cette thèse comme les distributions de probabilité, les variables aléatoires, l'entropie, l'entropie différentielle, la distance entre deux distributions ou deux variables aléatoires sans entrer dans les détails de la théorie des probabilités et de la mesure. Au chapitre 1, nous spécifions également le concept d'algorithme d'échantillonnage d'une distribution ou de génération d'une variable aléatoire. De plus, nous complétons le chapitre 1 en précisant les opérations que nous jugeons « gratuites », opérations qui ne sont pas comptabilisées dans nos calculs de la complexité de nos algorithmes, et l'opération que nous jugeons « coûteuse » qui est comptabilisée dans nos calculs de complexité. L'opération de générer un bit aléatoire et, par conséquent, le nombre de fois que nous devons générer un bit aléatoire est la quantité qui nous importe principalement. Aux chapitres 4 et 5, nous calculons également le nombre de bits relatifs à la description des paramètres des distributions, ces bits devant circuler sur un réseau de communication.

Le chapitre 2 aborde l'échantillonnage exact des distributions discrètes qui a été en grande partie résolu par Knuth et Yao [30] en 1976. L'article [30] contient un algorithme générique optimal (à deux bits près) quant au nombre espéré de bits requis. Nous donnons une nouvelle preuve plus simple de ce résultat majeur contenu dans [30]. Nous expliquons aussi une méthode simple due à Han et Hoshi [25] en 1997 pour échantillonner les distributions discrètes et nous donnons également une nouvelle preuve plus simple de sa complexité espérée. De plus, le cas important de la génération d'une variable uniforme discrète prenant ses valeurs dans $\{0, \ldots, n-1\}$ pour $n > 2$ est analysé. Notons que la thèse de doctorat de Lumbroso [33] en 2012 contient une analyse asymptotique très fine de l'espérance du nombre de bits

aléatoires pour générer une variable discrète uniforme. Nous y ajoutons notre part en obtenant la complexité exacte et en la comparant à celle de [33].

Le chapitre 3 aborde l'échantillonnage des distributions continues à un niveau fixe de précision arbitraire spécifié par l'utilisateur avant l'exécution d'un algorithme d'échantillonnage. Nous généralisons en quelque sorte les résultats de Knuth et Yao aux variables aléatoires continues. Le résultat majeur du chapitre 3, et de cette thèse, est un minorant pour l'espérance du nombre de bits aléatoires qu'il faut générer afin d'échantillonner une distribution continue à un niveau fixe de précision arbitraire. La minoration obtenue ne dépend pas de l'algorithme d'échantillonnage et ne dépend que du choix de la norme ainsi que du niveau de précision. C'est la première fois à notre connaissance qu'une telle borne inférieure est établie pour les distributions continues. Nous donnons également des majorants pour la complexité espérée de différentes méthodes comme l'inversion et la méthode de partitionnement. Nous regardons également le cas de certaines distributions continues importantes comme la loi uniforme, la loi exponentielle de moyenne 1 et la loi normale standard. Notons que le cas de loi exponentielle fut traité par Flajolet et Saheb [20] en 1986. Nous montrons que deux méthodes (la méthode de l'inversion et celle de la partition) sur trois sont meilleures en terme du nombre espéré de bits aléatoires que la méthode dans [20].

Le chapitre 4 concerne l'échantillonnage de la distribution quantique GHZ, des noms de Greenberger, Horne et Zeilinger [24]. Expliquons un peu le contexte en quelques lignes. La distribution de GHZ est une distribution discrète paramétrée par un vecteur de réels de dimension N. Les N paramètres sont distribués sur un réseau de type client-serveur se composant de N ordinateurs. Les N ordinateurs ont accès à une source de bits aléatoires. Nous étudions également au chapitre 5 le cas où le serveur est le seul à posséder un générateur de bits aléatoires. Le serveur peut également transmettre de l'information aux $N-1$ clients. En plus de calculer le nombre espéré de bits aléatoires requis par le serveur afin de pouvoir échantillonner la distribution, nous sommes intéressés, tel est le but, par le nombre espéré de bits de communication qui doivent être envoyés par les $N-1$ clients au

serveur et du serveur vers les $N-1$ clients. Ces bits de communications servent à décrire le vecteur des paramètres caractérisant la distribution échantillonnée par le serveur. Les bits des paramètres envoyés au serveur permettent à ce dernier d'en apprendre suffisamment à propos des paramètres décrivant ladite distribution qui de facto permet d'échantillonner *exactement* la distribution en question. Notons que ce problème de communication a des racines lointaines du temps d'Einstein, Podolsky et Rosen en 1935 dans leur célèbre article [17].

Le chapitre 5 généralise le résultat du chapitre 4 à toutes les distributions quantiques discrètes. Nous expliquons ce qu'est une distribution quantique discrète de probabilité au début du chapitre 5 et seules quelques notions simples de l'algèbre linéaire suffiront. Nous nous concentrons d'abord seulement sur la complexité espérée du nombre de bits de communications en supposant que le serveur peut générer des variables uniformes continues. Ensuite nous relaxons l'hypothèse que le serveur possède un générateur de variables aléatoires continues en le remplaçant par un générateur de bits aléatoires tel que nous y aurons été habitués aux chapitres précédents.

CHAPITRE 1

NOTIONS MATHÉMATIQUES, GÉNÉRATION ET MODÈLES INFORMATIQUES

Dans ce chapitre, nous exposons quelques notions de mathématiques au cœur de cette thèse comme les variables aléatoires, les distributions de probabilité et la distance entre deux variables aléatoires. Par la suite, nous éclaircissons la notion de génération d'une variable aléatoire ou de l'échantillonnage d'une distribution et de ce qu'est un algorithme de génération. Enfin, nous complétons ce chapitre par une discussion spécifiant les opérations, qui au sens large, sont considérées gratuites que nous ne comptabilisons pas dans nos calculs de complexités, et l'opération coûteuse dont nous tenons compte dans nos calculs de complexité. L'opération de générer un bit aléatoire est l'opération qui nous importe réellement.

Commençons par décrire les objets mathématiques au centre de cette thèse c'est-à-dire les distributions de probabilité ou les variables aléatoires. Le lecteur peut consulter entre autres Billingsley [4], Feller [18, 19] en ce qui concerne la théorie des probabilités et la théorie de la mesure. Nous n'expliquons aucune notion de la théorie de la mesure comme les espaces mesurables, etc.

Il y a trois grands types de distributions dites pures qui sont les distributions absolument continues, singulières (continues) et singulières (discrètes). Toute distribution de probabilités est une combinaison convexe des trois types purs. Dans cette thèse, nous nous restreignons aux distributions discrètes et absolument continues à l'exception de l'annexe V. Les variables aléatoires sont intrinsèquement liées aux distributions de probabilités et vice-versa. Dans cette thèse, toutes les variables aléatoires sont réelles. Généralement, nous dénotons par une lettre majuscule une variable aléatoire. Si X dénote une variable aléatoire, alors nous écrivons $\mathbf{P}\{X \in A\}$ pour signifier la probabilité que X appartienne à l'ensemble A. La lettre $\mathbf{P}$ est une lettre générique appliquée à l'événement $\{X \in A\}$. Nous pouvons avoir par exemple deux variables aléatoires X et Y et écrire $\mathbf{P}\{X \in A\}$, $\mathbf{P}\{Y \in B\}$ ou

encore $\mathbf{P}\{X \in A, Y \in B\}$.

Définition 1 (variable aléatoire absolument continue)**.** Une variable aléatoire X est absolument continue si pour tout ensemble de nombres réels A de mesure de Lebesgue non nulle, il existe une fonction intégrable f telle que

$$\mathbf{P}\{X \in A\} = \int_A f(x)dx.$$

Nous disons que la densité de X est f.

Définition 2 (variable aléatoire singulière (continue))**.** Une variable aléatoire X est singulière s'il existe un ensemble A de nombres réels de mesure de Lebesgue nulle tel que $\mathbf{P}\{X \in A\} = 1$.

Si $A = (-\infty, x]$ et $x \in \mathbb{R}$, alors la fonction $F : \mathbb{R} \to [0, 1]$ telle que $F(x) = \mathbf{P}\{X \in A\}$ est appelée la fonction de répartition. L'événement $\{X \in A\}$ est identique à $\{X \leq x\}$ et, par conséquent, nous écrivons $F(x) = \mathbf{P}\{X \leq x\}$. Si $X = (X_1, \ldots, X_d) \in \mathbb{R}^d$, que $A = (-\infty, x_1] \times \ldots \times (-\infty, x_d]$ et $(x_1, \ldots, x_d) \in \mathbb{R}^d$, alors nous écrivons $F(x_1, \ldots, x_d) = \mathbf{P}\{X \in A\} = \mathbf{P}\{\cap_{i=1}^d [X_i \leq x_i]\}$.

Définition 3 (variable aléatoire discrète)**.** Une variable aléatoire X est discrète s'il existe un ensemble dénombrable A de nombres réels tel que $\mathbf{P}\{X \in A\} = 1$.

Définition 4 (indépendance)**.** Des variables aléatoires X_i de distributions F_i avec $i \in \{1, \ldots, n\}$ sont indépendantes si pour tous les ensembles A_i

$$\mathbf{P}\left\{ \bigcap_{i=1}^{n} \{X_i \in A_i\} \right\} = \prod_{i=1}^{n} \mathbf{P}\{X_i \in A_i\}.$$

Puisque l'égalité de la définition 4 doit être vérifiée pour tous les ensembles A_i, nous pouvons entre autres prendre $A_i = (-\infty, x_i]$ pour $i \in \{1, \ldots, n\}$. Si F dénote la fonction de répartition du vecteur $(X_1, \ldots, X_n)$, c'est-à-dire que $F(x_1, \ldots, x_n) = \mathbf{P}\big\{ \bigcap_{i=1}^n \{X_i \in A_i\} \big\}$, alors

$$F(x_1, \ldots, x_n) = \prod_{i=1}^{n} F_i(x_i). \tag{1.1}$$

Nous ponvons démontrer que si la ligne (1.1) est vraie pour toutes les valeurs x_i, alors les variables X_i sont indépendantes. L'indépendance est particulièrement facile à vérifier dans les cas discret et absolument continu. Dans le cas discret, il suffit de vérifier pour tout $(x_1, \ldots, x_n)$ que

$$\mathbf{P}\left\{ \bigcap_{i=1}^{n} \{X_i = x_i\} \right\} = \prod_{i=1}^{n} \mathbf{P}\{X_i = x_i\}.$$

Dans le cas continu, si f_i dénote la densité de X_i, alors il faut vérifier pour tout $(x_1, \ldots, x_n)$ que

$$f(x_1, \ldots, x_n) = \prod_{i=1}^{n} f_i(x_i).$$

Définition 5 (espérance d'une variable aléatoire)**.** Si $X \in \mathbb{R}$ est une variable aléatoire, alors nous dénotons par $\mathbf{E}(X)$ son espérance. Si X est discrète et prend ses valeurs dans un ensemble dénombrable A, alors

$$\mathbf{E}(X) = \sum_{x \in A} x\mathbf{P}\{X = x\}.$$

Si X est absolument continue de densité f et prenant ses valeurs dans un ensemble de longueur non nulle A de réels, alors

$$\mathbf{E}(X) = \int_A xf(x)dx.$$

Pous plus d'information quant aux types de convergence de suites aléatoires et au théorème de convergence dominée de Lebesgue, veuillez consulter l'annexe I.

Une quantité très importante qui apparaît à plusieurs endroits dans cette thèse est l'entropie d'une distribution discrète ou de la variable aléatoire que représente cette dernière.

Définition 6 (entropie d'une distribution discrète)**.** Si la suite $(p_i)_{i \in \mathbb{Z}}$ définit une distribution discrète, alors l'entropie binaire de $(p_i)_{i \in \mathbb{Z}}$ est dénotée par $\mathcal{E}\big\{(p_i)_{i \in \mathbb{Z}}\big\}$

et définie par

$$\mathcal{E}\big\{(p_i)_{i\in\mathbb{Z}}\big\} = \sum_{i\in\mathbb{Z}} p_i \log_2\left(\frac{1}{p_i}\right).$$

Une propriété importante de l'entropie dans le cas discret que nous utilisons au chapitre 3 est celle de l'emboîtement (« nesting property »). Étant donné un vecteur de probabilités $p = (p_1, \ldots, p_i, \ldots, p_n)$ et n possiblement infini ainsi que $\alpha \in (0,1)$, considérons le vecteur de probabilités $p' = (p_1, \ldots, \alpha p_i, (1-\alpha)p_i, \ldots, p_n)$. La propriété d'emboîtement affirme que l'entropie de p' est supérieure ou égale à p tout simplement parce que

$$\begin{aligned}
-\alpha p_i \log_2(\alpha p_i) - (1-\alpha)p_i \log_2((1-\alpha)p_i) &= -p_i \log_2(p_i) \\
&\quad + p_i(-\alpha \log_2(\alpha) - (1-\alpha)\log(1-\alpha)) \\
&\geq -p_i \log_2(p_i).
\end{aligned}$$

Si h est une fonction mesurable (*cf.* Billingsley [4] pour les notions de mesure) dont le domaine coïncide avec l'ensemble des valeurs que peut prendre une variable aléatoire continue X, alors l'espérance de h est la quantité

$$\mathbf{E}(h(X)) = \int_A h(x)f(x)dx \quad \text{si cette dernière existe.}$$

Lorsque X est continue de densité f, une quantité d'intérêt est l'espérance de $\log_2\left(\frac{1}{f}\right)$ et cette quantité s'appelle l'entropie différentielle.

Définition 7 (entropie différentielle)**.** L'entropie différentielle d'une variable aléatoire $X \in \mathbb{R}$ de densité f, dénotée par $\mathcal{E}(f)$, est la quantité

$$\mathcal{E}(f) = \int_A f(t) \log_2\left(\frac{1}{f(t)}\right) dt.$$

Si X est une variable aléatoire de densité f, alors parfois nous écrivons $\mathcal{E}(X)$ au lieu de $\mathcal{E}(f)$.

Lorsque X est une variable aléatoire continue de densité f définie sur $A \subseteq \mathbb{R}$, il sera utile de partitionner A de sorte à obtenir une discrétisation du support

de f. Si $\mathcal{A} = \{A_i\}_{i\in\mathbb{Z}}$ est une partition de A c'est-à-dire que les sous-ensembles A_i sont disjoints et recouvrent entièrement A, alors les valeurs $\mathbf{P}\{X \in A_i\}$ pour $i \in \mathbb{Z}$ forment une distribution discrète. L'entropie de partition de X est la quantité dénotée par $\mathcal{E}_{\mathcal{A}}(X)$ et définie par

$$\mathcal{E}_{\mathcal{A}}(X) = \sum_{i\in\mathbb{Z}} \mathbf{P}\{X \in A_i\} \log_2 \left(\frac{1}{\mathbf{P}\{X \in A_i\}} \right).$$

Mentionnons au passage un théorème dû à Csiszár reliant l'entropie d'une grille et l'entropie différentielle. Dans le théorème qui suit, nous écrivons $i = (i_1, \ldots, i_d) \in \mathbb{Z}^d$ pour dénoter un vecteur d'entiers.

Théorème 1 (Csiszár, 1961)**.** *Soit $X = (X_1, \ldots, X_d) \in \mathbb{R}^d$ un vecteur aléatoire de densité f, $\epsilon > 0$ et la grille*

$$\bigcup_{i\in\mathbb{Z}^d} \big[i_1\epsilon, (i_1+1)\epsilon\big) \times \ldots \times \big[i_d\epsilon, (i_d+1)\epsilon\big) \stackrel{def}{=} \mathcal{A}_\epsilon^\star.$$

Si l'entropie de la distribution discrète des parties entières $(\lfloor X_1 \rfloor, \ldots, \lfloor X_d \rfloor)$ est finie, alors

$$\lim_{\epsilon\to 0} \left(\mathcal{E}_{\mathcal{A}_\epsilon^\star}(X) - \log_2 \left(\frac{1}{\epsilon^d} \right) \right) = \begin{cases} \mathcal{E}(f) & \textit{si } X \textit{ est absolument continue,} \\ -\infty & \textit{si } X \textit{ est singulière.} \end{cases}$$

Pour plus d'information sur la théorie asymptotique de la discrétisation des distributions de probabilités, consulter Rényi [39], Csiszár [14, 15], Linder et Zeger [32] et Cover et Thomas [13]. Nous pouvons créer une distribution absolument continue dont l'entropie différentielle est finie et dont l'entropie de partition est $+\infty$. L'annexe III montre un exemple différent de celui contenu dans Rényi [39].

Le théorème suivant concerne l'inverse F^{-1} d'une fonction de répartition F et, pour la preuve, consulter Billingsley [4].

Théorème 2 (inverse d'une distribution)**.** *Soit F une fonction de répartition d'une*

variable aléatoire $X \in \mathbb{R}$ continue définie sur $\mathbb{R}$ ayant son inverse F^{-1} défini par

$$F^{-1}(u) = \inf\{x \ : \ F(x) = u\}.$$

Si U est uniforme sur $[0,1]$ alors la variable aléatoire $F^{-1}(U)$ est de distribution F. Si X est de distribution F, alors $F(X)$ est uniformément distribuée sur $[0,1]$.

Pour donner un avant-goût de l'utilité du théorème 2, regardons le cas de la variable aléatoire exponentielle X de moyenne 1 et de distribution $F(x) = 1 - e^{-x}$. Pour cette loi, nous avons $F^{-1}(u) = -\log(1-u)$ et puisque U est distribuée comme $1 - U$, alors nous avons $F^{-1}(u) = -\log(u)$. Par conséquent, si nous disposions d'une variable continue uniforme sur l'intervalle $[0,1]$, alors nous pourrions générer $X = -\log(U)$. Si nous ne disposions que des bits aléatoires ou, de façon équivalente, si nous ne pouvions que générer séquentiellement les bits aléatoires de l'expansion binaire de U, alors nous serions intéressés à connaître le nombre de bits de U que nous devons générer afin que la distance de Wasserstein, définie plus loin, entre la sortie de notre algorithme et la variable cible X soit au plus un certain $\epsilon > 0$. Le paramètre ϵ est la précision fixée par l'utilisateur. Nous analysons entre autres ce cas au chapitre 3 ainsi que celui de la loi normale.

Avant de définir le problème de générer une variable aléatoire, nous nous proposons de regarder comment mesurer la distance entre deux variables. Dénotons le presque majorant de X par $\operatorname{ess\,sup} X$. En pratique, écrire $\operatorname{ess\,sup} X$ ou $\sup X$ signifie la même chose sauf sur les ensembles de mesure nulle. Pour $d \geq 1$, $p \geq 1$ et $v \in \mathbb{R}^d$, la norme ℓ_p du vecteur v, dénoté $\|v\|_p$, est donnée par $\|v\|_p = \left(\sum_{i=1}^{d} |v_i|^p\right)^{\frac{1}{p}}$. De plus, utilisons la lettre générique $\mathcal{L}$ appliquée à X pour signifier la distribution de X. Si nous avons deux variables aléatoires X et Y de lois respectives $\mathcal{L}(X)$ et $\mathcal{L}(Y)$ ayant le même support de définition, alors la quantité $\operatorname{ess\,sup} \|X - Y\|_p$ est la distance selon la norme ℓ_p entre X et Y.

Définition 8 (distance de Wasserstein)**.** Soit $I \subseteq \mathbf{R}^d$. Soit $X \in I$ et $Y \in I$ deux variables aléatoires de lois respectives F et G. Soit $\mathcal{M}$ la classe des distributions conjointes sur $\mathbb{R}^d \times \mathbb{R}^d$ et $p \geq 1$, alors la distance de Wasserstein selon la norme ℓ_p

entre F et G, dénotée $W_p(F,G)$, est

$$W_p(F,G) = \inf_{H\in\mathcal{M}} \Big\{ \operatorname{ess\,sup} \|X-Y\|_p \;\; : \;\; H(x_1,\ldots,x_d,\infty,\ldots,\infty) = F \text{ et } H(\infty,\ldots,\infty,x_{d+1},\ldots,x_{2d}) = G \Big\} \stackrel{\text{def}}{=} \operatorname{dist}(X,Y).$$

Remarquons qu'en dimension 1, toutes les normes ℓ_p sont identiques et, par conséquent, il n'y a qu'une distance de Wasserstein en dimension 1. Pour plus d'information concernant la distance de Wasserstein, consulter Rachev et Rüschendorf [38]. L'annexe IV fournit également un nouveau résultat concernant la distance de Wasserstein entre deux variables aléatoires et leurs inverses respectifs. Étant donné deux variables aléatoires X (« target »), Y (« output ») et $\epsilon > 0$, si $\operatorname{dist}(X,Y) < \epsilon$, alors X et Y sont couplées de sorte que $\operatorname{ess\,sup} \|X-Y\| \leq \epsilon$ c'est-à-dire que presque sûrement $\|X-Y\| \leq \epsilon$.

Remarque 1 (variation totale entre deux distributions (variables)). La variation totale entre deux variables aléatoires n'est d'aucune utilité pour nos travaux, car si X est continue et que Y est discrète, alors $\sup_A \|\mathbf{P}\{X \in A\} - \mathbf{P}\{Y \in A\}\| = 1$.

Nous sommes maintenant en mesure de préciser ce que générer une variable aléatoire signifie. Allons-y avec l'échantillonnage de distribution discrète et, pour cela, rappelons-nous notre exemple d'introduction qui consistait à simuler une variable binaire biaisée, nous voulons produire une instance d'une variable aléatoire X de distribution F étant donnée une suite $(B_1, B_2, \ldots,)$. Toute distribution discrète peut être simulée exactement, car un nombre fini de bits aléatoires suffisent à décrire n'importe quelle réalisation d'une variable discrète.

Définition 9 (algorithme de génération exacte pour variable discrète)**.** Étant donné une variable aléatoire discrète X représentée par un vecteur de probabilités $(p_1,\ldots,p_n)$ avec n possiblement infini, tout algorithme utilisant des bits aléatoires $B_1, B_2, \ldots, B_T$ et retournant $X = i$ avec probabilité p_i est un algorithme de génération exacte pour la variable discrète X.

Afin de préciser ce que signifie générer une instance d'une variable continue, nous devons incorporer la notion de précision, car il va de soi qu'il est impossible de générer exactement une variable continue puisqu'il faudrait retourner un nombre infini de bits pour décrire l'information d'une réalisation en général. Si nous dénotons par Y la sortie d'un algorithme et par G la distribution de Y, alors, pour $p \geq 1$ choisi selon nos besoins, la distance $\text{dist}(X, Y) = W_p(F, G)$ est une quantité qui influence le nombre de bits B_i à utiliser. Ainsi en fixant une valeur $\epsilon > 0$ appelée la précision, nous désirons des algorithmes produisant Y de sorte que $W_p(F, G) < \epsilon$. Notons que la sortie Y est discrète car chaque fonction de $B_1, B_2, \ldots, B_T$ est discrète. Comme dans notre exemple d'introduction, nous dénotons par T le nombre de bits aléatoires requis. Avec la notation des derniers paragraphes, nous avons plus formellement la définition suivante d'un algorithme de génération.

Définition 10 (algorithme de génération pour variable continue)**.** Étant donnés une variable (ou vecteur) aléatoire X continue de distribution F et un paramètre de précision $\epsilon > 0$, tout algorithme utilisant des bits aléatoires $B_1, B_2, \ldots, B_T$ et retournant $Y = y$ de distribution G telle que $\text{dist}(X, Y) < \epsilon$ est un algorithme de génération pour la variable continue X.

Remarque 2. Avec la notation qui précède, nous sommes intéressés aux algorithmes de génération pour lesquels la quantité $\mathbf{E}(T)$ est bornée. Dans le cas des variables continues, T dépend de la précision ϵ.

Définition 11 (algorithme optimal = espérance minimale)**.** Si T est le nombre de bits aléatoires utilisés par un algorithme d'échantillonnage, alors l'algorithme est dit optimal s'il minimise l'espérance $\mathbf{E}(T)$.

Précisons davantage ce qui nous pouvons faire « gratuitement » et ce que nous ne pouvons pas faire « gratuitement ». Générer un bit aléatoire ou lire un bit aléatoire à partir d'une suite donnée de bits non biaisés i.i.d. sont les seules opérations (équivalentes) que nous ne pouvons pas faire gratuitement. Chaque bit aléatoire utilisé est comptabilisé et seule la quantité $\mathbf{E}(T)$ nous intéresse. Afin de générer

une instance d'une variable aléatoire X, un algorithme de génération doit posséder la description de la distribution de F. Nous ne tenons pas compte de l'espace relatif à l'information concernant les instances de X ni du temps relatif aux manipulations concernant cette information. Plus précisément, nous utilisons des oracles afin de calculer ou d'obtenir de l'information sur la distribution de la variable X. Il va de soi que les lois de répartions, des fonctions de masse ou des densités avec lesquelles nous travaillons sont calculables au sens informatique du terme et, par conséquent, l'utilisation d'oracle afin d'obtenir de l'information relative à ces dernières est justifiée du point de vue de la calculabilité. Pour une distribution discrète représentée par un vecteur de probabilités $(p_1, \ldots, p_n)$ avec n possiblement infini, deux types d'oracles tabulaires peuvent être utilisés. Le premier type d'oracle prend $i \in \{1, \ldots, n\}$ et $j \in \mathbb{N}$ comme arguments et retourne le j^{e} bit de p_i. Plus spécifiquement, si $p_i = \sum_{j=1}^{\infty} p_{i,j} 2^{-j}$, alors l'oracle prend deux arguments que sont i et j et retourne $p_{i,j} \in \{0, 1\}$. Ce type d'oracle est implicitement utilisé dans les travaux de Knuth et Yao que nous présentons au chapitre 2. Le deuxième type d'oracle prend un nombre naturel i et retourne $\sum_{j=1}^{i} p_j$ et ce type d'oracle est utilisé pour implanter l'algorithme de Han et Hoshi au chapitre 2. Pour une distribution continue F, trois types d'oracle sont utilisés. Dans le cas continu, le premier type d'oracle est celui calculant $F(x)$ pour $x \in \mathbb{R}$. Pour une variable continue X de distribution F, si A est un hypercube, alors un oracle calculant F peut être utilisé afin de calculer $\mathbf{P}\{X \in A\}$, ce qui nous permet par exemple de partitionner le domaine de F. Un autre type d'oracle utile dans le cas continu est celui qui, en vertu du Théorème 2, calcule $F^{-1}(u)$ pour $u \in [0, 1]$. Si X est absolument continue de densité f définie sur $I \subseteq \mathbb{R}$, alors un oracle utile est celui calculant $f(x)$ avec $x \in I$. De plus, nous faisons l'hypothèse que le coût du calcul d'une fonction mathématique de base est constant et ce peu importe le niveau de précision requis. Les fonctions mathématiques de base utilisées sont les puissances réelles d'un nombre, les fonctions trigonométriques et exponentielles ainsi que leurs inverses. De plus, nous ne tenons pas compte de l'espace ou du temps que prendrait l'implantation des oracles en pratique. Encore une fois, tout ce qui nous préoccupe est le nombre

de bits aléatoires non biaisés indépendants et générés identiquement par la source et, dans le cas des chapitres 4 et 5, le nombre de bits de communication circulant sur le réseau.

À l'annexe I, les définitions des différents types de convergence sont données en plus de l'énoncé du théorème de convergence dominée de Lebesgue.

CHAPITRE 2

ÉCHANTILLONNAGE EXACT D'UNE DISTRIBUTION DISCRÈTE

Dans ce chapitre, nous exposons premièrement la théorie contenue dans l'article de Knuth et Yao [30] de 1976. Dans [30], le problème de générer exactement une variable aléatoire discrète y est résolu de façon optimale, c'est-à-dire en minimisant l'espérance du nombre de bits aléatoires. D'ailleurs, ils furent les premiers à introduire la terminologie « Random Bit Model » qui réfère à la génération de variables aléatoires en n'utilisant que des bits aléatoires. Ils introduisirent également le terme ou concept de « DDG tree » pour « Discrete Data Generator tree » afin de représenter l'exécution d'un algorithme d'échantillonnage. Nous donnons entre autres une autre preuve plus simple du résultat principal de [30], résultat qui établit des bornes inférieure et supérieure du nombre espéré de bits requis par un algorithme d'échantillonnage. Deuxièmement, nous expliquons une partie de la théorie contenue dans l'article de Han et Hoshi [25] de 1997. Han et Hoshi implantèrent la méthode de l'inversion pour les distributions discrètes et s'intéressèrent également à des sources produisant des bits biaisés i.i.d. ou corrélés (source Markovienne). L'inversion d'une loi discrète engendre également un arbre DDG comme nous le verrons. Nous donnons aussi une autre preuve plus simple du résultat de Han et Hoshi lorsque la source produit des bits non biaisés i.i.d. Enfin, nous nous intéressons à la complexité de produire une instance d'une variable uniforme discrète ayant un nombre d'atomes supérieur à deux. Notons que dans la thèse de Lumbroso [33] en 2012, nous y trouvons une analyse asymptotique très fine de la complexité de générer une variable discrète uniforme et nous comparons notre résultat exact au résultat dans [33]. Tous les arbres dans ce chapitre seront représentés du haut (racine) vers le bas. Il est implicitement sous-entendu que chaque arête gauche est étiquetée par 0 et chaque arête droite par 1. Une partie de ce chapitre se retrouve dans Devroye et Gravel [41].

Nous voulons échantillonner un vecteur de probabilités $(p_1, p_2, \ldots, p_n)$ avec la possibilité que n soit infini. Afin d'expliciter la théorie de Knuth et Yao, regardons quelques exemples numériques lorsque $n = 2$ et $n = 3$. Prenons $n = 2$ qui correspond à une loi de Bernoulli et

$$\begin{aligned} p_A &= \frac{2}{7} && (2.1)\\ &= (0.\overline{010})_2 \\ &= p_1, \\ p_B &= \frac{5}{7} && (2.2)\\ &= (0.\overline{101})_2 \\ &= p_2 \\ &= 1 - p_1. && (2.3) \end{aligned}$$

Nous prenons des lettres pour les symboles afin d'éviter temporairement certains conflits de notation. Si la suite de bits aléatoires débute par $0U_2U_3U_4\cdots$, alors nous ne pouvons pas nous arrêter. Si la suite de bits aléatoires est $1U_2U_3U_4\cdots$, nous retournons le symbole B. Si la suite de bits aléatoires est $01U_3U_4\cdots$, alors nous retournons le symbole A. Si la suite de bits aléatoire est $001U_4\ldots$, alors nous retournons le symbole B. Nous pouvons continuer ainsi ad vitam aeternam. Nous avons donc l'automate à la figure 2.1 qui contient une boucle puisque les probabilités p_A et p_B sont rationnelles.

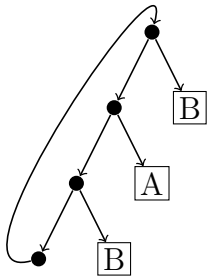

Figure 2.1 – 1er exemple - Knuth et Yao

L'élimination de la boucle engendre un arbre de hauteur infinie. Remarquons

que chaque arête droite issue d'un noeud interne, implicitement étiquetée par 1 selon notre convention au début de ce chapitre, mène à une feuille de l'arbre et donc au retour d'un symbole. Chaque arête gauche issue d'un noeud interne conduit à un autre noeud interne et donc aucun symbole n'est retourné. À chaque niveau, il y a au plus un symbole de chaque type A ou B qui peut être retourné.

Allons-y avec un autre exemple de distribution discrète sur trois atomes ($n = 3$) et prenons les symboles A, B, et C avec les probabilités respectives suivantes :

$$p_A = \frac{1}{\pi} \tag{2.4}$$
$$= (0.010100010111110\ldots)_2 = \sum_{j=1}^{\infty} p_{A,j} 2^{-j},$$

$$p_B = \frac{1}{e} \tag{2.5}$$
$$= (0.010111100010110\ldots)_2 = \sum_{j=1}^{\infty} p_{B,j} 2^{-j},$$

$$p_C = 1 - p_A - p_B \tag{2.6}$$
$$= (0.010100000101010\ldots)_2 = \sum_{j=1}^{\infty} p_{C,j} 2^{-j}.$$

L'automate de la figure 2.2 échantillonne la distribution (p_A, p_B, p_C).

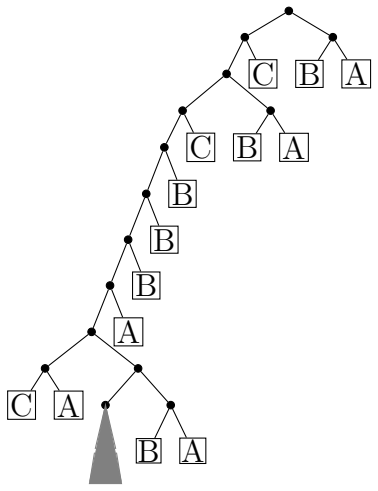

Figure 2.2 – 2e exemple - Knuth et Yao

Encore une fois, remarquons qu'à chaque niveau, il y a au plus un symbole

de chaque type pouvant être retourné. Le sous-arbre en gris représente le fait que l'arbre est infini. Les probabilités des lignes (2.4), (2.5) et (2.6) sont des nombres irrationnels et l'automate n'a aucun cycle. Puisque $p_{A,1} = p_{B,1} = p_{C,1} = 0$, il n'y a pas de symbole retourné au 1er niveau. Puisque $p_{A,2} = p_{B,2} = p_{C,2} = 1$, un symbole de chaque type peut être retourné au 2e niveau avec leur probabilité respective. De façon générale, si $p_{X,i} = 1$ et $X \in \{A, B, C\}$, alors l'automate peut retourner X au i^{e} niveau.

Nous voyons au travers des deux exemples précédents que nous pouvons générer exactement une variable discrète en utilisant un automate probabiliste ayant une représentation arboricole. Cette représentation arboricole, dans le contexte spécifique de la génération, fut baptisée « DDG tree » par Knuth et Yao en 1976. À ce jour, il existe deux « types » d'arbre DDG. L'arbre DDG de Knuth et Yao est basé sur le vecteur de probabilités ou la fonction de masse. Plus tard, Han et Hoshi [25] en 1997 ont construit un arbre DDG à partir de la fonction de répartition (« cumulative distribution function »).

Définition 12 (arbre DDG - type Knuth et Yao)**.** Étant donné un vecteur de probabilités $(p_1, p_2, \ldots, p_n)$ auquel est associé le vecteur de symboles $(x_1, x_2, \ldots, x_n)$ avec n possiblement infini, un arbre DDG pour $(p_1, p_2, \ldots, p_n)$ contient 0 ou 1 symbole x_i (feuille) pour chaque niveau. Il y a un symbole au m^{e} niveau si 2^{-m} est présent dans le développement binaire de p_i.

L'arbre de Han et Hoshi est très semblable à celui de Knuth et Yao à la différence, comme nous le verrons plus loin, qu'un niveau peut contenir jusqu'à deux symboles x_i.

Knuth et Yao [30] démontrèrent que l'algorithme générique implicitement décrit par un arbre DDG construit à partir de la fonction de masse, dorénavant appelé l'algorithme de Knuth et Yao, est optimal c'est-à-dire que l'espérance du nombre de bits requis est minimale.

Remarque 3 (algorithme de Knuth et Yao en pratique). En pratique, il est très difficile d'implanter l'algorithme optimal pour un vecteur arbitraire $(p_1, \ldots, p_n)$ de

probabilités surtout si n est infini. En effet, considérons l'expansion binaire de p_i c'est-à-dire $p_i = \sum_{j=1}^{\infty} p_{i,j} 2^{-j}$. Pour un entier naturel m, construire les m premiers niveaux de l'arbre DDG sous-jacent requiert de connaître tous les indices $i \in \mathbb{N}$ tels que les coefficients $p_{i,j} = 1$ pour $1 \leq j \leq m$. Il est clair qu'il n'existe qu'un nombre fini de probabilités p_i ayant un $p_{i,j} = 1$ pour j fixe, car sinon $\sum_{i=1}^{\infty} p_i$ divergerait. Le problème de calculer les quantités $p_{i,j}$ en cours d'exécution peut prendre beaucoup de temps et, en théorie, ceci explique pourquoi nous faisons appel à un oracle tabulaire qui pour chaque entrée (i, j) nous indique gratuitement si $p_{i,j} = 1$. Dans certains cas, nous pouvons trouver effectivement les bits des expansions sans avoir à les enregistrer avant l'exécution de l'algorithme puisqu'il peut exister des raccourcis mathématiques et calculatoires, mais ceci peut être prohibitif en temps. Nous rappelons toutefois le lecteur que nous ne tenons pas compte des aspects reliés au temps et à l'espace qu'il faudrait pour implanter les oracles.

Malgré la remarque 3, voici notre implantation par listes de l'algorithme de Knuth et Yao. Les feuilles d'un niveau de l'arbre DDG sont représentées par une liste. Pour chaque naturel $j \in \mathbb{N}$, nous définissons la liste L_j du j^{e} niveau par

$$L_j = \begin{cases} \emptyset & \text{si } j = 0, \\ \{i \ : \ p_{i,j} = 1\} & \text{si } j \geq 1. \end{cases}$$

Pour faciliter la notation, nous dénotons la taille de la j^{e} liste par $|L_j|$. Le i^{e} élément de la j^{e} liste est dénoté par $L_j(i)$.

Algorithme 2.1 : Implantation par listes de l'algorithme de Knuth et Yao

1: $N_0 \leftarrow 0$
2: **pour** $j = 1$ **à** ∞ {Parcours du j^{e} niveau.} **faire**
3: $B_j \leftarrow$ `bit aléatore`
4: $N_j \leftarrow 2(N_{j-1} - |L_{j-1}|) + B_j$
5: **si** $N_j \leq |L_j|$ **alors**
6: **retourner** $L_j(N_j + 1)$ {Arrêt aléatoire dans une feuille. Toutes feuilles sont équiprobables.}
7: **sinon** $\{N_j > |L_j|\}$

8: Continuer {Pas d'arrêt dans une feuille. « Sortie » de la liste L_j.}
9: **fin si**
10: **fin pour**

L'algorithme 2.1 termine avec probabilité 1, car

$$\sum_{j=0}^{\infty} \frac{|L_j|}{2^j} = 0 + \sum_{j=1}^{\infty} \frac{|L_j|}{2^j} = \sum_{j=1}^{\infty} \sum_{i=1}^{n} \frac{p_{i,j}}{2^j} = 1.$$

De plus, si l'algorithme s'arrête à la j^{e} itération, alors

$$N_j = \sum_{i=1}^{j-1} 2^i \big(B_{j-1} - |L_{j-1}|\big) + B_j \in \{0, 1, \ldots, |L_j| - 1\},$$

et

$$\mathbf{P}\big\{N_j \in \{0, 1, \ldots, |L_j| - 1\} \mid \text{arrêt } j^{\text{e}} \text{ tour}\big\} = \frac{1}{|L_j|} \frac{|L_j|}{2^j} = \frac{1}{2^j}.$$

Remarque 4 (génération optimale par paquets et extraction de bits aléatoires). Notre algorithme 2.1 peut être modifié afin de retourner la valeur j du compteur de la boucle qui est également le niveau de l'arbre DDG sur lequel l'algorithme de Knuth et Yao s'arrête. Le niveau de l'arbre est une variable aléatoire qui nous est donnée gratuitement à partir de laquelle nous extrayons des bits non biaisés i.i.d. La probabilité que le symbole x_i soit retourné au j^{e} niveau est $\frac{1}{p_i 2^{\mathrm{d}(x_i)}}$ où $\mathrm{d}(x_i)$ dénote la profondeur de x_i. S'il fallait générer une telle variable par la méthode de l'inversion par exemple, alors il faudrait générer des bits non biaisés i.i.d. pour l'inverser et c'est une partie de ces bits qu'il aurait fallu générer que nous pouvons récupérer de façon effective. La réutilisation de ces bits ouvre la porte à un algorithme optimal pour générer des variables i.i.d. par paquets (« batch generation »). À propos de la génération par paquets et de l'extraction de bits aléatoires, consulter l'annexe II.

Avant d'entrer dans les détails de la preuve, voici un exemple illustrant le lancer d'un dé non biaisé et qui est un exemple optimal. Nous utilisons cet exemple tout au long de ce chapitre pour illustrer certains faits.

Exemple (lancer d'un dé non biaisé) Lancer un dé est équivalent à simuler une loi discrète avec 6 réalisations équiprobables. Étant donné que les branches 000 et 111 sont équiprobables, nous ne rejetons pas complètement ces deux possibilités si elles se produisent et nous réutilisons le premier bit. Nous obtenons l'automate suivant :

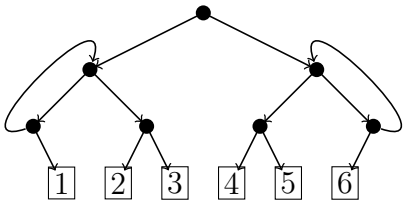

Figure 2.3 – Le lancer d'un dé non biaisé

Notons que si les boucles pointaient sur la racine, alors l'algorithme ne serait pas optimal. Si T est le nombre de bits aléatoires non biaisés i.i.d. utilisés pour générer aléatoirement une des six faces, alors nous pouvons calculer $\mathbf{E}(T)$ d'au moins trois façons. La troisième façon se fera à partir du résultat de Knuth et Yao directement. La première méthode consiste à trouver une équation pour $\mathbf{E}(T)$ et nous avons

$$\mathbf{E}(T) = 3 + \frac{2}{8}\big(\mathbf{E}(T) - 1\big),$$

car, nous espérons 3 bits et nous passons $\frac{2}{8}$ du temps à recommencer en réutilisant le premier bit. En résolvant, nous obtenons $\mathbf{E}(T) = \frac{11}{3}$. Pour la deuxième méthode, nous trouvons exactement $\mathbf{P}\{T = t\}$. En effet,

$$\mathbf{P}\{T = t\} = \frac{\text{nombre de feuilles au } t^{\text{e}} \text{ niveau}}{2^t},$$

et, par conséquent pour $k > 0$,

$$\mathbf{P}\{T = 1\} = 0,$$

et

$$\mathbf{P}\{T = 2k\} = 0,$$

$$\mathbf{P}\{T = 2k+1\} = \frac{6}{2^{2k+1}}.$$

Nous trouvons donc que

$$\begin{aligned}
\mathbf{E}(T) &= \sum_{t=0}^{\infty} t\mathbf{P}\{T=t\} \\
&= \sum_{k=1}^{\infty} (2k+1)\frac{6}{2^{2k+1}} \\
&= \sum_{k=1}^{\infty} \frac{6k}{4^k} + \sum_{k=1}^{\infty} \frac{3}{4^k} \\
&= 6\left(\frac{1}{4}\frac{1}{(1-(1/4))^2}\right) + 3\left(\frac{1}{4}\frac{1}{(1-(1/4))}\right) \\
&= \frac{11}{3}.
\end{aligned}$$

La troisième façon de calculer l'espérance du nombre de bits aléatoires requis pour simuler le lancer d'un dé est d'utiliser le résultat de Knuth et Yao dans [30]. Nous énonçons donc le résultat et nous donnons une preuve simple.

Théorème 3 (Knuth et Yao (1976))**.** *Si $(p_1, \ldots, p_n)$ est un vecteur de probabilités (n peut être infini) et que T est le nombre de bits aléatoires requis pour échantillonner $(p_1, \ldots, p_n)$ et que $\sum_{i=1}^{n} p_i \log_2\left(\frac{1}{p_i}\right)$ converge alors l'espérance $\mathbf{E}(T)$ pour un algorithme optimal de type DDG échantillonnant $(p_1, \ldots, p_n)$ est minorée et majorée comme suit :*

$$\sum_{i=1}^{n} p_i \log_2\left(\frac{1}{p_i}\right) \leq \mathbf{E}(T) \leq \sum_{i=1}^{n} p_i \log_2\left(\frac{1}{p_i}\right) + 2.$$

Si $\sum_{i=1}^{n} p_i \log_2\left(\frac{1}{p_i}\right)$ diverge, alors $\mathbf{E}(T)$ diverge.

Preuve du théorème 3. Étant donné un vecteur de probabilité $(p_1, p_2, \ldots, p_n)$ avec n possiblement infini, pour tout $i \in \{1, \ldots, n\}$ considérons l'expansion binaire de

p_i c'est-à-dire

$$p_i = \sum_{j=1}^{\infty} \frac{b_{i,j}}{2^j} \text{ et } b_{i,j} \in \{0,1\}.$$

Si T dénote le nombre de bits requis par un algorithme optimal pour échantillonner $(p_1, p_2, \ldots, p_n)$, alors, pour tout $t \geq 1$,

$$\begin{aligned}\mathbf{P}\{T = t\} &= \frac{\text{nombre de feuilles au niveau } t}{2^t} \\ &= \sum_{i=1}^{n} \frac{b_{i,t}}{2^t}.\end{aligned}$$

Par conséquent,

$$\begin{aligned}\mathbf{E}(T) &= \sum_{t=0}^{\infty} t\mathbf{P}\{T = t\} \\ &= \sum_{t=1}^{\infty} t \sum_{i=1}^{n} \frac{b_{i,t}}{2^t} \\ &= \sum_{i=1}^{n} \left(\sum_{t=1}^{\infty} \frac{t b_{i,t}}{2^t} \right). \end{aligned} \tag{2.7}$$

Nous montrons maintenant que la quantité entre deux parenthèses de la ligne (2.7) est majorée par $p_i \log_2(1/p_i)$ et minorée par $p_i \log_2(1/p_i) + 2p_i$ donc le résultat s'ensuit. Pour simplifier la notation et sans perte de généralité, soit $x \in [0,1]$ et son expansion binaire

$$x = \sum_{j=1}^{\infty} \frac{x_j}{2^j}.$$

Pour compléter la preuve, il reste donc à démontrer que

$$x \log_2(x) \leq \sum_{j=1}^{\infty} \frac{j x_j}{2^j} \leq x \log_2(x) + 2x. \tag{2.8}$$

Si m est l'indice du premier coefficient non nul de l'expansion de x, alors deux cas

sont possibles : (1) $x = 2^{-m}$ ou (2) $2^{-m} < x < 2^{1-m}$. Les inégalités sont strictes pour le deuxième cas, car $x \neq 2^{-m}$. Pour le premier cas, si x n'a seulement que le coefficient $x_m = 1$, alors la ligne (2.8) est trivialement vraie. Pour le deuxième cas,

$$\frac{1}{2^m} < x < \frac{2}{2^m} \Longleftrightarrow 0 < m + \log_2(x) < 1.$$

Ainsi pour la majoration,

$$\sum_{j=1}^{\infty} \frac{jx_j}{2^j} < \frac{m}{2^m} + \sum_{j=m+1}^{\infty} \frac{j}{2^j} = \frac{m+1}{2^m} < x(m+1) = x(2 - \log_2(x))$$

et, pour la minoration,

$$\sum_{j=1}^{\infty} \frac{jx_j}{2^j} = \sum_{j=m}^{\infty} \frac{jx_j}{2^j} \geq m \sum_{j=m}^{\infty} \frac{x_j}{2^j} = mx > x(-\log_2(x)).$$

Ce qui complète la preuve. □

Brièvement, Knuth et Yao obtinrent les bornes entropiques pour l'espérance $\mathbf{E}(T)$ en analysant la fonction $\nu : [0,1] \to [0,\infty)$ définie par

$$\nu(x) = \sum_{j=0}^{\infty} \frac{\{2^j x\}}{2^j}$$

où, comme dans Knuth et Yao [30], $\{y\} = y \bmod 1 = y - \lfloor y \rfloor$ est la partie fractionnaire de $y \in \mathbb{R}$. Pour $\{y\} \in [0,1)$ et $k \in \{1,2,\ldots\}$, dénotons par $\{y\}_k$ le k^{e} bit du développement binaire de $\{y\}$. Pour tout $y \in [0,\infty)$,

$$\{y\}_k = \lfloor 2^k y \rfloor \bmod 2,$$
$$\{y\} = \sum_{k=1}^{\infty} \frac{\lfloor 2^k y \rfloor \bmod 2}{2^k}.$$

Pour tout $x \in [0,1]$ dont le développement binaire est $x = 0.x_1x_2\cdots x_k\cdots$,

$$\begin{aligned}
\nu(x) &\stackrel{\text{def}}{=} \sum_{j=0}^{\infty} \frac{\{2^j x\}}{2^j} \\
&= \sum_{j=0}^{\infty} \frac{1}{2^j} \sum_{k=1}^{\infty} \frac{\lfloor 2^{k+j} x \rfloor \bmod 2}{2^k} \\
&= \sum_{j=0}^{\infty} \frac{1}{2^j} \sum_{k=j+1}^{\infty} \frac{\lfloor 2^{k} x \rfloor \bmod 2}{2^{k-j}} \\
&= \sum_{j=0}^{\infty} \sum_{k=j+1}^{\infty} \frac{\lfloor 2^{k} x \rfloor \bmod 2}{2^k} \\
&= \sum_{k=1}^{\infty} \sum_{j=0}^{k-1} \frac{\lfloor 2^{k} x \rfloor \bmod 2}{2^k} \\
&= \sum_{k=1}^{\infty} k \frac{\lfloor 2^{k} x \rfloor \bmod 2}{2^k} \\
&= \sum_{k=1}^{\infty} \frac{k x_k}{2^k}. \qquad (2.9)
\end{aligned}$$

L'expression (2.9) est égale à l'expression entre les parenthèses de (2.7). La fonction ν indique le nombre espéré de bits pour générer un symbole, disons $\bullet$ de probabilité x à partir d'un arbre DDG. La quantité $\lfloor 2^k x \rfloor \bmod 2$ indique si le coefficient 2^{-k} de x est présent, c'est-à-dire si $x_k = 1$, dans l'expansion binaire de x lequel cas il y a une branche de longueur k de probabilité 2^{-k} ayant une feuille étiquetée par le symbole $\bullet$ qui requiert donc k bits d'où le k multipliant $\lfloor 2^k x \rfloor \bmod 2 \in \{0,1\}$. Par conséquent,

$$\mathbf{E}(T) = \sum_{i=1}^{n} \nu(p_i). \qquad (2.10)$$

Nous avons donc le corollaire suivant qui donne l'expression exacte de la complexité de générer une variable aléatoire discrète uniforme sur l'ensemble $\{0,\ldots,n-1\}$. Le corollaire fournira également une troisième façon de calculer l'espérance du nombre de bits pour le lancer d'un dé.

Corollaire 1 (Complexité exacte pour la loi discrète uniforme sur $\{0, \ldots, n-1\}$)**.** *Soit $n \in \mathbb{N}$, $m \in \mathbb{N}$, $r \in \mathbb{N}$, $a \in \mathbb{N}$ et $k \in \mathbb{N}$ tels que $r > 1$ est impair, $n = 2^m r$*

$$\begin{aligned}
&\frac{1}{r} = \frac{a}{2^k} \sum_{i=0}^{\infty} \frac{1}{2^{ik}}, \\
&k = \min\{j \geq 1 \ \ t.q. \ \ 2^j = 1 \bmod r\} \ \ et \\
&a = \frac{2^k - 1}{r}.
\end{aligned}$$

Si T est le nombre de bits non biaisés i.i.d. requis pour générer une variable aléatoire uniforme dans l'ensemble $\{0, \ldots, n-1\}$ à partir de l'algorithme optimal de Knuth et Yao, alors

$$\mathbf{E}(T) = m + \frac{\sum_{i=1}^{k} i a_i 2^{-i}}{\sum_{i=1}^{k} a_i 2^{-i}} + \frac{k}{2^k - 1}.$$

Preuve du corollaire 1. Par la ligne (2.10), l'expression exacte du nombre espéré de bits pour l'algorithme optimal est donnée par

$$\mathbf{E}(T) = \sum_{k=1}^{n} \nu(p_k) = \sum_{k=1}^{n} \nu\left(\frac{1}{n}\right) = n\nu\left(\frac{1}{n}\right) = \sum_{i=0}^{\infty} \left\{\frac{2^i}{n}\right\} \frac{n}{2^i}.$$

Décomposons n en partie pair et impair c'est-à-dire $n = 2^m r$ où r est impair. Le développement périodique de $1/r$ en base 2 de période k est

$$\begin{aligned}
&\frac{1}{r} = \frac{a}{2^k} \sum_{i=0}^{\infty} \frac{1}{2^{ik}}, \\
&k = \min\{j \geq 1 \ \text{ t.q. } \ 2^j = 1 \bmod r\} \ \text{ et} \\
&a = \frac{2^k - 1}{r}.
\end{aligned}$$

Nous avons toujours que $\text{pgcd}(a, r) = 1$. Donc,

$$\mathbf{E}(T) = \sum_{i=0}^{\infty} \left\{\frac{2^i}{n}\right\} \frac{n}{2^i}$$

$$
\begin{aligned}
&= \sum_{i=0}^{m-1} \left\{ \frac{2^i}{2^m r} \right\} \frac{2^m r}{2^i} + \sum_{i=m}^{\infty} \left\{ \frac{2^i}{2^m r} \right\} \frac{2^m r}{2^i} \\
&= \sum_{i=0}^{m-1} \frac{2^i}{2^m r} \frac{2^m r}{2^i} + \sum_{i=m}^{\infty} \left\{ \frac{2^i}{2^m r} \right\} \frac{2^m r}{2^i} \\
&= m + \sum_{i=m}^{\infty} \left\{ \frac{2^i}{2^m r} \right\} \frac{2^m r}{2^i}. \qquad (2.11)
\end{aligned}
$$

En utilisant la cyclicité du développement périodique de $\frac{1}{r}$, la somme du côté droit de (2.11) peut être décomposée en k sous-sommes comme suit

$$
\sum_{i=m}^{\infty} \left\{ \frac{2^i}{2^m r} \right\} \frac{2^m r}{2^i} = \sum_{i=0}^{k-1} \sum_{j=0}^{\infty} \left\{ \frac{2^{i+jk}}{2^m r} \right\} \frac{2^m r}{2^{i+jk}}.
$$

Pour i fixe, $\left\{ \frac{2^{i+jk}}{2^m r} \right\}$ ne dépend pas de j, car

$$
\begin{aligned}
\frac{2^{i+jk}}{2^m r} &= 2^{i+jk} \frac{a}{2^k} \sum_{i=0}^{\infty} \frac{1}{2^{ik}}, \text{ et} \\
\frac{1}{r} &= 0.\overline{a_1 \cdots a_k}, \text{ donc} \\
2^{i+jk} \frac{1}{r} &= a_1 \cdots a_i . a_{i+1} \cdots a_k \overline{a_1 \cdots a_k}.
\end{aligned}
$$

Si ℓ est le bit le plus significatif a, alors nous « paddons » avec $k - \ell$ bits valant 0 pour obtenir $a = a_k \ldots a_1$. Par conséquent,

$$
\left\lfloor 2^{i+jk} \frac{1}{r} \right\rfloor = 2^{i-1} a_1 + 2^{i-2} a_2 + \ldots + 2^0 a_i
$$

et

$$
\left\{ 2^{i+jk} \frac{1}{r} \right\} = a_{i+1} 2^{-1} + a_{i+2} 2^{-2} + \ldots + a_k 2^{i-k} + \frac{1}{2^{k-i}} \frac{1}{r} = \left\{ \frac{a}{2^k} 2^i \right\} + \frac{1}{2^{k-i}} \frac{1}{r}.
$$

Enfin la somme du côté droit de (2.11) est

$$
\sum_{i=m}^{\infty} \left\{ \frac{2^i}{2^m r} \right\} \frac{2^m r}{2^i} = \sum_{i=0}^{k-1} \sum_{j=0}^{\infty} \left\{ \frac{2^{i+jk}}{2^m r} \right\} \frac{2^m r}{2^{i+jk}}
$$

$$
\begin{aligned}
&= \sum_{i=0}^{k-1} \left(\left\{ \frac{a}{2^k} 2^i \right\} + \frac{1}{2^{k-i}} \frac{1}{r} \right) \sum_{j=0}^{\infty} \frac{r}{2^i} \frac{1}{2^{jk}} \\
&= \sum_{i=0}^{k-1} \left(\left\{ \frac{a}{2^k} 2^i \right\} + \frac{1}{2^{k-i}} \frac{1}{r} \right) \frac{r}{2^i} \frac{2^k}{2^k - 1} \\
&= \sum_{i=0}^{k-1} \left\{ \frac{a}{2^k} 2^i \right\} \frac{r}{2^i} \frac{2^k}{2^k - 1} + \frac{k}{2^k - 1} \\
&= \frac{r}{2^k - 1} \sum_{i=0}^{k-1} \left(a_{i+1} 2^{-1} + a_{i+2} 2^{-2} + \ldots + a_k 2^{i-k} \right) 2^{k-i} + \frac{k}{2^k - 1} \\
&= \frac{r}{2^k - 1} \sum_{i=0}^{k-1} \sum_{j=i+1}^{k} a_j 2^{k-j} + \frac{k}{2^k - 1} \\
&= \frac{r 2^k}{2^k - 1} \sum_{i=1}^{k} i a_i 2^{-i} + \frac{k}{2^k - 1} \\
&= \frac{2^k}{a} \sum_{i=1}^{k} i a_i 2^{-i} + \frac{k}{2^k - 1} \\
&= \frac{\sum_{i=1}^{k} i a_i 2^{-i}}{\sum_{i=1}^{k} a_i 2^{-i}} + \frac{k}{2^k - 1}. \qquad (2.12)
\end{aligned}
$$

En combinant (2.11) et (2.12), nous obtenons que

$$
\mathbf{E}(T) = m + \frac{\sum_{i=1}^{k} i a_i 2^{-i}}{\sum_{i=1}^{k} a_i 2^{-i}} + \frac{k}{2^k - 1}.
$$

□

Montrer que $m + \frac{\sum_{i=1}^{k} i a_i 2^{-i}}{\sum_{i=1}^{k} a_i 2^{-i}} + \frac{k}{2^k - 1} \leq \log_2(n) + 2$ comme l'affirme le résultat d'optimalité de Knuth et Yao serait intéressant.

Revenons à notre exemple du lancer d'un dé qui s'agit de générer une variable uniforme discrète avec $n = 6$ et calculons $\mathbf{E}(T)$ en se servant de notre corollaire 1. Nous avons donc que $m = 1$, $r = 3$, $k = 2$, $a = 1 = (01)_2 = (a_1 a_2)_2$ et ainsi, comme nous l'avions déjà calculé,

$$
\mathbf{E}(T) = m + \frac{\sum_{i=1}^{k} i a_i 2^{-i}}{\sum_{i=1}^{k} a_i 2^{-i}} + \frac{k}{2^k - 1} = 1 + \frac{1/2}{1/4} + \frac{2}{3} = \frac{11}{3}.
$$

Mentionnons l'algorithme de Lumbroso, apparu en 2012 dans sa thèse de doctorat [33], appelé le « Fast Dice Roller » échantillonnant une loi discrète uniforme sur n points. La complexité espérée du « Fast Dice Roller » est dans l'intervalle d'optimalité $[\log_2(n), \log_2(n) + 2]$ donné par le résultat de Knuth et Yao.

Algorithme 2.2 : « Fast Dice Roller » de Lumbroso (2012)

```
 1: X ← 0
 2: Y ← 1
 3: répéter
 4:    Y ← 2Y
 5:    X ← 2X + B {B est un bit aléatoire.}
 6:    si Y ≥ n alors
 7:       si X < n alors
 8:          retourner X
 9:       sinon
10:          Y ← Y − n
11:          X ← X − n
12:       fin si
13:    fin si
14: fin répéter
```

L'exactitude du « Fast Dice Roller » est due au fait que, pour toute itération de la boucle éternelle, X est uniformément distribuée dans $\{0, \ldots, n-1\}$. En effet, initialement X ne prend qu'une valeur donc X est uniformément distribué dans $\{0\}$. Les lignes 10 et 11 sont atteintes si et seulement si $X \geq n$ et, conditionnellement au fait que $X \geq n$, nous avons que X est uniforme sur $\{n, \ldots, Y-1\}$. De plus, étant donné que $X \geq n$, l'ensemble $\{n, \ldots, Y-1\} \neq \emptyset$ car $Y > X \geq n$ et l'ensemble $\{n, \ldots, Y-1\}$ est translaté de n unités qui permet de réutiliser les bits qui peuvent l'être comme dans notre exemple précédent du lancer d'un dé. Dans Lumbroso [33], nous y trouvons l'expression asymptotique de la complexité du « Fast Dice Roller ».

Théorème 4 (Lumbroso (2012))**.** *Pour tout $\alpha > 0$, si T est le nombre de bits requis non biaisés i.i.d. par le « Fast Dice Roller », alors*

$$\mathbf{E}(T) = \log_2(n) + \frac{1}{2} + \frac{1}{\log 2} - \frac{\gamma}{\log 2} + P(\log_2(n)) + O(n^{-\alpha}).$$

P est un polynôme trigonométrique périodique et γ est la constante d'Euler. De plus,

$$0 \leq \mathbf{E}(T) - \log_2(n) \leq 2.$$

Selon le résultat de Knuth et Yao, pour un algorithme DDG optimal pour la loi uniforme

$$\log_2(n) \leq \mathbf{E}(T) \leq \log_2(n) + 2.$$

La complexité espérée du « Fast Dice Roller » est donc dans l'intervalle d'optimalité selon le résultat de Knuth et Yao.

Nous complétons ce chapitre avec une preuve d'un résultat du Han et Hoshi en 1997 dans [25]. Han et Hoshi implantèrent la méthode de l'inversion pour les lois discrètes et ils montrèrent que l'exécution de leur méthode peut être représentée par un arbre DDG. Dans Han et Hoshi [25], nous y trouvons un algorithme adapté aux sources Markoviennes, c'est-à-dire lorsque les bits sont corrélés selon une chaîne de Markov, sujet auquel nous ne nous sommes pas attardés. En effet, nous explicitons leur algorithme et nous donnons une preuve de leur algorithme qui ne requiert pas entre autres d'utiliser la théorie de la majorisation des distributions discrètes.

Nous rappelons brièvement l'algorithme avant d'établir sa complexité. Étant donné un vecteur $(p_1, \ldots, p_n)$ avec n dénombrable, l'algorithme de Han et Hoshi partitionne l'intervalle $[0, 1]$, qui est l'image de la distribution, en sous-intervalles disjoints $[Q_{i-1}, Q_i)$ tels que

$$\begin{aligned} Q_0 &= 0, \text{ et} \\ Q_i &= \sum_{k=1}^{i} p_k \text{ pour } i \in \{1, \ldots, n\}. \end{aligned}$$

Rappelons que si U est uniformément distribuée sur l'intervalle $[0, 1]$, alors il existe un unique $i \in \{1, \ldots, n\}$ tel que $Q_{i-1} \leq U < Q_i$ par le théorème 2 du chapitre 1. L'algorithme sélectionne aléatoirement (itérativement) un sous-intervalle $I \subset [0, 1)$

et s'arrête dès que $I \subset [Q_{i-1}, Q_i)$ pour un certain entier $i \in \{1, \ldots, n\}$. L'algorithme est le suivant :

Algorithme 2.3 : Han et Hoshi (1997)

1: $T \leftarrow 0$
2: $\alpha_T \leftarrow 0$
3: $\beta_T \leftarrow 1$
4: **répéter**
5: $\quad T \leftarrow T + 1$
6: $\quad B \leftarrow$ `bit aléatoire`
7: $\quad \alpha_T \leftarrow \alpha_{T-1} + (\beta_{T-1} - \alpha_{T-1})(B/2)$
8: $\quad \beta_T \leftarrow \alpha_{T-1} + (\beta_{T-1} - \alpha_{T-1})((B+1)/2)$
9: $\quad I \leftarrow [\alpha_T, \beta_T)$
10: **tant que** $I \subset [Q_{i-1}, Q_i)$
11: **retourner** i (ou le symbole x_i).

Étant donné un vecteur de probabilités $(p_1, \ldots, p_n)$, remarquons que l'algorithme de Han et Hoshi utilise un oracle tabulaire qui sur entrée i retourne $\sum_{j=1}^{i} p_j$. À la différence de Knuth et Yao, le type d'oracle utilisé par Han et Hoshi est plus pratique puisque seule la connaissance des sommes $p_1 + \ldots + p_j$ pour $j \in \{1, \ldots, n\}$ est requise.

Par exemple, pour un vecteur de probabilité (p_1, p_2, p_3, p_4) dont

$$\begin{aligned} p_1 &= 0.001\cdots, \\ p_1 + p_2 &= 0.0101\cdots, \\ p_1 + p_2 + p_3 &= 0.1011\cdots, \end{aligned}$$

nous avons la figure 2.4 qui représente l'exécution de Han et Hoshi pour des valeurs de $0 < T \leq 4$.

L'image de la distribution qui est l'intervalle $[0, 1]$ est représentée horizontalement. Si les valeurs de U sont telles que $\lfloor 2^4 U \rfloor = 0000$, alors 1 est retourné. Si les valeurs de U sont telles que $\lfloor 2^4 U \rfloor = 0001$, alors l'algorithme ne s'est pas arrêté. Si les valeurs de U sont telles que $\lfloor 2^3 U \rfloor = 001$ ou $\lfloor 2^4 U \rfloor = 0100$, alors l'algorithme

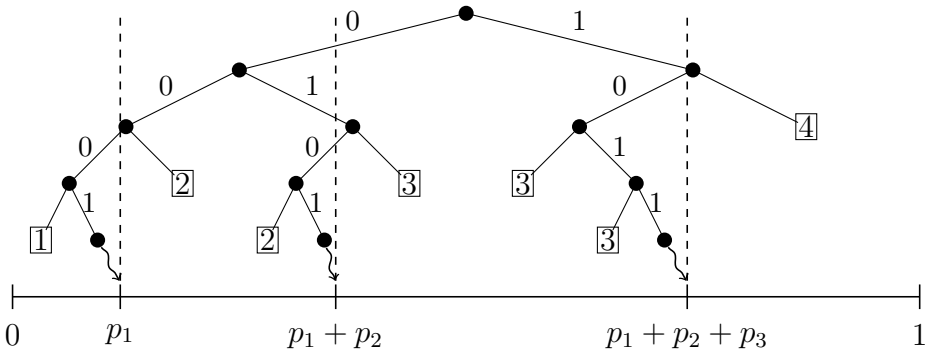

Figure 2.4 – 1er exemple - Han et Hoshi

retourne 2. Si les valeurs de U sont telles que $\lfloor 2^4 U \rfloor = 0101$, alors l'algorithme ne s'est pas arrêté.

Donnons un autre exemple illustrant l'exécution de Han et Hoshi et, cette fois-ci, avec le vecteur de probabilités

$$(p_1, p_2, p_3, p_4, p_5, p_6, p_7) = \Big(\frac{1}{16}, \frac{5}{32}, \frac{5}{32}, \frac{9}{32}, \frac{3}{16}, \frac{1}{32}, \frac{1}{8}\Big). \tag{2.13}$$

Puisque les probabilités de la ligne (2.13) sont des multiples de 2^{-5}, alors les valeurs cumulées $q_i = p_1 + \ldots + p_i$ pour $i \in \{1, \ldots, 7\}$ sont exactement :

$$\begin{aligned}
q_1 &= \frac{2}{32} = (0.00010)_2 \\
q_2 &= \frac{7}{32} = (0.00111)_2 \\
q_3 &= \frac{12}{32} = (0.01100)_2 \\
q_4 &= \frac{21}{32} = (0.10101)_2 \\
q_5 &= \frac{27}{32} = (0.11011)_2 \\
q_6 &= \frac{28}{32} = (0.11100)_2 \\
q_7 &= \frac{32}{32} = (1.00000)_2.
\end{aligned}$$

Ainsi nous avons donc l'arbre « DDG » représenté à la figure 2.5.

L'exactitude de l'algorithme de Han et Hoshi découle directement du théorème de l'inversion pour les discrètes. Son exécution est représentée par un arbre DDG.

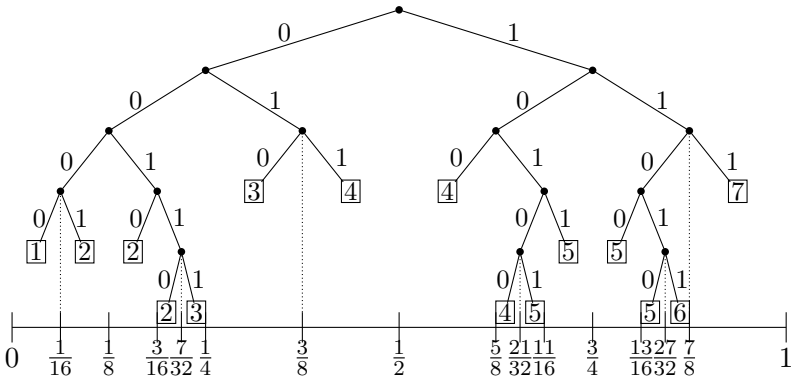

Figure 2.5 – 2e exemple - Han et Hoshi

En effet, si T est le nombre de bits aléatoires requis de la source, alors T est également le nombre d'itérations de la boucle. Pour $T \geq 1$, l'intervalle $\big[\alpha_T, \beta_T\big)$ contient $\big[\alpha_{T+1}, \beta_{T+1}\big)$. À chaque noeud ou chaque feuille correspond un intervalle de la forme $\big[\alpha_T, \beta_T\big)$. Par exemple, la racine représente l'intervalle $[0, 1]$. L'enfant gauche de la racine représente l'intervalle $[0, 1/2)$ et l'enfant droit de la racine représente l'intervalle $[1/2, 1)$. Pour un noeud interne v correspond un intervalle $\big[\alpha_T, \beta_T\big)$ qui n'est pas contenu dans l'intervalle $\big[Q_i, Q_{i+1}\big)$. Si la source produit $B = 0$, alors l'enfant gauche de v correspond à l'intervalle $\big[\alpha_T, (\alpha_T + \beta_T)/2)\big) = \big[\alpha_{T+1}, \beta_{T+1}\big)$. Si la source produit $B = 1$, alors l'enfant droit de v correspond à l'intervalle $\big[(\alpha_T + \beta_T)/2), \beta_T\big) = \big[\alpha_{T+1}, \beta_{T+1}\big)$. Chaque feuille correspond à un intervalle $\big[\alpha_T, \beta_T\big)$ entièrement contenu dans $\big[Q_i, Q_{i+1}\big)$ provoquant l'arrêt de la boucle et s'ensuit ainsi le retour du symbole X_i avec probabilité Q_i.

Théorème 5 (Han et Hoshi (1997))**.** *Étant donné un vecteur de probabilités $(p_1, \ldots, p_n)$ avec n possiblement infini, si T est le nombre de bits non biaisés i.i.d. requis par l'algorithme de Han et Hoshi, alors*

$$\mathbf{E}(T) \leq \mathcal{E}\big(\{p_i\}_{i \in \mathbb{Z}}\big) + 3.$$

Preuve du théorème 5. Soit L_i l'ensemble des feuilles rattachées au symbole i dans l'arbre DDG. Nous partionnons L_i en deux ensembles A_i et B_i de sorte que, pour chaque niveau de l'arbre, chaque ensemble A_i et B_i possède au plus une feuille.

Soit les sommes

$$\alpha_i = \sum_{u \in A_i} p(u)$$
$$\beta_i = \sum_{u \in B_i} p(u),$$

où $p(u)$ est la probabilité de la feuille u c'est-à-dire $1/2^{\mathrm{d}(u)}$ et $\mathrm{d}(u)$ est la profondeur de u. Nous avons donc que $p_i = \alpha_i + \beta_i$. Par la propriété d'emboîtement et des calculs élémentaires,

$$\begin{aligned}\sum_{i=1}^{\infty} p_i \log_2\left(\frac{1}{p_i}\right) &\leq \sum_{i=1}^{\infty} \alpha_i \log_2\left(\frac{1}{\alpha_i}\right) + \sum_{i=1}^{\infty} \beta_i \log_2\left(\frac{1}{\beta_i}\right) \\ &\leq \sum_{i=1}^{\infty} p_i \log_2\left(\frac{1}{p_i}\right) + 1. \end{aligned} \tag{2.14}$$

Soit $(\alpha_i)_j$ le j^{e} bit de l'expansion binaire de α_i et $(\beta_i)_j$ le j^{e} bit de l'expansion binaire de β_i. Alors

$$\mathbf{E}(T) = \sum_{j=1}^{\infty} \frac{j(\alpha_i)_j}{2^j} + \sum_{j=1}^{\infty} \frac{j(\beta_i)_j}{2^j} \stackrel{\mathrm{def}}{=} \mathrm{I} + \mathrm{II}.$$

Comme nous l'avons fait pour la preuve du résultat de Knuth et Yao,

$$\sum_{i=1}^{\infty} \alpha_i \log_2\left(\frac{1}{\alpha_i}\right) \leq \mathrm{I} \leq \sum_{i=1}^{\infty} \alpha_i \log_2\left(\frac{1}{\alpha_i}\right) + 2\alpha_i,$$
$$\sum_{i=1}^{\infty} \beta_i \log_2\left(\frac{1}{\beta_i}\right) \leq \mathrm{II} \leq \sum_{i=1}^{\infty} \beta_i \log_2\left(\frac{1}{\beta_i}\right) + 2\beta_i,$$

de sorte qu'en utilisant (2.14),

$$\begin{aligned}\sum_{i=1}^{\infty} p_i \log_2\left(\frac{1}{p_i}\right) \leq \mathbf{E}(T) &\leq \sum_{i=1}^{\infty} p_i \log_2\left(\frac{1}{p_i}\right) + 1 + 2\sum_{i=1}^{\infty} p_i \\ &= \sum_{i=1}^{\infty} p_i \log_2\left(\frac{1}{p_i}\right) + 3.\end{aligned}$$

□

Enfin, nous complétons ce chapitre en décrivant une autre méthode pour générer une loi uniforme discrète sur l'ensemble $\{0, \ldots, n-1\}$, méthode dont la complexité semble n'avoir jamais été analysée. Considérons une suite infinie de bits $(U_1, U_2, \ldots)$ aléatoires et la variable aléatoire continue $U = \sum_{j=1}^{\infty} U_j 2^{-j}$ représentée par ladite suite. Pour générer une variable discrète uniforme ayant n réalisations, si nous possédions U, alors nous n'avons qu'à calculer $\lfloor nU \rfloor$. L'algorithme s'arrête aussitôt qu'il est possible de décider de la valeur de $\lfloor nU \rfloor$. Si nous dénotons par $U(t)$ la troncation de U avec les t premiers bits, c'est-à-dire $U(t) = 0.U_1 \cdots U_t$, alors, la variable aléatoire T du temps d'arrêt est

$$\begin{aligned} T &= \min\Big\{t \in \mathbb{N} \;:\; \Big(nU(t), n\Big(U(t) + \frac{1}{2^t}\Big)\Big) \not\subseteq \mathbb{N}\Big\} \\ &= \min\Big\{t \in \mathbb{N} \;:\; \text{l'intervalle } \Big(nU(t), n\Big(U(t) + \frac{1}{2^t}\Big)\Big) \text{ ne contient pas d'entier}\Big\}. \end{aligned}$$

Si $nU(t) \in \mathbb{N}$, alors $\lfloor nU \rfloor = nU(t)$ avec probabilité 1. De même, si $n(U(t)+2^{-t}) \in \mathbb{N}$ alors nous avons toujours que $\lfloor nU \rfloor = nU(t)$ avec probabilité 1, car la probabilité est nulle d'avoir une suite infinie de bits qui vaudraient 1. Voici donc l'algorithme lequel s'ensuivra d'une analyse de l'espérance du nombre de bits aléatoires, en d'autres termes, de la suite $(U_1, U_2, \ldots)$.

Algorithme 2.4 : loi uniforme discrète (bis.)

---

si $n = 2^m$ pour $m \in \mathbb{N}$ **alors**
  Générer les bits aléatoires $U_0, \ldots U_{m-1}$.
  retourner $\lfloor \frac{U_0 + U_1 2 + \ldots + U_{m-1} 2^{m-1}}{2^m} n \rfloor$
sinon
  $y \leftarrow 0$
  $U \leftarrow$ `bit aléatoire`
  pour $t \leftarrow 1$ **à** ∞ **faire**
    $y \leftarrow 2y + U$
    si $n > 2^t$ **alors**
      $x \leftarrow \lfloor ny/2^t \rfloor$
      si ny ou $n(y+1)$ est un multiple de 2^t **alors**
        retourner $\frac{ny}{2^t}$

13: **sinon**
14: **si** $\lfloor ny/2^t \rfloor = \lfloor n(y+1)/2^t \rfloor$ **alors**
15: **retourner** $\lfloor \frac{ny}{2^t} \rfloor$
16: **sinon**
17: **continuer**
18: **fin si**
19: **fin si**
20: **fin si**
21: **fin pour**
22: **fin si**

Théorème 6. *Soit $n = 2^m r$ avec r impair et une variable uniforme continue $U = \sum_{j=1}^{\infty} U_j 2^{-j}$. Si T dénote le nombre de bits, pour décider de la valeur $\lfloor nU \rfloor$, nous avons*

$$\mathbf{E}(T) = \lceil \log_2(n) \rceil + \frac{2r-2}{2^{\lceil \log_2(r) \rceil}}.$$

Si $r \neq 1$, alors

$$\lceil \log_2(n) \rceil + \frac{2}{3} \leq \mathbf{E}(T) \leq \lceil \log_2(n) \rceil + 2.$$

Notons que si $r = 1$ dans le théorème 6, alors $\mathbf{E}(T) = m = \lceil \log_2(n) \rceil$.

Preuve du théorème 6. D'abord, le cas $n = 2^m$ est évident. Nous supposons ainsi que $n \neq 2^m$ pour tout $m \in \mathbb{N}$ et utilisons le fait que

$$\mathbf{E}(T) = \sum_{t=0}^{\infty} \mathbf{P}\{T > t\}.$$

Afin de cerner le comportement de $\mathbf{P}\{T > t\}$, nous analysons les deux cas $n < 2^t$ et $n > 2^t$ puisque $n \neq 2^t$. Si $n > 2^t$, alors tout sous-intervalle I de $(0, n)$ de longueur $\frac{n}{2^t}$ contient au moins un naturel élément de $\{1, \ldots, n-1\} \cap I^\circ$ (où I° dénote l'intérieur ouvert de I) et, par conséquent, l'algorithme ne s'arrête pas et $\{T > t\}$ est un événement de probabilité 1.

Si $n < 2^t$, alors définissons

$$Z_t = \text{card}\Big\{ i \ : \ i \in \{1, \ldots, n-1\} \text{ et } i = j\frac{n}{2^t} \text{ avec } 1 \leq j \leq 2^t - 1 \Big\},$$

$$\mathcal{I} = \Big\{ I \ : \ I \subset (0,n),\, |I| = \frac{n}{2^t} \text{ et } I^\circ \cap \{1,\dots,n-1\} \neq \emptyset \Big\}.$$

Ainsi

$$\mathbf{P}\{T > t\} = \sum_{I \in \mathcal{I}} |I| = \frac{n-1-Z_t}{2^t}.$$

Puisque $n = 2^m r$, nous avons que $n\frac{n}{2^t} = j\frac{r}{2^{k-m}}$ et, par conséquent, j est un multiple de 2^{k-m}. Donc

$$Z_t = \left\lfloor \frac{2^t - 1}{2^{t-m}} \right\rfloor = \left\lfloor 2^m - \frac{1}{2^{t-m}} \right\rfloor = 2^m - 1,$$

et, pour $n < 2^t$,

$$\mathbf{P}\{T > t\} = \frac{n - 2^m}{2^t} = \frac{n - (n/r)}{2^t}.$$

Enfin,

$$\begin{aligned}
\mathbf{E}(T) &= \sum_{t=0}^{\infty} \mathbf{P}\{T > t\} \\
&= \sum_{t=0}^{\lfloor \log_2(n) \rfloor} 1 + \sum_{t=\lceil \log_2(n) \rceil}^{\infty} \frac{1}{2^t}\left(n - \frac{n}{r}\right) \\
&= \lfloor \log_2(n) \rfloor + 1 + \frac{2}{2^{\lceil \log_2(n) \rceil}}\left(n - \frac{n}{r}\right) \\
&= \lceil \log_2(n) \rceil + \frac{2}{2^{m + \lceil \log_2(r) \rceil}}\left(n - \frac{n}{r}\right) \\
&= \lceil \log_2(n) \rceil + \frac{2r}{2^{\lceil \log_2(r) \rceil}}\left(\frac{n - (n/r)}{n}\right) \\
&= \lceil \log_2(n) \rceil + \frac{2r-2}{2^{\lceil \log_2(r) \rceil}}.
\end{aligned}$$

Enfin, si $r \neq 1$, alors

$$\begin{aligned}
\sup_r \frac{2r-2}{2^{\lceil \log_2(r) \rceil}} &\leq \frac{2r}{2^{\lceil \log_2(r) \rceil}} = 2, \\
\inf_r \frac{2r-2}{2^{\lceil \log_2(r) \rceil}} &\geq \inf_r \frac{2r-2}{2r} \geq 1 - \frac{1}{3} = \frac{2}{3}.
\end{aligned}$$

□

Enfin voici un tableau comparant les complexités de la méthode précédente et celle du « Fast Dice Roller » pour certaines valeurs de n.

Tableau 2.1: Comparaison des complexités du « Fast Dice Roller » et l'inversion de $\lfloor nU \rfloor$

n	m	r	k	$\log_2(n)$	Fast Dice Roller	$\lfloor nU \rfloor$
1	0	1	1	0.000000	0.000000	0.000000
2	1	1	1	1.000000	1.000000	1.000000
3	0	3	2	1.584963	2.666667	3.000000
4	2	1	1	2.000000	2.000000	2.000000
5	0	5	4	2.321928	3.600000	4.000000
6	1	3	2	2.584963	3.666667	4.000000
7	0	7	3	2.807355	3.428571	4.500000
8	3	1	1	3.000000	3.000000	3.000000
9	0	9	6	3.169925	4.666667	5.000000
10	1	5	4	3.321928	4.600000	5.000000
11	0	11	10	3.459432	4.848485	5.250000
12	2	3	2	3.584963	4.666667	5.000000
13	0	13	12	3.700440	4.707692	5.500000
14	1	7	3	3.807355	4.428571	5.500000
15	0	15	4	3.906891	4.266667	5.750000
16	4	1	1	4.000000	4.000000	4.000000
17	0	17	8	4.087463	5.764706	6.000000
18	1	9	6	4.169925	5.666667	6.000000
19	0	19	18	4.247928	5.723197	6.125000
20	2	5	4	4.321928	5.600000	6.000000
21	0	21	6	4.392317	5.428571	6.250000
22	1	11	10	4.459432	5.848485	6.250000
23	0	23	11	4.523562	5.702003	6.375000
24	3	3	2	4.584963	5.666667	6.000000
25	0	25	20	4.643856	5.551220	6.500000
26	1	13	12	4.700440	5.707692	6.500000
27	0	27	18	4.754888	5.614035	6.625000
28	2	7	3	4.807355	5.428571	6.500000
29	0	29	28	4.857981	5.436314	6.750000
30	1	15	4	4.906891	5.266667	6.750000
31	0	31	5	4.954196	5.161290	6.875000
32	5	1	1	5.000000	5.000000	5.000000
33	0	33	10	5.044394	6.848485	7.000000

34	1	17	8	5.087463	6.764706	7.000000
35	0	35	12	5.129283	6.737973	7.062500
36	2	9	6	5.169925	6.666667	7.000000
37	0	37	36	5.209453	6.790051	7.125000
38	1	19	18	5.247928	6.723197	7.125000
39	0	39	12	5.285402	6.593407	7.187500
40	3	5	4	5.321928	6.600000	7.000000
41	0	41	20	5.357552	6.550244	7.250000
42	1	21	6	5.392317	6.428571	7.250000
43	0	43	14	5.426265	6.945736	7.312500
44	2	11	10	5.459432	6.848485	7.250000
45	0	45	12	5.491853	6.794139	7.375000
46	1	23	11	5.523562	6.702003	7.375000
47	0	47	23	5.554589	6.750699	7.437500
48	4	3	2	5.584963	6.666667	7.000000
49	0	49	21	5.614710	6.628039	7.500000
50	1	25	20	5.643856	6.551220	7.500000
51	0	51	8	5.672425	6.431373	7.562500
52	2	13	12	5.700440	6.707692	7.500000
53	0	53	52	5.727920	6.625925	7.625000
54	1	27	18	5.754888	6.614035	7.625000
55	0	55	20	5.781360	6.512267	7.687500
56	3	7	3	5.807355	6.428571	7.500000
57	0	57	18	5.832890	6.538012	7.750000
58	1	29	28	5.857981	6.436314	7.750000
59	0	59	58	5.882643	6.370009	7.812500
60	2	15	4	5.906891	6.266667	7.750000
61	0	61	58	5.930737	6.259336	7.875000
62	1	31	5	5.954196	6.161290	7.875000
63	0	63	6	5.977280	6.095238	7.937500

CHAPITRE 3

ÉCHANTILLONNAGE AVEC PRÉCISION ARBITRAIRE D'UNE DISTRIBUTION CONTINUE

Dans ce chapitre, nous expliquons comment échantillonner des distributions continues avec un niveau de précision arbitraire. La précision est donnée en argument aux algorithmes d'échantillonnage et ne change jamais durant l'exécution des algorithmes. Nous établissons de nouveaux résultats concernant la quantité de bits non biaisés i.i.d. requis pour échantillonner avec une précision arbitraire une distribution continue F. Le résultat majeur de ce chapitre est la minoration de l'espérance du nombre de bits pour générer une variable X de distribution F avec un niveau de précision $\epsilon > 0$. Nous analysons de même la complexité de certains algorithmes comme la méthode de l'inversion et la méthode des partitions. Ce chapitre se retrouve intégralement dans Devroye et Gravel [41].

Rappelons d'abord la définition de la distance entre deux variables qui nous importe telle que décrite au chapitre 1. Étant données deux variables aléatoires X et Y de distributions respectives F et G ayant le même support de définition, la distance entre X et Y qui nous incombe est celle de Wasserstein. Nous avions que $\text{dist}(X, Y) = W_p(F, G)$ (définition 8 du chapitre 1). De plus, nous avions qu'un algorithme de génération (définition 10 du chapitre 1) pour une variable continue X de distribution F est un algorithme qui étant donné un paramètre fixe de précision $\epsilon > 0$ retourne Y de distribution G en utilisant T bits non biaisés i.i.d. telle que $\text{dist}(X, Y) < \epsilon$. Le premier résultat de ce chapitre minore l'espérance de T. Voici donc le théorème 7 qui est le résultat principal de cette thèse. L'énoncé du résultat est suivi d'une définition et de deux lemmes préalables à sa preuve.

Théorème 7. *Soit $X \in \mathbb{R}^d$ de densité f. Soit Y la sortie d'un algorithme de génération pour X telle que* $\operatorname{ess\,sup} \|X - Y\|_p \leq \epsilon$. *Si $\mathcal{E}(\lfloor X_1 \rfloor, \ldots, \lfloor X_d \rfloor) < \infty$,*

alors

$$\mathbf{E}(T) \geq \mathcal{E}(f) + d \log_2 \left(\frac{1}{\epsilon} \right) - \log_2 V_{d,p},$$

où

$$V_{d,p} = \frac{2^d \Gamma\left(\frac{1}{p} + 1\right)}{\Gamma\left(\frac{d}{p} + 1\right)}$$

est le volume de boule de rayon 1 *dans l'espace* $\mathbb{R}^d$ *et* T *est le nombre de bits aléatoires requis pour générer* Y*.*

Définition 13 (graphe des ϵ-voisins)**.** Soit $\mathcal{A}$ une partition quelconque de $\mathbf{R}^d$, $\epsilon > 0$ un paramètre fixe de précision et $p \geq 1$. Un graphe G est appelé le graphe des ϵ-voisins de $\mathcal{A}$ si les sommets de G sont les ensembles disjoints $A \in \mathcal{A}$ et si les arêtes de G sont les couples $(A, B) \in \mathcal{A} \times \mathcal{A}$ telles que

$$\inf_{(x,y) \in A \times B} \|x - y\|_p < \epsilon.$$

Étant donné le graphe G des ϵ-voisins d'une partition quelconque $\mathcal{A}$ de $\mathbb{R}^d$, le couple (A, B) n'est pas une arête de G si $\|x - y\|_p \geq \epsilon$ pour tout $x \in A$ et $y \in B$. Dénotons par Δ le sommet de degré maximal de G. Également, rappelons qu'étant donné une partition $\mathcal{A}$ de $\mathbb{R}^d$ et une une variable aléatoire X, nous avions que $\mathcal{E}_A(X)$ dénote l'entropie de partition de X telle que mentionnée au chapitre 1. Voici donc deux lemmes qui en conjonction avec le théorème 1 du chapitre 1 permettront d'obtenir le résultat principal de cette thèse, le théorème 7.

Lemme 1. *Soit* $X \in \mathbb{R}^d$ *un vecteur aléatoire et* Y *la sortie d'un algorithme de génération pour* X *telle que presque sûrement* $\|X - Y\|_p < \epsilon$*. Nous avons avec la notation précédente que*

$$\mathbf{E}(T) \geq \sup_{\mathcal{A}} \left\{ \mathcal{E}_{\mathcal{A}}(X) - \log_2(\Delta + 1) \right\}.$$

Avant de prouver le lemme précédent, remarquons que nous pouvons maximiser la borne inférieure en sélectionnant la partition $\mathcal{A}$ et le degré maximal Δ qui nous seraient les plus avantageux selon le cas désiré. De plus, la minoration du lemme

1 coïncide avec le résultat de Shannon dans [40] lorsque X est discrète et ayant un nombre fini d'atomes, car en choisissant ϵ suffisamment petit, nous avons que $\Delta = 0$.

Preuve du lemme 1. Soit X et Y deux vecteurs aléatoires dans $\mathbb{R}^d$ et dénotons par $p_{AB} = \mathbf{P}\{X \in A, Y \in B\}$. Nous avons que $p_{AB} = 0$ si (A, B) n'est pas une arête du graphe G des voisins. Donc

$$|\mathcal{E}_{\mathcal{A}}(X) - \mathcal{E}_{\mathcal{A}}(Y)| = \left| \sum_{A \in \mathcal{A}} \mathbf{P}\{X \in A\} \log_2 \left(\frac{1}{\mathbf{P}\{X \in A\}} \right) - \sum_{A \in \mathcal{A}} \mathbf{P}\{Y \in A\} \log_2 \left(\frac{1}{\mathbf{P}\{Y \in A\}} \right) \right| \tag{3.1}$$

$$= \sum_{(A,B) \in \mathcal{A} \times \mathcal{A}} \mathbf{P}\{X \in A, Y \in B\} \log_2 \left(\frac{\mathbf{P}\{Y \in B\}}{\mathbf{P}\{X \in A\}} \right) \tag{3.2}$$

$$\leq \log_2 \sum_{(A,B) \in \mathcal{A} \times \mathcal{A}} \left(\frac{\mathbf{P}\{X \in A, Y \in B\}}{\mathbf{P}\{X \in A\}} \mathbf{P}\{Y \in B\} \right) \tag{3.3}$$

$$= \log_2 \left(\left(\sum_{B \in \mathcal{A}} \mathbf{P}\{Y \in B\} \right) \left(\sum_{A \in \mathcal{A}} \frac{\mathbf{P}\{X \in A, Y \in B\}}{\mathbf{P}\{X \in A\}} \right) \right) \tag{3.4}$$

$$\leq \log_2 \left(\left(\sum_{B \in \mathcal{A}} \mathbf{P}\{Y \in B\} \right) (\Delta + 1) \right) \tag{3.5}$$

$$= \log_2(\Delta + 1). \tag{3.6}$$

Le passage de la ligne (3.2) à la ligne (3.3) s'explique par l'inégalité de Jensen (cf. Hewitt et Stromberg [26]). Le passage de la ligne (3.4) à la ligne (3.5) s'explique par le fait qu'il y a au plus $\Delta + 1$ valeurs de $p_{AB} = \mathbf{P}\{X \in A, Y \in B\}$ et que $p_{AB} \leq 1$.

Si T dénote le nombre de bits aléatoires pour générer Y d'instance A avec probabilité $\mathbf{P}\{Y \in A\}$, alors

$$\begin{aligned} \mathbf{E}(T) &\geq \mathcal{E}_{\mathcal{A}}(Y) \text{ par Knuth et Yao} \\ &\geq \mathcal{E}_{\mathcal{A}}(X) - \log_2(\Delta + 1) \text{ par la ligne (3.6).} \end{aligned} \tag{3.7}$$

Puisque la partition $\mathcal{A}$ est arbitraire, nous avons que

$$\mathbf{E}(T) \geq \sup_{\mathcal{A}} \big\{ \mathcal{E}_{\mathcal{A}}(X) - \log_2(\Delta + 1) \big\}.$$

□

Dans le lemme suivant, λ dénote la mesure de Lebesgue. Pour $h > 0$, la grille de $\mathbb{R}^d$ est dénotée par $\mathcal{A}_h^\star$ c'est-à-dire

$$\mathcal{A}_h^\star = \bigcup_{i \in \mathbb{Z}^d} \big[i_1 h, (i_1 + 1)h \big) \times \ldots \times \big[i_d h, (i_d + 1)h \big).$$

Lemme 2. *Soit une partition $\mathcal{A}$ de $\mathbb{R}^d$ et une variable aléatoire $X \in \mathbb{R}^d$ de densité f. Si $\mathcal{E}(\lfloor X_1 \rfloor, \ldots, \lfloor X_d \rfloor) < \infty$, alors*

$$\mathcal{E}_A(X) \geq \mathcal{E}(f) + \sum_{A \in \mathcal{A}} \mathbf{P}\{X \in A\} \log_2 \left(\frac{1}{\lambda(A)} \right).$$

Si $\mathcal{A} = \mathcal{A}_h^\star$ et $h > 0$, alors

$$\mathcal{E}_{\mathcal{A}_h^\star}(X) \geq \mathcal{E}(f) + d \log_2 \left(\frac{1}{h} \right).$$

Démonstration. Fixons $A \in \mathcal{A}$. Si Z est uniforme sur A et $Y = f(Z)$, alors

$$\mathbf{P}\{X \in A\} = \int_A f = \lambda(A) \mathbf{E}(Y).$$

Ainsi

$$\frac{\mathbf{P}\{X \in A\}}{\lambda(A)} \log_2 \left(\frac{\lambda(A)}{\mathbf{P}\{X \in A\}} \right) = \mathbf{E}(Y) \log_2 \left(\frac{1}{\mathbf{E}(Y)} \right),$$

et, par l'inégalité de Jensen et la concavité de $x \log_2 \big(1/x\big)$,

$$\begin{aligned} \mathbf{E}(Y) \log_2 \left(\frac{1}{\mathbf{E}(Y)} \right) &\geq \mathbf{E}\left(Y \log_2 \left(\frac{1}{Y} \right) \right) \\ &= \frac{1}{\lambda(A)} \int_A f \log_2 \left(\frac{1}{f} \right). \end{aligned}$$

En sommant sur tous les ensembles $A \in \mathcal{A}$, nous complétons ainsi la preuve. □

Preuve du théorème 7. Soit la grille $\mathcal{A}_h^\star$ de $\mathbf{R}^d$ avec $h > 0$. Par le lemme 1,

$$\mathbf{E}(T) \geq \sup_h \left(\mathcal{E}_{\mathcal{A}_h^\star}(X) - \log_2 \left(\Delta_h + 1 \right) \right)$$

où Δ_h est le degrée maximal du graphe des h-voisins défini sur $\mathcal{A}_h^\star \times \mathcal{A}_h^\star$ en reliant deux noeuds $A \in \mathcal{A}_h^\star$ et $B \in \mathcal{A}_h^\star$ si $\inf_{x \in A, y \in B} \|x - y\|_p < \epsilon$. Posons $h = \epsilon/n$ et ainsi

$$\mathbf{E}(T) \geq \limsup_{n \to \infty} \left(\mathcal{E}_{\mathcal{A}_{\epsilon/n}^\star}(X) - \log_2 \left(\Delta_{\epsilon/n} + 1 \right) \right).$$

Si B_r dénote la boule de rayon r centrée en 0 selon la norme ℓ_p, alors

$$\frac{\lambda(B_\epsilon)}{h^d} \leq \Delta_h \leq \frac{\lambda\big(B_{\epsilon + 2hd^{1/p}}\big)}{h^d}$$

de sorte que lorsque $n \to \infty$,

$$\Delta_{\epsilon/n} \sim \left(\frac{n}{\epsilon} \right)^d \lambda(B_\epsilon) = V_{d,p} n^d.$$

Aussi, par le lemme 2,

$$\mathcal{E}_{\mathcal{A}_{\epsilon/n}^\star(X)} \geq \mathcal{E}(f) + d \log_2 \left(\frac{n}{\epsilon} \right)$$

de sorte que

$$\begin{aligned}
&\mathcal{E}_{\mathcal{A}_{\epsilon/n}^\star(X)} - \log_2 \left(\Delta_{\epsilon/n} + 1 \right) \\
&\quad \geq \mathcal{E}(f) + d \log_2 \left(\frac{1}{\epsilon} \right) + \log_2 \left(\frac{n^d}{1 + V_{d,p} n^d (1 + o(1))} \right) \\
&\quad \overset{n \to \infty}{\to} \mathcal{E}(f) + d \log_2 \left(\frac{1}{\epsilon} \right) - \log_2 V_{d,p}.
\end{aligned}$$

□

Nous allons maintenant considérer la *méthode de partition* et obtenir une ma-

joration pour la complexité espérée de l'algorithme sous-jacent. Considérons une variable aléatoire $X \in \mathbb{R}^d$. Étant donné $\epsilon > 0$, nous appelons $\mathcal{A}_\epsilon$ une ϵ-partition si pour tout ensemble $A \in \mathcal{A}_\epsilon$, il existe $x_A \in A$ (le centre) tel que

$$\sup_{y \in A} \|x_A - y\|_p \leq \epsilon.$$

Tout algorithme qui sélectionne aléatoirement $A \in \mathcal{A}_\epsilon$ avec probabilité $p(A) \stackrel{\text{def}}{=} \mathbf{P}\{X \in A\} = \int_A f$ peut être utilisé pour générer une variable aléatoire Y qui approxime X avec une précision ϵ. Une fois que A est généré, l'algorithme retourne

$$Y = x_A.$$

Par conséquent, X et Y sont couplées de sorte que $\|X - Y\|_p \leq \epsilon$. L'algorithme de Knuth et Yao peut être utilisé pour sélectionner A. Si T dénote comme à l'habitude le nombre de bits aléatoires requis pour sélectionner A, alors

$$\mathbf{E}(T) \leq \mathcal{E}_{\mathcal{A}_\epsilon}(X) + 2.$$

Pour $p = \infty$, nous prenons $\mathcal{A}_\epsilon = \mathcal{A}^\star_{2\epsilon}$, la grille dont les côtés sont de longueur 2ϵ. Lorsque $d = 1$, peu importe la valeur de p, la partition en sous-intervalles de longueur 2ϵ peut être utilisée.

Si la densité de X est f et que $p = \infty$ ou $d = 1$, alors, lorsque $\epsilon \downarrow 0$, nous avons pour la procédure suggérée précédemment que

$$\begin{aligned}
\mathbf{E}(T) &\leq \mathcal{E}_{\mathcal{A}^\star_{2\epsilon}}(X) + 2 && (3.8)\\
&\leq \mathcal{E}(f) + d \log_2 \left(\frac{1}{2\epsilon}\right) + 2 + o(1) && (3.9)\\
&= \mathcal{E}(f) + d \log_2 \left(\frac{1}{\epsilon}\right) + 2 - d + o(1). && (3.10)
\end{aligned}$$

Le passage de la ligne (3.9) à la ligne (3.10) est possible sous l'hypothèse que $\mathcal{E}(\lfloor X_1 \rfloor, \ldots, \lfloor X_d \rfloor) < \infty$ et $\mathcal{E}(f) > -\infty$.

Notons qu'en comparant (3.10) avec la minoration

$$\mathbf{E}(T) \geq \mathcal{E}(f) + d \log_2 \left(\frac{1}{\epsilon}\right) - d,$$

nous constatons une différence de $2 + o(1)$.

Nous rappelons les valeurs de $\mathcal{E}(f)$ pour les distributions suivantes :

$$\begin{aligned} \text{Uniform}[0,1]: &\quad \mathcal{E}(f) = 0, \\ \text{Exponential}(1): &\quad \mathcal{E}(f) = \log_2(e), \\ \text{Normal}(0,1): &\quad \mathcal{E}(f) = \log_2 \sqrt{2\pi e}. \end{aligned}$$

Lorsque $X \in \mathbb{R}^1$ et $a > 0$, nous avons que

$$\mathcal{E}(aX) = \mathcal{E}(X) + \log_2(a).$$

Lorsque $p \in [1, \infty)$, nous considérons la grille dont les côtés sont de longueur $2\epsilon/d^{\frac{1}{p}}$ c'est-à-dire

$$\mathcal{A}_\epsilon = \mathcal{A}^\star_{2\epsilon/d^{\frac{1}{p}}}$$

de sorte que si $\mathcal{E}(\lfloor X_1 \rfloor, \ldots, \lfloor X_d \rfloor) < \infty$ et $\mathcal{E}(f) > -\infty$, alors

$$\begin{aligned} \mathbf{E}(T) &\leq \mathcal{E}_{\mathcal{A}^\star_{2\epsilon/d^{\frac{1}{p}}}} + 2 \\ &\leq d \log_2 \left(\frac{1}{\epsilon}\right) + \mathcal{E}(f) + 2 - d + \frac{d}{p} \log_2(d) + o(1). \end{aligned}$$

En comparant avec la minoration, nous obtenons une différence

$$D = 2 + \frac{d}{p} \log_2(d) + d \log_2 \Gamma\left(\frac{1}{p} + 1\right) - \log_2 \Gamma\left(\frac{d}{p} + 1\right) + o(1).$$

En utilisant l'inégalité $\Gamma\big(1+u\big) \geq \big(u/e\big)^u \sqrt{2\pi u}$, $u > 0$, nous obtenons

$$D \leq 2 + d \log_2 \left(\Gamma\left(\frac{1}{p} + 1\right)(ep)^{\frac{1}{p}}\right) - \frac{1}{2} \log_2 \left(2\pi \frac{d}{p}\right) + o(1),$$

qui croît linéairement avec d. Pour éviter cette croissance en d que nous n'avons pas lorsque $p = \infty$, il semble nécessaire de considérer des partitions ayant des plus petites densités de recouvrement. Pour plus d'information, à propos des problèmes d'empilement et de recouvrement, voir Conway et Sloane [12].

Nous allons maintenant considérer la *méthode de l'inversion* qui génère une variable X de loi de répartion F en utilisant la propriété que la variable

$$X = F^{-1}(U)$$

est distribuée comme F où F^{-1} dénote l'inverse de F et U uniformément distribuée sur $[0, 1]$. Si

$$U = 0.U_1U_2\cdots = \sum_{j=1}^{\infty} \frac{U_j}{2^j},$$

et que U_1, U_2, … sont des bits aléatoires indépendants non biaisés, alors en posant

$$U_{(t)} = 0.U_1 \cdots U_t,$$
$$U^+_{(t)} = 0.U_1 \cdots U_t + \frac{1}{2^t}$$

nous avons que

$$U_{(t)} \leq U \leq U^+_{(t)}.$$

Remarquons que $U_{(0)} = 0$, $U_{(1)} = 0.1111\cdots = 1$. Visuellement, nous avons la figure 3.1.

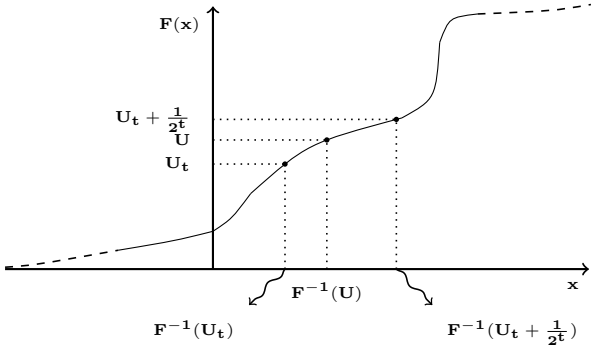

Figure 3.1 – Illustration de la méthode de l'inversion

Le nombre de bits aléatoires est

$$T = \min\{t \geq 0 \ : \ F^{-1}\big(U_{(t)}^{+}\big) - F^{-1}\big(U_{(t)}\big) \leq 2\epsilon\}.$$

Si nous définissons

$$Y = \frac{F^{-1}\big(U_{(t)}^{+}\big) + F^{-1}\big(U_{(t)}\big)}{2},$$

alors X et Y sont couplées de sorte que

$$|X - Y| \leq \epsilon.$$

La variable T (temps d'arrêt) définie précédemment est également le nombre de bits pour générer Y.

Observons que l'inversion requiert un oracle pour le calcul de F^{-1}. Cependant, nous n'avons pas à tabuler ou calculer les probabilités de chaque cellule comme c'est le cas pour la méthode de discrétisation afin d'exécuter l'algorithme de Knuth et Yao. Ainsi la méthode de l'inversion est plus simple à implanter.

Dans le même esprit, l'inversion mime la méthode de Han et Hoshi et cette observation mène naturellement et simplement à une majoration. Soit la grille $\mathcal{A}_{2\epsilon}^{\star}$ de $\mathbb{R}^1$ en sous-intervalles de longueur 2ϵ. Dénotons les probabilités des sous-intervalles par

$$p(A) = \mathbf{P}\{X \in A\}, \ \ A \in \mathcal{A}_{2\epsilon}^{*}.$$

Supposons qu'un intervalle de $\mathcal{A}_{2\epsilon}^{\star}$ est sélectionné aléatoirement selon la méthode de Han et Hoshi qui aurait utilisé les bits aléatoires U_1, U_2, U_3, … également utilisés par la méthode de l'inversion. Le nombre de bits utilisés par la méthode de l'inversion est plus petit que le nombre de bits utilisés par l'algorithme de Han et Hoshi. Par conséquent,

$$\mathbf{E}(T) \leq \mathcal{E}_{\mathcal{A}_{2\epsilon}^{\star}}(X) + 3$$

et en conjonction avec le lemme 2, nous concluons que

Théorème 8. *Si f est la densité de X, que $\mathcal{E}(\lfloor X_1 \rfloor, \ldots, \lfloor X_d \rfloor) < \infty$ et*

que $\int f \log_2(1/f) > -\infty$*, alors, lorsque* $\epsilon \downarrow 0$*, nous avons que*

$$\mathbf{E}(T) \leq \log_2\left(\frac{1}{\epsilon}\right) + \int f \log_2\left(\frac{1}{f}\right) + 2 + o(1).$$

Remarque 5. En comparant le théorème 8 avec la minoration

$$\mathbf{E}(T) \geq \log_2\left(\frac{1}{\epsilon}\right) + \mathcal{E}(f) - 1,$$

nous avons une différence de $3 + o(1)$.

Remarque 6. La méthode de partition a une majoration plus petite d'une unité pour $\mathbf{E}(T)$. La simplicité de la méthode de l'inversion ne peut être sous-estimée. Nous pouvons rapetisser la différence en ajoutant des hypothèses sur la densité f.

Théorème 9. *Supposons que* X *a une densité bornée et décroissante sur l'intervalle* $[0, \infty)$*. Lorsque* $\epsilon \downarrow 0$*, la complexité espérée de la méthode de l'inversion est*

$$\mathbf{E}(T) \leq \log_2\left(\frac{1}{\epsilon}\right) + \int f \log_2\left(\frac{1}{f}\right) + o(1).$$

Preuve du théorème 9. Définissons $X_t = F^{-1}\big(U_{(t)}\big)$ et $X_t^+ = F^{-1}\big(U_{(t)}^+\big)$ comme sur la figure 3.1. Ainsi

$$\begin{aligned}
\mathbf{E}(T) &= \sum_{t=0}^{\infty} \mathbf{P}\big\{X_t^+ - X_t > 2\epsilon\big\} \\
&\leq \sum_{t=0}^{\infty} \mathbf{P}\Big\{f(X_t^+) < \frac{1}{2^t 2\epsilon}\Big\} \\
&\quad \left(\text{parce que } \frac{1}{2^t} = \int_{X_t}^{X_t^+} f \geq f(X_t^+)(X_t^+ - X_t)\right) \\
&\leq \sum_{t=0}^{\infty} \mathbf{P}\Big\{f(X) < \frac{1}{2^t 2\epsilon}\Big\} + \sum_{t=0}^{\infty} \mathbf{P}\Big\{f(X_t^+) < \frac{1}{2^t 2\epsilon} < f(X)\Big\} \\
&= \mathrm{I} + \mathrm{II}.
\end{aligned}$$

Maintenant

$$\begin{aligned}\mathrm{I} &\leq \mathbf{E}\left\{1 + \log_2 \frac{1}{2\epsilon f(X)}\right\} \\ &= \log_2\left(\frac{1}{\epsilon}\right) + \int f \log_2\left(\frac{1}{f}\right)\end{aligned}$$

et ce même si la dernière intégrale diverge. Le résultat s'ensuit si nous pouvons démontrer que $\mathrm{II} = o(1)$ et, pour cela, notons que

$$\mathrm{II} \leq \sum_{t=0}^{\infty} \mathbf{P}\Big\{f(X_t^+) < \frac{1}{2^t 2\epsilon} \leq f(X_t)\Big\}.$$

Pour une valeur fixe de t, nous constatons que $f(X_t^+) < \frac{1}{2^t 2\epsilon} \leq f(X_t)$ seulement si X est élément de l'intervalle contenant la valeur $\frac{1}{2^t 2\epsilon}$, intervalle qui peut ne pas exister aussi. La probabilité de chaque intervalle est exactement $1/2^t$. Cependant, si $\frac{1}{2^t 2\epsilon} > f(0)$, cet intervalle n'existe pas. Par conséquent,

$$\begin{aligned}\mathrm{II} &\leq \sum_{t=0}^{\infty} \frac{1}{2^t} \mathbb{1}\left\{t \geq \log_2\left(\frac{1}{2\epsilon f(0)}\right)\right\} \\ &\leq 4\epsilon f(0) = o(1).\end{aligned}$$

□

Pour la loi uniforme sur l'intervalle $[0, 1]$, nous avons que $\mathcal{E}(f) = 0$ et, par le théorème 9, que

$$\mathbf{E}(T) \leq \log_2\left(\frac{1}{\epsilon}\right) + o(1).$$

Nous pouvons nous débarrasser du « o(1) » comme suit :

$$\begin{aligned}\mathbf{E}(T) &= \sum_{t=0}^{\infty} \mathbf{P}\{T > t\} \\ &= \sum_{t=0}^{\infty} \sum_{i=0}^{2^t-1} \mathbb{1}_{\{F^{-1}(\frac{i+1}{2^t}) - F^{-1}(\frac{i}{2^t}) > 2\epsilon\}} \frac{1}{2^t}\end{aligned}$$

$$
\begin{aligned}
&= \sum_{t=0}^{\infty} \sum_{i=0}^{2^t-1} \mathbb{1}_{\{\frac{i+1}{2^t} - \frac{i}{2^t} > 2\epsilon\}} \frac{1}{2^t} \\
&= \sum_{t=0}^{\infty} \mathbb{1}_{\{t \leq \log_2(\frac{1}{2\epsilon})\}} \\
&= \left\lfloor \log_2 \left(\frac{1}{2\epsilon}\right) \right\rfloor + 1 \\
&\leq \log_2 \left(\frac{1}{2\epsilon}\right) + 1 \\
&= \log_2 \left(\frac{1}{\epsilon}\right).
\end{aligned}
$$

Le théorème 9 est une amélioration de $1 + o(1)$ bits par rapport à la méthode de partition (avec $d = 1$). Probablement qu'il est possible d'obtenir des majorations similaires pour la méthode des partitions si nous ajoutons des conditions supplémentaires de régularité.

Pour la loi exponentielle, la complexité espérée de la méthode de l'inversion est

$$
\begin{aligned}
\mathbf{E}(T) &\leq \log_2 \left(\frac{1}{\epsilon}\right) + \mathcal{E}(f) + o(1) \\
&= \log_2 \left(\frac{1}{\epsilon}\right) + \log_2(e) + o(1),
\end{aligned}
$$

où $\log_2(e) = 1.443\ldots$ Flajolet et Saheb [20] ont proposé une méthode pour la loi exponentielle telle que

$$
\mathbf{E}\{T\} = \log_2 \left(\frac{1}{\epsilon}\right) + 5.4 + \varphi(\epsilon),
$$

où $|\varphi(\epsilon)| \leq 0.2$ lorsque $\epsilon \downarrow 0$.

Pour la loi normale, Karney [29] a proposé une méthode qui règle le problème de la précision, mais qui n'offre aucune majoration explicite de sa complexité. La complexité espérée pour la méthode de l'inversion est

$$
\mathbf{E}\{T\} = \log_2 \left(\frac{1}{\epsilon}\right) + \log_2 \sqrt{2\pi e} + o(1),
$$

mais le problème est qu'il faut recourir à un oracle pour le calcul de F^{-1}. Même la méthode de partition requiert un oracle non trivial pour le calcul de F. Afin de pallier à ces difficultés, nous proposer d'adapter la méthode, légèrement plus coûteuse, de Box-Müller [5]. La méthode de Box-Müller utilise le fait que si E est une variable exponentielle de moyenne 1 et que (V_1, V_2) est distribuée uniformément sur le cercle de rayon 1, alors la paire

$$\left(\sqrt{2E}V_1, \sqrt{2E}V_2\right)$$

est distribuée selon une loi normale dans $\mathbb{R}^2$ dont le vecteur des moyennes est zéro et dont la matrice de covariance est l'identité. La variable aléatoire $\sqrt{2E}$ possède le nom de Maxwell, sa densité est $re^{-r^2/2}$, $r > 0$, et son entropie différentielle est

$$\begin{aligned}
\mathcal{E}(f_{\text{Maxwell}}) &= \int_0^\infty f_{\text{Maxwell}}(r) \log_2 \left(\frac{1}{f_{\text{Maxwell}}(r)} \right) dr \\
&= \frac{1}{\log 2} \int_0^\infty f_{\text{Maxwell}}(r) \left(\log \left(\frac{1}{r} \right) + \frac{r^2}{2} \right) dr \\
&= \frac{1}{\log 2} \left(\frac{1}{2} \left(\gamma - \log(2) \right) + 1 \right) \\
&= 1.359068\ldots,
\end{aligned}$$

où $\gamma = 0.577215\ldots$ est la constante de Euler-Mascheroni.

Nous ébauchons la procédure qui peut servir également pour des problèmes plus compliqués de génération de variables aléatoires. Supposons que les deux variables normales doivent avoir une précision de ϵ c'est-à-dire que $d = 2$ et $p = \infty$. Nous générons d'abord une variable de Maxwell M par la méthode de l'inversion en notant que

$$\begin{aligned}
F(r) &= 1 - e^{-\frac{r^2}{2}}, \\
F^{-1}(u) &= \sqrt{-2\log(1-u)}.
\end{aligned}$$

La variable M de Maxwell doit avoir une précision de $\frac{\epsilon}{2}$. La loi de Maxwell est

unimodale et son mode est $r = 1$. La probabilité du côté gauche du mode est $1 - \frac{1}{\sqrt{e}}$. Nous sélectionnons aléatoirement entre le côté gauche ou le côté droit en utilisant au plus deux bits. Ensuite, nous appliquons l'inversion sur le côté ainsi sélectionné. Par le théorème 9, nous utilisons T_1 bits aléatoires et

$$\mathbf{E}(T_1) \leq \log_2\left(\frac{2}{\epsilon}\right) + \mathcal{E}\big(f_{\text{Maxwell}}\big) + 2 + o(1).$$

Appelons par M' l'approximation ainsi générée.

Par la suite, nous générons une variable uniforme $U \in [0, 2\pi)$ avec une précision

$$\frac{\epsilon/2}{M' + \big(\epsilon/2\big)}.$$

Appelons par U' l'approximation ainsi générée. $U' \in [0, 2\pi)$ et $|U - U'| \leq \frac{\epsilon/2}{M'+(\epsilon/2)}$. Puisque l'entropie différentielle de U est $\log_2(2\pi)$, nous avons que le nombre, T_2, de bits aléatoires a une espérance majorée par

$$\begin{aligned}
\mathbf{E}(T_2) &\leq \mathbf{E}\left(\log_2\left(\frac{M' + \big(\epsilon/2\big)}{\epsilon/2}\right)\right) + \log_2(2\pi) + o(1) \\
&\leq \mathbf{E}\left(\log_2\left(\frac{M + \big(\epsilon/2\big)}{\epsilon/2}\right)\right) + \log_2(2\pi) + o(1) \\
&= \log_2\left(\frac{2}{\epsilon}\right) + \mathbf{E}\big(\log_2(M)\big) + \log_2(2\pi) + o(1) \\
&\qquad \text{(par le théorème de convergence dominée).}
\end{aligned}$$

L'algorithme retourne

$$\big(M' \sin(U') \ , \ M' \cos(U')\big),$$

et

$$\begin{aligned}
|M' \sin(U') - M \sin(U)| &\leq \epsilon \\
|M' \cos(U') - M \cos(U)| &\leq \epsilon,
\end{aligned}$$

car

$$|\sin(U') - \sin(U)| \leq |U - U'| \leq \frac{\epsilon/2}{M' + \big(\epsilon/2\big)}$$

et de même pour la partie du cosinus. Ensuite,

$$\begin{aligned}
|M'\sin(U') - M\sin(U)| &\leq |M' - M||\sin(U')| + M|\sin(U') - \sin(U)| \\
&\leq |M' - M| + M|U' - U| \\
&\leq \frac{\epsilon}{2} + M\frac{\epsilon/2}{M' + \big(\epsilon/2\big)} \\
&\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} \\
&= \epsilon.
\end{aligned}$$

Enfin, nous avons que la complexité totale espérée est

$$\begin{aligned}
\mathbf{E}(T_1) + \mathbf{E}(T_2) &\leq 2\log_2\left(\frac{2}{\epsilon}\right) + \mathbf{E}\big(\log_2(M)\big) + \mathcal{E}(f_{\text{Maxwell}}) + 2 + \log_2(2\pi) + o(1) \\
&= 2\log_2\left(\frac{1}{\epsilon}\right) + 2 + \log_2(2\pi e) + o(1) \\
&= 2\log_2\left(\frac{1}{\epsilon}\right) + 6.094191\ldots + o(1).
\end{aligned}$$

En comparant avec la minoration pour générer deux normales indépendantes, nous obtenons une différence de $4 + o(1)$ c'est-à-dire

$$2\log_2\left(\frac{1}{\epsilon}\right) + 2\log_2\sqrt{2\pi e} - 2 = 2\log_2\left(\frac{1}{\epsilon}\right) + 2.094191\ldots$$

CHAPITRE 4

ÉCHANTILLONNAGE EXACT DE LA DISTRIBUTION QUANTIQUE DISCRÈTE DE GHZ ET COMPLEXITÉ DE LA COMMUNICATION

Dans ce chapitre, il n'y aura qu'une seule distribution, qui est issue du monde de la physique quantique. Elle est fondamentale en informatique quantique. Le contexte est particulier dans la mesure où nous devons tenter d'échantillonner exactement une distribution discrète en ne possédant qu'une partie de l'information décrivant ladite distribution. La distribution porte le nom de GHZ des noms de Greenberger, Horne et Zeilinger [24]. C'est une distribution caractérisée par un vecteur de paramètres de taille n, chaque paramètre étant une paire d'angles (θ, φ). La caractérisation complète pour des valeurs arbitraires de n fut réalisée par Gravel [22, 23].

Les réalisations sont des éléments de l'ensemble $\{+1, -1\}^n$ au lieu de l'ensemble plus conventionnel $\{0, 1\}^n$. Étant donné un vecteur $b = (b_1, \ldots, b_n) \in \{-1, +1\}^n$, nous dénotons la distribution de probabilité par p,

$$p(b) = \cos^2\left(\frac{\theta}{2}\right) p_1(b) + \sin^2\left(\frac{\theta}{2}\right) p_2(b) \text{ avec} \tag{4.1}$$

$$\theta = \sum_{j=1}^{n} \theta_j \ ,$$

$$p_1(b) = \frac{1}{2}\big(\mathrm{a}_1(b) + \mathrm{a}_2(b)\big)^2 \ ,$$

$$p_2(b) = \frac{1}{2}\big(\mathrm{a}_1(b) - \mathrm{a}_2(b)\big)^2 \ , \tag{4.2}$$

$$\mathrm{a}_1(b) = \prod_{j=1}^{n} \cos\left(\tfrac{1}{2}\big(\varphi_j - \tfrac{\pi}{2} b_j\big)\right) \ ,$$

$$\mathrm{a}_2(b) = \prod_{j=1}^{n} -\sin\left(\tfrac{1}{2}\big(\varphi_j - \tfrac{\pi}{2} b_j\big)\right). \tag{4.3}$$

La distribution $p(b)$ est une combinaison convexe des deux distributions $p_1(b)$ et $p_2(b)$. Notons que les distributions p_1 et p_2 ne dépendent que des paramètres φ_j et que les coefficients $\cos^2(\theta/2)$ ainsi que $\sin^2(\theta/2)$ ne dépendent que des paramètres θ_j. De plus, notons que a_1^2 et a_2^2 sont des distributions de probabilités discrètes.

Expliquons le contexte de génération dès maintenant.

1 Supposons n ordinateurs (ou personnes) où chaque ordinateur possède un et un seul (φ_j, θ_j) pour $j \in \{1, \ldots, n\}$.

2 Désignons le 1er ordinateur comme le *serveur*. Les $n-1$ autres ordinateurs (*clients*) peuvent communiquer au serveur et vice-versa. Cependant les $n-1$ clients ne peuvent pas communiquer entre eux.

3 Supposons que le serveur et les clients peuvent générer des bits non biaisés i.i.d.

Visuellement, nous avons le schéma de communication suivant :

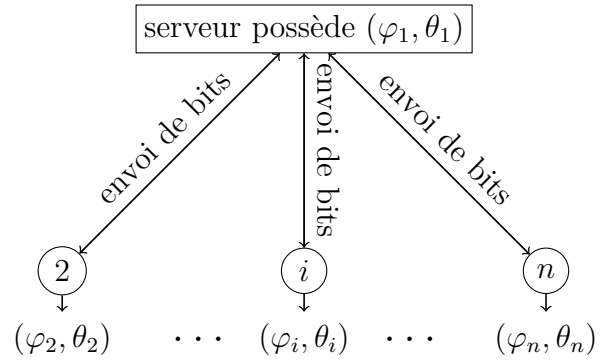

Figure 4.1 – Contexte communicationnel de l'échantillonnage de la distribution de GHZ

Pour plus d'information sur la complexité de la communication, consulter Kushilevitz et Nisan [31].

Comment le serveur doit-il procéder afin d'échantillonner p *exactement* ? Quelle est l'espérance du nombre de bits que le serveur échange avec les clients ? Quelle est l'espérance du nombre de bits aléatoires non-biaisés i.i.d. à générer ? Ce sont

ces questions auxquelles nous répondons. Notons que le problème a des origines lointaines qui remontent jusqu'à l'article célèbre [17] d'Einstien, Podolsky et Rosen en 1935. S'ensuivit une suite d'articles dont les plus connus sont : Bell [2], Greenberger, Horne et Zeilinger [24], Maudlin [35], Brassard, Cleve et Tapp [10], Steiner [42], Massar, Bacon, Cerf et Cleve [34], Brassard [7], Broadbent, Chouha et Tapp [11], Bancal, Branciard et Gisin [1], Branciard et Gisin [6] et Brassard et Kaplan [8].

Un fait crucial concernant p est que pour tout $b \in \{-1,+1\}^n$,

$$p(b) \leq p_1(b) + p_2(b) = 2\bigg(\frac{1}{2}\Big(p_1(b) + p_2(b)\Big)\bigg). \tag{4.4}$$

Pour tout $b \in \{-1,+1\}^n$, définissons la distribution q par

$$q(b) = \frac{1}{2}\Big(p_1(b) + p_2(b)\Big).$$

Par conséquent, $p(b) \leq 2q(b)$. L'inégalité (4.4) nous amène à considérer l'utilisation de l'algorithme d'acceptation et de rejet de von Neumann, algorithme que nous modifions et adaptons à notre contexte. Remarquons, et cela sera également important pour la suite, que, pour tout $b \in \{-1,+1\}^n$,

$$\varphi_1 \mapsto \varphi_1 + 2\pi \implies \ a_2(b) \mapsto -a_2(b) \implies \ p_1(b) \mapsto p_2(b). \tag{4.5}$$

Pour plus d'information concernant d'autres propriétés de la distribution de GHZ, consulter Gravel [22, 23].

De façon générale, supposons deux fonctions de masse f et g (ou bien des densités, mais seul le cas discret nous intéresse ici) ayant le même support de définition. S'il existe une constante $C > 1$ telle que pour tout x élément du support de f, nous avons $f(x) \leq Cg(x)$, alors l'algorithme suivant dû à von Neumann paru originellement en 1951 dans [37] échantillonne f.

Algorithme 4.1 : Algorithme d'acceptation et de rejet de von Neumann

1: **répéter**

2: Générer X selon g.
3: Générer U uniformément continue sur $[0, 1]$.
4: **si** $UCg(X) \leq f(X)$ **alors**
5: **retourner** X
6: **sinon** {Rejet de X}
7: Continuer.
8: **fin si**
9: **fin répéter**

La fonction de masse f est « difficile » à échantillonner. La fonction de masse g est supposée « facile » à échantillonner. En ce qui nous concerne, la distribution facile à échantillonner est q. Par convention, supposons que le serveur est l'ordinateur numéroté par 1 et les clients sont respectivement numérotés par $2, \ldots, n$. Pour échantillonner $q = \frac{1}{2}\big(a_1^2 + a_2^2\big)$ (rappel : $p_1 + p_2 = a_1^2 + a_2^2$, nous procédons ainsi :

1 Pour $j \in \{2, \ldots, n\}$, chaque client génère sa variable de Rademacher B_j selon $\mathbf{P}\{B_j = b_j\} = \cos\big(\frac{1}{2}\big(\varphi_j - \frac{\pi}{2} b_j\big)\big)$ et l'envoie au serveur.

2 Le serveur génère aussi sa variable de Rademacher c'est-à-dire B_1.

3 Le serveur génère un bit aléatoire non biaisé qu'il envoie aux clients. Si le bit est 1, alors, les clients changent le signe de leur variable B_j sinon le signe reste inchangé.

Le problème est d'échantillonner p exactement étant donné le vecteur $(B_1, \ldots, B_n) = B$ distribué selon q. En effet, le serveur doit apprendre suffisamment du coefficient $\cos^2\big(\frac{\theta}{2}\big)$ des lignes (4.1) ainsi que des valeurs $\cos\big(\frac{1}{2}\big(\varphi_j - \frac{\pi}{2} B_j\big)\big)$ afin d'échantillonner p_1 ou p_2 exactement. Que voulons-nous dire par apprendre suffisamment ? Cela signifie connaître des approximations suffisamment précises de p_1 et p_2. De plus, par la ligne (4.5), simuler p_2 se réduit à simuler p_1.

Définition 14 (k-approximation)**.** Une k-approximation d'une quantité v est n'importe quelle quantité $\hat{v}$ telle que $|v - \hat{v}| \leq 2^{-k}$.

Un cas spécial d'une k-approximation est une k-troncation et, dans ce cas, $\hat{v} = \lfloor v2^k \rfloor / 2^k$. Parfois, nous utiliserons le terme k-approximation pour k-troncation et, évidemment, jamais l'inverse.

Nous allons maintenant procéder par étapes pour modifier l'algorithme de von Neumann. Le plan, espérons-le didactique, à suivre pour se rendre au protocole est le suivant :

1. Analyser le cas simple de générer une variable de Bernoulli avec paramètre $p \notin \{0, 1\}$ (à ne pas confondre avec le p précédent) lorsque les bits de l'expansion de p nous parviennent séquentiellement sur un canal de communication et que nous avons une variable continue uniforme $U \in [0, 1]$. Quelle est la complexité espérée du nombre de bits à lire du développement binaire de U ? Combien de bits de p (troncation) devons-nous connaître ?

2. Analyser le cas de générer une variable de Bernoulli avec paramètre p lorsque nous possédons une approximation de p et une variable continue uniforme $U \in [0, 1]$. Quelle est la complexité espérée du nombre de bits à lire du développement binaire U ? Quelle précision d'approximation pour p devons-nous atteindre avant de s'arrêter (accepter ou rejeter) ? À cette étape, nous aurons résolu le problème de choisir entre p_1 et p_2. En effet, pour échantillonner p, il faut choisir entre p_1 ou p_2 selon la probabilité $\cos^2\big(\theta/2\big)$ comme l'indique la ligne (4.1).

3. Analyser le cas de générer une variable (vecteur) selon $p_1(b)$ étant donné $B = (B_1, \ldots, B_n)$ distribué selon q ainsi que les approximations de p et de q, plus spécifiquement de $2q$? Quelle la complexité espérée du nombre de bits à lire du développement binaire de U et quelle(s) précision(s) respective(s) de p_1 et de Cq devons-nous atteindre avant de nous arrêter (accepter ou rejeter) ? Nous montrons également comment obtenir ces approximations.

Allons-y avec **l'étape 1** de notre plan. Supposons deux personnes dont l'une possède une source de bits aléatoires et l'autre la valeur du paramètre $p \notin \{0, 1/$

$2, 1\}$ définissant une loi de Bernoulli. Cette étape correspond à notre exemple de l'introduction. Étant donnée une variable U uniformément distribuée sur $[0, 1]$ dont l'expansion binaire est $U = 0.U_1U_2\cdots = \sum_{k=1}^{\infty} U_k 2^{-k}$ ainsi que $p = \sum_{k=1}^{\infty} p_k 2^{-k}$, nous comparons U_k et p_k et s'arrêtons dès que $U_k \neq p_k$. Si K est la variable aléatoire du temps d'arrêt, alors nous retournons 1 si $U_K > p_K$ sinon nous retournons 0. Ainsi, comme nous l'avons fait dans l'introduction, $\mathbf{E}(K) = 2$.

Allons-y avec **l'étape 2** de notre plan. Dans l'algorithme qui suit, $U(k)$ dénote la k-troncation de U c'est-à-dire $0.U_1\cdots U_k = U(K)$ et $p(k)$ dénote une k-approximation par rapport à p.

Algorithme 4.2 : Modification de la condition d'arrêt dans l'algorithme de von Neumann pour une loi de Bernoulli

```
1: k ← 1
2: U(0) ← 0
3: répéter
4:    Générer un bit non biaisé i.i.d. U_k
5:    U(k) ← U(k − 1) + U_k/2^k {donc U(k) = 0.U_1 ... U_k}
6:    Obtenir p(k) telle que |p(k) − p| ≤ 1/2^k
7:    si U(k) ≤ p(k) − 2/2^k alors
8:       retourner Y = 1
9:    sinon si U(k) ≥ p(k) + 1/2^k alors
10:      retourner Y = 0
11:   sinon {Impossible de prendre une décision donc continuer}
12:      k ← k + 1
13:   fin si
14: fin répéter
```

Pour tout naturel $k \geq 0$, supposons obtenue $p(k)$ telle que $|p(k) - p| \leq 1/2^k$. Nous avons que $U(k) \leq U \leq U(k) + 2^{-k}$ pour tout k. Si $U(k) + 1/2^k \leq p(k) - 1/2^k$, qui est équivalent à $U(k) \leq p(k) - 2/2^k$, alors $U \leq p$. Si $U(k) \geq p(k) + 1/2^k$ alors $U \geq p$. Le cas $U = p$ a une probabilité de 0. Ainsi l'algorithme 4.2 est correct. Le nombre d'itérations avant de décider de l'acceptation ou du rejet de Y est une variable aléatoire. Dénotons par K ce temps d'arrêt. Puisque

$$\mathbf{P}\{K > k\} \leq \mathbf{P}\left\{|U(k) - p(k)| \leq \frac{2}{2^k}\right\} \leq \mathbf{P}\left\{|U - p| \leq \frac{4}{2^k}\right\} \leq \frac{8}{2^k},$$

on

$$\mathbf{E}\{K\} = \sum_{k=0}^{\infty} \mathbf{P}\{K > k\} \leq \sum_{k=0}^{\infty} \min\left(1, \frac{8}{2^k}\right) = 5.$$

Ainsi, il faut espérer générer 5 bits aléatoires et être capable d'obtenir une 5-approximation du paramètre p. Comme nous nous y attendions, il faut donc davantage de bits aléatoires avec une approximation que si nous possédions une troncation.

Allons-y avec **l'étape 3** de notre plan. Supposons que nous avons généré un vecteur $B = (B_1, \ldots, B_n)$ selon la distribution q. Rappelons que $q(b) = (1/2)(p_1(b) + p_2(b))$ et p_1, p_2, ainsi que p sont données aux lignes (4.1), (4.2) et (4.3). L'inégalité $2q(B) \leq p(B)$ permet de décider si nous rejetons ou acceptons. Supposons que L_k est une k-approximation de $2q(B) = 2q(B) = a_1^2(B) + a_2^2(B)$ et R_k est une k-approximation de $p(B)$. L'algorithme suivant échantillonne p étant donnés B, L_k et R_k et, de plus, $q_1 \stackrel{\text{def}}{=} a_1^2$ et $q_2 \stackrel{\text{def}}{=} a_2^2$.

Algorithme 4.3 : Modification de la condition d'arrêt dans l'algorithme de von Neumann pour la distribution de GHZ

---

entrée(s) : $B \in \{-1, +1\}^n$ distribué selon $q = \frac{q_1+q_2}{2}$

1: $k \leftarrow 1$
2: $U(0) \leftarrow 0$
3: **répéter**
4: Générer un bit non biaisé i.i.d. U_k
5: $U(k) \leftarrow U(k-1) + U_k/2^k$ {donc $U(k) = 0.U_1 \ldots U_k$}
6: Obtenir L_k et R_k {L_k et R_k dépendent de B}
7: **si** $U(k)\, L_k - R_k < -\frac{4}{2^k}$ **alors**
8: **retourner** B {B est accepté}
9: **sinon si** $U(k)\, L_k - R_k > \frac{4}{2^k}$ **alors**
10: Recommencer sur une nouvelle entrée. { B est rejetée}
11: **sinon** {Impossible de décider - continuer}
12: $k \leftarrow k + 1$
13: **fin si**
14: **fin répéter**

L'exactitude de l'algorithme s'explique de façon similaire à celle de l'algorithme 4.3. Remarquons que l'algorithme 4.3 s'arrête avec probabilité 1 puisque $|Cq - L_k| \leq 2^{-k} \to 0$ et $|p_1 - R_k| \leq 2^{-k} \to 0$ lorsque $k \to \infty$ pour tout $B \in \{-1, +1\}^n$.

Nous procédons immédiatement avec la majoration de complexité espérée $\mathbf{E}(K)$. S'ensuivra une explication quant à l'obtention des k-approximations de p et Cq par le serveur et enfin le protocole d'échantillonnage.

Étant donné B, si l'algorithme ne s'est pas arrêté c'est-à-dire si $K > k$, alors

$$\begin{aligned}
|U(q_1(B)+q_2(B))-p_1(B)| &= \big|\big(U(q_1(B)+q_2(B))-U(k)L_k\big)+\big(R_k-p_1(B)\big)+\big(-R_k+U(k)L_k\big)\big| \\
&\leq |U(q_1(B)+q_2(B))-U(k)L_k|+|R_k-p_1(B)|+|R_k-U(k)L_k| \\
&\leq \frac{3}{2^k}+\frac{1}{2^k}+\frac{4}{2^k} \\
&= \frac{8}{2^k}.
\end{aligned}$$

Par conséquent,

$$\begin{aligned}
\mathbf{P}\{K > k \mid B\} &\leq \mathbf{P}\{|U(q_1(B)+q_2(B))-p_1(B)| \leq 8/2^k \mid B\} \\
&= \mathbf{P}\left\{U \in \left(\frac{p_1(B)}{2q(B)}-\frac{1}{2}\frac{8}{2^k}\frac{1}{q(B)}\ ,\ \frac{p_1(B)}{2q(B)}+\frac{1}{2}\frac{8}{2^k}\frac{1}{q(B)}\right)\right\} \\
&\leq \frac{8}{2^k}\frac{1}{q(B)}\,.
\end{aligned}$$

Posons $k_0 = \left\lceil 3+\log_2(\frac{1}{q(B)})\right\rceil$ et

$$\begin{aligned}
\mathbf{E}(K \mid B) &= \sum_{k=0}^{\infty}\mathbf{P}\{K > k \mid B\} \\
&\leq \sum_{k=0}^{\infty}\min\left(1, \frac{8}{2^k q(B)}\right) \\
&\leq \sum_{k<k_0} 1 + \sum_{k\geq k_0}\frac{8}{2^k q(B)} \\
&\leq 5+\log_2\left(\frac{1}{q(B)}\right).
\end{aligned}$$

Maintenant, en déconditionnant

$$\mathbf{E}(K) \leq 5+\sum_{b\in\{-1,+1\}^n} q(b)\log_2\left(\frac{1}{q(b)}\right)$$

$$= 5 + \mathcal{E}(q) \qquad \leq n + 5. \tag{4.6}$$

Comment le serveur obtient les k-approximations de p_1 et de $2q$? Selon les lignes (4.1), (4.2) et (4.3), les distributions p et q ne sont que des produits de cosinus et de sinus. En effet, si le serveur peut approximer a_1 et a_2, alors il lui est facile d'approximer p et q. Étant donné $B = (B_1, \ldots, B_n)$ où B_j est distribuée selon $\mathbf{P}\{B_j = b_j\} = \cos^2\left(\frac{1}{2}\left(\varphi_i - b_j\frac{\pi}{2}\right)\right)$ il faut calculer des k-approximations de $\mathrm{a}_1(B)$ et $\mathrm{a}_2(B)$. Posons

$$c_j = \cos\left(\frac{1}{2}\left(\varphi_j - \frac{\pi}{2}B_j\right)\right) \text{ et}$$
$$s_j = -\sin\left(\frac{1}{2}\left(\varphi_j - \frac{\pi}{2}B_j\right)\right).$$

Dénotons $\hat{c}_j$ et $\hat{s}_j$ les ℓ-troncations de c_j et s_j respectivement. Il faut déterminer la valeur de ℓ telle que $\prod_{j=1}^n \hat{c}_j$ et $\prod_{j=1}^n \hat{s}_j$ sont des k-approximations de $\mathrm{a}_1(B)$ et $\mathrm{a}_2(B)$ respectivement. Remarquons que chacun (client et serveur) connaît exactement ses valeurs de c_j et s_j et, par conséquent, les clients peuvent transmettent $\hat{c}_j$ et $\hat{s}_j$ directement au serveur. Pour tout c_j, il existe $\epsilon_j \in [-1, 1]$ tel que $c_j = \hat{c}_j + \frac{\epsilon_j}{2^\ell}$. Dénotons par I l'ensemble d'indices $\{1, 2, \ldots, n\}$ et nous avons que

$$\prod_{j=1}^n c_j \;=\; \sum_{A\in\mathcal{P}(I)} \prod_{j\in A} \hat{c}_j \prod_{j\notin A} \frac{\epsilon_j}{2^\ell} \;=\; \prod_{j=1}^n \hat{c}_j \;+\; \sum_{A\in\mathcal{P}(I)\setminus I} \prod_{j\in A} \hat{c}_j \prod_{j\notin A} \frac{\epsilon_j}{2^\ell}.$$

Nous pouvons borner l'erreur :

$$\left|\prod_{j=1}^n c_j - \prod_{j=1}^n \hat{c}_j\right| \;\leq\; \sum_{j=1}^n \left(\binom{n}{j}\frac{1}{2^{j\ell}}\right) - 1 \;=\; \left(1 + \frac{1}{2^\ell}\right)^n - 1\,.$$

Si nous choisissons $\ell = \left\lceil -\log_2\left(\left(1 + 2^{-k}\right)^{1/n} - 1\right)\right\rceil$, alors

$$\left|\prod_{j=1}^n c_j - \prod_{j=1}^n \hat{c}_j\right| \leq \frac{1}{2^k}\,.$$

Remarquons que si $k \geq 1$ et $n \geq 1$, alors

$$-\log_2\Big(\big(1+2^{-k}\big)^{1/n}-1\Big) \leq k+\log_2 n+\frac{1}{2\log 2}.$$

En tenant compte des signes c_j et s_j que nous devons transmettre ainsi que les ℓ bits de leur expansion binaire respective, il faut ainsi communique $2(n-1)(\ell+1) = O(kn+n\log n)$ bits. La valeur espérée de k est $\mathbf{E}(K) \leq n+5$ et il faut ainsi $O(n^2)$ bits de communications pour échantillonner p_1.

Le protocole est donc le suivant :

Algorithme 4.4 : Protocole d'échantillonnage de la distribution GHZ

1: Le serveur, ordinateur numéroté 1, communique avec les $n-1$ clients afin d'obtenir des troncations suffisamment précises des quantités θ_j. Le serveur génère une variable de Bernoulli, dénoté Z, de paramètre $\cos^2(\theta/2)$.
2: **si** $Z = 1$ **alors**
3: $\quad\varphi_1 \leftarrow \varphi_1 + 2\pi$ {p_2 est échantillonnée au lieu de p_1}
4: **fin si**
{Début de l'utilisation de l'algorithme distribué de von Neumann modifié convenablement pour échantillonner p_1.}
5: **répéter**
6: $\quad$Le serveur génère un bit non biaisé S et le diffuse aux $n-1$ clients.
{Le bit S détermine lequel de q_1 ou q_2 doit être échantillonné.}
7: $\quad$Localement et indépendamment, chacun (clients et serveur) génère sa variable de Rademacher $B_j \in \{-1,+1\}$ distribué selon $\cos^2\big(\frac{1}{2}\big(\varphi_j - B_j\frac{\pi}{2}\big)\big)$.
{À ce stade, $(B_1,\ldots,B_n)$ est distribué selon q_1.}
8: $\quad$**si** $S = 1$ **alors**
9: $\quad\quad B_j \leftarrow -B_j$ {Dans ce cas, $(B_1,\ldots,B_n)$ est distribué selon q_2.}
10: $\quad$**fin si**
{Le vecteur $B = (B_1,\ldots,B_n)$ est distribué selon $q = \frac{\mathrm{a}_1^2+\mathrm{a}_2^2}{2}$.}
{Le serveur communique avec les clients afin de déterminer s'il accepte ou rejette B.}
11: $\quad$Chaque client calcule $c_j = \cos\big(\frac{1}{2}\big(\varphi_j - \frac{\pi}{2}B_j\big)\big)$ et $s_j = -\sin\big(\frac{1}{2}\big(\varphi_j - \frac{\pi}{2}B_j\big)\big)$
12: $\quad$Le serveur assigne $k \leftarrow 1$.
13: $\quad$Le serveur assigne $U(0) \leftarrow 0$.
14: $\quad$**répéter**
15: $\quad\quad$Le serveur génère un bit non biaisé i.i.d. U_k.
16: $\quad\quad$Le serveur calcule $U(k) \leftarrow U(k-1) + U_k/2^k$ donc $U(k) = 0.U_1\ldots U_k$.

17: Le serveur demande les $(k+3+\lceil \log_2 n \rceil)$-troncations de c_j et s_j de chaque client $j \geq 2$.
18: Le serveur utilise ces approximations pour calculer les $(k+2)$-approximations de $\mathrm{a}_1(B)$ et $\mathrm{a}_2(B)$ qui sont à leur tour utilisées pour calculer les k-approximations L_k de $\mathrm{a}_1^2(B) + \mathrm{a}_2^2(B)$ et R_k de $p_1(B)$.
19: **si** $U(k)L_k - R_k < -\frac{4}{2^k}$ **alors**
20: $Y \leftarrow 1$ et **sortir de la boucle « répéter »**. {B est accepté.}
21: **sinon si** $U(k)L_k - R_k > \frac{4}{2^k}$ **alors**
22: $Y \leftarrow 0$ et **sortir de la boucle « répéter »**. {B est rejeté.}
23: **sinon** {Impossible de décider}
24: $k \leftarrow k+1$ et **continuer dans la boucle « répéter »**
25: **fin si**
26: **fin répéter**
27: **tant que** $Y = 1$ {Acceptation de B}
28: Le serveur informe les clients que la simulation est complète et, par conséquent, le serveur et les clients révèlent leur variable B_j.

Exactitude du protocole : La partie du protocole avant la boucle de la ligne 5 échantillonne une distribution de Bernoulli avec paramètre $\cos^2\left(\sum_{i=1}^n \theta_i/2\right)$ qui permet donc au serveur de savoir lequel de p_1 ou p_2 il devra échantillonner. La deuxième étape de notre plan montre comment échantillonner exactement une loi de Bernoulli si nous possédions une approximation du paramètre.

La partie du protocole à l'intérieur de la boucle débutant à la ligne 5 et finissant à la ligne 27 est essentiellement l'algorithme de von Neumann. Le serveur doit savoir premièrement laquelle de q_1 ou q_2 doit être échantillonné, ce qui ce fait grâce au bit S qui est diffusé aux $n-1$ clients. Échantillonner q_1 ou q_2 se fait localement et indépendamment, ce qui explique entre autres pourquoi les valeurs de B_j ne sont révélées qu'à la fin lorsque le serveur est sûr que le vecteur B est accepté. Puisque les variables B_j sont obtenues localement, chaque client s'en sert pour calculer ses valeurs de c_j et s_j. Chaque client transmet donc ainsi au serveur les bits de c_j et s_j séquentiellement (bit à bit) au serveur qui obtient donc les troncations $\hat{c}_j$ et $\hat{s}_j$, troncations suffisamment précises permettant d'atteindre la précision requise pour approximer $q_1(B) + q_2(B)$ et $p_1(B)$ à partir de L_k et R_k respectivement. Les approximations L_k et R_k permettent de décider si B est accepté ou rejeté. La

troisième étape de notre plan montre qu'il faut espérer des $(n+5)$-approximations de $q_1(B) + q_2(B)$ et $p_1(B)$ pour décider et, qu'avec probabilité 1, le protocole termine. La variable Y est utilisée comme une indicatrice de la terminaison du protocole. Si $Y = 1$, alors chaque client révèle au serveur sa variable B_j. Le nombre espéré de tours de la boucle « répéter » de la ligne (5) est 2, car $C = 2$.

Complexités espérées du nombre de bits aléatoires : L'espérance du nombre de bits aléatoires est majorée par $17 + 4n + 2\mathcal{E}(q)$. Il suffit de 5 bits en moyenne pour générer la variable de Bernoulli $\cos^2\big(\theta/2\big)$ permettant de décider lequel de p_1 ou de p_2 il faut échantillonner. Ensuite, le serveur génère 1 bit pour le signe des variables B_j et il suffit de 2 bits espérés pour chaque B_j. Ensuite le serveur génère $(5 + \mathcal{E}(q))$ pour échantillonner q. Le nombre de fois qu'il faut recommencer l'échantillonnage de p_1 ou de p_2 est 2. Par conséquent, il suffit de

$$5 + 2\big(1 + 2n + (5 + \mathcal{E}(q))\big) = 17 + 4n + 2\mathcal{E}(q).$$

Rappelons que $\mathcal{E}(q) \leq n$ car la loi uniforme maximise l'entropie de Shannon.

Complexités espérées du nombre de bits de communication : L'espérance du nombre de bits de communication est majoré $(n-1)(10 + 2\lceil\log_2(n)\rceil + \mathcal{E}(q))$. Il suffit de $(n-1)(5+\lceil\log_2(n)\rceil)$ bits espérés à envoyer au serveur pour échantillonner la variable de Bernoulli $\cos^2\big(\theta/2\big)$. Ensuite, pour échantillonner p_1 ou p_2, chaque client envoie au départ $2(5+\lceil\log_2(n)\rceil)$ bits (le 2 provient du fait qu'il faut obtenir des approximations des cosinus et des sinus pour échantillonner p_1 ou p_2). Il faut 2 bits (1 bit pour chaque cos et sin) par itération. Le nombre espéré d'itération avant de pouvoir décider avec certitude est majoré par $5 + \mathcal{E}(q)$. Par conséquent, il suffit de

$$(n-1)(5+\lceil\log_2(n)\rceil)+2(n-1)(\mathcal{E}(q)+5+\lceil\log_2(n)\rceil) = (n-1)(10+2\lceil\log_2(n)\rceil+\mathcal{E}(q)).$$

Enfin, notons que nous pouvons créer d'autres variantes du protocole. Par exemple, nous pouvons restreindre le serveur à être le seul ordinateur possédant

une source de bits aléatoires, ce qui sera fait au chapitre 5. Une variante en parallèle existe dans [9] également. Le but principal dans [9] ainsi que dans cette thèse était de montrer comment échantillonner exactement des distributions discrètes en ne possédant qu'une partie de l'information les caractérisant.

Remarque 7. La façon de borner $p(b)$ par $q(b)$ telle que $p(b) \leq 2q(b)$ est basée sur le type de majoration fourni à l'annexe VI. La majoration du type que nous retrouvons à l'annexe VI est une majoration basée sur la théorie spectrale des opérateurs et elle stipule que n'importe quelle distribution quantique discrète (le résultat est aussi vrai pour le cas continu) est majorée par une distribution qui est la combinaison convexe de produits de Bernoulli indépendantes (ou de Rademacher). Malheureusement, ce type de majoration est inefficace (complexité espérée exponentielle en n) pour résoudre la cas général de toutes les distributions quantiques dans notre contexte particulier inhérent à la communication.

CHAPITRE 5

ÉCHANTILLONNAGE EXACT ET DISTRIBUÉ DES DISTRIBUTIONS QUANTIQUES DISCRÈTES

Dans ce chapitre, nous montrons comment échantillonner exactement n'importe quelles distributions quantiques discrètes dans le même contexte de communication qu'au chapitre 4.

Nous procéderons selon les étapes suivantes :

1. Établir une formulation générale et opérationnelle d'une distribution discrète quantique. Par le terme « opérationnel », nous voulons signifier une formulation permettant l'échantillonnage et surtout ouvrant la porte à l'étape qui suit. La formulation sera établie directement à partir des axiomes de la mécanique quantique.

2. Étant donnée une distribution quantique discrète $p_\theta(a)$ arbitraire avec $a \in \{0,1\}^n$ et θ un vecteur de paramètres de taille n, borner p_θ par une distribution « facile » à échantillonner de la forme $q\tilde{p}_\theta + (1-q)2^{-n}$ pour un certain $q \in (0,1)$. Ce type de majoration est la clef de voûte afin d'utiliser l'algorithme de von Neumann convenablement modifié et adapté au contexte de communication sous-jacent au problème.

3. En supposant que seul le serveur puisse générer des variables continues i.i.d. uniformes dans $[0,1]$, modifier l'algorithme d'acceptation et de rejet de von Neumann afin de simuler p_θ. À cette étape, nous nous concentrerons seulement sur la complexité espérée de bits de communication échangés entre le serveur et les clients.

4. Reprendre l'étape précédente en restreignant le serveur à ne pouvoir générer que des bits non biaisés i.i.d. À cette étape, nous tiendrons compte également du nombre de bits aléatoires utilisés par le serveur.

Allons-y avec **l'étape 1** de notre plan. Spécifions ce qu'est une distribution quantique discrète qui est un sous-ensemble particulier du polyèdre convexe que forment toutes les distributions discrètes. Un peu d'algèbre linéaire s'impose et Godement [21] suffit. Soit ρ une matrice de taille $d \times d$ (très bientôt, $d = 2^n$) définie positive satisfaisant $\mathrm{Tr}(\rho) = 1$ qui est appelée dans le cadre de la mécanique quantique une matrice de densité. Une matrice définie positive est une matrice Hermitienne ayant toutes ses valeurs propres positives. Une matrice Hermitienne a toutes ses valeurs propres positives si et seulement la forme quadratique engendrée par ladite matrice est positive. De plus, soit U une matrice unitaire de taille $d \times d$ c'est-à-dire que $UU^\dagger = U^\dagger U = I$ où $\dagger$ dénote l'opération de transconjugaison. La propriété que $UU^\dagger = U^\dagger U = I$ est équivalente à l'invariance du produit scalaire de deux vecteurs transformés par U. Les rangées et les colonnes de U forment respectivement des bases orthonormées transconjuguées l'une par rapport à l'autre pour l'espace vectoriel $\mathbb{C}^d$. Ici ρ représente l'état d'un système quantique discret. La matrice $U\rho U^\dagger$ représente l'évolution du système une fois transformée par U et initialement dans l'état ρ. Nous dénotons comme en mécanique quantique par $|e_i\rangle$ le vecteur colonne de la base canonique dont la i^{e} coordonnée vaut 1 et toutes ses autres coordonnées valant 0. Pour des raisons de commodité, nous commençons à compter le rang des coordonnées d'un vecteur à 0. De plus $\langle e_i|$ dénote la transconjuguée de $|e_i\rangle$. Une distribution quantique discrète est représentée par le vecteur

$$\big(\langle e_0|U\rho U^\dagger|e_0\rangle, \ldots, \langle e_i|U\rho U^\dagger|e_i\rangle, \ldots, \langle e_{d-1}|U\rho U^\dagger|e_{d-1}\rangle\big). \tag{5.1}$$

Les quantités $\langle e_i|U\rho U^\dagger|e_i\rangle$ sont les éléments de la diagonale de la matrice $U\rho U^\dagger$. Si $X \in \{0, \ldots, d-1\}$ est une variable aléatoire, alors $\mathbf{P}\{X = i\} = \langle e_i|U\rho U^\dagger|e_i\rangle$. Pour davantage d'information sur les fondements statistiques de la mécanique quantique, nous vous suggérons l'excellent livre [27] d'Holevo.

En utilisant la définition du produit matriciel,

$$\begin{aligned}\mathbf{P}\{X=i\} &= \left(U\rho U^{\dagger}\right)_{ii} \\ &= \sum_{k=0}^{d-1}\sum_{\ell=0}^{d-1} u_{ik}\rho_{k\ell}\bar{u}_{i\ell} \\ &\stackrel{\text{def}}{=} p_{\theta}(i). \end{aligned} \tag{5.2}$$

Remarque 8 (rappel du contexte de communication). Le serveur connaît les entrées $\rho_{k\ell}$ de la matrice ρ. Le serveur doit communiquer avec les $(n-1)$ clients afin d'apprendre suffisamment d'information concernant les entrées u_{ik} de la matrice U qui est le produit de Kronecker de n matrices unitaires de taille 2×2. Par conséquent, $U=\bigotimes_{j=1}^{n} U_j$ et U_j est une matrice unitaire de taille 2×2. Le fait que U s'exprime comme le produit tensoriel de n matrices unitaires de taille 2×2 découle directement des axiomes de la mécanique quantique. La taille de U est donc $2^n\times 2^n$ et la distribution possède 2^n atomes. Par convention, le serveur possède l'information relative à U_1 et le client i possède l'information concernant U_i.

Notre résultat permet d'établir une nouvelle borne pour p_{θ} qui sera utilisée à la 2^{e} étape. Le lemme suivant utilise la paramétrisation standard du groupe orthogonal et unitaire. Pour plus d'information concernant la paramétrisation du groupe orthogonal et unitaire, une excellente référence est [36] de Murnaghan. Dans ce qui suit, $\otimes$ dénote le produit de Kronecker pour les matrices et $\odot$ dénote le produit terme à terme d'Hadamard pour les matrices de même taille. Dans ce qui suit, si A est une matrice de taille $m\times m$, alors $(A)_{ij}$ dénote l'entrée située à la i^{e} ligne et la j^{e} colonne de A pour tout $(i,j)\in\{0,\ldots,m-1\}^2$.

Lemme 3 (Produit de Kronecker de matrices unitaires 2×2)**.** *Soit $n\in\mathbb{N}$ et, pour tout $j\in\{1,\ldots,n\}$, U_j des matrices unitaires de taille 2×2 c'est-à-dire*

$$U_j \;=\; \begin{pmatrix} e^{\imath\varphi_j}\cos\omega_j & -\imath^{\tau_j}e^{-\imath\psi_j}\sin\omega_j \\ \imath^{\tau_j}e^{\imath\psi_j}\sin\omega_j & (-1)^{\tau_j}e^{-\imath\varphi_j}\cos\omega_j \end{pmatrix},$$

où $\tau_j \in \{0,1\}$, $\varphi_j \in [0,2\pi)$, $\psi_j \in [0,2\pi)$, *et* $\omega_j \in [0,2\pi)$. *Si* $U = \bigotimes_{j=1}^{n} U_j$, *alors, avec* $d = 2^n$ *et pour tout* $(k,\ell) \in \{0, d-1\}^2$, *nous avons que l'entrée* $(U)_{k\ell}$ *est donnée par*

$$\left(\prod_{j=1}^{n} \cos\left(\omega_j - \frac{\pi}{2}(k_j \oplus l_j)\right)\right) \exp\left(\imath \sum_{j=1}^{n} \left((-1)^{\ell_j}\chi_j(k_j \oplus \ell_j) + \frac{\pi}{2}(k_j + \ell_j)\tau_j\right)\right)(-1)^{W(\neg k \wedge \ell)}$$

où k_j *et* ℓ_j *dénotent les* j^e *bits de* k *et* ℓ *respectivement,* $\neg$ *dénote la négation bit à bit,* $\wedge$ *dénote la conjonction bit à bit et* $\oplus$ *est la disjonction exclusive,* $+$ *dénote l'addition usuelle,* W *dénote le poids de Hamming et*

$$\chi_j(x) \;=\; \begin{cases} \varphi_j & \text{if } x = 0 \\ \psi_j & \text{if } x = 1 \end{cases}.$$

Preuve du lemme 3. Remarquons d'abord que si M_1, N_1, M_2, et N_2 sont des matrices de même taille, alors

$$(M_1 \odot N_1) \otimes (M_2 \odot N_2) = (M_1 \odot M_2) \otimes (N_1 \odot N_2). \tag{5.3}$$

La preuve de (5.3) ne repose que sur les définitions du produit d'Hadamard et du produit de Kronecker et nous pouvons consulter entre autres Godement [21]. L'utilisation du produit d'Hadamard nous permet de décomposer le produit de Kronecker en termes plus simples. Nous écrivons d'abord $U_j = A_j \odot B_j \odot C_j \odot D_j$ où

$$A_j = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$
$$B_j = \begin{pmatrix} \cos\omega_j & \sin\omega_j \\ \sin\omega_j & \cos\omega_j \end{pmatrix},$$
$$C_j = \begin{pmatrix} 1 & \imath^{\tau_j} \\ \imath^{\tau_j} & (-1)^{\tau_j} \end{pmatrix},$$

$$D_j = \begin{pmatrix} e^{\imath\varphi_j} & e^{-\imath\psi_j} \\ e^{\imath\psi_j} & e^{-\imath\varphi_j} \end{pmatrix}.$$

Si, de plus, nous dénotons $A = \bigotimes_{j=1}^{n} A_j$, $B = \bigotimes_{j=1}^{n} B_j$, $C = \bigotimes_{j=1}^{n} C_j$ et $D = \bigotimes_{j=1}^{n} D_j$, alors, par (5.3), nous avons $U = A \odot B \odot C \odot D$, et

$$(U)_{k\ell} = (A)_{k\ell}(B)_{k\ell}(C)_{k\ell}(D)_{k\ell}.$$

Pour $(k, \ell) \in \{0, \ldots, d-1\}^2$, le problème se réduit à trouver les entrées $(A)_{k\ell}$, $(B)_{k\ell}$, $(C)_{k\ell}$ et $(D)_{k\ell}$. Par induction, il est facile de montrer que

$$\begin{aligned}
(A)_{k\ell} &= (-1)^{W(\neg k \wedge \ell)}, \\
(B)_{k\ell} &= \prod_{j=1}^{n} \cos\Big(\omega_j - \frac{\pi}{2}(k_j \oplus l_j)\Big), \\
(C)_{k\ell} &= \exp\left(\imath \sum_{j=1}^{n} \left((-1)^{\ell_j} \chi_j(k_j \oplus \ell_j)\right)\right), \\
(D)_{k\ell} &= \exp\left(\imath \sum_{j=1}^{n} \left(\frac{\pi}{2}(k_j + \ell_j)\tau_j\right)\right).
\end{aligned}$$

□

Allons-y maintenant avec **l'étape 2** de notre plan. Maintenant, notre prochain lemme 4 borne le gradient par rapport à θ de $p_\theta(i)$ pour tout i. Par conséquent, nous pouvons borner p_θ en vertu du théorème de Taylor pour les fonctions multivariées. Pour borner ∇p_θ, nous faisons appel à notre lemme précédent 3. Borner ∇p_θ permet de borner l'espérance du nombre de bits des paramètres qui doivent être communiqués des clients vers le serveur.

Dans le lemme suivant, la taille de θ est $3n$, car chaque partie possède trois paramètres réels. La taille du gradient de p_θ est donc $3n$. Notons que les paramètres booléens τ_j n'ajouteront que $n-1$ bits de communication.

Lemme 4 (Majoration de $\|\nabla p_\theta\|$)**.** *Soit ρ une matrice de taille $2^n \times 2^n$ définie*

positive telle que $\mathrm{Tr}(\rho) = 1$ *et, pour* $j \in \{1, \dots n\}$*, des transformations unitaires* U_j *de tailles* 2×2*. Si* $U = \bigotimes_{j=1}^{n} U_j$ *alors, pour tout* $i \in \{0,1\}^n$*,*

$$\|\nabla p_\theta(i)\|^2 = \|\nabla (U\rho U^\dagger)_{ii}\| \leq 36n.$$

Démonstration. Avec la notation du lemme 3, pour $(k, \ell) \in \{0, 2^n - 1\}^2$, posons

$$\xi_{k\ell} = \frac{\pi}{2}(W(\neg k \wedge \ell) \bmod 2) + \sum_{j=1}^{n} \left((-1)^{\ell_j} \chi_j(k_j \oplus \ell_j) + \frac{\pi}{2}(k_j + \ell_j)\tau_j \right),$$

et écrivons

$$u_{k\ell} = \left(\prod_{j=1}^{n} \cos\left(\omega_j - \frac{\pi}{2}(k_j \oplus l_j)\right) \right) e^{i\xi_{k\ell}}.$$

Notons que les quantités $\xi_{k\ell}$ sont indépendantes des paramètres ω_j pour $j \in \{1, \dots, n\}$.

Premièrement, pour tout $m \in \{1, \dots, n\}$ et pour tout $(k, \ell) \in \{0, \dots 2^n - 1\}^2$,

$$\frac{\partial u_{k\ell}}{\partial \omega_m} = u_{k\ell}\left(\omega_m + \frac{\pi}{2}\right)$$

où $u_{k\ell}(\omega_m + \frac{\pi}{2})$ est $u_{k\ell}$ avec ω_m translaté par $\frac{\pi}{2}$ et tous les autres paramètres restent inchangés. Par le théorème de dérivation des fonctions composées, nous avons deuxièmement pour $m \in \{1, \dots, n\}$ que

$$\frac{\partial u_{ik}\bar{u}_{i\ell}}{\partial \omega_m} = u_{ik}\frac{\partial \bar{u}_{i\ell}}{\partial \omega_m} + \bar{u}_{i\ell}\frac{\partial u_{ik}}{\partial \omega_m}.$$

Troisièmement,

$$\begin{aligned}
u_{ik}\left(\omega_m - \frac{\pi}{4}\right) &= \frac{1}{\sqrt{2}}u_{ik} + \frac{1}{\sqrt{2}}\frac{\partial u_{ik}}{\partial \omega_m}, \\
u_{ik}\left(\omega_m - \frac{\pi}{4}\right)\bar{u}_{i\ell}\left(\omega_m - \frac{\pi}{4}\right) &= \frac{1}{2}u_{ik}\bar{u}_{i\ell} + \frac{1}{2}u_{ik}\frac{\partial \bar{u}_{i\ell}}{\partial \omega_m} + \frac{1}{2}\frac{\partial u_{ik}}{\partial \omega_m}\bar{u}_{i\ell} + \frac{1}{2}\frac{\partial u_{ik}}{\partial \omega_m}\frac{\partial \bar{u}_{i\ell}}{\partial \omega_m}, \\
&= \frac{1}{2}u_{ik}\bar{u}_{i\ell} + \frac{1}{2}\frac{\partial (u_{ik}\bar{u}_{i\ell})}{\partial \omega_m} + \frac{1}{2}u_{ik}\big(\omega_m + \frac{\pi}{2}\big)\bar{u}_{i\ell}\big(\omega_m + \frac{\pi}{2}\big).
\end{aligned}$$

Rappelons que $p_\theta(i) = \mathbf{P}\{X = i\}$. Pour des raisons de commodités, nous écrivons $\mathbf{P}_{\omega_m+\gamma}\{i\}$ pour dénoter la distribution obtenue lorsque ω_m translaté par γ et tous les autres les paramètres restent inchangés. Par conséquent $2\mathbf{P}_{\omega_m-\frac{\pi}{4}}\{X = i\}$ vaut

$$2\sum_{k=0}^{2^n-1}\sum_{\ell=0}^{2^n-1}\rho_{k\ell}u_{ik}\left(\omega_m-\frac{\pi}{4}\right)\bar{u}_{i\ell}\left(\omega_m-\frac{\pi}{4}\right)=\mathbf{P}\{X=i\}+\frac{\partial\mathbf{P}\{X=i\}}{\partial\omega_m}+\mathbf{P}_{\omega_m+\frac{\pi}{2}}\{X=i\},$$

et pour tout $m \in \{1, \ldots, n\}$,

$$\left|\frac{\partial\mathbf{P}\{X=i\}}{\partial\omega_m}\right| \leq 2.$$

Pour les paramètres φ_m avec $m \in \{1, \ldots, n\}$,

$$\begin{aligned}\frac{\partial\xi_{ik}}{\partial\varphi_m} &\in \{-2,-1,0,1,2\},\\ \frac{\partial u_{ik}}{\partial\varphi_m} &= u_{ik}\left(\imath\frac{\partial\xi_{ik}}{\partial\varphi_m}\right).\end{aligned}$$

Par conséquent,

$$\begin{aligned}\frac{\partial\mathbf{P}\{X=i\}}{\partial\varphi_m} &= \sum_{k=0}^{2^n-1}\sum_{\ell=0}^{2^n-1}\rho_{k\ell}\frac{\partial u_{ik}\bar{u}_{i\ell}}{\partial\varphi_m}\\ &= \sum_{k=0}^{2^n-1}\sum_{\ell=0}^{2^n-1}\rho_{k\ell}\left(u_{ik}\frac{\partial\bar{u}_{i\ell}}{\partial\varphi_m}+\bar{u}_{i\ell}\frac{\partial u_{ik}}{\partial\varphi_m}\right)\end{aligned}$$

et

$$\left|\frac{\partial\mathbf{P}\{X=i\}}{\partial\varphi_m}\right| \leq 4\mathbf{P}\{X=i\} \leq 4.$$

De la même façon que pour $|\frac{\partial\mathbf{P}}{\partial\varphi_m}|$ nous bornons $|\frac{\partial\mathbf{P}}{\partial\psi_m}|$ pour $m \in \{1, \ldots, n\}$. La preuve est complétée comme suit :

$$\nabla\mathbf{P}=\left(\frac{\partial\mathbf{P}}{\partial\omega_1},\ldots,\frac{\partial\mathbf{P}}{\partial\omega_n},\frac{\partial\mathbf{P}}{\partial\varphi_1},\ldots,\frac{\partial\mathbf{P}}{\partial\varphi_n},\frac{\partial\mathbf{P}}{\partial\psi_1},\ldots,\frac{\partial\mathbf{P}}{\partial\psi_n}\right),$$

$$\begin{aligned}\|\nabla \mathbf{P}\|^2 &= \sum_{i=1}^{n} \left|\frac{\partial \mathbf{P}}{\partial \omega_i}\right|^2 + \sum_{i=1}^{n} \left|\frac{\partial \mathbf{P}}{\partial \varphi_i}\right|^2 + \sum_{i=1}^{n} \left|\frac{\partial \mathbf{P}}{\partial \psi_i}\right|^2 \\ &\leq n(4+16+16) \\ &= 36n.\end{aligned}$$

□

Allons-y avec **l'étape 3** de notre plan. Notre but est de générer une instance d'une variable X dont la fonction de masse est $\mathbf{P}\{X = i\} \stackrel{\text{def}}{=} p_\theta(i)$. Nous avons n ordinateurs et chaque ordinateur possède 4 paramètres dont 3 angles et 1 booléen. Nous ne considérons que les parties fractionnaires des angles. En effet, les paramètres booléens ainsi que les parties entières des angles n'ajoutent que $4(n-1)$ bits (1 bit pour un booléen et 3 bits pour une partie entière). De plus, notons que, dans les algorithmes qui suivent, seul le serveur possède la source de bits aléatoires à la différence du chapitre 4 où chaque client possède également sa source de bits aléatoires.

Pour tout naturel $k > 0$, dénotons par θ_k la k-troncation du vecteur θ c'est-à-dire en prenant les k-troncations de chaque coordonnée du vecteur θ. Puisque θ_k est une troncation, il est évident que pour tout $i \in \{0,1\}^n$

$$\lim_{k\to\infty} p_{\theta_k}(i) \to p_\theta(i).$$

En vertu du lemme 4, pour tout $\theta \in [0, 2\pi)^3$ et pour tout $i \in \{0,1\}^n$,

$$\sup |p_\theta(i) - p_{\theta_k}(i)| \leq \alpha \stackrel{\text{def}}{=} \frac{6\sqrt{3}n}{2^k} \text{ for } k \in \mathbb{N}.$$

Remarquons que

$$\sum_{i=0}^{2^n-1} \big(p_{\theta_k}(i) + \alpha\big) = 1 + 2^n\alpha$$

et définissons $q \stackrel{\text{def}}{=} \frac{1}{1+2^n\alpha} \in (0,1)$ et $C \stackrel{\text{def}}{=} (1+2^n\alpha)$. Par conséquent, pour tout

$i \in \{0,1\}^n$

$$\begin{aligned} p_\theta(i) &\leq p_{\theta_k}(i) + \alpha \\ &= C\left(qp_{\theta_k}(i) + (1-q)\frac{1}{2^n}\right). \end{aligned} \tag{5.4}$$

L'inégalité de la ligne (5.4) indique une fois de plus que l'algorithme de von Neumann est approprié pour nos besoins. En effet, il est facile de générer une variable Y de distribution $\mathbf{P}\{Y = i\} \stackrel{\text{def}}{=} qp_{\theta_k}(i) + (1-q)2^{-n}$ étant donné le vecteur θ_k. En connaissant θ_k, le serveur génère :

(i) avec probabilité $\frac{1}{1+2^n\alpha}$ une instance de Y de distribution p_{θ_k}. L'échantillonnage de p_{θ_k} peut se faire entre autres en utilisant la méthode de Knuth et Yao explicitée au chapitre 2,

(ii) avec probabilité $\frac{2^n\alpha}{1+2^n\alpha}$ le serveur génère Y uniformément dans l'ensemble $\{0,1\}^n$.

Le prochain algorithme suppose que le serveur a colligé l'information décrivant θ_k.

Algorithme 5.1 : Algorithme de von Neumann pour distribution quantique (modèle standard de génération par variables uniformes)

---

entrée(s) : θ_k {la k-troncation de θ}

répéter
  Générer X de distribution $qp_{\theta_k}(i) + (1-q)2^{-n}$ proportionnelle à $p_{\theta_k}(i) + \alpha$.
  Générer U uniformément dans $[0,1]$.
  Calculer $D \leftarrow \mathbb{1}_{\{U(p_{\theta_k}(i)+\alpha) \leq p_\theta(X)\}}$ {Ceci requiert la procédure ci-après.}
  si $D = 1$ **alors**
    retourner X {X est acceptée}
  sinon {$D = 0$}
    continuer {X est rejetée}
  fin si
fin répéter

La quantité $U(p_{\theta_k}(i)+\alpha) \stackrel{\text{def}}{=} R$ est connue du serveur. Si les bits de U sont générée séquentiellement, alors un changement simple à l'algorithme suivant permet d'échantillonner p_θ et c'est ce que nous ferons à la prochaine étape de notre plan. L'analyse qui suivra le prochain sous-programme, lequel est appelé par l'algorithme 5.1, portera sur l'espérance du nombre de bits de communication.

Algorithme 5.2 : Procédure pour la variable d'arrêt D dans l'algorithme 5.1

1: **pour** $t = k$ **à** ∞ **faire**
2: Le serveur s'assure qu'il possède t bits de chaque coordonnée du vecteur θ afin de connaître la t-troncation θ_t. Le serveur calcule $p_{\theta_t}(X)$.
3: **si** $R \le p_{\theta_t}(X) - \frac{6\sqrt{3}n}{2^t}$ **alors**
4: **retourner** $D = 1$.
5: **sinon si** $R \ge p_{\theta_t}(X) + \frac{6\sqrt{3}n}{2^t}$ **alors**
6: **retourner** $D = 0$.
7: **fin si**
8: **fin pour**

Théorème 10. *Si A dénote le nombre espéré de bits de communication, alors*

$$A \le (4 + o(1))n^2.$$

Démonstration. La procédure 5.2 s'arrête avec probabilité 1 car $\frac{6\sqrt{3}n}{2^t} \to 0$ lorsque $t \to \infty$. Soit T la variable aléatoire de la valeur de t (temps d'arrêt) de la ligne (1) dans la procédure c'est-à-dire la valeur de t lorsque l'indicatrice D est déterminée. À ce stade, le nombre total de bits transmis dans un protocole séquentiel est au plus Tn.

Pour l'analyse de $\mathbf{E}(T)$, nous avons que

$$\begin{aligned}
\mathbf{P}\{T > t \mid X\} &\le \mathbf{P}\left\{ |R - p_{\theta_t}(X)| \le \frac{6\sqrt{3}n}{2^t} \;\middle|\; X \right\} \\
&\le \min\left\{ 1, \mathbf{P}\left\{ \frac{-(6\sqrt{3}n/2^t) + p_{\theta_t}(X)}{p_{\theta_k}(X)+\alpha} \le U \le \frac{(6\sqrt{3}n/2^t) + p_{\theta_t}(X)}{p_{\theta_k}(X)+\alpha} \right\} \right\} \\
&\le \min\left\{ 1, \frac{12\sqrt{3}n}{2^t(p_{\theta_k}(X)+\alpha)} \right\}. \qquad (5.5)
\end{aligned}$$

Si N dénote le nombre total d'itérations de von Neumann (les itérations débutant à la ligne (1) de l'algorithme 1) et que T_j dénote le nombre de bits transmis à la j^{e} itération pour $j \in \{1, \ldots, N\}$, alors le nombre total de bits transmis des paramètres est borné par

$$n(T_1 + \ldots + T_N).$$

Les variables T_j sont i.i.d. donc par, l'identité de Wald (consulter [18]), la complexité espérée A du nombre total de bits est au plus

$$A \stackrel{\text{def}}{=} n\mathbf{E}(N)\mathbf{E}(T_1). \tag{5.6}$$

Puisque les variables T_j sont i.i.d. nous écrirons dorénavant $T \stackrel{\text{def}}{=} T_1$. Maintenant

$$\mathbf{E}(N) = 1 + 2^n\alpha.$$

Par conséquent,

$$\begin{aligned}
A &= n(1 + 2^n\alpha) \sum_{t=0}^{\infty} \mathbf{P}\{T > t\} \\
&= n \sum_i (p_{\theta_k}(i) + \alpha) \sum_{t=0}^{\infty} \mathbf{P}\{T > t \mid X = i\} \\
&\leq n \sum_i (p_{\theta_k}(i) + \alpha) \sum_{t=k+1}^{\infty} \min\left\{1, \frac{12\sqrt{3}n}{2^t(p_{\theta_k}(X) + \alpha)}\right\} + n(k+1)(1 + 2^n\alpha) \\
&\stackrel{\text{def}}{=} \mathrm{I} + \mathrm{II},
\end{aligned}$$

et

$$\begin{aligned}
\mathrm{I} &= n \sum_i (p_{\theta_k}(i) + \alpha) \sum_{t=k+1}^{\infty} \min\left\{1, \frac{12\sqrt{3}n}{2^t(p_{\theta_k}(X) + \alpha)}\right\}, \\
\mathrm{II} &= n(k+1)(1 + 2^n\alpha).
\end{aligned}$$

Soit $t^* \in (0, \infty)$ tel que

$$\frac{12\sqrt{3}n}{2^{t^*}} = p_{\theta_k}(i) + \alpha.$$

Dans l'expression de I, la contribution des naturels $t \in \mathbb{N}$ tels que $t \leq \lfloor t^* \rfloor$ est au plus

$$\begin{aligned} n \sum_i (p_{\theta_k}(i) + \alpha) \sum_{t \leq \lfloor t^* \rfloor} 1 &= n \sum_i (p_{\theta_k}(i) + \alpha) \lfloor t^* \rfloor \\ &\leq n \sum_i (p_{\theta_k}(i) + \alpha) t^* \\ &\leq n(1 + 2^n \alpha) \log_2 \left(\frac{12\sqrt{3}n}{\alpha} \right). \end{aligned}$$

Dans l'expression de I, la contribution des naturels $t \in \mathbb{N}$ tels que $t > \lfloor t^* \rfloor$ est au plus

$$\begin{aligned} & n \sum_i (p_{\theta_k}(i) + \alpha) \sum_{t \geq \lceil t^* \rceil} \frac{12\sqrt{3}n}{(p_{\theta_{k(i)}} + \alpha)} \frac{1}{2^t} \\ \leq & n \sum_i (p_{\theta_{k(i)}} + \alpha) \left(\frac{2}{2^{t^*}} \frac{12\sqrt{3}n}{(p_{\theta_{k(i)}} + \alpha)} \right) \\ = & 2n \sum_i (p_{\theta_k}(i) + \alpha) \text{ (par définition de } t^*) \\ = & 2n(1 + 2^n \alpha). \end{aligned}$$

Ainsi est majorée

$$\mathrm{I} \leq n(1 + 2^n \alpha) \left(2 + \log_2 \left(\frac{12\sqrt{3}n}{\alpha} \right) \right).$$

Se rappelant notre définition de $\alpha = 6\sqrt{3}n/2^k$,

$$\mathrm{I} + \mathrm{II} \leq n(1 + 2^n \alpha) \left(k + 3 + \log_2 \left(\frac{12\sqrt{3}n}{\alpha} \right) \right)$$

$$= n(1 + 2^n\alpha)(2k + 4).$$

En choisissant par exemple $k = 2n$, nous avons que

$$\mathrm{I} + \mathrm{II} \leq (4 + o(1))n^2.$$

□

Allons-y avec **l'étape 4** de notre plan. Pour nous débarrasser des variables continues uniformes, nous modifions l'algorithme 5.2 en remplaçant les inégalités impliquant R et U par des inégalités impliquant des t-troncations de U pour $t \geq 1$. Ces inégalités nouvelles convergent vers les inégalités utilisées du modèle standard de génération lorsque $t \to \infty$. Se rappelant le lemme 4, pour tout $i \in \{0, \ldots, 2^n - 1\}$ et pour toutes k-troncations de θ,

$$\begin{aligned}
\frac{p_{\theta_k}(i) + \alpha}{1 + 2^n\alpha} &= \frac{1}{1 + 2^n\alpha} p_{\theta_k}(i) + \frac{2^n\alpha}{1 + 2^n\alpha} \frac{1}{2^n}, \\
p_\theta(i) &\leq (1 + 2^n\alpha) \frac{p_{\theta_k}(i) + \alpha}{1 + 2^n\alpha} \\
&= p_{\theta_k}(i) + \alpha, \text{ et} \\
\alpha &= \frac{6\sqrt{3}n}{2^k}.
\end{aligned}$$

Dans l'algorithme suivant, nous avons la liberté de choisir la valeur k tout comme nous l'avons fait auparavant pour la procédure 5.2. La valeur de k sera une fois de plus $2n$ et ce choix implique des complexités asymptotiques qui ne sont pas exponentielles.

Algorithme 5.3 : Algorithme de von Neumann pour distribution quantique (modèle de génération par bits)

1: À ce stade, X est générée selon $\frac{p_{\theta_k}(i)+\alpha}{1+2^n\alpha}$. {L'algorithme de Knuth et Yao peut être utilisé à cet effet.}
2: Le serveur génère k bits non biaisés i.i.d. $U_1, \ldots, U_k$.
3: $U(k) \leftarrow \sum_{j=1}^{t} U_j 2^{-j}$.

4: Le serveur obtient k bits de chaque paramètre et apprend θ_k. Le serveur calcule $p_{\theta_k}(X)$.
5: **pour** $t = k$ **à** ∞ **faire**
6: **si** $U(t)\big(p_{\theta_k}(X) + \alpha\big) - p_{\theta_t}(X) \leq -\frac{6\sqrt{3}n}{2^t} - \frac{1}{2^t}\big(p_{\theta_k}(X) + \alpha\big)$ **alors**
7: **retourner** X {X est accepté}
8: **sinon si** $U(t)\big(p_{\theta_k}(X) + \alpha\big) - p_{\theta_t}(X) \geq \frac{6\sqrt{3}n}{2^t} + \frac{1}{2^t}\big(p_{\theta_k}(X) + \alpha\big)$ **alors**
9: Recommencer à ligne 1 avec un nouveau X. {X est rejeté}
10: **sinon** {impossible de décider}
11: Générer le bit U_{t+1}
12: $U(t+1) \leftarrow U(t) + U_{t+1}2^{-(t+1)}$.
13: Le serveur obtient les $(t+1)^{\text{es}}$ bits de chaque paramètre et apprend θ_{t+1}. Le serveur calcule $p_{\theta_{t+1}}(X)$.
14: **fin si**
15: **fin pour**

Théorème 11. *Si A dénote l'espérance du nombre total de bits transmis des paramètres et B l'espérance du nombre de bits aléatoires, alors*

$$A \leq (4 + o(1))n^2 \text{ et}$$
$$B \leq (5 + o(1))n^2.$$

Démonstration. Nous obtenons d'abord des majorations pour k et choisirons $k = 2n$ à la fin de sorte à obtenir le résultat. En ayant remplacé la variable uniforme U dans l'expression de R de la procédure 5.2 par une t-troncation, nous avons que $U(t) - \frac{1}{2^t} \leq U(t) \leq U \leq U(t) + \frac{1}{2^t}$. Si

$$\left(U(t) + \frac{1}{2^t}\right)\big(p_{\theta_k}(X) + \alpha\big) \leq p_{\theta_t}(X) - \frac{6\sqrt{3}n}{2^t},$$

ou de façon équivalente,

$$U(t)\big(p_{\theta_k}(X) + \alpha\big) - p_{\theta_t}(X) \leq -\frac{6\sqrt{3}n}{2^t} - \frac{1}{2^t}\big(p_{\theta_k}(X) + \alpha\big) \tag{5.7}$$

alors la variable X est acceptée. Si

$$\left(U(t) - \frac{1}{2^t}\right)\big(p_{\theta_k}(X) + \alpha\big) \geq p_{\theta_t}(X) + \frac{6\sqrt{3}n}{2^t},$$

ou de façon équivalente,

$$U(t)\big(p_{\theta_k}(X) + \alpha\big) - p_{\theta_t}(X) \geq \frac{6\sqrt{3}n}{2^t} + \frac{1}{2^t}\big(p_{\theta_k}(X) + \alpha\big) \tag{5.8}$$

alors la variable X est rejetée. La procédure ainsi modifiée s'arrête si (5.7) ou (5.8) sont vérifiées c'est-à-dire si

$$\big|U(t)\big(p_{\theta_k}(X) + \alpha\big) - p_{\theta_t}(X)\big| \geq \frac{6\sqrt{3}n}{2^t} + \frac{1}{2^t}\big(p_{\theta_k}(X) + \alpha\big),$$

et elle ne s'arrête pas si

$$\begin{aligned}\big|U(t)\big(p_{\theta_k}(X) + \alpha\big) - p_{\theta_t}(X)\big| &< \frac{6\sqrt{3}n}{2^t} + \frac{1}{2^t}\big(p_{\theta_k}(X) + \alpha\big) \\ &\leq \frac{6\sqrt{3}n + 1 + \alpha}{2^t}.\end{aligned} \tag{5.9}$$

Par conséquent, étant donnée la variable X distribuée selon $\frac{p_{\theta_k}+\alpha}{1+2^n\alpha}$, nous avons avant l'arrêt de la procédure de décision que

$$\begin{aligned}&\big|U\big(p_{\theta_k}(X) + \alpha\big) - p_{\theta_t}(X)\big| \\ &\leq |U - U(t)|\big|p_{\theta_k}(X) + \alpha\big| + \big|U(t)\big(p_{\theta_k}(X) + \alpha\big) - p_{\theta_t}(X)\big| \\ &\leq \frac{1+\alpha}{2^t} + \frac{6\sqrt{3}n + 1 + \alpha}{2^t} \ \text{(par (5.9))} \\ &= \frac{2 + 2\alpha + 6\sqrt{3}n}{2^t}.\end{aligned} \tag{5.10}$$

La quantité (5.10) est cruciale pour majorer $\mathbf{P}\{T > t \mid X\}$ où T est le nombre de bits de communications pour une itération. En effet, si N dénote une fois de plus

le nombre d'itérations de von Neumann avant l'acceptation, alors

$$\begin{aligned} &A = n\mathbf{E}(N)\mathbf{E}(T) \text{ et} \\ &B \leq n\mathbf{E}(N)\left(\mathbf{E}(T) + \mathcal{E}\left(\frac{p_{\theta_k}+\alpha}{1+2^n\alpha}\right) + 2\right), \end{aligned}$$

où la quantité $\mathcal{E}\left(\frac{p_{\theta_k}+\alpha}{1+2^n\alpha}\right)$ est l'entropie binaire de $\frac{p_{\theta_k}+\alpha}{1+2^n\alpha}$. Pour la complexité B, le serveur génère une instance de la variable distribuée selon $\frac{p_{\theta_k}+\alpha}{1+2^n\alpha}$, et, par le théorème 3 de Knuth et Yao du chapitre 2, il faut espérer au plus $\mathcal{E}\left(\frac{p_{\theta_k}+\alpha}{1+2^n\alpha}\right)+2$ bits. Comme dans le cas du modèle standard de génération par variables uniformes, $\mathbf{E}(N) = 1+2^n\alpha$. En utilisant (5.10),

$$\mathbf{E}(T) \leq k + 5 + \mathcal{E}\left(\frac{p_{\theta_k}+\alpha}{1+2^n\alpha}\right) + \log_2\left(\frac{1+\alpha+6\sqrt{3}n}{1+2^n\alpha}\right). \tag{5.11}$$

Le terme contenant $\log_2$ dans l'expression de la ligne (5.11) est majoré une fois que le choix de k est établi. Pour montrer (5.11), notons que $\mathbf{E}(T) = \mathbf{E}_X(\mathbf{E}(T|X))$ et, par conséquent,

$$\begin{aligned} \mathbf{E}(T) &= \mathbf{E}_X(\mathbf{E}(T|X)) \\ &= \mathbf{E}_X\left(\sum_{t=0}^{\infty}\mathbf{P}\{T > t \mid X\}\right) \\ &= \mathbf{E}_X\left(k+1+\sum_{t=k+1}^{\infty}\mathbf{P}\{T > t \mid X\}\right) \\ &= k+1+\sum_{i=0}^{2^n-1}\frac{p_{\theta_k}(i)+\alpha}{1+2^n\alpha}\sum_{t=k+1}^{\infty}\mathbf{P}\{T > t \mid X = i\}. \end{aligned} \tag{5.12}$$

Donc

$$\begin{aligned} &\mathbf{P}\{T > t \mid X\} \\ &\leq \min\left\{1, \mathbf{P}\left\{\left|U\big(p_{\theta_k}(X)+\alpha\big) - p_{\theta_t}(X)\right| \leq \frac{2(1+\alpha+3\sqrt{3}n)}{2^t}\right\}\right\} \end{aligned}$$

$$= \min\left\{1, \frac{4(1+\alpha+3\sqrt{3}n)}{2^t}\frac{1}{p_{\theta_k}(X)+\alpha}\right\} \quad \text{(par 5.10)}.$$

Posons $t^* \in (0,\infty)$ tel que $1 = \frac{4(1+\alpha+3\sqrt{3}n)}{2^{t^*}}\frac{1}{p_{\theta_k}(X)+\alpha}$. Pour les naturels $t \in \mathbb{N}$ tels que $t \leq \lfloor t^* \rfloor$, la double somme de la ligne (5.12) vaut au plus

$$\begin{aligned}
&\sum_{i=0}^{2^n-1} \frac{p_{\theta_k}(i)+\alpha}{1+2^n\alpha} \sum_{t\leq\lfloor t^*\rfloor} 1 \\
&\leq \sum_{i=0}^{2^n-1} \frac{p_{\theta_k}(i)+\alpha}{1+2^n\alpha} t^* \\
&= \mathcal{E}\left(\frac{p_{\theta_k}+\alpha}{1+2^n\alpha}\right) + 2 + \log_2\left(\frac{1+\alpha+3\sqrt{3}n}{1+2^n\alpha}\right). \qquad (5.13)
\end{aligned}$$

Pour les naturels $t \in \mathbb{N}$ tels que $t > \lfloor t^* \rfloor$, la double somme de la ligne (5.12) vaut au plus

$$\begin{aligned}
&\sum_{i=0}^{2^n-1} \frac{p_{\theta_k}(i)+\alpha}{1+2^n\alpha} \sum_{t\geq\lceil t^*\rceil}^{\infty} \frac{4(1+\alpha+3\sqrt{3}n)}{2^t}\frac{1}{p_{\theta_k}(X)+\alpha} \\
&\leq \sum_{i=0}^{2^n-1} \frac{p_{\theta_k}(i)+\alpha}{1+2^n\alpha}\left(\frac{4(1+\alpha+3\sqrt{3}n)}{2^{t^*}}\frac{1}{p_{\theta_k}(X)+\alpha}\right)2 \\
&= 2\sum_{i=0}^{2^n-1} \frac{p_{\theta_k}(i)+\alpha}{1+2^n\alpha}\left(1\right) \text{ (par la définition de } t^*) \\
&= 2. \qquad (5.14)
\end{aligned}$$

En combinant (5.12), (5.13) et (5.14),

$$\mathbf{E}(T) \leq k + 5 + \mathcal{E}\left(\frac{p_{\theta_k}+\alpha}{1+2^n\alpha}\right) + \log_2\left(\frac{1+\alpha+3\sqrt{3}n}{1+2^n\alpha}\right).$$

Par conséquent,

$$A = n\mathbf{E}(N)\mathbf{E}(T), \quad \text{et}$$

$$B \leq n\mathbf{E}(N)\left(\mathbf{E}(T) + 2 + \mathcal{E}\left(\frac{p_{\theta_k} + \alpha}{1 + 2^n\alpha}\right)\right).$$

En se rappelant que $\alpha = \frac{6\sqrt{3}n}{2^k}$, que $\mathbf{E}(N) = 1 + 2^\alpha n$, que l'entropie vaut au plus n et en choisissant $k = 2n$,

$$\log_2\left(\frac{1 + \alpha + 3\sqrt{3}n}{1 + 2^n\alpha}\right) = n + O\big(\log_2(n)\big).$$

Finalement,

$$A \leq (4 + o(1))n^2 \text{ et}$$
$$B \leq (5 + o(1))n^2.$$

□

CONCLUSION

Nous avons étudié la complexité espérée du nombre de bits aléatoires requis pour échantillonner des distributions. Le modèle standard suppose l'existence d'un générateur pouvant produire des variables aléatoires continues et uniformes sur l'intervalle $[0, 1]$. La grande différence dans cette thèse est d'avoir remplacé le générateur de variables aléatoires continues et uniformes par un générateur de bits non biaisés i.i.d. Le cas des distributions discrètes ayant été déjà traité en détail entre autres par Knuth et Yao ainsi que Han et Hoshi, nous avons reprouvé de façon simple plusieurs résultats déjà existants pour le cas discret. Notre but était d'élargir la théorie aux distributions continues. Contrairement au cas discret, il est évidemment impossible de générer exactement une variable continue donc cela nous conduisit dans un premier temps à penser au concept de précision ainsi qu'au choix d'une métrique appropriée nous permettant de mesurer la distance entre la variable continue théorique et la variable discrète produite par un algorithme d'échantillonnage. Les principaux résultats de cette thèse en ce qui concerne l'échantillonnage d'une distribution continue sont :

1. minoration de la complexité espérée pour l'échantillonnage d'une distribution continue avec une précision arbitraire et la relation entre la complexité et l'entropie différentielle lorsque la distribution est absolument continue,

2. majoration de la complexité espérée pour la méthode de partition du support d'une distribution continue et la relation entre la complexité et l'entropie différentielle lorsque la distribution est absolument continue,

3. majoration de la complexité espérée pour la méthode de l'inversion d'une distribution continue et la relation entre la complexité et l'entropie différentielle lorsque la distribution est absolument continue.

De façon générale pour un niveau de précision $\epsilon > 0$, si la distribution est absolument continue, que l'entropie des parties entières converge et que son entropie

différentielle converge, alors nos résultats montrent que la complexité est la somme de $\log_2(1/\epsilon)$, de l'entropie différentielle et d'un terme négligeable.

En ce qui concerne l'échantillonnage exact d'une distribution discrète, nous avons les résultats qui suivent :

1. nouvelle preuve simple du résultat de Knuth et Yao de 1976,
2. nouvelle preuve simple du résultat de Han et Hoshi de 1997 pour les sources produisant des bits non biaisées i.i.d.,
3. formulation exacte de la complexité du « Fast Dice Roller » de Lumbroso de 2012, méthode optimale pour produire des variables uniformes discrètes dans l'ensemble $\{0, \ldots, n-1\}$,
4. génération d'une variable discrète uniforme dans $\{0, \ldots, n-1\}$ en inversant $\lfloor nU \rfloor$ où U est une variable uniforme continue dans l'intervalle $[0, 1]$. Plus précisément, calculer exactement l'espérance du nombre de bits nécessaires de l'expansion binaire de U pour décider de la valeur de $\lfloor nU \rfloor$.

Nous avons également analysé le problème d'échantillonner exactement une distribution discrète en ne possédant qu'une partie de l'information décrivant ladite distribution, problème utile pour l'informatique quantique et la complexité de la communication. Le nombre espéré de bits à communiquer sur le réseau fut également analysé. Dans un premier temps, nous nous sommes concentrés sur la distribution particulière de GHZ vue son importance capitale en informatique quantique. Dans le cas de GHZ, nous avons supposé que les clients et le serveur peuvent générer des bits. Nous avons montré que les complexités espérées du nombre de bits aléatoires et du nombre de bits de communication sont respectivement de $O(n)$ et de $O(n^2)$ pour la distribution de GHZ. Par la suite, nous avons généralisé le résultat à toutes les distributions quantiques discrètes définies sur $\{0, 1\}^n$. Les grandes étapes pour résoudre le problème furent :

1. À partir des axiomes de la mécanique quantique, trouver une formulation appropriée de p_θ permettant d'opérer des calculs différentiels sur p_θ par rap-

port au vecteur θ dans le but de borner p_θ par une autre distribution facile à échantillonner dans notre contexte communicationnel.

2. Modifier l'algorithme d'acceptation et de rejet de von Neumann qui échantillonne p_θ sur un réseau où le vecteur θ est distribué en utilisant la distribution facile fournie par la majoration en (1).

Nous avons montré que les complexités du nombre de bits de bits aléatoires et de communication sont respectivement $(5 + o(1))n^2$ et $(4 + o(1))n^2$ peu importe la distribution quantique.

Parmi les problèmes ouverts ou plutôt la recherche à venir, il y a :

1. Adapter la méthode d'acception et de rejet de von Neumann pour les distributions absolument continues lorsque la source ne produit que des bits non biaisés i.i.d.

2. Étudier des modèles où la complexité du nombre de requêtes aux oracles serait prise en compte en plus du nombre de bits.

3. Montrer que la méthode de l'inversion est optimale dans le cas continu.

4. Obtenir des résultats semblables à ceux que Rényi et/ou Csiszàr fournissent avec une hypothèse plus générale. Rappelons qu'il faut que l'entropie des distributions des parties entières converge.

5. Pour l'échantillonnage distribué des distributions quantiques discrètes, le cas des distributions définies dans $\{0, \ldots, k-1\}^n$ reste à faire. Rappelons que nous avons fait $k = 2$. Nous conjecturons que les complexités sont $O\big(k^2 n^2\big)$.

6. Encore plus général que le cas précédent serait de simuler les POVMs.

BIBLIOGRAPHIE

[1] J.-D. Bancal, C. Branciard et N. Gisin. Simulation of equatorial von Neumann measurements on GHZ states using nonlocal resources. *Advances in Mathematical Physics*, 293245, 2010.

[2] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1(3):195–200, 1964.

[3] R. Bhatia. *Matrix Analysis*. Springer, 1997.

[4] P. Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics. Wiley, 2012.

[5] G. E. Box and M. E. Muller. A note on the generation of random normal deviates. *Ann. Math. Stat*, 29:610–611, 1958.

[6] C. Branciard et N. Gisin. Quantifying the nonlocality of Greenberger-Horne-Zeilinger quantum correlations by a bounded communication simulation protocol. *Physical Review Letters*, 107 :020401, 2011.

[7] G. Brassard. Quantum communication complexity. *Foundations of Physics*, 33(11):1593–1616, 2003.

[8] G. Brassard et M. Kaplan. Simulating equatorial measurements on ghz states with finite expected communication cost. Dans *Proceedings of 7th Conference on Theory of Quantum Computation, Communication, and Cryptography (TQC)*, pages 65–73, 2012.

[9] G. Brassard, L. Devroye et C. Gravel. Exact classical simulation of the ghz distribution. Dans *Proceedings of 9th Conference on Theory of Quantum Computation, Communication, and Cryptography (TQC)*, 2014.

[10] G. Brassard, R. Cleve et A. Tapp. Cost of exactly simulating quantum entanglement with classical communication. *Physical Review Letters*, 83:1874–1877, 1999.

[11] A. Broadbent, P. R. Chouha et A. Tapp. The GHZ state in secret sharing and entanglement simulation. Dans *Proceedings of the Third International Conference on Quantum, Nano and Micro Technologies*, pages 59–62, 2009.

[12] J. H. Conway et N. J. A. Sloane. *Sphere packings, lattices and groups*. Springer, third édition, 1998.

[13] T. M. Cover et J. A. Thomas. *Elements of Information Theory*. Wiley, New-York, 1991.

[14] I. Csiszár. Some remarks on the dimension and entropy of random variables. *Acta Mathematica Academiae Scientiarum Hungarica*, 12:399–408, 1961.

[15] I. Csiszár. On the dimension and entropy of order α of the mixture of probability distributions. *Acta Mathematica Academiae Scientiarum Hungarica*, 13:245–255, 1962.

[16] L. Devroye. *Non-Uniform Random Variate Generation*. Springer, 1986.

[17] A. Einstein, B. Podolsky et N. Rosen. Can quantum-mechanical description of physical reality be considered complete ? *Physical Review*, 47:777–780, 1935.

[18] W. Feller. *An Introduction To Probability Theory And Its Application*, volume 1. Wiley, 3^{e} édition, 2008.

[19] W. Feller. *An Introduction To Probability Theory And Its Application*, volume 2. Wiley, 2^{e} édition, 2008.

[20] P. Flajolet et N. Saheb. The complexity of generating an exponentially distributed variate. *Journal of Algorithms*, 7:463–488, 1986.

[21] R. Godement. *Cours d'algèbre*. Hermann, 1966.

[22] C. Gravel. Structure de la distribution de probabilité de l'état ghz sous l'action de mesures de von neumann locales. Mémoire de maîtrise, Université de Montréal, 2011. URL `https://papyrus.bib.umontreal.ca/jspui/handle/1866/5511`.

[23] C. Gravel. Structure of the probability distribution for the ghz quantum state under local von neumann measurements. *Quantum Physics Letters*, 1(3):87–96, 2012.

[24] D. M. Greenberger, M. A. Horne et A. Zeilinger. Going beyond Bell's theorem. Dans M. Kafatos, éditeur, *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, pages 69–72. Kluwer Academic Publishers, 1989.

[25] T. S. Han et M. Hoshi. Interval algorithm for random number generation. *IEEE Transactions on Information Theory*, 43(2):599–611, 1997.

[26] E. Hewitt and K. Stromberg. *Real and Abstract Analysis*. Springer-Verlag, New York, 1965.

[27] A. S. Holevo. *Statistical Structure of Quantum Theory*. Lecture Notes in Physics Monographs. Springer, 2001. ISBN 9783540420828.

[28] S. Kakutani. On equivalence of infinite product measures. *Annals of Mathematics*, pages 214–224, 1948.

[29] C. F. F. Karney. Sampling exactly from the normal distribution. URL `http://arxiv.org/abs/1303.6257`. 2013.

[30] D. E. Knuth et A. C.-C. Yao. The complexity of nonuniform random number generation. Dans J. F. Traub, éditeur, *Algorithms and Complexity : New Directions and Recent Results.*, pages 357–428, New York, 1976. Carnegie-Mellon University, Computer Science Department, Academic Press. Reprinted in Knuth's *Selected Papers on Analysis of Algorithms* (CSLI, 2000).

[31] E. Kushilevitz et N. Nisan. *Communication Complexity*. Cambridge University Press, 1997. ISBN 9780521560672.

[32] T. Linder et K. Zeger. Asymptotic entropy-constrained performance of tessellating and universal randomized lattice quantization. *IEEE Transactions of Information Theory*, 40(2), 1994.

[33] J. Lumbroso. *Probabilistic Algorithms for Data Sreaming and Random Generation*. Thèse de doctorat, Université Pierre et Marie Curie - Paris 6, 2012.

[34] S. Massar, D. Bacon, N. Cerf et R. Cleve. Classical simulation of quantum entanglement without local hidden variables. *Physical Review A*, 63(5) :052305, 2001.

[35] T. Maudlin. Bell's inequality, information transmission, and prism models. Dans *Proceedings of the Biennial Meeting of the Philosophy of Science Association*, pages 404–417, 1992.

[36] F. Murnaghan. *The unitary and rotation groups*. Lectures on applied mathematics. Spartan, Washington D.C., 1962.

[37] J. von Neumann. Various techniques used in connection with random digits. monte carlo methods. *National Bureau Standards*, 12:36–38, 1951.

[38] S. T. Rachev et L. Rüschendorf. *Mass Transportation Problems : Volume 1 : Theory*. Springer (Probability and Its Applications), 1998.

[39] A. Renyi. On the dimension and entropy of probability distributions. *Acta Mathematica Academiae Scientiarum Hungarica*, 10:193–215, 1959.

[40] C. E. Shannon. A mathematical theory of communication. *Bell. Sys. Tech. Journal*, 27:379–423, 623–656, 1948.

[41] L. Devroye et C. Gravel. Sampling with arbitrary precision. URL `http://arxiv.org/abs/1502.02539`. 2015.

[42] M. Steiner. Towards quantifying non-local information transfer : finite-bit non-locality. *Physics Letters A*, 270:239–244, 2000.

Annexe I

Types de convergence et théorème de convergence dominée de Lebesgue

Pour la notion d'espace de probabilité, consulter [4]. Cette annexe est un complément non essentiel à la compréhension des résultats contenus dans cette thèse.

Définition 15. Soit une suite X_1, X_2, … de variables aléatoires définies sur le même espace de probabilité $(\Omega, \mathcal{F}, \mathbf{P})$. Soit X une variable aléatoire définie sur $(\Omega, \mathcal{F}, \mathbf{P})$.

1. $X_n \to X$ *presque sûrement*, dénoté par $X_n \overset{\text{p.s.}}{\to} X$, si

$$\mathbf{P}\{\omega \in \Omega \ : \ X_n(\omega) \to X(\omega) \text{ lorsque } n \to \infty\} = 1.$$

2. $X_n \to X$ en *espérance d'ordre* r et $r \geq 1$, dénoté par $X_n \overset{\text{r}}{\to} X$, si, pour tout nombre naturel n, les quantités $\mathbf{E}|X_n^r|$ convergent et

$$\mathbf{E}|X_n - X|^r \to 0 \text{ lorsque } n \to \infty.$$

3. $X_n \to X$ en *probabilité*, dénoté par $X_n \overset{\text{p}}{\to} X$, si, pour tout nombre réel $\delta > 0$,

$$\mathbf{P}\big\{|X_n - X| > \delta\big\} \to 0 \text{ lorsque } n \to \infty.$$

4. $X_n \to X$ en *distribution*, dénoté par $X_n \overset{\mathcal{L}}{\to} X$, si, pour tout x tel que $\mathbf{P}\{X \leq x\}$ est continue,

$$\mathbf{P}\{X_n \leq x\} \to \mathbf{P}\{X \leq x\} \text{ lorsque } n \to \infty.$$

Théorème 12 (convergence dominée de Lebesgue)**.** *Soit* $(X_1, X_2, \ldots)$ *une suite de variables aléatoires et une variable* X *telles que* $X_n \to X$ *presque sûrement lorsque*

$n \to \infty$. S'il existe une variable aléatoire Y telle que $|X_n| \leq Y$ presque sûrement et $\mathbf{E}(|Y|) < \infty$, alors

$$\lim_{n\to\infty} \mathbf{E}(X_n) = \mathbf{E}(X).$$

Le corollaire suivant sera utilisé ultérieurement.

Corollaire 2. *Si $(X_1, X_2, \ldots)$ est une suite de variables aléatoires qui satisfait*

$$\mathbf{E}\Big(\sum_{i=1}^{\infty} |X_i|\Big) < \infty,$$

alors

$$\sum_{i=1}^{\infty} \mathbf{E}(X_i) = \mathbf{E}\Big(\sum_{i=1}^{\infty} X_i\Big).$$

Pour prouver le corollaire 2, simplement poser $Z_n = \sum_{i=1}^{n} X_i$ et $Z = \sum_{i=1}^{\infty} |X_i|$ de sorte que $|Z_n| \leq Z$ et ainsi le résultat est impliqué par le théorème 12.

Annexe II

Génération d'un paquet et extraction de bits aléatoires

Nous montrons dans cette annexe comment générer n variables i.i.d. X_1, X_2, …, X_n de distribution $(p_1, p_2, \ldots)$ en n'utilisant que $n \sum p_i \log_2 \left(1/p_i\right) + O(1)$ bits. Pour arriver à ce résultat, il faut remarquer que tout algorithme de type DDG peut être facilement modifié pour retourner la variable aléatoire du niveau d'arrêt en plus de la variable aléatoire générée. De cette information supplémentaire, c'est-à-dire le niveau d'arrêt, nous pouvons extraire les bits aléatoires i.i.d. qui permettent d'atteindre la complexité espérée $n \sum p_i \log_2 \left(1/p_i\right) + O(1)$ au lieu de $n \sum p_i \log_2 \left(1/p_i\right) + O(n)$. Cette annexe se retrouve dans Devroye et Gravel [41].

Extraction de bits aléatoires. Extraire une suite i.i.d. de variables aléatoires de Bernoulli à partir d'une suite i.i.d. de variables aléatoires X_1, X_2, … est un sujet bien connu. Réexpliquons ici à notre façon comment extraire une telle suite afin de l'adapter à notre contexte de génération. Soit F_1, F_2, … une suite de distributions définies sur les entiers positifs. Soit $(p_1, p_2, \ldots)$ un vecteur de probabilités à partir duquel sont générés n variables i.i.d. X_1, X_2, …, X_n avec $n \in \mathbb{N}$. De plus, soit Y_1, Y_2, …, Y_n des variables indépendentes distribuées non-identiquement selon F_{X_1}, F_{X_2}, …, F_{X_n}.

Un cas spécial est celui où $p_1 = 1$ et Y_1, Y_2, …, Y_n sont i.i.d. selon F_1.

Supposons que les entropies binaires respectives des distributions F_1, F_2, … sont $\mathcal{E}_1$, $\mathcal{E}_2$, … et toutes finies. F_i est la distribution de $Y \mid X = i$. Faisons l'hypothèse que

$$\mathcal{E} \stackrel{\text{def}}{=} \sum_{i=1}^{\infty} p_i \mathcal{E}_i < \infty.$$

Théorème 13. *Il existe un algorithme, décrit ci-après, qui sur entrées* (X_1, Y_1), (X_2, Y_2), …, (X_n, X_n) *retourne* R_n *bits i.i.d. non biaisées où*

$$R_n \stackrel{\text{p}}{\to} \mathcal{E} \text{ as } n \to \infty.$$

De plus ces bits sont indépendents de $(X_1, \ldots, X_n)$.

Le théorème 13 décrit combien de bits parfaits nous pouvons extraire des variables Y_1, Y_2, ..., Y_n et cette quantité, R_n, est l'entropie de $(Y_1, Y_2, \ldots Y_n)$.

Nous pouvons supposer que $Y_i = F_{X_i}^{-1}(V_i)$ pour $i \in \{1, \ldots, n\}$ où $V_1, V_2, \ldots, V_n$ sont des variables i.i.d. uniformément sur l'intervalle $[0, 1]$. Pour tout $i \in \mathbb{N}$, soit l'expansion binaire de p_i c'est-à-dire

$$p_i = \sum_{j=1}^{\infty} \frac{b_{ij}}{2^j} \text{ où } b_{ij} \in \{0, 1\}.$$

Pour des raisons de commodités, définissons $p_{ij} \stackrel{\text{def}}{=} \frac{b_{ij}}{2^j}$ pour tout $(i, j) \in \mathbb{N} \times \mathbb{N}$. Aussi, pour tout $i \in \mathbb{N}$, soit

$$F_i(j) = \begin{cases} 0 & \text{if } j = 0, \\ \frac{1}{p_i} \sum_{k=1}^{j} p_{ik} & \text{if } j \geq 1. \end{cases}$$

Il existe une variable continue uniforme $U \in [0, 1]$ telle que toute variable Y_1, Y_2, ..., Y_n puisse être reconstruite uniquement. La variable uniforme continue U et $(Y_1, Y_2, \ldots, Y_n)$ sont reliés par une application bijective donnée par l'algorithme suivant.

Algorithme II.1 : Extraction de bits aléatoires

---

entrée(s) : Une suite de paires (X_1, Y_1), ..., (X_n, Y_n) avec X_ℓ et Y_ℓ telles que décrites précédemment pour chaque $\ell \in \{1, \ldots, n\}$.

1: $U_0^- \leftarrow 0$
2: $U_0^+ \leftarrow 1$
3: **pour** $\ell = 1$ **à** n **faire**
4: $\quad U_\ell^- \leftarrow U_{\ell-1}^- + \big(U_{\ell-1}^+ - U_{\ell-1}^-\big) F_{X_\ell}(Y_\ell - 1)$
5: $\quad U_\ell^+ \leftarrow U_{\ell-1}^- + \big(U_{\ell-1}^+ - U_{\ell-1}^-\big) F_{X_\ell}(Y_\ell)$
6: **fin pour**
7: $R_n \leftarrow \max\big\{t \geq 0 \ : \ \lfloor 2^t U_n^- \rfloor = \lfloor 2^t U_n^+ \rfloor\big\}$ {R_n est le plus long préfixe commun des quantités U_n^- et U_n^+.}
8: **retourner** $\lfloor 2^{R_n} U_n^- \rfloor$

Pour l'**exactitude** de l'algorithme II.1, les intervalles $[U_\ell^-, U_\ell^+]$ sont emboîtés. Plus précisément,

$$[U_0^-, U_0^+] \supset \ldots \supset [U_\ell^-, U_\ell^+] \supset [U_{\ell+1}^-, U_{\ell+1}^+] \supset \ldots \supset [U_n^-, U_n^+].$$

Définissons $U = \liminf_{n\to\infty} U_n^- = \limsup_{n\to\infty} U_n^+$. U est la variable aléatoire uniformément distribuée sur $[0,1]$ utilisée pour l'inversion de la distribution conjointe des $(Y_1 \mid X_1, \ldots, Y_n \mid X_n)$. Nous avons

$$U \in [U_n^-, U_n^+] = \bigcap_{\ell=1}^{n} [U_\ell^-, U_\ell^+].$$

En plus, pour toute itération,

$$\frac{U - U_j^-}{U_j^+ - U_j^-} \stackrel{\mathcal{L}}{=} \text{Uniforme}[U_j^-, U_j^+].$$

Le symbole $\stackrel{\mathcal{L}}{=}$ signifie « est distribué comme ». Nous avons donc que $R_n = \max\left\{t \geq 0 \;:\; \lfloor 2^t U_n^- \rfloor = \lfloor 2^t U_n^+ \rfloor\right\}$, et par conséquent,

$$\frac{1}{2^{R_n}} \lfloor 2^{R_n} U \rfloor \leq U_n^- \leq U \leq U_n^+ \leq \frac{1}{2^{R_n}} \Big(\lfloor 2^{R_n} U \rfloor + 1 \Big).$$

Les bits U_1, U_2, ..., U_{R_n} sont clairement i.i.d.

La **complexité** de l'algorithme II.1 découle théorème 13 que nous prouvons immédiatement.

Preuve du théorème 13. Soit $t \in \mathbb{R}$ et considérons les deux cas : $\{R_n \geq t\}$ et $\{R_n < t\}$. Nous montrons que t est concentrée autour de $n\mathcal{E}$ avec la quantité $\mathcal{E}$ telle que définie précédemment. Avant de considérer les deux cas en question, voici un calcul qui sera utile éventuellement.

$$\mathbf{E}\left(\log_2 \left(\frac{p_{X_1}}{p_{X_1 Y_1}} \right) \right) = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \left(\log_2(p_i) + \log_2 \left(\frac{1}{p_{ij}} \right) \right) p_{ij}$$

$$
\begin{aligned}
&= \sum_{i=1}^{\infty} p_i \log_2(p_i) + \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} p_{ij} \log_2\left(\frac{1}{p_{ij}}\right) \\
&= -\mathcal{E}(X) + \sum_{i=1}^{\infty} p_i \sum_{j=1}^{\infty} \frac{p_{ij}}{p_i}\left(\log_2\left(\frac{p_i}{p_{ij}}\right) + \log_2\left(\frac{1}{p_i}\right)\right) \\
&= -\mathcal{E}(X) + \sum_{i=1}^{\infty} p_i \mathcal{E}_i + \mathcal{E}(X) \\
&\stackrel{\text{def}}{=} \mathcal{E}.
\end{aligned}
$$

Pour l'événement $\{R_n \geq t\}$, nous avons

$$
\begin{aligned}
\Big\{R_n \geq t\Big\} &\subseteq \Big\{\big(U_n^+ - U_n^-\big) < \frac{1}{2^t}\Big\} \\
&= \left\{\prod_{\ell=1}^{n} \frac{p_{X_\ell Y_\ell}}{p_{X_\ell}} < \frac{1}{2^t}\right\} \\
&= \left\{\sum_{\ell=1}^{n} \log_2\left(\frac{p_{X_\ell}}{p_{X_\ell Y_\ell}}\right) > t\right\}. \qquad \text{(II.1)}
\end{aligned}
$$

Les pairs (X_1, Y_1), … (X_n, Y_n) sont i.i.d. et, par conséquent,

$$
\mathbf{E}\left(\log_2\left(\frac{p_{X_1}}{p_{X_1 Y_1}}\right)\right) = \mathcal{E}.
$$

Par la loi des grands nombres, pour tout $\epsilon > 0$, si $t = n(c + \epsilon)$, alors $\mathbf{P}\{R_n > n(\mathcal{E} + \epsilon)\} \to 0$ lorsque $n \to \infty$.

Pour l'autre cas, l'événement $\{R_n < t\}$, nous avons

$$
\Big\{R_n \geq t\Big\} \subseteq \Big\{\lfloor 2^t U_n^+ \rfloor > \lfloor 2^t U_n^- \rfloor\Big\}.
$$

Par la loi des grands nombres, pour tout $\epsilon > 0$ si $t = n(\mathcal{E} - \epsilon)$, alors

$$
\mathbf{P}\Big\{U_n^+ - U_n^- \geq \frac{1}{2^t}\Big\} = \mathbf{P}\left\{\sum_{\ell=i}^{n} \log_2\left(\frac{1}{p_{X_\ell Y_\ell}}\right) \leq t\right\} \to 0.
$$

Fixons arbitrairement un entier $k > 0$,

$$\begin{aligned}\mathbf{P}\{R_n < t+k\} &\leq \mathbf{P}\left\{U_n^+ - U_n^- \geq \frac{1}{2^{t+k}}\right\} + \mathbf{P}\left\{U_n^+ - U_n^- < \frac{1}{2^{t+k}}, \lfloor 2^t U_n^+ \rfloor > \lfloor 2^t U_n^- \rfloor\right\} \\ &\leq o(1) + \frac{2}{2^k} \\ &\to 0 \quad \text{pour } k \text{ et } n \text{ suffisamment grands.}\end{aligned}$$

L'événement $\left\{U_n^+ - U_n^- < \frac{1}{2^{t+k}}, \lfloor 2^t U_n^+ \rfloor > \lfloor 2^t U_n^- \rfloor\right\}$ se produit seulement si $|U - \frac{m}{2^t}| \leq \frac{1}{2^{t+k}}$ et $m \in \mathbb{N}$. □

Génération d'un paquet à partir d'un algorithme de type DDG.

Soit une variable aléatoire $X \in \mathbb{N}$ dont le vecteur de probabilités est $(p_1, p_2, \ldots)$ et d'entropie $\mathcal{E}(X)$ finie. Supposons qu'un algorithme de type DDG est employé pour générer X. Soit L l'ensemble des feuilles de l'arbre de génération sous-jacent à l'algorithme et soit $\texttt{label}(u)$ l'étiquette d'une feuille $u \in L$. Définissons

$$L_i = \{u \in L \ : \ \texttt{label}(u) = i\} \text{ et } i \in \mathbb{N}.$$

Si $d(u)$ dénote la profondeur de $u \in L$, alors

$$p_i = \sum_{u \in L_i} \frac{1}{2^{d(u)}}.$$

Si l'algorithme retourne la variable X, alors l'algorithme s'est arrêté à une feuille élément de L_X. Étant donné $X = i$,

$$\mathbf{P}\big\{\text{Arrêter via } u \in L_i\big\} = \frac{1/2^{d(u)}}{p_i}.$$

Si Y dénote la hauteur de la feuille de sortie étant donnée X et que l'algorithme retourne (X, Y), alors

$$\sum_{i=1}^{\infty} p_i \mathcal{E}(Y \mid X = i) = \sum_{i=1}^{\infty} p_i \sum_{u \in L_i} \frac{1}{p_i} \frac{1}{2^{d(u)}} \log_2 \left(p_i 2^{d(u)}\right)$$

$$
\begin{aligned}
&= \sum_{i=1}^{\infty} \sum_{u \in L_i} \frac{1}{2^{d(u)}} \log_2 \left(2^{d(u)}\right) + \sum_{i=1}^{\infty} p_i \log_2(p_i) \\
&= \sum_{u \in L} \frac{1}{2^{d(u)}} \log_2 \left(2^{d(u)}\right) - \mathcal{E}(X) \\
&= \mathcal{E}(Y) - \mathcal{E}(X).
\end{aligned}
$$

Par exemple, pour l'algorithme de Knuth et Yao, nous avons que $\mathcal{E}(Y) - \mathcal{E}(X) \leq 2$ tandis que pour l'algorithme de Han et Hoshi nous avons que $\mathcal{E}(Y) - \mathcal{E}(X) \leq 3$. Notre méthode pour générer un paquet est correcte pour tout algorithme de type DDG tel que l'entropie $\mathcal{E}(Y)$ est finie.

L'algorithme ci-dessous génère n variables i.i.d. X_1, X_2, ..., X_n en utilisant une file Q contenant les bits extraits qui peuvent être réutilisés. L'algorithme utilise une opération `FetchBit` qui obtient d'abord un bit de la file Q si elle est non vide. Si la file Q est vide, alors `FetchBit` obtient un bit du générateur de Bernoulli (1/2). Nous constatons évidemment que l'opération `FetchBit` est requise à l'intérieur de l'algorithme de génération.

Algorithme II.2 : Génération de paquets

1: $Q \leftarrow \emptyset$ {Initialement, la file est vide.}
2: $R_0 \leftarrow 0$ {Initialement, il n'y aucun bit « recyclé ».}
3: **pour** $i = 1$ **à** n **faire**
4: Générer (X_i, Y_i) par un algorithme de type DDG. {L'algorithme utilise intrinsèquement l'opération `FetchBit` afin d'obtenir des bits du générateur ou de la file Q.}
5: **retourner** X_i
6: Appeler l'algorithme d'extraction sur entrée (X_i, Y_i) et ajouter $R_i - R_{i-1}$ bits à Q.
7: **fin pour**

Nous rappelons que l'annexe I contient la définition de la convergence en probabilité. La convergence en probabilité est utilisée dans le théorème qui suit immédiatement.

Théorème 14. *L'algorithme II.2 utilise N_n bits et*

$$\frac{N_n}{n} \xrightarrow{p} \mathcal{E}(X) \text{ lorsque } n \to \infty \text{ tant que } \mathcal{E}(Y) < \infty.$$

Remarque 9. La minoration fournie par le résultat de Knuth et Yao donne que

$$\mathcal{E}(N_n) \geq n\mathcal{E}(X),$$

et, par conséquent, la procédure est asymptotiquement optimale.

Preuve du théorème 14. Choisissons un entier k large et concentrons-nous sur la valeur de N_{nk}. Soit Q_t la taille de la file au temps t et $Q_0 = 0$. Pour $j \in \{1, \ldots, nk\}$, soit T_j le nombre de bits requis pour générer X_j sans avoir recours à l'extraction. Les variables aléatoires T_j sont i.i.d. et ainsi :

$$N_{nk} = \left(\sum_{j=1}^{nk} T_j \right) - R_{nk} + Q_{nk}.$$

Par la loi des grands nombres

$$\frac{T_1 + T_2 + \ldots + T_{nk}}{\mathbf{E}(T_1 + T_2 + \ldots + T_{nk})} \xrightarrow{\mathrm{p}} 1 \text{ lorsque } n \to \infty.$$

Notons que $\mathbf{E}(T_1 + T_2 + \ldots + T_{nk}) = nk\mathcal{E}(Y)$ car les variables T_j sont i.i.d. Par le théorème 13, nous avons que $\big(R_{nk}/nk\big) \xrightarrow{\mathrm{p}} \mathcal{E}(Y) - \mathcal{E}(X)$ lorsque $n \to \infty$. Par conséquent,

$$\begin{aligned} \frac{N_{nk}}{nk} &= \mathcal{E}(Y) + o_{\mathrm{p}}(1) - \big(\mathcal{E}(Y) - \mathcal{E}(X)\big) + o_{\mathrm{p}}(1) + \frac{Q_{nk}}{nk} \\ &= \mathcal{E}(X) + o_{\mathrm{p}}(1) + \frac{Q_{nk}}{nk}. \end{aligned}$$

Le résultat est prouvé si $\big(Q_{nk}/nk\big) \xrightarrow{\mathrm{p}} 0$ lorsque $n \to \infty$. Pour montrer que $\big(Q_{nk}/nk\big) \xrightarrow{\mathrm{p}} 0$, nous n'avons qu'à considérer une majoration de Q_{nk} car $Q_{nk} \geq 0$. Par

conséquent,

$$Q_{nj} \leq Q_{n(j-1)} + \big(R_{nj} - R_{n(j-1)}\big) - \min_{1\leq j\leq k}\{T_{nj}, Q_{n(j-1)}\}.$$

Puisque $\big(R_n/n\big) \xrightarrow{\text{p}} \mathcal{E}(Y) - \mathcal{E}(X)$ et que $\big(V_n/n\big) \xrightarrow{\text{p}} \mathcal{E}(Y)$, nous avons que

$$\max_{1\leq j\leq k}\left|\frac{R_{nj} - R_{n(j-1)}}{n} - \Big(\mathcal{E}(Y) - \mathcal{E}(X)\Big)\right| \xrightarrow{\text{p}} 0 \text{ et} \tag{II.2}$$

$$\max_{1\leq j\leq k}\left|\frac{T_{nj}}{n} - \mathcal{E}(Y)\right| \xrightarrow{\text{p}} 0. \tag{II.3}$$

Fixons $\epsilon > 0$ et soit A la probabilité que les deux événements (II.2) et (II.3) soient plus petits que ϵ de sorte que $\mathbf{P}\{A^c\} = o(1)$. Il est important de remarquer que sur A,

$$\begin{aligned} Q_{nj} &\leq \begin{cases} Q_{n(j-1)} + \big(\mathcal{E}(Y) - \mathcal{E}(X) + \epsilon\big)n - \big(\mathcal{E}(Y) - \epsilon\big)n & \text{si } Q_{n(j-1)} \geq \big(\mathcal{E}(Y) - \epsilon\big)n, \\ Q_{n(j-1)} + \big(\mathcal{E}(Y) - \mathcal{E}(X) + \epsilon\big)n - Q_{n(j-1)} & \text{sinon.} \end{cases} \\ &\leq \max\big\{Q_{n(j-1)}, \big(\mathcal{E}(Y) - \mathcal{E}(X) + \epsilon\big)\big\} \quad \text{si } 2\epsilon \leq \mathcal{E}(X). \end{aligned}$$

et, par conséquent,

$$\begin{gathered} \max_{1\leq j\leq k} Q_{nj} \leq \big(\mathcal{E}(Y) - \mathcal{E}(X) + \epsilon)n \text{ et} \\ \frac{Q_{nk}}{nk} \leq \frac{\mathcal{E}(Y) - \mathcal{E}(X) + \epsilon}{k}. \end{gathered}$$

Si nous choisissons k suffisamment grand de sorte que $\big(\big(\mathcal{E}(Y) - \mathcal{E}(X) + \epsilon\big)/k\big) \leq \epsilon$, alors

$$\mathbf{P}\left\{\frac{Q_{nk}}{nk} > \epsilon\right\} \leq \mathbf{P}\{A^c\} = o(1).$$

□

Annexe III

Construction d'une densité avec entropie de partition divergente et entropie différentielle nulle

Nous allons construire une densité de probabilité ayant une entropie différentielle finie et pour laquelle l'entropie de partitionnement pour une partition arbitraire $\mathcal{A}$ est infinie. En d'autres termes, l'utilisation de l'entropie différentielle n'est pas justifiée si l'entropie de partition diverge.

Pour $k \in \mathbb{N}$, soit $p_k > 0$ tel que

$$\sum_{k=1}^{\infty} p_k = 1 \quad \text{et} \quad \sum_{k=1}^{\infty} p_k \log_2 \left(\frac{1}{p_k}\right) = +\infty,$$

comme, par exemple, $p_k = C\frac{1}{k\log(k)}$ où C est la constance de normalisation.

Soit une suite croissante (a_k) de réels tels que $a_k + p_k \leq a_{k+1}$. Définissons $A_k \subset \mathbb{R}$ par $A_k = [a_k, a_k + p_k)$ de sorte que l'ensemble des A_k forme une collection d'intervalles disjoints. Utilisons les ensembles A_k pour définir la densité f d'une variable aléatoire X telle que $\mathcal{E}(f) = 0$ et $\mathcal{E}_{\mathcal{R}}(X) = +\infty$ où $\mathcal{R}$ est une partition de $\mathbb{R}$ en intervalles de taille ϵ c'est-à-dire $\mathcal{R} = \bigcup_{i\in\mathbb{Z}} R_i$ et $R_i = [i\epsilon, (i+1)\epsilon)$ pour tout $i \in \mathbb{Z}$. La densité de X est

$$f(x) = \sum_{k=1}^{\infty} \mathbb{1}_{\{x\in A_k\}}.$$

Pour tout $x \in \mathbb{R}$, $f(x) \in \{0, 1\}$ et, de plus,

$$\mathcal{E}(f) = \int_{-\infty}^{+\infty} f(x)\log_2\left(\frac{1}{f(x)}\right)dx = \sum_{k=1}^{\infty}\int_{x\in A_k} f(x)\log_2\left(\frac{1}{f(x)}\right)dx = 0.$$

Nous obtenons $\mathcal{E}_{\mathcal{R}}(X) = +\infty$ en créant des « trous » entre les A_k, mais, avant de procéder ainsi, donnons la distribution F de X. Afin d'identifier F, définissons les

quatre quantités suivantes :

$$\begin{aligned} q_0 &= 0, \\ q_k &= \sum_{j=1}^{k} p_j \quad \text{pour } k \geq 1, \\ B_0 &= (-\infty, a_1), \text{ et} \\ B_k &= [a_k + p_k, a_{k+1}) \quad \text{pour } k \geq 1. \end{aligned}$$

La distribution de X est donc

$$F(x) = \sum_{k=1}^{\infty} (x - a_k + q_{k-1}) \mathbb{1}_{\{x \in A_k\}} + \sum_{k=0}^{\infty} q_k \mathbb{1}_{\{x \in B_k\}}. \tag{III.1}$$

Pour $k \geq 1$, si $B_k = \emptyset$, alors il n'y a aucun trou entre les intervalles A_k c'est-à-dire que $a_k + p_k = a_{k+1}$ pour $k \geq 1$ et, visuellement, nous avons la figure III.1.

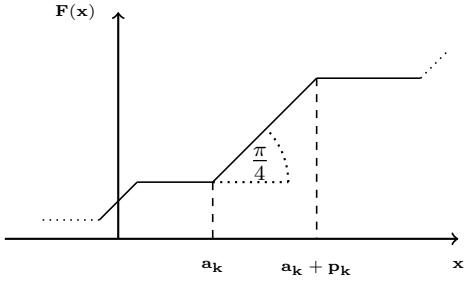

Figure III.1 – Distribution absolument continue, entropie de partition divergente et entropie différentielle nulle

Il faut un nombre infini dénombrable de trous entre les intervalles A_k et, par conséquent, étant donnée une suite (k_j) d'indices pour $j \in \mathbb{N}$, nous avons que $a_{k_j} + p_{k_j} < a_{k_j+1}$. Les quantités p_k définissant une distribution discrète, on a que pour tout $\epsilon > 0$, il existe $K \in \mathbb{N}$ tel que pour tout $k \geq K$, l'inégalité $p_k < \epsilon$ est vraie. Puisque $\mathcal{R}$ est une partition de $\mathbb{R}$, alors il existe i_K tel que pour tout $i \geq i_K$, $A_i = [a_i, a_i + p_i) \subset R_i$, ce qui est toujours possible par la liberté de choix des

quantités a_k. Par conséquent, $\mathbf{P}\{X \in R_i\} = p_i$ et

$$\begin{aligned}
\mathcal{E}_{\mathcal{R}}(X) &= \sum_{i=1}^{\infty} \mathbf{P}\{X \in R_i\} \log_2 \left(\frac{1}{\mathbf{P}\{X \in R_i\}} \right) \\
&> \sum_{i=i_K}^{\infty} \mathbf{P}\{X \in R_i\} \log_2 \left(\frac{1}{\mathbf{P}\{X \in R_i\}} \right) \\
&= \sum_{i=i_K}^{\infty} p_i \log_2 \left(\frac{1}{p_i} \right) \\
&= +\infty.
\end{aligned}$$

Comme nous le verrons ultérieurement, générer une instance de X avec précision d'au moins ϵ prend donc un nombre infini de bits aléatoires.

Il est intéressant de remarquer que s'il n'y a aucun trou entre les A_k, alors la distribution de X est uniforme sur l'intervalle $[a_1, a_1 + 1)$ et $\mathcal{E}_{\mathcal{R}}(X)$ est ainsi fini. En effet, $\mathcal{E}_{\mathcal{R}}(X)$ est finie tant qu'il y a un nombre fini de trous.

Nous complétons cette annexe en donnant une condition suffisante pour la convergence de l'entropie différentielle.

Lemme 5. *Soit $X \in \mathbb{R}$ une variable aléatoire de densité f.*

$$\textit{Si} \quad \mathbf{E}\big(\log_2(1 + |X|) \big) < \infty \ \textit{alors} \quad \int_{\mathbb{R}} f(x) \log_2 \left(\frac{1}{f(x)} \right) dx < \infty.$$

Preuve du lemme 5. Soit $A = \{x \in \mathbb{R} \ : \ f(x) \leq 1\}$ et g la densité de Cauchy centrée en zéro c'est-à-dire

$$g(x) = \frac{1}{\pi} \frac{1}{1 + x^2} \text{ for } x \in \mathbb{R}.$$

Grâce aux faits que

$$\begin{aligned}
\big(1 + |x|\big)^2 &\geq 1 + |x|^2 \\
&= 1 + x^2 \quad \text{et}
\end{aligned}$$

$$t \log_2\left(\frac{1}{t}\right) \leq \frac{1}{e \log(2)} \text{ pour tout } t > 0,$$

si $\mathbf{E}\big(\log_2(1+|X|)\big) < \infty$, alors

$$\begin{aligned}
\int_{\mathbb{R}} f(x) \log_2\left(\frac{1}{f(x)}\right) dx &< \int_A f(x) \log_2\left(\frac{1}{f(x)}\right) dx \\
&= \int_A f(x) \log_2\left(\frac{1}{g(x)} \frac{g(x)}{f(x)}\right) dx \\
&= \int_A f(x) \log_2\left(\frac{1}{g(x)}\right) dx + \int_A g(x) \frac{f(x)}{g(x)} \log_2\left(\frac{g(x)}{f(x)}\right) dx \\
&< \int_{\mathbb{R}} \log_2(\pi) f(x) dx + 2\mathbf{E}\big(\log_2(1+|X|)\big) + \frac{1}{e \log(2)} \int_{\mathbb{R}} g(x) dx \\
&< \infty.
\end{aligned}$$

□

Annexe IV

Distance de Wasserstein et l'inversion

Étant données les variables aléatoires X et Y de distributions respectives F et G, soit $\mathcal{M}$ la classe de toutes les distributions conjointes définies sur le plan réel tel que les distributions marginales coïncident avec X et Y. Nous rappelons donc la définition de la distance de Wasserstein qui est

$$W_\infty(F,G) \stackrel{\text{def}}{=} \inf_{(X,Y)\in\mathcal{M}} \operatorname{ess\,sup} \|X-Y\|_\infty = \inf_{(X,Y)\in\mathcal{M}} \operatorname{ess\,sup} |X-Y|.$$

Nous écrirons $W(F,G)$ à partir de maintenant.

Nous avons le nouveau théorème suivant :

Théorème 15.

$$W(F,G) = \sup_{u\in[0,1]} |F^{-1}(u) - G^{-1}(u)|.$$

Preuve du théorème 15. Si $U \in [0,1]$ est une variable uniforme et continue, alors $F^{-1}(U) \stackrel{\mathcal{D}}{=} X$ et $G^{-1}(U) \stackrel{\mathcal{D}}{=} Y$. Par conséquent,

$$W(F,G) \le \operatorname{ess\,sup} |F^{-1}(U) - G^{-1}(U)| \le \sup_{u\in[0,1]} |F^{-1}(u) - G^{-1}(u)|. \qquad \text{(IV.1)}$$

Si F (ou G) a des plateaux, alors n'importe quelle valeur de $F^{-1}(u)$ telle que

$$\inf\{x \ : \ F(x) \ge u\} \le F^{-1}(u) \le \sup\{x \ : \ F(x) \ge u\}$$

satisfait la ligne (IV.1) en vertu de ess sup.

Nous démontrons que $\sup_{u\in[0,1]} |F^{-1}(u) - G^{-1}(u)| \le W(F,G)$ par contradiction. Rappelons la notation suivante :

$$F(x) = \mathbf{P}\{X \le x\}, \qquad \text{(IV.2)}$$

$$F^{-1}(u) = \inf\{x \ : \ F(x) \ge u\}, \qquad \text{(IV.3)}$$

$$G(y) = \mathbf{P}\{Y \leq x\} \text{ et} \tag{IV.4}$$

$$G^{-1}(u) = \inf\{y \ : \ G(y) \geq u\}. \tag{IV.5}$$

Étant donnée une valeur réelle $\delta > 0$ telle $W(F, G) = \delta$, il existe par la définition de W une distribution conjointe du couple (X, Y) satisfaisant

$$\mathbf{P}\{|X - Y| \leq \delta\} = 1. \tag{IV.6}$$

Afin de rendre non probabiliste (déterministe) la ligne (IV.6), nous utilisons des ensembles de mesure nulle pour altérer la distribution du couple (X, Y) de sorte que $|X - Y| \leq \delta$. Supposons (pour une contradiction) qu'il existe $u \in (0, 1)$ tel que

$$F^{-1}(u) = G^{-1}(u) - \theta \text{ et } \theta > \delta.$$

Par souci d'une notation commode et plus compacte, écrivons $x_u = F^{-1}(u)$ et $y_u = G^{-1}(u)$. Visuellement, nous avons la figure IV.1 suivante :

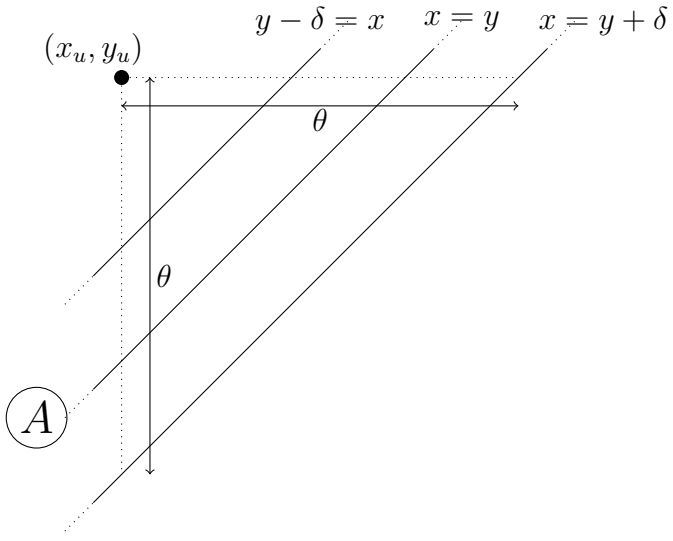

Figure IV.1 – Wasserstein et l'inversion

Soit l'ensemble $A = \{(x, y) \in \mathbb{R}^2 \ : \ |x - y| \leq \delta\}$ tel que représenté sur la figure

IV.1. Le plan est ainsi séparé en régions de sorte que

$$\begin{aligned}\mathbf{P}\{(X,Y)\in A\} &= 1,\\ \mathbf{P}\{(X,Y)\in A\cap\big((-\infty,x_u)\times\mathbb{R}\big)\} &= u \text{ et}\\ \mathbf{P}\{(X,Y)\in A\cap\big(\mathbb{R}\times(y_u,\infty)\big)\} &= 1-u.\end{aligned}$$

Par conséquent,

$$\mathbf{P}\{(X,Y)\in A\cap\big((x_u,\infty)\cap(-\infty,y_u)\big)\}=0, \text{ mais}$$

$$\mathbf{P}\{Y\le y_u\}=\mathbf{P}\{Y\le x_u+\delta\}\Longrightarrow G(y_u)=G(x_u+\delta).$$

Puisque $G^{-1}(u)\stackrel{\text{def}}{=}\inf\{y \ : \ G(y)\ge u\}$ donc $y_u\neq G^{-1}(u)$ (contradiction). Par conséquent pour tout $u\in[0,1]$, nous avons que $|F^{-1}(u)-G^{-1}(u)|\le\delta$.

Le résultat est généralisé à toutes les distributions par un argument d'approximation. Soit X et Y des variables continues de distributions respectives F et G et supposons que $W(F,G)=\operatorname{ess\,sup}|X-Y|$. Soit $\epsilon>0$ arbitraire, soit U continûment uniforme sur $[0,1]$ indépendante de X et Y et soit les variables $X_\epsilon=X+\epsilon U$ et $Y_\epsilon=Y+\epsilon U$ de distributions respectives F_ϵ et G_ϵ. Remarquons que

$$W(F_\epsilon,G_\epsilon)=\operatorname{ess\,sup}|F_\epsilon^{-1}(U)-G_\epsilon^{-1}(U)|. \tag{IV.7}$$

Si X et Y sont couplées de sorte que $\operatorname{ess\,sup}|X-Y|=W(F,G)$ et puisque $|X-X_\epsilon|\le\epsilon$ et $|Y-Y_\epsilon|\le\epsilon$, alors par l'inégalité du triangle,

$$\big|\operatorname{ess\,sup}|X_\epsilon-Y_\epsilon|-\operatorname{ess\,sup}|X-Y|\big|\le 2\epsilon.$$

Aussi $|F_\epsilon^{-1}(U)-F^{-1}(U)|\le\epsilon$ et $|G_\epsilon^{-1}(U)-G^{-1}(U)|\le\epsilon$ et, par la ligne (IV.7),

$$\operatorname{ess\,sup}|F^{-1}(U)-G^{-1}(U)|\le W(F_\epsilon,G_\epsilon)\le\operatorname{ess\,sup}|F^{-1}(U)-G^{-1}(U)|+4\epsilon.$$

Puisque ϵ est arbitraire, la preuve est complétée. □

Annexe V

Génération par convolution d'une exponentielle tronquée

Cette annexe se retrouve dans Devroye et Gravel [41]. Si X est une variable exponentielle de moyenne 1, alors $\lfloor X \rfloor$ et $X - \lfloor X \rfloor \stackrel{\text{def}}{=} \{X\}$ sont indépandentes, consulter Devroye [16]. De plus, $\lfloor X \rfloor$ est une variable géométrique de paramètre $1/e$ et $\{X\}$ est une variable exponentielle tronquée sur l'intervalle $[0, 1]$ de moyenne 1. Si f dénote la densité de $\{X\}$, alors

$$f(x) = \frac{e^{-x}}{1 - e^{-1}}.$$

Le prochain lemme indique que $\{X\}$ est la convolution de variables de Bernoulli indépendantes et non identiquement distribuées.

Dorénavant, pour ce qui est de cette annexe, X dénotera une variable exponentielle tronquée de moyenne 1 sur l'intervalle $[0, 1]$.

Théorème 16. *Soit $(X_1, \ldots, X_j, \ldots)$ une suite de variables de Bernouilli indépendantes et non identiquement distribuées telles que*

$$\begin{aligned} \mathbf{P}\{X_j = 2^{-j}\} &= p_j \in [0, 1], \\ \mathbf{P}\{X_j = 0\} &= 1 - p_j, \text{ et} \\ \frac{p_j}{1 - p_j} &= e^{-1/2^j} \text{ pour tout } j \in \mathbb{N}. \end{aligned}$$

Si $X = \sum_{j=1}^{\infty} X_j$, alors X est une variable exponentielle tronquée de moyenne 1.

Preuve du lemme 16. La preuve est basée principalement sur le fait que la transformée de Fourier d'une somme de variables indépendantes est le produit des transformées de Fourier des variables. De l'expression $\frac{p_j}{1-p_j}$, nous déduisons que

$$p_j = \frac{e^{-1/2^j}}{e^{-1/2^j} + 1},$$

$$1 - p_j = \frac{1}{e^{-1/2^j} + 1}.$$

La transformée de Fourier de X_j est

$$\begin{aligned}\mathbf{E}(e^{\imath X_j t}) &= p_j e^{\imath t/2^j} + (1 - p_j) \\ &= \frac{e^{((-1+\imath t)/2^j)} + 1}{e^{-1/2^j} + 1}.\end{aligned}$$

Puisque X est la somme des variables indépendantes X_j, nous avons que $F = F_{X_1} * F_{X_2} * \cdots$, où $*$ dénote l'opération de convolution et, par conséquent,

$$\begin{aligned}\mathbf{E}(e^{\imath X t}) &= \prod_{j=1}^{\infty} \mathbf{E}(e^{\imath X_j t}) \\ &= \prod_{j=1}^{\infty} \frac{e^{((-1+\imath t)/2^j)} + 1}{e^{-1/2^j} + 1} \\ &= \lim_{n\to\infty} \prod_{j=1}^{n} \frac{e^{((-1+\imath t)/2^j)} + 1}{e^{(-1/2^j)} + 1} \qquad \text{(V.1)} \\ &= \frac{1 - e^{(-1+\imath t)}}{1 - e^{-1}} \frac{1}{-1 + \imath t}. \qquad \text{(V.2)}\end{aligned}$$

La ligne (V.2) est la transformée de Fourier de $f_X(x)$. Le passage de la ligne (V.1) à la ligne (V.2) s'explique par les deux faits qui suivent :

$$\frac{e^{((-1+\imath t))} - 1}{e^{((-1+\imath t)/2^n)} - 1} = \prod_{j=1}^{n-1} \left(1 + e^{((-1+\imath t)/2^j)}\right) \text{ et } \lim_{n\to\infty} \frac{e^{(-1/2^n)} - 1}{e^{((-1+\imath t)/2^n)} - 1} = \frac{-1}{-1 + \imath t}.$$

□

Étant donné un paramètre $\epsilon > 0$, posons $k = \lceil \log_2 (1/\epsilon) \rceil$. La variable aléatoire $X = \sum_{j=1}^{k} X_j$ peut prendre 2^k valeurs. Pour chaque valeur de X correspond une et une seule valeur du vecteur $(2X_1, \ldots, 2^k X_k)$. La fonction de masse conjointe de

$(2X_1, \ldots, 2^k X_k)$ se factorise puisque les X_i sont indépendants. Par conséquent,

$$\mathbf{P}\left\{X = \sum_{j=1}^{k} x_j\right\} = \prod_{j=1}^{k} p_j^{(x_j 2^j)}(1-p_j)^{1-(x_j 2^j)}.$$

Nous avons donc les ingrédients pour l'algorithme qui suit et qui génère une variable exponentielle tronquée et, de facto, une variable exponentielle.

Algorithme V.1 : Génération d'une variable exponentielle tronquée de moyenne 1

---

entrée(s) : Un paramètre $\epsilon > 0$ de précision.

1: $k \leftarrow \lceil \log_2\left(1/\epsilon\right) \rceil$.
2: Exécuter l'algorithme de Knuth et Yao pour échantillonner $\mathbf{P}\{X = i\}$ avec $i \in \frac{1}{2^k}\{0, \ldots, 2^k - 1\}$.
3: **retourner** La valeur retournée par l'algorithme de Knuth et Yao.

L'entropie de X est la même que celle du vecteur $(2^1 X_1, \ldots, 2^k X_k)$ qui la somme des entropies de X_j. Si $\mathbf{P}\{X_j = 0\} = 1/\left(e^{-(1/2^j)}+1\right) = p_j$ et $\mathbf{P}\{X_j = 2^{-j}\} = 1-p_j$, alors

$$\begin{aligned}
\mathcal{E}(X_j) &= p_j \log_2(1/p_j) + (1-p_j)\log_2(1/(1-p_j)) \\
&= \log_2\left(1 + e^{-1/2^j}\right) + \frac{1}{\log(2)}\frac{p_j}{2^j}.
\end{aligned}$$

Par conséquent,

$$\begin{aligned}
\mathcal{E}(X) &= \sum_{j=1}^{k} \mathcal{E}(X_j) \\
&= \log_2 \prod_{j=1}^{k}\left(1 + e^{-1/2^j}\right) + \frac{1}{\log(2)}\sum_{j=1}^{k}\frac{p_j}{2^j} \\
&\leq k + \log_2(1-e^{-1}) + \frac{1}{\log(2)}\left(\frac{1}{2^k} + 1 - \frac{1}{2^{k+1}}\right) \qquad \text{(V.3)} \\
&\leq k + \log_2(1-e^{-1}) + \frac{1}{\log(2)} + \frac{1}{\log(2)}\frac{1}{2^k}.
\end{aligned}$$

Afin d'établir la ligne (V.3), nous avons que

$$\log_2 \prod_{j=1}^{k} \left(1 + e^{-1/2^j}\right) = \log_2 \left(1 - e^{-1}\right) - \log_2 \left(1 - e^{-1/2^k}\right) \tag{V.4}$$

$$\leq \log_2 \left(1 - e^{-1}\right) + k + \frac{1}{\log(2)} \frac{1}{2^k}. \tag{V.5}$$

Pour la ligne (V.4), nous utilisons plusieurs fois l'identité $(1 - e^z) = (1 + e^{z/2})(1 - e^{-z/2})$ valide pour tout $z \in \mathbb{C}$. Pour la ligne (V.5), la fonction $1 - e^{-z} - ze^{-z}$ est croissante sur $z \in [0, 1]$ puisque sa dérivée première est positive donc $ze^{-z} \leq 1 - e^{-z}$. Pour $z = 1/2^k$, nous avons que $-\log(1 - e^{-1/2^k}) \leq k \log(2) + \frac{1}{2^k}$. Pour établir la ligne (V.3), nous avons que

$$\sum_{j=1}^{k} \frac{p_j}{2^j} \leq \sum_{j=1}^{k} \frac{1}{2^j} = 1 - \frac{1}{2^{k+1}}.$$

Par le théorème 3,

$$\begin{aligned}
\mathbf{E}(T) &\leq \mathcal{E}_{\mathcal{A}}(X) + 2 \\
&= \left(k + \log_2(1 - e^{-1}) + \frac{1}{\log(2)} + \frac{1}{\log(2)} \frac{1}{2^k}\right) + 2 \\
&= k + \log_2(1 - e^{-1}) + \log_2(e) + 2 + \frac{\log_2(e)}{2^k}.
\end{aligned}$$

En tenant compte de la partie entière,

$$\begin{aligned}
\mathbf{E}(T) &\leq \left\lceil \log_2 \left(1/\epsilon\right) \right\rceil + \log_2(e) + (2 - e) \log_2(1 - e^{-1}) + 4 + \frac{\log_2(e)}{2^k} \\
&= \left\lceil \log_2 \left(1/\epsilon\right) \right\rceil + 7.360698 + o(1).
\end{aligned}$$

Il semble qu'en vertu du théorème qui suit de Kakutani en 1948 dans [28], les seules lois absolument continues utiles en général qui peuvent être engendrées par la somme de variables de Bernoulli indépendantes non identiquement distribuées sont les lois uniformes et exponentielle tronquée.

Théorème 17 (Kakutani (1948))**.** *Pour tout $i \in \mathbb{N}$, soit $p_i \in [0,1]$ et X_i des variables de Bernoulli indépendantes telles $\mathbf{P}\{X_i = 1\} = p_i$. Si $X = \sum_{i=1}^{\infty} X_i 2^{-i}$, alors*

$$\begin{aligned} X \text{ est singulière} &\Leftrightarrow \sum_{i=1}^{\infty} \left(p_i - \frac{1}{2}\right)^2 \text{ diverge,} \\ X \text{ est absolue continue} &\Leftrightarrow \sum_{i=1}^{\infty} \left(p_i - \frac{1}{2}\right)^2 \text{ converge,} \\ X \text{ est discrète} &\Leftrightarrow \prod_{i=1}^{\infty} \left(\frac{1}{2} + \left|p_i - \frac{1}{2}\right|\right) > 0. \end{aligned}$$

Un exemple intéressant de deux distributions singulières dont la somme est absolument continue est le suivant. Reprenons la suite de variables indépendantes non identiquement distribuées pour générer une variable exponentielle tronquée de moyenne 1 sur l'intervalle $[0,1]$ c'est-à-dire

$$\begin{aligned} \mathbf{P}\{X_j = 2^{-j}\} &= p_j \\ &= \frac{e^{-1/2^j}}{e^{-1/2^j} + 1}, \\ \mathbf{P}\{X_j = 0\} &= 1 - p_j \\ &= \frac{1}{e^{-1/2^j} + 1}. \end{aligned}$$

Considérons

$$\begin{aligned} X &= \sum_{j=1}^{\infty} X_{2j}, \\ Y &= \sum_{j=1}^{\infty} X_{2j-1} \text{ et} \\ Z &= X + Y. \end{aligned}$$

En vertu du théorème de Kakutani, X et Y ont des distributions singulières. Quant à Z, sa distribution est une exponentielle tronquée de moyenne 1.

Remarquons enfin que certaines variables singulières peuvent être générées en pire cas. La variable $X \in [0,1]$ de Cantor telle

$$X \stackrel{\text{def}}{=} \sum_{j=1}^{\infty} \frac{X_j}{3^j},$$

et X_j sont i.i.d. pour tout entier $j \geq 1$ selon

$$\begin{aligned} \mathbf{P}\{X_j = 0\} &= \frac{1}{2}, \\ \mathbf{P}\{X_j = 1\} &= 0, \\ \mathbf{P}\{X_j = 2\} &= \frac{1}{2}. \end{aligned}$$

Pour générer une variable de Cantor de précision ϵ, il suffit en pire cas de $\lceil \log_3 \left(\frac{1}{\epsilon}\right) \rceil$ bits aléatoires non biaisées i.i.d.

Un autre exemple de variable singulière qui puisse être générée en pire est la variable $X \in [0,1]$ telle que

$$X \stackrel{\text{def}}{=} \sum_{j=1}^{\infty} \frac{X_{2j}}{2^{2j}} = 0.0X_20X_4\cdots,$$

et X_{2j} sont non biaisées et i.i.d. pour tout entier $j \geq 1$. Pour générer une instance de X de précision ϵ, il suffit en pire cas de $\lceil \log_2 \left(\frac{1}{\epsilon}\right) \rceil$ bits aléatoires non biaisées i.i.d.

Annexe VI

Majoration spectrale d'une distribution quantique discrète

Dans cette annexe, nous montrons comment il est toujours possible de borner une distribution discrète quantique par une distribution discrète ayant une structure plus simple permettant d'établir un protocole simple, mais qui malheureusement n'est pas efficace dans la majorité des cas.

Soit ρ une matrice de densité de taille $d \times d$ c'est-à-dire une matrice définie positive telle que $\mathrm{tra}(\rho) = 1$. Une matrice définie positive est une matrice Hermitienne ayant toutes ses valeurs propres positives. Soit U une matrice unitaire de taille $d \times d$. Une transformation unitaire représentée par une matrice unitaire préserve le produit scalaire, ce qui est équivalent en dimension finie au fait que $UU^\dagger = I$ où I est la matrice identité. Les rangées de U forment une base de l'espace vectoriel $\mathbb{C}^d$. Le symbole $\dagger$ dénote la transconjuguée U. La matrice $U\rho U^\dagger$ représente l'évolution du système initialement dans l'état ρ. Soit $|e_i\rangle$ les vecteurs colonnes canoniques ayant un 1 à la i^{e} rangée et 0 ailleurs pour $i \in \{0, \ldots, d-1\}$. Pour des raisons de commodité, le rang d'une coordonnée d'un vecteur débute à zéro. Aussi $\langle e_i|$ (vecteur ligne) dénote la transconjuguée de $|e_i\rangle$ c'est-à-dire $|e_i\rangle = \langle e_i|^\dagger$. Le vecteur de probabilités d'intérêt est donné par la diagonale de $U\rho U^\dagger$ qui est

$$(\langle e_0|U\rho U^\dagger|e_0\rangle, \ldots, \langle e_i|U\rho U^\dagger|e_i\rangle, \ldots, \langle e_{d-1}|U\rho U^\dagger|e_{d-1}\rangle). \tag{VI.1}$$

La quantité $\langle e_i|U\rho U^\dagger|e_i\rangle$ est la i^{e} entrée de la diagonale de $U\rho U^\dagger$. Si X est une variable aléatoire discrète représentée par (VI.1), alors $\mathbf{P}\{X = i\} = \langle e_i|U\rho U^\dagger|e_i\rangle$. Pour plus de détails concernant les fondements probabiliste et statistique de la mécanique quantique, consulter Holevo [27].

En utilisant la définition de la multiplication de matrice

$$\mathbf{P}\{X = i\} = \left(U\rho U^\dagger\right)_{ii}$$

$$= \sum_{k=0}^{d-1} \sum_{\ell=0}^{d-1} u_{ik} \rho_{k\ell} \bar{u}_{i\ell}. \tag{VI.2}$$

Nous allons obtenir la distribution d'une autre variable aléatoire Y ayant le même ensemble de valeurs que X et telle que $\forall i \in \{0, \ldots, d-1\}$, $\exists C$ telle que $\mathbf{P}\{X = i\} \leq C\mathbf{P}\{Y = i\}$. Si $d = 2^n$ et $U = \bigotimes_{j=1}^{n} U_j$ où U_j sont des matrices unitaires de taille 2×2, alors la distribution de Y est une combinaison convexe de produits de Bernoulli.

La constante C, bien qu'indépendante de tout i, dépend indirectement de ρ. *En effet, C est le nombre de valeurs strictement positives de la diagonale de ρ.* Si l'algorithme distribué de von Neumann du *chapitre 4* (et non celui du chapitre 5) est appliqué directement alors il est inefficace si C est grand.

$$\begin{aligned}
\left(U\rho U^\dagger\right)_{ii} &= \sum_{k=0}^{d-1} \sum_{\ell=0}^{d-1} u_{ik} \rho_{k\ell} \bar{u}_{i\ell} \\
&\leq \sum_{k=0}^{d-1} \sum_{\ell=0}^{d-1} u_{ik} \sqrt{\rho_{kk}} \sqrt{\rho_{\ell\ell}} \bar{u}_{i\ell} \text{ cf. Bhatia [3]} \\
&= \sum_{k=0}^{d-1} u_{ik} \sqrt{\rho_{kk}} \left(\sum_{k=0}^{d-1} u_{ik} \sqrt{\rho_{kk}} \right)^\dagger \\
&= \left| \sum_{k=0}^{d-1} u_{ik} \sqrt{\rho_{kk}} \right|^2 \\
&= \left| C \sum_{k=0}^{d-1} \frac{1}{C} u_{ik} \sqrt{\rho_{kk}} \right|^2 \\
&\leq C^2 \sum_{k=0}^{d-1} \frac{1}{C} \left| u_{ik} \sqrt{\rho_{kk}} \right|^2 \\
&= C \sum_{k=0}^{d-1} |u_{ik}|^2 \rho_{kk} \\
&= C\mathbf{P}\{Y = i\}.
\end{aligned}$$

La deuxième inégalité ci-dessus découle des inégalités des moyennes de Hölder.

Dans le cas de GHZ, $C = 2$, et nous pouvions s'en sortir facilement avec les

techniques du chapitre 4. Une autre distribution quantique discrète pour laquelle $C = n$ utile en informatique quantique est celle de Werner. Pour la distribution de Werner, les techniques du chapitre 4 sont inefficaces. Pour tout $b \in \{-1, +1\}^n$, et $i \in \{1, \ldots, n\}$ et pour tout $(\theta_i, \varphi_i) \in [0, 2\pi) \times [-\pi/2, \pi/2]$, la distribution de Werner est donnée par

$$\mathbf{P}(b) = \frac{1}{n}\left(\left(\sum_{i=1}^{n} r_i(b)\cos(\theta_i)\right)^2 + \left(\sum_{i=1}^{n} r_i(b)\sin(\theta_i)\right)^2\right),$$

où

$$r_i(b) = \left(\prod_{j\neq i}\cos\left(\frac{1}{2}\big(\varphi_j - (\pi/2)b_j\big)\right)\right)\left(-\sin\left(\frac{1}{2}\big(\varphi_i - (\pi/2)b_i\big)\right)\right)$$

Par l'inégalité précédente (ou même Cauchy-Schwarz dans ce cas-ci), $\mathbf{P}(b) \leq n\mathbf{Q}(b)$ et $\mathbf{Q}(b) = \frac{1}{n}\sum_{i=1}^{n} r_i^2(b)$. Les entrées, pour $i, j \in \{0, \ldots, 2^n - 1\}$, de la matrice ρ de l'état de Werner sont données par

$$(\rho)_{ij} = \begin{cases} \frac{1}{n} & \text{si } i \text{ et } j \text{ sont des puissances de 2,} \\ 0 & \text{sinon.} \end{cases}$$

Il y a donc n entrées de la diagonale de ρ qui ne sont pas nulles.