ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

SUR LE CALCUL ET LA MAJORATION DE LA DISCRÉPANCE À L'ORIGINE

THÈSE N° 2259 (2000)

PRÉSENTÉE AU DÉPARTEMENT DE MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Eric THIÉMARD

Ingénieur mathématicien diplômé EPFL
originaire de Chénens (FR)

acceptée sur proposition du jury :

Prof. Th. M. Liebling, directeur de thèse
Prof. H. Faure, rapporteur
Prof. R. Ingold, rapporteur
Prof. S. Morgenthaler, rapporteur
Prof. L. E. Trotter, rapporteur

Lausanne, EPFL
2000

à mes parents

Je remercie le Professeur Thomas Liebling pour m'avoir fait bénéficier de ses conseils bienveillants et pour m'avoir laissé libre d'entreprendre un travail de recherche dans un domaine a priori peu en rapport avec ses préoccupations habituelles. Je tiens également à rendre hommage à son excellent cours de modèles de décision, ainsi qu'à celui d'algorithmique d'Alain Prodon. C'est en les fréquentant que m'est venue l'envie de m'intéresser aux mathématiques discrètes et à la recherche opérationnelle.

Je remercie les Professeurs Henri Faure, Rolf Ingold, Stephan Morgenthaler et Leslie Trotter pour avoir accepté de faire partie de mon jury de thèse, pour leurs commentaires avisés et pour l'ambiance détendue qu'ils ont su instaurer lors de l'examen.

Un grand merci à Michela Spada, Sophie Liron et Lionel Pournin qui ont patiemment relu et corrigé une version préliminaire de ce document. Les deux charmantes lectrices en question remarqueront avec soulagement et vraisemblablement stupéfaction l'absence de « il est clair que » dans cette page. Je suis par ailleurs redevable à Christine Lütolf pour son aide précieuse lors de la traduction anglaise du résumé et pour le rôle déterminant qu'elle a joué dans la genèse du thiémardeutsch.

J'ai la chance de partager depuis un certain nombre d'années un bureau avec Jean-François Hêche. Que puis-je vous en dire, sinon que le Jeff est une marmotte insomniaque et bufferisante d'excellente compagnie. Plus sérieusement, il est en partie responsable des progrès que j'ai éventuellement pu faire en mathématiques et je le remercie d'avoir répondu, avec une clairvoyance qui m'étonne toujours, à d'incessantes interrogations trop peu souvent pertinentes. Au moment de quitter l'EPFL, j'ai bien sûr une pensée émue pour les agréables moments passés avec plusieurs générations de membres du ROSO : Vincent, François, Christian, Didier, Antonio, Andrea, Paul-Jean, Nicolas, Jean-Albert, Kim, Gérard, Frank, Michel, ainsi que l'inestimable M$^{\text{me}}$ Lieber.

Je tiens également à remercier chaleureusement mes parents, mon frère et mes amis pour le soutien qu'ils m'ont toujours apporté. Enfin, et pour terminer en beauté, un merci très particulier à Michela, l'inaltérable rayon de soleil de cet été de rédaction, pour sa patience et ses encouragements.

Résumé

La discrépance est une mesure de la non-uniformité d'une séquence de points distribués dans un cube unité multidimensionnel. Cette grandeur est intéressante à bien des égards, mais sa détermination s'avère particulièrement délicate. Les nouveaux algorithmes proposés dans ce travail permettent de la calculer ou de la majorer dans des cas inaccessibles jusqu'alors.

En dehors de ces résultats, cette présentation ne se limite pas à l'énoncé de quelques définitions et propriétés classiques, mais discute également certaines implications pratiques du sujet. En effet, la notion de discrépance étant liée à de nombreuses autres disciplines, un important effort de synthèse a été consenti afin de mettre en perspective quelques-unes de ces connexions. La première partie de ce document est un tour d'horizon qui commence par une introduction à la théorie de l'équirépartition, une discipline vouée à l'étude du concept d'uniformité. Puis, après une brève description de la très populaire méthode de Monte-Carlo pour l'évaluation d'intégrales multiples, la version déterministe de cette technique est discutée plus en détail. Cette dernière repose sur l'utilisation de séquences de points particulièrement bien distribuées, les suites à discrépance faible, dont quelques exemples sont donnés à la fin de cette introduction.

La seconde partie de ce travail commence par la présentation des deux méthodes connues pour le calcul de la discrépance. La complexité de ces algorithmes est exponentielle et l'impraticabilité de l'un d'entre eux est illustrée numériquement. Apparemment, ces difficultés semblent contrebalancées par l'existence de majorations pour certains types de séquences. Hélas, le nombre minimal de points menant à l'obtention d'une borne non triviale par ce biais croît de manière exponentielle avec la dimension.

Nous proposons un nouveau principe permettant de calculer des bornes inférieures et supérieures pour la discrépance. La largeur maximale de l'intervalle en question pouvant être spécifiée a priori, cette approche permet d'atteindre une précision arbitraire. Cette construction repose sur la considération de partitions du cube unité en intervalles. Deux formes de décompositions sont envisagées. Tout d'abord, dans le cas particulier des grilles, il est possible d'établir des bornes pour des séquences de très grande taille en exploitant judicieusement cette structure. En revanche, pour un même effort de calcul, le second type de partition proposé permet d'obtenir des intervalles de meilleure qualité, mais uniquement pour des ensembles de points de cardinalité plus modeste. Dans les deux cas, des techniques de comptage multidimensionnelles spécialisées ont été développées. En fin de compte, dans de nombreuses situations, cette méthode est tout simplement la seule approche connue applicable.

Nous montrons ensuite que le calcul de la discrépance peut également se ramener à la résolution d'une famille d'instances d'un problème de géométrie combinatoire qui consiste à déterminer l'intervalle ancré à l'origine de volume minimal ou maximal contenant un nombre fixé de points d'une séquence donnée. Après avoir exprimé ces problèmes sous forme de programmes linéaires en nombres entiers, leur résolution est abordée à l'aide d'heuristiques et de techniques de génération de coupes et d'énumération par séparation et évaluation. De plus, nous proposons une approche pour traiter de manière implicite une partie importante des configurations impliquées. La méthode obtenue permet de calculer la discrépance de séquences dans des cas qui n'avaient jamais été atteints auparavant. D'autre part, comme le processus

consiste en une suite d'améliorations apportées à un intervalle initial, il peut être interrompu à chaque instant, fournissant alors une borne inférieure et supérieure reflétant l'effort de calcul consenti jusque-là.

On peut regretter que les techniques proposées dans ce document mènent à des algorithmes qui ne sont pas polynomiaux. Toutefois, les résultats des expériences numériques effectuées montrent qu'elles permettent de traiter des cas qui, jusqu'alors, semblaient totalement hors de portée. Ajoutons que ces méthodes n'ont certainement pas encore atteint leur version définitive. En effet, l'approche algorithmique par calcul (ou amélioration) d'intervalles introduite dans ce travail est inédite et son développement ne demande qu'à être poursuivi. Ainsi, cette thèse inaugure une voie de recherche prometteuse pour le calcul et la majoration de la discrépance.

Abstract

Star discrepancy is a measure for the irregularity of point sequences in a multidimensional unit cube. This measure is interesting in many respects, but its computation is known to be very difficult. With the new algorithms proposed here, the star discrepancy or bounds for it can be computed in cases so far inaccessible.

In the first part of this thesis, we discuss classical definitions and properties as well as practical implications of the subject. Indeed, the notion of discrepancy being linked to many other disciplines as well, a substantial effort has been made to show some of these connections. An overview is given, starting with an introduction to the theory of irregularities of distribution, a discipline concerned with the study of the uniformity concept. Then, after a short presentation of the very popular Monte Carlo method for numerical integration, the deterministic counterpart of this technique is discussed in more detail. The latter relies on the use of low-discrepancy sequences, i.e. point sets with strong regularity properties that are briefly presented at the end of this introduction.

The second part of this thesis starts with the presentation of the two known approaches to compute the star discrepancy. The complexity of these algorithms is exponential and the impracticability of one of them is illustrated numerically. One might have hoped that these difficulties are partially counterbalanced by the existence of upper bounds for some special types of low-discrepancy sequences. Unfortunately, the minimum number of points for which these classical bounds become meaningful grows exponentially with the dimension.

Here we propose a new principle for computing upper and lower bounds for the star discrepancy. Since the maximum width of such an interval can be specified beforehand, this approach yields arbitrary high precision. Our construction is based on the consideration of finite partitions of the unit cube into subintervals. Two types of decompositions are studied here. First, in the case of grids it is possible to compute bounds for the star discrepancy of very large sequences by exploiting this special decomposition structure. For the same computational effort, a second partition algorithm is proposed, which yields much better results, but for smaller point sets only. Moreover, in both cases, specialized multidimensional counting techniques needed to be developed. Note that, in many situations, our technique is simply the only known method which is applicable.

We then show that the computation of the star discrepancy is amenable to solving a set of instances of a problem in computational geometry, namely finding the subinterval (anchored at the origin) of minimum or maximum volume that contains a specific number of points of a given sequence. After the formulation of these problems as integer linear programs, their solution is tackled with heuristics, cutting plane generation, and branch and bound methods. We also propose a technique to enumerate most configurations involved implicitly. Our method allows the computation of the star discrepancy in cases out of reach thus far. Moreover, since a lower and an upper bound are maintained throughout and gradually improved during the process, the algorithm can be stopped at any time, providing an interval whose quality reflects the computational effort spent up to this point.

While both proposed techniques lead to algorithms which are not polynomial, the results of our numerical experiments clearly demonstrate their practical efficiency. Moreover, the novel algorithmic approach proposed here for computing (or improving) a lower and an upper bound has certainly not yet been carried to its limit. We are convinced this development is worth further efforts.

Table des matières

Remarques

Un certain nombre de conventions d'écriture ont été respectées et l'usage de plusieurs symboles a été réservé à la désignation de grandeurs ou d'objets mathématiques précis. La signification de la plupart de ces symboles, ainsi que quelques définitions importantes sont données en pages 5 et 6. La majeure partie des notations retenues recoupent celles de Niederreiter [Nie92]. Par ailleurs, la terminologie utilisée s'inspire essentiellement de certains travaux en langue française de Faure [Fau94] et de Tuffin [Tuf97].

De nombreux résultats classiques sont cités dans ce travail. Nous avons choisi de ne reproduire aucune des démonstrations correspondantes. En revanche, une mention de l'auteur ainsi qu'un renvoi au document original contenant la preuve manquante sont donnés dans chaque cas.

Dans le cadre de la comparaison du comportement asymptotique des fonctions, les notations standard sont utilisées. Si f et g sont deux fonctions à valeurs réelles non négatives de la variable n, on note

- ▷ $f(n) = O(g(n))$ si et seulement s'il existe une valeur n_0 et une constante réelle c telles que
$$\forall n \geq n_0,\ f(n) \leq cg(n)\,;$$
- ▷ $f(n) = \Omega(g(n))$ si et seulement s'il existe une valeur n_0 et une constante réelle c telles que
$$\forall n \geq n_0,\ f(n) \geq cg(n)\,;$$
- ▷ $f(n) = \Theta(g(n))$ si et seulement si $f(n) = O(g(n))$ et $f(n) = \Omega(g(n))$.

Partie 1

La discrépance : un tour d’horizon

CHAPITRE 1

L'équirépartition des séquences

Le mot *discrépance* signifie « désaccord » (du latin *discrepare* : rendre un son différent, ne pas être d'accord). Dans un contexte mathématique, il s'agit d'une mesure de l'écart existant entre une situation de référence (généralement l'uniformité parfaite) et une configuration donnée.

Les questions de discrépance constituent une partie essentielle de la théorie de l'équirépartition (voir les excellents ouvrages de Kuipers et Niederreiter [KN74], Beck et Chen [BC87], Drmota et Tichy [DT97] et Matoušek [Mat99]). Nous commençons ce chapitre par une brève introduction aux problèmes typiques que l'on étudie dans cette discipline. Cette partie est suivie de l'énoncé de quelques définitions et notations relatives à la discrépance, ainsi que de la description des rares cas que l'on sait manipuler facilement. Ce chapitre se termine par l'évocation de quelques résultats asymptotiques importants.

1.1 Introduction

L'étude de l'équirépartition des séquences est née de la volonté d'obtenir des réponses quantitatives à des questions relatives à la distribution uniforme. Son origine remonte au début du siècle (le vingtième), avec les travaux de Weyl sur l'équirépartition dans l'intervalle unité [Wey16]. Une autre contribution majeure apportée à cette théorie est la conjecture énoncée en 1935 par van der Corput [vdC35].

CONJECTURE 1.1 *Si* $\{x^1, x^2, \ldots\}$ *est une suite de nombres réels dans l'intervalle* $[0,1]$*, alors pour tout entier* k*, il existe un entier* n *et deux intervalles de même taille, tels que le nombre d'éléments de la séquence* $\{x^1, \ldots, x^n\}$ *contenus dans ces deux intervalles diffère d'au moins* k *(voir figure 1.1).*

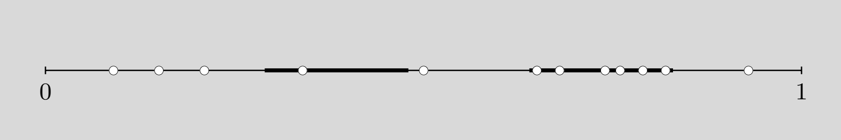


FIG. 1.1. Une séquence de $n = 12$ nombres réels compris entre 0 et 1, ainsi que deux intervalles de même taille. Celui de droite contient $k = 5$ points de plus que celui de gauche.

Cette conjecture exprime l'idée qu'aucune suite ne peut être arbitrairement bien distribuée ou, de manière équivalente, qu'une certaine irrégularité est nécessairement présente dans toute séquence de nombres réels. Elle a été démontrée pour la première fois en 1945 par van Aardenne-Ehrenfest [vAE45].

Considérons maintenant la question fondamentale suivante dans le cas d'une séquence $\{x^1, \ldots, x^n\}$ dans l'intervalle $[0,1]$: quelle est la manière la plus uniforme de choisir ces n valeurs ? Il est clair que l'ensemble de n points équidistants donné par

$$x^i = \frac{2i-1}{2n}, \text{ pour } i = 1, \ldots, n \tag{1}$$

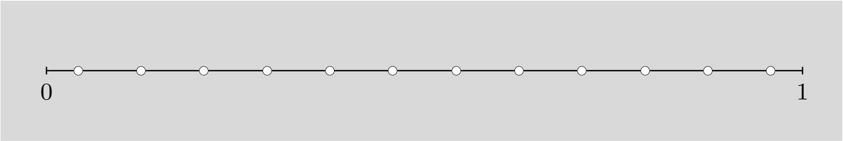


FIG. 1.2. Séquence de $n = 12$ valeurs uniformément distribuées entre 0 et 1.

est à peu près le seul prétendant sérieux (avec peut-être $x^i = (i-1)/(n-1)$) au titre de séquence la mieux distribuée dans l'intervalle unité (voir figure 1.2). Par contre, si l'on considère la même question transposée au cas du carré unité, la situation est nettement moins tranchée. En effet, il est alors possible d'énoncer plusieurs définitions défendables, mais inconciliables de l'uniformité. Par exemple, il paraît tout à fait raisonnable de choisir un critère du type suivant :

> On note $\mathcal{I}_2^*$ l'ensemble des rectangles P de la forme $[0, \beta_1^P) \times [0, \beta_2^P)$ dans le carré unité (voir figure 1.3) et $\lambda(P) = \beta_1^P \beta_2^P$ l'aire d'un tel rectangle. Une séquence de n points $x = \{x^1, \dots, x^n\}$ est dite *optimalement distribuée* si elle minimise le supremum sur tous les rectangles $P \in \mathcal{I}_2^*$ de la *déviation*

$$\big||P \cap x| - n\lambda(P)\big|. \tag{2}$$

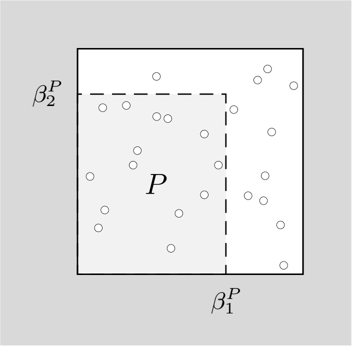


FIG. 1.3. Une séquence de $n = 25$ points et un rectangle $P \in \mathcal{I}_2^*$ dans le carré unité.

Comme on s'attend à ce que la proportion des points situés dans un rectangle $P \in \mathcal{I}_2^*$ soit proche de son aire $\lambda(P)$, il semble en effet naturel de juger peu uniforme une séquence pour laquelle il est possible d'exhiber un rectangle avec une importante déviation (2). Malheureusement, à ce jour, on ne connaît pas de méthode permettant de construire une séquence optimalement distribuée dans le carré unité pour n quelconque (pour $n \in \{1, \dots, 6\}$, le problème a néanmoins été résolu par White [Whi77]).

Une question légèrement moins complexe est de décider si, pour n'importe quelle valeur de n, il existe une séquence $x = \{x^1, \dots, x^n\}$ telle que le supremum sur $P \in \mathcal{I}_2^*$ de la déviation (2) soit inférieur à une constante indépendante de n. L'énoncé paraît très simple, mais c'est un lauréat de la médaille Fields, K. F. Roth[1], qui réfuta l'affirmation en 1954 [Rot54]. Toutefois, le problème n'a été résolu de manière réellement satisfaisante qu'en 1972 avec les travaux de Schmidt [Sch72]. Ce dernier a pu déterminer, pour une séquence optimalement distribuée, l'ordre exact de croissance de ce supremum en fonction de n.

Pour la généralisation de cette assertion en dimension supérieure, Roth a montré que la réponse est également négative. Malheureusement, il n'existe pas à ce jour de résultat équivalent au pas

---

[1] Roth a reçu cette insigne distinction en 1958, mais pour la résolution d'autres problèmes non moins complexes.

supplémentaire que Schmidt a su franchir pour le carré unité. Ces questions sont discutées de manière moins informelle dans la section 1.4.

Soulignons le fait que dans l'approche ci-dessus, il aurait été parfaitement légitime de remplacer $\mathcal{I}_2^*$ par la famille des triangles ou des sous-ensembles convexes inclus dans le carré unité. Il aurait également été possible d'utiliser un critère d'uniformité reposant sur une tout autre mesure, comme par exemple la distance minimale entre deux points de la séquence.

1.2 Quelques discrépances

Les notations suivantes seront constamment utilisées pour désigner certaines grandeurs associées aux intervalles multidimensionnels considérés :

- s est un entier positif représentant la *dimension* d'un espace.
- I^s désigne le cube unité semi-ouvert $[0,1)^s$ en dimension s.
- $\bar{I}^s$ désigne le cube unité fermé $[0,1]^s$ en dimension s.
- P représente un *intervalle* multidimensionnel semi-ouvert (ou intervalle tout court) de la forme

$$P = \prod_{j=1}^{s} \left[\alpha_j^P, \beta_j^P\right), \text{ où } 0 \leq \alpha_j^P \leq \beta_j^P \text{ pour tout } j \in \{1, \dots, s\}.$$

On utilise également la notation simplifiée

$$P = \left[\alpha^P, \beta^P\right), \text{ où } \alpha^P = (\alpha_1^P, \dots, \alpha_s^P) \text{ et } \beta^P = (\beta_1^P, \dots, \beta_s^P).$$

- α^P et β^P
  sont deux points qui désignent respectivement le « coin inférieur-gauche » et « supérieur-droit » d'un intervalle P.
- P^- et P^+
  désignent respectivement les intervalles $\left[0, \alpha^P\right)$ et $\left[0, \beta^P\right)$ associés à un intervalle P.
- $\bar{P}$ est la fermeture $\left[\alpha^P, \beta^P\right]$ d'un intervalle P.
- $\lambda(P)$
  est le *volume* d'un intervalle P :

$$\lambda(P) = \prod_{j=1}^{s} \left(\beta_j^P - \alpha_j^P\right).$$

- $\mathcal{I}_s$ est l'ensemble des intervalles inclus dans I^s :

$$\mathcal{I}_s = \left\{ P = \prod_{j=1}^{s} \left[\alpha_j^P, \beta_j^P\right) : 0 \leq \alpha_j^P \leq \beta_j^P \leq 1 \right\}.$$

- $\mathcal{I}_s^*$ désigne l'ensemble des intervalles inclus dans I^s et ancrés à l'origine :

$$\mathcal{I}_s^* = \left\{ P = \prod_{j=1}^{s} \left[0, \beta_j^P\right) : 0 \leq \beta_j^P \leq 1 \right\}.$$

Pour les grandeurs relatives aux séquences de points, les notations retenues sont les suivantes :

- ▷ x est une *séquence* (c'est-à-dire un ensemble ordonné d'éléments non nécessairement distincts) dénombrable de points que l'on note

$$x = \{x^1, x^2, \dots\}.$$

Les points considérés appartiennent toujours au cube unité $\bar{I}^s$.

- ▷ x est généralement appelée une *suite* dans le cas d'une séquence infinie.
- ▷ $x(n)$ (où x est une séquence avec $|x| \geq n$) désigne la sous-séquence constituée des n premiers points de x : $x(n) = \{x^1, \dots, x^n\}$.
- ▷ n est utilisé pour un nombre de points. Lorsque x est une séquence finie, on utilise généralement n pour désigner sa cardinalité.
- ▷ x^i est le i^{e} point de x.
- ▷ x^i_j est la j^{e} coordonnée du point x^i, où $j \in \{1, \dots, s\}$.
- ▷ $A(E, x)$ désigne le nombre de points de la séquence x appartenant à l'ensemble $E \subset \bar{I}^s$.

REMARQUE 1.2 On se permet parfois d'emprunter quelques notions ensemblistes telles que $\in$, $\exists$, $\subset$, $\setminus$ ou $\cup$ et de les appliquer à une séquence de points. On ne s'autorise de tels abus de notation que lorsque cela n'induit pas d'ambiguïté : par exemple lorsque l'ordre des éléments ne nous intéresse pas ou que la multiplicité de tous les points est supposée égale à 1. Par contre, il convient de souligner le fait que, dans le cas de la cardinalité $|\,.\,|$ d'une séquence ou d'une sous-séquence possédant une propriété donnée, la multiplicité des points doit être prise en compte lors du comptage. C'est notamment le cas pour $A(P, x(n))$ dans les définitions ci-dessous.

DÉFINITION 1.3 Une suite de points x est dite *équirépartie* si pour tout intervalle[2] $P \in \mathcal{I}_s$, on a

$$\frac{A(P, x(n))}{n} \xrightarrow{n\to\infty} \lambda(P).$$

La notion de discrépance découle directement de cette idée.

DÉFINITION 1.4 Soit une séquence x d'au moins n points dans $\bar{I}^s$. La *discrépance* $D^*_n(x)$ est une mesure de la non-uniformité de x restreinte à ses n premiers points (voir figure 1.3) :

$$D^*_n(x) = \sup_{P \in \mathcal{I}^*_s} \left| \frac{A(P, x(n))}{n} - \lambda(P) \right|. \tag{3}$$

La détermination de cette grandeur revient à chercher l'intervalle ancré à l'origine qui contient la densité la plus anormalement faible ou élevée de points comparativement à son volume. Il est clair que

$$0 < D^*_n(x) \leq 1$$

et que plus la discrépance est petite, plus la séquence est uniforme. Cette définition, énoncée pour une séquence en dimension quelconque, correspond au critère formulé en page 4 dans le cas du carré unité. La question du calcul et de l'évaluation de bornes pour la discrépance d'une séquence constitue le sujet principal de ce travail.

REMARQUE 1.5 Dans une partie de la littérature francophone sur le sujet, ainsi que dans le titre de ce document, $D^*_n(x)$ est appelée *discrépance à l'origine*. Cependant, étant donné notre utilisation récurrente de ce terme, il nous a paru préférable d'opter pour la dénomination simplifiée de *discrépance* afin d'alléger la présentation.

---

[2] La définition correspondante où l'on se limite aux intervalles $P \in \mathcal{I}^*_s$ est équivalente à celle-ci.

Comme suggéré dans l'introduction, la discrépance n'est pas la seule grandeur susceptible d'être utilisée pour caractériser la non-uniformité d'une séquence. Cependant, il se trouve qu'elle est impliquée dans un domaine d'application (la simulation de quasi-Monte-Carlo présentée au chapitre 3) justifiant l'intérêt particulier dont elle jouit depuis une trentaine d'années. Voici néanmoins les mesures alternatives les plus courantes de non-uniformité :

DÉFINITION 1.6 La *discrépance extrême* $D_n(x)$ d'une séquence x d'au moins n points dans $\bar{I}^s$ est donnée par

$$D_n(x) = \sup_{P \in \mathcal{I}_s} \left| \frac{A(P, x(n))}{n} - \lambda(P) \right| .$$

DÉFINITION 1.7 La *discrépance isotrope* $J_n(x)$ d'une séquence x d'au moins n points dans $\bar{I}^s$ est donnée par

$$J_n(x) = \sup_{C \in \mathcal{C}_s} \left| \frac{A(C, x(n))}{n} - \lambda_s(C) \right|$$

où $\mathcal{C}_s$ est la famille des sous-ensembles convexes de $\bar{I}^s$ et λ_s désigne la mesure de Lebesgue dans $\mathbb{R}^s$. Niederreiter a montré [Nie72] qu'il était possible de restreindre le supremum ci-dessus à un certain sous-ensemble de $\mathcal{C}_s$.

Il découle directement de ces définitions que pour toute séquence x d'au moins n points dans $\bar{I}^s$, on a

$$0 < D_n^*(x) \leq D_n(x) \leq J_n(x) \leq 1.$$

De plus, on a pu montrer les relations suivantes :

THÉORÈME 1.8 (Kuipers et Niederreiter [KN74]) *Pour toute séquence x d'au moins n points dans le cube unité $\bar{I}^s$, on a*

$$D_n^*(x) \leq D_n(x) \leq 2^s D_n^*(x).$$

THÉORÈME 1.9 (Niederreiter et Wills [NW75]) *Pour toute séquence x d'au moins n points dans le cube unité $\bar{I}^s$, on a*

$$D_n(x) \leq J_n(x) \leq 4s \left[D_n(x)\right]^{1/s} .$$

Le résultat suivant fournit un lien supplémentaire entre l'équirépartition d'une suite et les différentes discrépances définies ci-dessus :

THÉORÈME 1.10 (Kuipers et Niederreiter [KN74]) *Si $x = \{x^1, x^2, \dots\}$ est une suite de points dans $\bar{I}^s$, alors les propositions suivantes sont équivalentes :*

1° *la suite x est équirépartie ;*

2° $\lim_{n \to \infty} D_n^*(x) = 0$ *;*

3° $\lim_{n \to \infty} D_n(x) = 0$ *;*

4° $\lim_{n \to \infty} J_n(x) = 0.$

La discrépance (3) peut être vue comme la norme L^∞ de la fonction *discrépance locale* qui associe à tout intervalle $P \in \mathcal{I}_s^*$ la valeur

$$\left| \frac{A(P, x(n))}{n} - \lambda(P) \right| \tag{4}$$

Si l'on préfère utiliser la norme L^2, on obtient une nouvelle mesure de non-uniformité :

DÉFINITION 1.11 La *discrépance carrée moyenne* $T_n^*(x)$ d'une séquence x d'au moins n points dans $\bar{I}^s$ est donnée par

$$T_n^*(x) = \left[\int_{P \in \mathcal{I}_s^*} \left(\frac{A(P, x(n))}{n} - \lambda(P) \right)^2 dP \right]^{\frac{1}{2}}. \tag{5}$$

On a clairement

$$0 < T_n^*(x) \leq D_n^*(x) \leq 1. \tag{6}$$

1.3 Les cas abordables

1.3.1 En dimension un. Il est clair que pour $x = \{x^1, \ldots, x^n\} \subset \bar{I}^s$, les valeurs de la discrépance $D_n^*(x)$ et de la discrépance extrême $D_n(x)$ ne dépendent pas de l'ordre des n points dans la séquence. De plus, en dimension $s = 1$, ces grandeurs sont faciles à calculer :

THÉORÈME 1.12 (Niederreiter [Nie72]) *Pour une séquence* $x = \{x^1, \ldots, x^n\} \subset \bar{I}$ *avec*

$$0 \leq x^1 \leq \cdots \leq x^n \leq 1,$$

on a

$$D_n^*(x) = \frac{1}{2n} + \max_{1 \leq i \leq n} \left| x^i - \frac{2i-1}{2n} \right|.$$

THÉORÈME 1.13 (Niederreiter [Nie92]) *Pour une séquence* $x = \{x^1, \ldots, x^n\} \subset \bar{I}$ *avec*

$$0 \leq x^1 \leq \cdots \leq x^n \leq 1,$$

on a

$$D_n(x) = \frac{1}{n} + \max_{1 \leq i \leq n} \left(\frac{i}{n} - x^i \right) - \min_{1 \leq i \leq n} \left(\frac{i}{n} - x^i \right).$$

1.3.2 En dimension deux. De Clerck [Cle86] a pu établir une formule relativement similaire pour la discrépance des séquences $x = \{x^1, \ldots, x^n\} \subset \bar{I}^2$ dont les composantes sont distinctes, *i.e.*

$$x_1^i \neq x_1^j \text{ et } x_2^i \neq x_2^j, \text{ pour tout } 1 \leq i < j \leq n.$$

Bundschuh et Zhu ont simplifié et généralisé cette formule au cas des séquences ne satisfaisant pas forcément cette contrainte :

THÉORÈME 1.14 (Bundschuh et Zhu [BZ93]) *Soit une séquence* $x = \{x^1, \ldots, x^n\} \subset \bar{I}^2$ *dont les points ont été préalablement triés dans l'ordre croissant de leur première composante :*

$$0 \leq x_1^1 \leq \cdots \leq x_1^n \leq 1.$$

On se donne la paire de points auxiliaires $x^0 = (0,0)$ *et* $x^{n+1} = (1,1)$*. Maintenant, pour chaque indice* $i \in \{0, \ldots, n\}$*, on note* $\{\xi_i^0, \ldots, \xi_i^{i+1}\}$ *la séquence obtenue en réordonnant le sous-ensemble de secondes composantes* $\{x_2^0, \ldots, x_2^i, x_2^{n+1}\}$ *de manière à satisfaire*

$$0 = \xi_i^0 \leq \xi_i^1 \leq \cdots \leq \xi_i^i \leq \xi_i^{i+1} = 1.$$

Avec ces notations, on a

$$D_n^*(x) = \max_{0 \leq i \leq n} \max_{0 \leq k \leq i} \max \left\{ \left| \frac{k}{n} - x_1^i \xi_i^k \right|, \left| \frac{k}{n} - x_1^{i+1} \xi_i^{k+1} \right| \right\}.$$

1.3.3 En dimension quelconque. Dans le cas général, le calcul de la discrépance est une tâche réputée très difficile. Il a pourtant été montré (voir Niederreiter [Nie72]) que le problème est discrétisable et résoluble en un nombre fini d'étapes. Malheureusement, la complexité des différents algorithmes connus croît de manière exponentielle avec la dimension (voir chapitres 5 et 8). Par ailleurs, la difficulté du problème est telle que l'on se contente souvent de bornes inférieures et supérieures sur la discrépance (voir chapitres 6 et 7).

En revanche, il se trouve que la discrépance carrée moyenne $T_n^*(x)$ se laisse facilement calculer dans n'importe quelle dimension :

THÉORÈME 1.15 (Warnock [War72]) *Pour une séquence $x = \{x^1, \ldots, x^n\} \subset \bar{I}^s$, on a*

$$(T_n^*(x))^2 = 3^{-s} - \frac{1}{n\,2^{s-1}} \sum_{i=1}^{n} \prod_{d=1}^{s} \left(1 + x_d^i\right)\left(1 - x_d^i\right) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \prod_{d=1}^{s} \left(1 - \max\left\{x_d^i, x_d^j\right\}\right).$$

La formule de Warnock montre que la discrépance carrée moyenne d'une séquence de n points peut être calculée en $O(n^2)$ opérations. Retravaillant le dernier terme de cette expression, Heinrich [Hei96] a obtenu un algorithme en $O(n(\log n)^s)$. Hélas, comme le laissait déjà supposer sa complexité théorique, les expériences numériques de Heinrich ont révélé qu'en dimension élevée ($s > 8$), sa méthode n'est pas plus performante que l'approche directe. D'autre part, se fondant sur une analyse de complexité détaillée, Matoušek [Mat98] a montré que l'algorithme de Heinrich n'est réellement plus efficace qu'une application directe de la formule de Warnock que pour des séquences comprenant au moins 2^{2^s} points.

1.4 Quelques résultats asymptotiques

Les théorèmes 1.12 et 1.13 impliquent que pour toute séquence x de n points dans l'intervalle unité $\bar{I}$

$$D_n^*(x) \geq \frac{1}{2n} \quad \text{et} \quad D_n(x) \geq \frac{1}{n}.$$

On remarque que ces bornes sont atteintes pour la séquence définie dans l'introduction par (1) et représentée sur la figure 1.2. Il existe donc des séquences finies dans $\bar{I}$ telles que $D_n(x) = \Theta(n^{-1})$. Par contre, le théorème suivant montre qu'il n'existe aucune suite x dans $\bar{I}$ pour laquelle $D_n(x) = O(n^{-1})$ pour tout $n \geq 1$:

THÉORÈME 1.16 (Schmidt [Sch72]) *Il existe une constante $c > 0$ telle que pour toute suite x dans l'intervalle unité $\bar{I}$, on a*

$$D_n(x) \geq c\,\frac{\log n}{n} \text{ pour une infinité de valeurs de } n.$$

La meilleure borne connue pour la valeur de cette constante est $c = 0.12$ (Béjian [Béj82]). En utilisant le théorème 1.8, on obtient un résultat correspondant pour la discrépance d'une suite quelconque dans l'intervalle unité :

$$D_n^*(x) \geq 0.06\,\frac{\log n}{n} \text{ pour une infinité de valeurs de } n.$$

Ainsi, il n'existe aucune suite pour laquelle la discrépance décroît plus rapidement que $O(n^{-1}\log n)$. Cependant, on connaît depuis longtemps (voir van der Corput [vdC35]) des suites possédant exactement ce taux (voir section 4.1). En dimension $s = 1$, on connaît donc des suites présentant la plus rapide décroissance possible de la discrépance à une constante près. La meilleure suite connue dans l'intervalle unité est une suite de van der Corput généralisée proposée par Faure [Fau81] (voir théorème 4.6).

THÉORÈME 1.17 (Schmidt [Sch72]) *Il existe une constante* $c' > 0$ *telle que pour toute séquence de points* $x = \{x^1, \dots, x^n\} \subset \bar{I}^2$*, on a*

$$D_n^*(x) \geq c' \, \frac{\log n}{n}.$$

Généralisant en dimension quelconque s les suites données pour $s = 1$ par van der Corput [vdC35], Halton [Hal60] a proposé une famille de suites satisfaisant $D_n^*(x) = O(n^{-1}(\log n)^s)$ (voir section 4.2). Ces suites ont ensuite été adaptées par Hammersley [Ham60] de manière à engendrer des séquences de taille n pour lesquelles $D_n^*(x) = O(n^{-1}(\log n)^{s-1})$ (voir section 4.3).

Considérons maintenant le résultat fondamental suivant :

THÉORÈME 1.18 (Roth [Rot54]) *Il existe une constante* $B_s > 0$ *ne dépendant que de* s*, telle que pour toute séquence* $x = \{x^1, \dots, x^n\} \subset \bar{I}^s$*, on a*

$$D_n^*(x) \geq B_s \, \frac{(\log n)^{(s-1)/2}}{n}.$$

On a également une propriété correspondante pour le cas des séquences infinies :

THÉORÈME 1.19 (Roth [Rot54]) *Il existe une constante* $B_s' > 0$ *ne dépendant que de* s*, telle que pour toute suite* $x = \{x^1, x^2, \dots\} \subset \bar{I}^s$*, on a*

$$D_n^*(x) \geq B_s' \, \frac{(\log n)^{s/2}}{n} \quad \textit{pour une infinité de valeurs de } n.$$

On remarque qu'il existe un écart sérieux entre la minoration $\Omega(n^{-1}(\log n)^{(s-1)/2})$ du théorème 1.18 de Roth et la majoration $O(n^{-1}(\log n)^{s-1})$ sur la discrépance des séquences de Hammersley. Pour les séquences finies en dimension $s = 2$, cet écart a été comblé par le théorème 1.17 de Schmidt. En revanche, pour $s \geq 3$ l'équivalent de ce pas supplémentaire est à l'état de conjecture depuis une quarantaine d'années :

CONJECTURE 1.20 *Il existe une constante* $B_s > 0$ *ne dépendant que de* s*, telle que pour toute séquence* $x = \{x^1, \dots, x^n\} \subset \bar{I}^s$*, on a*

$$D_n^*(x) \geq B_s \, \frac{(\log n)^{s-1}}{n}.$$

Cette affirmation semble particulièrement difficile à aborder, mais il est communément admis qu'elle est sans doute vraie. Un pas, qualitativement important, dans cette direction a été franchi récemment par Baker :

THÉORÈME 1.21 (Baker [Bak99]) *Il existe une constante* $B_s > 0$ *ne dépendant que de* s*, telle que pour toute séquence* $x = \{x^1, \dots, x^n\} \subset \bar{I}^s$ *avec* $s \geq 3$ *et* $n > e^e$*, on a*

$$D_n^*(x) \geq B_s \, \frac{(\log n)^{(s-1)/2}}{n} \left(\frac{\log \log n}{\log \log \log n} \right)^{1/(2s-2)}.$$

Notons également que, pour une suite particulière en dimension $s = 2$, Faure et Chaix [FC96] ont réussi à démontrer que $n^{-1}(\log n)^2$ est bien l'ordre exact de la discrépance. Il s'agit de la suite de Faure correspondante (voir section 4.5), mais on peut également la voir comme une certaine suite de Sobol (voir section 4.4) ou encore comme une construction de Srinivasan [Sri78].

REMARQUE 1.22 Le théorème 1.18 n'est qu'un corollaire. En fait, ce résultat a été établi par Roth pour le cas de la discrépance carrée moyenne $T_n^*(x)$, puis déduit pour la discrépance $D_n^*(x)$ en utilisant la relation (6). De plus, abordant le problème par l'autre extrémité, Roth a également montré [Rot80] qu'il existe des séquences permettant d'obtenir $T_n^*(x) = O(n^{-1}(\log n)^{(s-1)/2})$. Ainsi, pour la discrépance carrée moyenne, les bornes inférieures et supérieures coïncident. Une construction explicite de séquences finies satisfaisant $T_n^*(x) = \Theta(n^{-1}(\log n)^{(s-1)/2})$ a été récemment proposée par Chen et Skriganov [CS]. En revanche, dans le cas des séquences infinies, des constructions explicites ne sont connues qu'en dimension $s = 1$ (voir Proinov [Pro83], ainsi que Chaix et Faure [CF93]).

Il a également été montré que, pour tout $p > 1$, il existe des constructions présentant exactement les mêmes propriétés pour la discrépance moyenne obtenue en prenant la norme L^p à la place de L^2 dans la définition 1.11. Cette généralisation est due aux travaux de Schmidt [Sch77] pour la borne inférieure et de Chen [Che80] pour la borne supérieure.

Ces résultats, ainsi que le théorème 1.21, montrent qu'il se produit un saut lors du passage à la norme L^∞ et donc à la discrépance $D_n^*(x)$ (d'après la conjecture 1.20, il s'agirait même d'un saut important). Tout ceci indique que la discrépance locale (4) est faible pour la majorité des intervalles $P \in \mathcal{I}_s^*$, mais qu'elle est nécessairement d'un ordre de grandeur plus élevé pour au moins l'un d'entre eux.

Partant d'un point de vue totalement différent, un résultat important sur la taille de certaines séquences optimales de points a été obtenu très récemment.

THÉORÈME 1.23 (Heinrich, Novak, Wasilkowski et Woźniakowski [HNWW]) *Pour tout choix de $d \in (0, 1/2)$, on note $n(s, d)$ la cardinalité de la plus courte séquence dans le cube unité $\bar{I}^s$ ayant une discrépance inférieure ou égale à d. Cette taille minimale $n(s, d)$ croît linéairement avec s et au pire de manière quadratique avec d^{-1}.*

Malheureusement, la preuve de ce résultat est basée sur des arguments probabilistes et non sur une approche constructive. Le problème de la génération des séquences optimales reste donc entier. Les résultats d'une expérience présentée dans la section 8.5.4 semblent même indiquer que la construction de ces ensembles minimaux nécessitera probablement le développement de techniques spécifiques.

1.4.1 Les séquences aléatoires. Kiefer a établi quelques résultats sur la distribution asymptotique de la discrépance d'une suite aléatoire de points en dimension quelconque s.

THÉORÈME 1.24 (Kiefer [Kie61]) *Si $x = \{x^1, x^2, \dots\}$ est une suite de variables aléatoires indépendantes et identiquement distribuées selon la loi uniforme dans le cube unité $\bar{I}^s$, alors on a*

$$\limsup_{n\to\infty} \frac{\sqrt{2n}\, D_n^*(x)}{\sqrt{\log\log n}} = 1 \quad p.s.$$

D'autre part, pour tout $\varepsilon > 0$, il existe une constante $c > 0$ dépendant uniquement de ε et de s telle que

$$\text{Prob}\left(\sqrt{n}\, D_n^*(x) \le u\right) \ge 1 - ce^{-(2-\varepsilon)u^2}, \text{ pour tout } u \ge 0.$$

Ces résultats montrent que la discrépance d'une suite aléatoire décroît approximativement comme $O(n^{-1/2})$. De plus, le comportement est similaire pour l'espérance de la discrépance carrée moyenne d'une suite x de variables aléatoires indépendantes et identiquement distribuées selon la loi uniforme dans le cube unité $\bar{I}^s$ (voir par exemple Halton [Hal72] ou Morokoff et Caflisch [MC94]) :

$$E\left[(T_n^*(x))^2\right] = \frac{(1/2)^s - (1/3)^s}{n}. \tag{7}$$

CHAPITRE 2

La méthode de Monte-Carlo

La méthode de Monte-Carlo compte sans doute parmi les outils les plus puissants et les plus utilisés par l'ingénieur d'aujourd'hui. Schématiquement, le principe de base consiste en l'utilisation du hasard pour aborder l'étude d'un problème déterministe. Il s'agit d'une approche particulièrement flexible et possédant un champ d'application extrêmement large. Elle est l'un des piliers d'une discipline tirant pleinement parti de la puissance des ordinateurs actuels : la simulation. La méthode de Monte-Carlo est utilisée avec succès depuis plus d'un demi-siècle dans des domaines aussi variés que la physique, l'analyse numérique, la statistique, la chimie, la finance et la recherche opérationnelle. Elle est même devenue quasiment incontournable dans l'approche de certains problèmes en dimension élevée et pour l'évaluation de modèles stochastiques complexes.

Le problème généralement utilisé pour illustrer la puissance de la méthode de Monte-Carlo est celui de l'intégration numérique[1]. En dimension $s = 1$, les formules de quadrature classiques comme celles du rectangle, du trapèze ou de Simpson, permettent d'approcher l'intégrale d'une fonction par une somme pondérée de ses valeurs prises en différents points. Considérons par exemple la formule du trapèze dans le cas particulier d'une intégrale sur l'intervalle $[0, 1]$ et d'une subdivision en k parties (voir figure 2.1) :

$$\int_0^1 f(t)\,dt \approx \frac{1}{k}\sum_{i=1}^{k}\left(f\left(\frac{i-1}{k}\right)+f\left(\frac{i}{k}\right)\right).$$

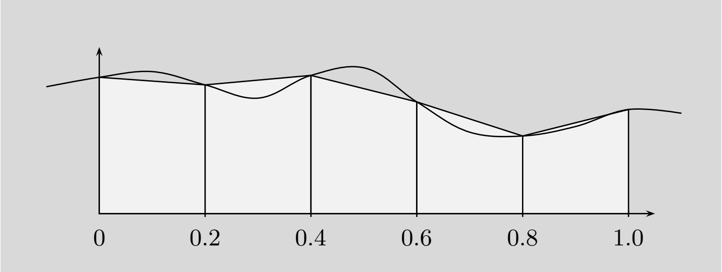


FIG. 2.1. Illustration de la formule du trapèze pour $k = 5$.

Cette méthode nécessite l'évaluation de la fonction f en $n = k + 1$ points. Pour une fonction deux fois continûment différentiable sur $[0, 1]$, on peut montrer (voir Davis et Rabinowitz [DR84]) que l'erreur d'approximation décroît comme $O(n^{-2})$. La généralisation de cette approche en dimension quelconque s par le produit cartésien de s subdivisions de ce type conduit à la considération d'une grille de taille $n = (k + 1)^s$ et garantit une décroissance de l'erreur en $O(n^{-2/s})$. Ainsi, le nombre de points requis pour assurer une certaine précision à une approximation par la méthode du trapèze multidimensionnelle croît de manière exponentielle avec la dimension. Il est communément admis que ce phénomène (qui se reproduit en partant de n'importe quelle formule de quadrature unidimensionnelle) rend cette approche impraticable en dimension $s \geq 5$.

---

[1] Notre présentation se limite à cette application dans le cas particulier d'une intégrale sur le cube unité $\bar{I}^s$.

2.1 Utilisation de l'aléatoire

Soit $f : \bar{I}^s \to \mathbb{R}$, une fonction intégrable sur $\bar{I}^s$. On considère le problème du calcul de l'intégrale

$$\int_{\bar{I}^s} f(t)\,dt. \tag{8}$$

La méthode de Monte-Carlo est basée sur une observation très simple. Si z est une variable aléatoire uniformément distribuée dans le cube unité $\bar{I}^s$ (on note $z \sim U(\bar{I}^s)$), alors l'espérance de la variable aléatoire $M = f(z)$ est égale à la valeur de cette intégrale :

$$E(M) = E\left(f(z)\right) = \int_{\bar{I}^s} f(t)\,dt.$$

Cette transformation revient à remplacer le problème du calcul d'une intégrale par celui de la détermination d'une espérance. Maintenant, considérant n variables aléatoires $x^1, \ldots, x^n$ i.i.d. (indépendantes et identiquement distribuées) $U(\bar{I}^s)$ et utilisant le principe statistique de base consistant à estimer l'espérance d'une variable aléatoire par la moyenne d'un échantillon, on définit l'estimateur

$$M_n = \frac{1}{n}\sum_{i=1}^{n} f(x^i). \tag{9}$$

Il se trouve que M_n est un estimateur non biaisé de l'intégrale (8) :

$$E\left(M_n\right) = \frac{1}{n}\sum_{i=1}^{n} E\left(f(x^i)\right) = E(M) = \int_{\bar{I}^s} f(t)\,dt.$$

On a donc l'*approximation de Monte-Carlo*

$$\frac{1}{n}\sum_{i=1}^{n} f(x^i) \approx \int_{\bar{I}^s} f(t)\,dt \quad \text{avec } x^1, \ldots, x^n \text{ i.i.d. } U(\bar{I}^s). \tag{10}$$

De plus, si la *variance de la fonction* f

$$\sigma^2 = \int_{\bar{I}^s} \left(f(t) - E(M)\right)^2 dt \tag{11}$$

est finie, on obtient

$$\text{Var}\left(M_n\right) = \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}\left(f(x^i)\right) = \frac{\sigma^2}{n} \tag{12}$$

comme variance de l'estimateur de Monte-Carlo M_n. Pour de grandes valeurs de n, le théorème central limite peut alors être invoqué. Il permet de construire un intervalle de confiance du type

$$\left[M_n \pm z_{1-\alpha/2}\,\frac{\sigma}{n^{1/2}}\right] \tag{13}$$

avec couverture à $(1-\alpha)\%$ pour l'intégrale (8), où $z_{1-\alpha/2}$ est le quantile d'une loi normale centrée réduite. Le fait remarquable qui a fait toute la popularité de la méthode de Monte-Carlo est que la taille de cet intervalle décroît comme $O(n^{-1/2})$ indépendamment de la dimension s. Notons qu'en pratique, la valeur de σ doit généralement être estimée.

L'histoire « officielle » de cette technique ne commence qu'en 1949 avec la publication d'un article de Metropolis et Ulam [MU49]. En effet, le début de son utilisation intensive date de cette époque, mais l'approche était en fait déjà connue et appliquée depuis bien plus longtemps (voir Tuffin [Tuf97]). Les ouvrages de Bratley, Fox et Schrage [BFS83], Fishman [Fis96] et Ross [Ros97] sont des références classiques présentant diverses extensions et variations relatives à la simulation de Monte-Carlo.

2.2 Simulation du hasard

Un point crucial pour une bonne mise en œuvre de la méthode de Monte-Carlo est la qualité de la source de hasard utilisée. En effet, l'emploi d'un échantillon de points apparemment aléatoire, mais présentant en fait (de manière sous-jacente) de mauvaises propriétés, peut mener à des résultats incorrects (voir Paskov [Pas97] ou Tezuka [Tez98] pour une illustration expérimentale).

En sa qualité de technique mathématique gourmande en calcul, la méthode de Monte-Carlo se prête naturellement à une mise en œuvre informatique. Cependant, l'ordinateur étant une machine purement déterministe, donc a priori non conçue pour engendrer un comportement aléatoire, on voit poindre les premières difficultés. De plus, les différentes tentatives d'utilisation d'une source de hasard externe issue d'un processus physique n'ont pas été concluantes. C'est ainsi que de nombreux mathématiciens se sont tournés vers le problème de la *simulation du hasard*, c'est-à-dire de la génération d'un processus ayant uniquement l'apparence de l'aléatoire, à partir de moyens exclusivement déterministes. On parle alors de générateurs de nombres *pseudo-aléatoires*. Une telle entreprise requiert néanmoins une définition claire du but poursuivi, à savoir une caractérisation suffisamment précise du concept de séquence aléatoire. Cette problématique est brièvement abordée dans la section 2.2.1. En guise d'introduction, citons la célèbre définition en ce sens énoncée par Lehmer [Leh51] :

> « Une séquence aléatoire est une notion imprécise couvrant l'idée d'une séquence dont chaque élément est impossible à prévoir pour un non-initié et passant un certain nombre de tests statistiques dépendant éventuellement de l'usage pour lequel elle est destinée. »

La plupart des gens jugeraient sans doute la séquence

$$\{0.537, 0.117, 0.836, 0.402, 0.966, 0.271, 0.187, 0.608, 0.727, 0.343\}$$

plus « aléatoire » que celle-ci :

$$\{0.000, 0.200, 0.400, 0.600, 0.800, 0.100, 0.300, 0.500, 0.700, 0.900\}.$$

Cependant, du point de vue de la théorie des probabilités, ce sont deux réalisations rigoureusement équiprobables de l'expérience consistant à générer une séquence de 10 variables aléatoires i.i.d. $U(I)$. Pourtant, bien qu'elles couvrent toutes deux le domaine de manière satisfaisante, la seconde paraît trop « lisse » pour être le fruit du hasard. Généralement, comme Lehmer le suggère, ce genre d'intuition se laisse formaliser sous forme de tests statistiques.

2.2.1 Les suites aléatoires. Qu'entend-on vraiment par *suite aléatoire* ? Plus précisément, que peut-on raisonnablement s'attendre à observer en considérant une suite de réalisations de variables aléatoires i.i.d. $U(I)$? Cette question mène inévitablement à des considérations d'ordre métaphysique sur la nature du hasard.

Le sujet préoccupait déjà certains mathématiciens du début du siècle, mais il se trouve que la théorie moderne des probabilités est construite de manière à éviter scrupuleusement cet écueil, ramenant la question à des notions générales comme celles de distribution et d'indépendance. Pourtant, bien qu'il soit essentiel de savoir préciser avec quelle probabilité une configuration donnée est censée se produire dans le cadre d'une expérience stochastique, il est regrettable de négliger le fond du problème consistant à caractériser un comportement aléatoire.

Bien sûr, le hasard est par essence insaisissable, mais il est néanmoins possible d'énoncer quelques caractéristiques minimales qu'une suite de variables aléatoires i.i.d. $U(I)$ doit nécessairement présenter :

1° une grande imprédictibilité ;

2° certaines propriétés d'équirépartition.

La formalisation de ces concepts est une question particulièrement délicate ayant fait l'objet de nombreux travaux. Un compte rendu accessible et relativement détaillé du sujet est disponible dans l'ouvrage de référence de Knuth [Knu69]. Il s'agit essentiellement d'une synthèse de certains résultats fondamentaux de Church, Kolmogorov, Loveland et Schnorr (entre autres).

Schématiquement, la caractérisation finale (R6) présentée dans [Knu69] stipule qu'une suite peut être qualifiée d'« aléatoire » si elle satisfait la propriété d'équirépartition (définition 1.3) pour une certaine famille infinie de sous-suites infinies de la suite considérée. La principale difficulté consiste à définir une famille qui soit assez générale pour assurer l'imprédictibilité et l'équirépartition de la suite, mais qui soit également suffisamment restreinte pour que la notion ne s'étouffe pas dans l'œuf, c'est-à-dire qui garantisse leur existence (en choisissant par exemple comme famille l'ensemble de toutes les sous-suites infinies, on arrive à la conclusion qu'aucune suite « aléatoire » n'existe).

2.2.2 Les générateurs pseudo-aléatoires. Les générateurs pseudo-aléatoires les plus répandus sont basés sur la méthode des *congruences linéaires* proposée par Lehmer [Leh51]. En partant d'un *germe* quelconque $y^1 \in \{0, \dots, m\}$, la séquence $\{y^1, y^2, \dots\}$ est générée suivant la relation de récurrence

$$y^i = (ay^{i-1} + c) \mod m, \tag{14}$$

où a, m et c sont des entiers non négatifs appelés respectivement *multiplicateur*, *module* et *incrément*. Lorsque $c = 0$, on parle d'un générateur *multiplicatif à congruences linéaires*. Il est clair que la séquence engendrée par (14) est périodique et que la longueur de cette période est au maximum égale à m.[2] Dans tous les cas, on espère que la séquence $x = \{x^1, x^2, \dots\}$ définie par

$$x^i = \frac{y^i}{m}$$

présente des caractéristiques proches de celles d'une suite de variables aléatoires i.i.d. $U(I)$. Le choix des paramètres a, c et m permettant d'atteindre ce but est particulièrement délicat. Néanmoins, les propriétés théoriques de la méthode ont été largement étudiées et l'on dispose de techniques éprouvées permettant de tester les paramètres utilisés (voir Knuth [Knu69] et Niederreiter [Nie92]).

Un exemple de générateur multiplicatif à congruences linéaires ayant eu son heure de gloire est donné par les paramètres $a = 16\,807$ et $m = 2^{31} - 1$ (voir Bratley, Fox et Schrage [BFS83] ainsi que la figure 3.1). Cependant, il a été mis en évidence depuis (voir L'Ecuyer [L'E94]) que ce générateur est en fait de qualité médiocre et que son usage est par conséquent à proscrire.

Il existe de nombreux autres types de générateurs pseudo-aléatoires susceptibles de présenter de meilleures propriétés, notamment une plus longue période (voir L'Ecuyer [L'E98]). Citons par exemple, les générateurs *à congruences linéaires multiples* qui sont basés sur des récurrences du type

$$y^i = (a_1 y^{i-1} + \cdots + a_k y^{i-k}) \mod m,$$

où k est appelé *ordre* du générateur. Plus récemment, on s'est également intéressé à des générateurs faisant intervenir plusieurs récurrences linéaires multiples : les générateurs *combinés à congruences linéaires multiples*. Par exemple, le générateur MRG32k3a de L'Ecuyer [L'E99] (considéré dans les expériences numériques des chapitres 7 et 8) combine 2 générateurs d'ordre 3 de la manière suivante :

$$\begin{aligned} y^i &= (1\,403\,580\,y^{i-2} - 810\,728\,y^{i-3}) \mod (2^{32} - 209) \\ z^i &= (527\,612\,z^{i-1} - 1\,370\,589\,z^{i-3}) \mod (2^{32} - 22\,853) \\ x^i &= \frac{(y^i - z^i) \mod (2^{32} - 209)}{2^{32} - 209} \end{aligned} \tag{15}$$

---

[2] En pratique, les choix $m = 2^{32}$ et (du nombre premier) $m = 2^{31} - 1$ sont particulièrement courants car ils peuvent permettre (suivant la valeur des autres paramètres) d'assurer une période relativement longue et une mise en œuvre sur 32 bits.

Il s'agit d'un générateur rapide, facilement implémentable, de période 2^{191} et ayant été soumis à des tests statistiques exigeants jusqu'en dimension 45.

2.2.3 Les tests. Revenons quelques instants sur la définition R6 (voir section 2.2.1) du concept de suite « aléatoire ». A priori, on pourrait espérer s'en servir pour homologuer ou invalider un générateur pseudo-aléatoire donné. Malheureusement, cette définition n'est d'aucune utilité sur ce point, car elle ne s'applique tout simplement pas aux séquences finies. On en revient donc à l'approche plus pragmatique de la définition de Lehmer (page 15). Dans cette optique, un bon générateur pseudo-aléatoire produit une séquence ne pouvant être, en un temps limité, distinguée d'une séquence réellement aléatoire. Le tri s'effectue sur la base de tests d'uniformité et d'indépendance. Notons que d'un certain point de vue, cette approche tient de la schizophrénie. En effet, elle revient à tester si une séquence que l'on sait purement déterministe, périodique et à valeurs discrètes est en fait aléatoire et continue. En d'autres termes, on évalue une hypothèse que l'on sait clairement fausse dès le départ. En revanche, elle tient également du bon sens : si un générateur est capable de duper une batterie de tests statistiques sophistiqués, on peut raisonnablement penser qu'il simule relativement bien un comportement réellement aléatoire.

Un générateur pseudo-aléatoire étant par nature déterministe et périodique, si suffisamment de temps est mis à disposition, il sera toujours possible de construire un test spécifique pour lequel il échouera lamentablement. Par contre, la méthode de validation en temps fini préconisée pour un générateur donné consiste en l'exigence d'une suite de non-échecs à différents tests réputés difficiles. Une telle approche ne permettra jamais de prouver que le générateur en question est parfaitement fiable pour la simulation ou la méthode de Monte-Carlo, mais chaque test réussi augmentera un peu la confiance qu'il nous inspire.

On distingue essentiellement deux familles de tests : les tests *théoriques* qui touchent à la structure mathématique sous-jacente du générateur (généralement sur toute sa période) et les tests *empiriques* qui considèrent le générateur comme une boîte noire et appliquent un test d'hypothèse statistique aux valeurs retournées dans le but d'y déceler des anomalies significatives. Ci-dessous, nous ne faisons qu'effleurer le sujet (en mettant délibérément l'accent sur certains résultats en rapport avec la discrépance) et renvoyons le lecteur à Knuth [Knu69] et à L'Ecuyer et Hellekalek [LH98] pour une présentation approfondie.

Niederreiter [Nie92] a mené une analyse théorique détaillée des générateurs à congruences linéaires. Par exemple, dans le cas particulier des générateurs multiplicatifs

$$y^i = (ay^{i-1}) \mod m$$

de période maximale $m-1$ (avec m premier), il a montré [Nie77][Nie78] que la discrépance moyenne sur tous les modules a primitifs[3] modulo m de la séquence $x = \{x^1, \ldots, x^{m-1}\} \subset I^s$ donnée par

$$x^i = \left(\frac{y^i}{m}, \ldots, \frac{y^{i+s-1}}{m}\right)$$

est de l'ordre de $O(m^{-1}(\log m)^s \log\log(m+1))$ avec une constante ne dépendant que de s. Considérant les résultats de la section 1.4.1, on en conclut qu'une telle séquence est beaucoup trop régulière pour imiter le comportement typique d'une suite aléatoire. Heureusement, Niederreiter [Nie85] a également montré que pour une sous-séquence de longueur n constituée d'une petite fraction de la période, la discrépance obtenue est plus en accord avec nos espérances, à savoir de l'ordre de $O(n^{-1/2}(\log\log n)^{1/2})$. Il s'agit d'un argument théorique qui renforce le principe empirique stipulant qu'il ne faudrait jamais utiliser plus qu'une fraction négligeable de la période d'un tel générateur.

---

[3] Pour m premier, a est dit *primitif* modulo m si le plus petit λ tel que $a^\lambda = 1 \mod m$ est égal à $m-1$. D'ailleurs, pour m premier, il est nécessaire que a soit primitif modulo m pour que la période du générateur soit maximale.

Un test empirique standard, appelé test d'*uniformité*, consiste à évaluer l'équirépartition d'une sous-séquence $x = \{x^1, \ldots, x^n\} \in I$ obtenue à l'aide du générateur à tester. Soit la fonction de distribution empirique $F_n(t)$ de l'échantillon étudié x :

$$F_n(t) = \frac{A([0,t],x)}{n}, \text{ pour tout } t \in \bar{I}.$$

On veut tester l'hypothèse nulle H_0 : « l'échantillon est issu d'une collection de variables aléatoires i.i.d. selon la loi F ». Dans notre cas, F est la loi uniforme

$$F(t) = t, \text{ pour tout } t \in \bar{I}.$$

Considérons maintenant la statistique du test de Kolmogorov-Smirnov

$$\begin{aligned} K_n &= \sqrt{n} \sup_{t \in \bar{I}} |F_n(t) - F(t)| \\ &= \sqrt{n} \sup_{t \in \bar{I}} \left| \frac{A([0,t],x)}{n} - t \right|. \end{aligned}$$

Ce test non paramétrique bien connu (voir par exemple Pratt et Gibbons [PG81]) permet d'évaluer la qualité de la répartition de n'importe quel générateur pseudo-aléatoire. De plus, on remarque que cette statistique de test (dans le cas particulier de la loi uniforme) n'est autre que la discrépance (3) de la séquence x (on peut donc se servir du théorème 1.12 pour la calculer). Une conséquence intéressante de cette observation est que pour une suite x constituée de variables aléatoires i.i.d. $U(I)$, la distribution de la variable $\sqrt{n}D_n^*(x)$ tend vers la loi de Kolmogorov-Smirnov :

$$\lim_{n \to \infty} \text{Prob}\left(\sqrt{n}\, D_n^*(x) \le t\right) = 1 - 2\sum_{k=1}^{\infty} (-1)^{k+1} e^{-2k^2t^2}, \text{ pour tout } t > 0.$$

Parmi les techniques actuellement utilisées pour l'évaluation de générateurs pseudo-aléatoires, le test *spectral* (Coveyou et MacPherson [CM67]) est peut-être la plus efficace. On sait déjà depuis longtemps que les s-uples de valeurs successives d'un générateur à congruences linéaires multiples définissent une séquence de points dans I^s disposés sur un treillis et qu'il existe des familles d'hyperplans parallèles et équidistants recouvrant l'ensemble de ces points (sur toute la période du générateur). La statistique considérée dans le test spectral est la distance entre une paire d'hyperplans voisins dans une famille maximisant cette distance. Plus cette statistique est petite, meilleur est le générateur. Généralement, afin d'obtenir une sécurité maximale, ce test est successivement appliqué dans un grand nombre de dimensions, afin d'y détecter d'éventuelles singularités[4]. L'approche est néanmoins coûteuse du point de vue du temps de calcul. Par exemple, la sélection des paramètres du générateur MRG32k3a (15) a nécessité plusieurs mois de calcul (voir L'Ecuyer [L'E99]). Les meilleures implémentations actuelles du test spectral permettent toutefois d'atteindre des dimensions de l'ordre de 45 (voir L'Ecuyer et Couture [LC97]).

2.3 Discussion

Dans un article aujourd'hui classique, car à l'origine du développement de la méthode de quasi-Monte-Carlo présentée au chapitre 3, Zaremba [Zar68a] adopte une attitude extrêmement critique, mais néanmoins tout à fait défendable, face à la méthode de Monte-Carlo. En résumé, argumentant sur le fait que malgré tous les efforts consentis dans la construction de bons générateurs pseudo-aléatoires, il

---

[4] L'illustration la plus classique est celle du générateur multiplicatif à congruences linéaires RANDU donné par $a = 65\,539$ et $m = 2^{31}$. Le cas est tristement célèbre aujourd'hui, mais ce générateur a été frénétiquement utilisé dans les années soixante, jusqu'à ce que l'on s'aperçoive qu'en dimension 3, l'ensemble des points obtenus sont répartis sur 15 plans parallèles situés à une distance $1/\sqrt{118} \approx 0.092$ les uns des autres. Il s'agit donc d'un très mauvais générateur.

n'en reste pas moins que les séquences obtenues sont purement déterministes et à valeurs discrètes. Il rejette donc en bloc toute l'analyse probabiliste de la section 2.2. Les mathématiques ne contiennent en effet aucun théorème justifiant un raisonnement probabiliste tel que la considération de l'intervalle de confiance (13) s'il est construit à partir d'une séquence ayant seulement *l'air aléatoire*, quoique cela puisse vouloir dire.

Par contre, le principe de la méthode de Monte-Carlo consistant à approcher l'intégrale (8) par la moyenne M_n (9) des valeurs prises par la fonction en différents points suffisamment bien distribués n'est pas contesté. En effet, pour toute fonction f intégrable au sens de Riemann, il suffit que la séquence $x = \{x^1, x^2, \dots\}$ soit équirépartie pour garantir la convergence

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} f(x^i) = \int_{\bar{I}^s} f(t)\, dt. \tag{16}$$

D'autre part, même en supposant valide l'analyse probabiliste des résultats de la méthode de Monte-Carlo (10), il n'en reste pas moins que les garanties sur l'erreur commise ne sont que probabilistes. En considérant des échantillons de points suffisamment larges, il est clair que l'on peut construire un intervalle de confiance arbitrairement étroit. Par contre, on n'aura jamais la certitude que la vraie valeur de l'intégrale estimée se trouve réellement dans l'intervalle obtenu, aussi petit soit-il. La méthode de Monte-Carlo n'est donc d'aucun secours dans les applications très exigeantes où l'on requiert la valeur exacte de l'intégrale (8) ou une approximation associée à une borne déterministe sur l'erreur commise.

L'intérêt majeur de la méthode de Monte-Carlo réside dans le fait que l'erreur carrée moyenne σ^2/n (12) associée à l'estimateur statistique M_n décroît linéairement avec la taille n de l'échantillon considéré, ceci indépendamment de la dimension s du problème (en revanche, l'influence de la dimension peut se manifester à travers l'innocente constante σ). On notera au passage qu'en pratique, on se permet une petite entorse en appliquant le théorème central limite pour un échantillon de taille finie (dans ce cas, M_n n'est qu'approximativement distribué selon une loi normale). De plus, la valeur de σ^2 est presque toujours inconnue et doit être estimée ; généralement on utilise

$$\hat{\sigma}_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} \big(f(x^i) - M_n\big)^2 .$$

Bien que $\hat{\sigma}_n^2$ soit un estimateur non biaisé de σ^2, il n'en reste pas moins que cette approximation est une source d'erreur supplémentaire (voir Fishman [Fis96]). D'autre part, on a l'assurance que la considération de l'intervalle de confiance (13) est valide pour toute fonction dont la variance σ^2 est finie, ce qui est très général. Malheureusement, la méthode de Monte-Carlo ne permet pas de tirer parti d'autres propriétés de régularité que la fonction f pourrait éventuellement posséder (cette objection est à nuancer compte tenu de la remarque suivante).

Un ingrédient supplémentaire couramment utilisé en simulation de Monte-Carlo est celui des méthodes de *réduction de variance* (voir Bratley, Fox et Schrage [BFS83], Fishman [Fis96] et Ross [Ros97]). Correctement appliquées, ces techniques s'avèrent très efficaces et permettent de réduire l'erreur carrée moyenne (12) d'un facteur constant. En revanche, leur utilisation ne change rien à la décroissance linéaire de cette erreur carrée en fonction de la taille de l'échantillon.

En résumé, la méthode de Monte-Carlo possède des avantages :

- elle est très générale et extrêmement flexible ;
- son taux de convergence est indépendant de la dimension du problème ;
- elle fournit des garanties sur l'erreur d'estimation commise ;
- elle a fait ses preuves ;

ainsi que certains inconvénients :

- ▷ elle fournit des garanties sur l'erreur qui ne sont que probabilistes ;
- ▷ elle ne permet pas d'exploiter les éventuelles propriétés de régularité de la fonction f ;
- ▷ sa convergence est très lente ; l'erreur décroît comme $O(n^{-1/2})$ (ainsi, en moyenne, il est nécessaire de multiplier la taille de l'échantillon considéré par 100 pour espérer réduire l'erreur d'un facteur 10) ;
- ▷ elle repose entièrement sur l'utilisation de nombres aléatoires ; comme on ne sait pas comment en obtenir, on se rabat habituellement sur des générateurs pseudo-aléatoires, ce qui devrait poser certains problèmes de conscience en cas d'analyse probabiliste des résultats ;
- ▷ compte tenu des sources potentielles d'erreur inhérentes à sa mise en œuvre, elle laissera toujours un doute à l'utilisateur quant à la validité des résultats obtenus.

CHAPITRE 3

La méthode de quasi-Monte-Carlo

La méthode de Monte-Carlo (chapitre 2) est une technique d'échantillonnage statistique qui permet, entre autres, d'approcher la valeur d'une intégrale multidimensionnelle $\int_{\bar{I}^s} f(t)\,dt$ par la moyenne des valeurs prises par la fonction f en un ensemble de points générés aléatoirement. Cependant, considérant l'expression (16), il est clair que la propriété fondamentale que l'on cherche à exploiter en utilisant une séquence de variables aléatoires i.i.d. $U(\bar{I}^s)$ n'est pas son imprédictibilité, mais bien son équirépartition. En d'autres termes, la nature stochastique de l'échantillon ne nous intéresse que dans la mesure où elle mène asymptotiquement au « remplissage » du cube unité. Cependant, comme l'illustre la figure 3.1 (au centre) et comme nous l'avons déjà mentionné au chapitre 1, il existe d'autres suites, déterministes, bien mieux distribuées qu'une suite aléatoire typique.

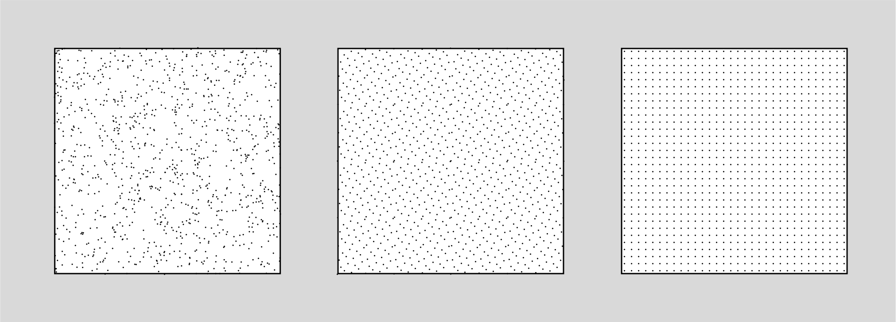


FIG. 3.1. Chacun de ces carrés contient $1\,024$ points provenant (de gauche à droite) d'une suite de variables aléatoires i.i.d. $U(\bar{I}^2)$ (en fait du générateur pseudo-aléatoire de Bratley, Fox et Schrage vu à la section 2.2.2), d'une séquence de Hammersley en base 2 (voir section 4.3) et d'une grille régulière 32×32 (obtenue par simple généralisation de l'expression (1) en dimension 2).

Un simple coup d'œil à la figure 3.1 (partie de gauche) suffit à mettre en évidence la médiocre qualité de la distribution d'une suite aléatoire typique. En effet, par sa nature même (*i.e.* l'indépendance entre les différents points dont elle est constituée), une telle séquence présente de nombreuses singularités : certaines régions contiennent d'importants amas de points, alors que de larges zones restent entièrement vides. Cela se traduit par une discrépance asymptotique en $O(n^{-1/2}(\log\log n)^{1/2})$ (voir section 1.4.1), alors que l'on connaît des séquences x bien meilleures pour lesquelles on a $D_n^*(x) = O(n^{-1}(\log n)^{s-1})$ (voir chapitre 4).

Signalons au passage que dans le cas des grilles régulières (partie droite de la figure 3.1) de $n = k^s$ points obtenues par simple généralisation de l'expression (1) en dimension s (les coordonnées des points en question sont donc de la forme $1/2k, 3/2k, \ldots, (2k-1)/2k$), la qualité de la distribution obtenue est particulièrement mauvaise en dimension élevée. En effet, Sobol [Sob82] a montré que pour une telle grille x, on a $D_n^*(x) = \frac{1}{2}n^{-1/s}$. Précisons que le supremum dans la définition de la discrépance (3) est atteint pour un intervalle vide de la forme $[0,1) \times \cdots \times [0,1) \times [0,1/2k)$.

3.1 L'échantillonnage déterministe

Une technique susceptible d'exploiter les propriétés de ces « bonnes » suites déterministes est présentée dans ce chapitre. En résumé, il s'agit de remplacer l'approximation de Monte-Carlo

$$\frac{1}{n}\sum_{i=1}^{n} f(x^i) \approx \int_{\bar{I}^s} f(t)\,dt \quad \text{avec } x^1,\dots,x^n \text{ i.i.d. } U(\bar{I}^s),$$

par l'*approximation de quasi-Monte-Carlo*

$$\frac{1}{n}\sum_{i=1}^{n} f(x^i) \approx \int_{\bar{I}^s} f(t)\,dt \quad \text{où } x^1,\dots,x^n \in \bar{I}^s \text{ est une séquence à discrépance faible.} \tag{17}$$

Le déterminisme est la spécificité principale de cette seconde méthode ; il se manifeste à deux niveaux :

1° sur le choix de la séquence utilisée $x = \{x^1, x^2, \dots\}$;

2° sur l'estimation de l'erreur d'approximation

$$\left| \frac{1}{n}\sum_{i=1}^{n} f(x^i) - \int_{\bar{I}^s} f(t)\,dt \right|. \tag{18}$$

Le théorème 3.1 est à l'origine de la popularité de la méthode de quasi-Monte-Carlo. Il stipule que l'erreur (18) est dans le pire des cas proportionnelle à la discrépance de la séquence x utilisée. De plus, étant donné que l'on sait construire des séquences avec $D_n^*(x) = O(n^{-1}(\log n)^{s-1})$, on entrevoit la possibilité d'une convergence déterministe asymptotiquement quasi linéaire en la taille de l'échantillon.

Ce résultat est à considérer à la lumière de la vitesse de convergence probabiliste en $O(n^{-1/2})$ de la méthode de Monte-Carlo. Ainsi, en théorie, les performances de la méthode de quasi-Monte-Carlo paraissent extrêmement prometteuses. Hélas, on s'aperçoit rapidement que d'un point de vue pratique ces heureuses perspectives sont à nuancer.

Commençons par énoncer ce résultat fameux, connu sous le nom d'*inégalité de Koksma-Hlawka*. Il a d'abord été établi par Koksma pour $s = 1$, puis généralisé en dimension quelconque par Hlawka.

THÉORÈME 3.1 (Hlawka [Hla61]) *Si $f : \bar{I}^s \to \mathbb{R}$ est une fonction à variation $V(f)$ bornée au sens de Hardy et Krause, alors pour toute séquence de points $x = \{x^1, \dots, x^n\} \subset I^s$, on a*

$$\left| \frac{1}{n}\sum_{i=1}^{n} f(x^i) - \int_{\bar{I}^s} f(t)\,dt \right| \le V(f) D_n^*(x).$$

Ainsi, l'erreur d'approximation (18) est dans le pire des cas égale au produit de la variation $V(f)$ (une grandeur qui ne reflète que l'irrégularité de la fonction f) et de la discrépance $D_n^*(x)$ (qui mesure uniquement la qualité de la répartition de la séquence).

L'évaluation de cette borne soulève deux nouvelles questions particulièrement difficiles :

1° Le calcul de la discrépance : ce problème est le sujet principal de ce travail. Malheureusement, on ne connaît actuellement aucun algorithme polynomial (en n et s) capable de déterminer ou même de majorer[1] de manière satisfaisante $D_n^*(x)$. Les principaux résultats classiques sur les techniques de majoration et de calcul de la discrépance font l'objet des chapitres 5 et 6, alors que nos propres contributions à ces questions sont présentées dans les chapitres 7 et 8.

2° Le calcul de la variation au sens de Hardy et Krause : la définition de cette grandeur est énoncée ci-dessous, mais la question de sa détermination (ou de sa majoration) n'est pas étudiée dans ce travail. Le sujet est néanmoins discuté plus longuement dans la section 3.2.

---

[1] Il est tout à fait envisageable d'exploiter l'inégalité de Koksma-Hlawka à l'aide d'une borne supérieure sur la discrépance.

La définition de la variation au sens de Hardy et Krause $V(f)$ d'une fonction $f : \bar{I}^s \to \mathbb{R}$ requiert l'introduction de quelques notions et notations préliminaires. Ainsi, à tout ensemble de s séquences (de nombres réels) de la forme

$$0 = z_j^0 < z_j^1 < \cdots < z_j^{n_j} = 1, \text{ pour tout } j \in \{1, \ldots, s\},$$

on associe la partition $\mathcal{P}$ (en intervalles) du cube unité I^s donnée par

$$\mathcal{P} = \left\{ P = \prod_{j=1}^{s} \left[z_j^{i_j}, z_j^{1+i_j} \right) : 0 \le i_j < n_j, \, \forall j \in \{1, \ldots, s\} \right\}.$$

De même, à tout intervalle

$$P = \prod_{j=1}^{s} [p_j^0, p_j^1) \subset I^s,$$

on fait correspondre la somme alternée

$$\Delta(f, P) = \sum_{e_1=0}^{1} \cdots \sum_{e_s=0}^{1} (-1)^{e_1 + \cdots + e_s} \, f\left(p_1^{e_1}, \ldots, p_s^{e_s}\right).$$

Définition 3.2 Pour toute fonction $f : \bar{I}^s \to \mathbb{R}$, sa *variation s-dimensionnelle au sens de Vitali* $V^{(s)}(f)$ est donnée par

$$V^{(s)}(f) = \sup_{\mathcal{P}} \sum_{P \in \mathcal{P}} \left| \Delta(f, P) \right|,$$

où le supremum est pris sur toutes les partitions $\mathcal{P}$ de I^s possibles. Si la grandeur $V^{(s)}(f)$ est finie, alors f est dite *à variation bornée au sens de Vitali*.

Cependant, pour qu'une telle fonction soit à variation bornée au sens de Hardy et Krause, il faut non seulement qu'elle soit à variation bornée au sens de Vitali, mais également que certaines de ses restrictions à différentes faces de $\bar{I}^s$ le soient aussi. Ainsi, pour tout $j \in \{1, \ldots, s\}$ et tout ensemble de composantes $1 \le i_1 < \cdots < i_j \le s$, on note $V^{(j)}(f, i_1, \ldots, i_j)$ la variation j-dimensionnelle au sens de Vitali de la restriction de f à $I^s_{i_1, \ldots, i_j} = \{t \in \bar{I}^s : t_d = 1, \, \forall d \notin \{i_1, \ldots, i_j\}\}$.

Définition 3.3 Pour toute fonction $f : \bar{I}^s \to \mathbb{R}$, sa *variation au sens de Hardy et Krause $V(f)$* est donnée par

$$V(f) = \sum_{j=1}^{s} \sum_{1 \le i_1 < \cdots < i_j \le s} V^{(j)}(f, i_1, \ldots, i_j).$$

Si cette quantité est finie, alors f est dite *à variation bornée au sens de Hardy et Krause*.

Remarque 3.4 Lorsque la dérivée partielle $\frac{\partial^s f}{\partial t_1 \cdots \partial t_s}$ est continue sur $\bar{I}^s$, on peut utiliser la définition suivante pour la variation s-dimensionnelle au sens de Vitali :

$$V^{(s)}(f) = \int_0^1 \cdots \int_0^1 \left| \frac{\partial^s f}{\partial t_1 \cdots \partial t_s} \right| dt_1 \cdots dt_s.$$

Remarque 3.5 La borne sur l'erreur donnée par l'inégalité de Koksma-Hlawka est essentiellement la meilleure possible. En effet, pour toute séquence $x = \{x^1, \ldots, x^n\} \subset I^s$ et tout $\varepsilon > 0$, il existe une fonction $f \in C^\infty(\bar{I}^s)$ (*i.e.* indéfiniment continûment différentiable sur $\bar{I}^s$) avec $V(f) = 1$ telle que

$$\left| \frac{1}{n} \sum_{i=1}^{n} f(x^i) - \int_{\bar{I}^s} f(t) \, dt \right| > D_n^*(x) - \varepsilon.$$

Ce résultat se montre facilement (voir Niederreiter [Nie92]) pour une fonction f définie comme suit : par définition, il existe un intervalle $P^+ = [0, \beta)$ tel que

$$\left| \frac{A(P^+, x)}{n} - \lambda(P^+) \right| > D_n^*(x) - \frac{\varepsilon}{2}.$$

De plus, il est clair qu'il existe un intervalle $P^- = [0, \alpha)$ avec $0 \leq \alpha_j < \beta_j$ et $(\beta_j - \alpha_j) < \varepsilon/(2s)$ pour tout $j \in \{1, \ldots, s\}$ tel que $P^+ \setminus P^-$ ne contient aucun point de x. Pour tout $j \in \{1, \ldots, s\}$, soit $f_j : \bar{I} \to \bar{I}$ une fonction non croissante indéfiniment continûment différentiable sur$\bar{I}$ telle que

$$f_j(t_j) = \begin{cases} 1 & \text{pour } t_j \in [0, \alpha_j], \\ 0 & \text{pour } t_j \in [\beta_j, 1]. \end{cases}$$

On peut en déduire que la fonction $f(t) = \prod_{j=1}^{s} f_j(t_j)$ possède la propriété énoncée.

Dans le cas d'une fonction possédant de fortes propriétés de différentiabilité et de continuité, notons que d'autres inégalités que celle de Koksma-Hlawka permettent de borner l'erreur (18) par le produit de deux grandeurs, l'une ne dépendant que de la fonction et l'autre que de la séquence (voir Zaremba [Zar68b], Hickernell [Hic98] et Sloan et Woźniakowski [SW98]). En général, ces bornes ne sont pas plus faciles à calculer que celle de Koksma-Hlawka. D'autre part, pour des domaines d'intégration autres que le cube unité, Zaremba [Zar70] et De Clerck [Cle81] ont obtenu des inégalités similaires, mais faisant intervenir la discrépance isotrope (définition 1.7) de la séquence considérée.

Dans un registre différent, Proinov a établi la borne suivante pour les fonctions continues :

THÉORÈME 3.6 (Proinov [Pro88]) *Si $f : \bar{I}^s \to \mathbb{R}$ est une fonction continue sur $\bar{I}^s$, alors pour toute séquence de points $x = \{x^1, \ldots, x^n\} \subset I^s$, on a*

$$\left| \frac{1}{n} \sum_{i=1}^{n} f(x^i) - \int_{\bar{I}^s} f(t)\, dt \right| \leq 4\, \omega(f, D_n^*(x)^{1/s}),$$

où, pour tout $\delta \geq 0$, $\omega(f, \delta)$ désigne le module de continuité

$$\omega(f, \delta) = \sup_{\substack{y,z \in \bar{I}^s \\ \max_{j \in \{1,\ldots,s\}} |y_j - z_j| \leq \delta}} |f(y) - f(z)| \,.$$

La valeur « 4 » qui apparaît dans cette expression n'est peut-être pas la meilleure possible. Proinov a néanmoins démontré que la constante optimale est nécessairement supérieure à 1. Ce théorème est très intéressant d'un point de vue théorique, mais vraisemblablement inutilisable en pratique (du moins en dimension élevée).

Les inégalités de Koksma-Hlawka et de Proinov font intervenir la discrépance dans le calcul d'une borne dans le pire des cas. Cependant, comme l'illustre le résultat suivant, il est également possible de considérer le problème de la convergence de l'erreur (18) dans le cas moyen.

THÉORÈME 3.7 (Woźniakowski [Woź91]) *Si l'on note $T_n^*(1 - x)$ la discrépance carrée moyenne* (5) *de la séquence symétrique de $x = \{x^1, \ldots, x^n\} \subset \bar{I}^s$ par rapport au centre du cube unité (i.e. chaque point x^i est remplacé par son symétrique $1 - x^i$), alors on obtient le résultat suivant pour l'espérance du carré de l'erreur commise dans l'approximation* (17) *sur les processus de Wiener définis sur $\bar{I}^s$:*

$$E\left[\left(\frac{1}{n} \sum_{i=1}^{n} f(x^i) - \int_{\bar{I}^s} f(t)\, dt \right)^2 \right] = \left(T_n^*(1 - x)\right)^2.$$

Ainsi, dans cette application particulière (l'intégration sur $\bar{I}^s$ d'une fonction f issue d'un processus de Wiener[2]), la moyenne du carré de l'erreur d'approximation (18) est égale au carré de la discrépance carrée moyenne du symétrique de la séquence utilisée. Ce résultat caractérise l'erreur moyenne pour une séquence x fixée et pour une fonction f distribuée aléatoirement (conformément à la mesure associée au processus). Il est clair que pour l'approximation de quasi-Monte-Carlo correspondante, il est préférable d'utiliser une séquence déterministe dont la discrépance carrée moyenne est en $\Theta(n^{-1}(\log n)^{(s-1)/2})$ (voir remarque 1.22), plutôt qu'une séquence de variables aléatoires i.i.d. $U(\bar{I}^s)$ dont on sait qu'elle conduit à une convergence bien moins rapide (voir section 1.4.1).

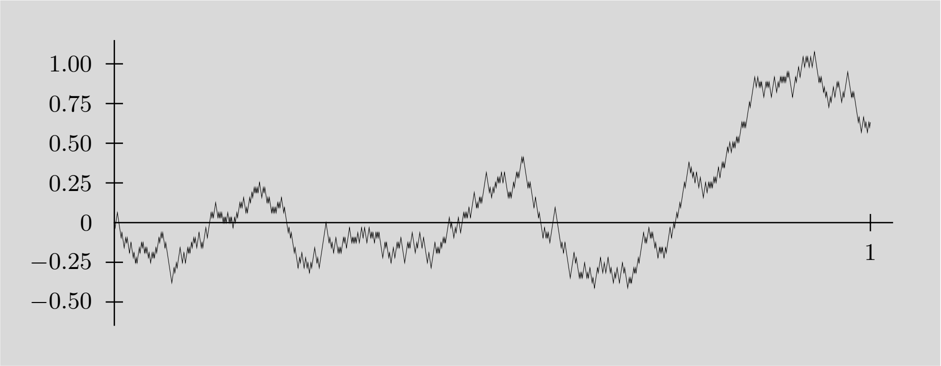


FIG. 3.2. Discrétisation (sur $1\,000$ points) d'une réalisation d'un processus de Wiener en dimension 1.

Une réalisation d'un processus de Wiener est une fonction continue, mais qui n'est différentiable nulle part. D'autre part, la mesure de Wiener est concentrée sur des fonctions dont la variation au sens de Hardy et Krause n'est pas bornée (voir Morokoff et Caflisch [MC94]). Néanmoins, le théorème de Woźniakowski stipule qu'en moyenne l'intégration d'une telle fonction est facile à appréhender par la méthode de quasi-Monte-Carlo (17). Malheureusement, bien que $T_n^*(x) \leq D_n^*(x)$, l'inégalité de Koksma-Hlawka ne permet pas d'aborder l'analyse de l'erreur d'approximation (18) dans ce cas. Ces considérations illustrent le fait qu'en général le problème de l'estimation d'une erreur dans le pire des cas est intrinsèquement plus difficile que dans le cas moyen (voir Traub et Woźniakowski [TW94]).

3.2 Discussion

Commençons cette discussion par une critique de la notion de variation au sens de Hardy et Krause en tant que mesure de l'irrégularité d'une fonction. Cette grandeur a comme défaut rédhibitoire de ne pas être invariante sous des transformations élémentaires comme la symétrie. Par exemple, pour

$$f(t) = \prod_{j=1}^{s}(1-t_j) \quad \text{et} \quad g(t) = \prod_{j=1}^{s} t_j,$$

on a respectivement $V(f) = 1$ et $V(g) = 2^s - 1$, alors que ces deux fonctions sont essentiellement les mêmes (elles ont notamment la même intégrale sur $\bar{I}^s$). Bien qu'il s'agisse d'un cas extrême, cet exemple suggère qu'au-delà du fait d'être ou non à variation bornée au sens de Hardy et Krause, ce concept ne donne aucune assurance que d'éventuelles propriétés supplémentaires de régularité de la fonction soient reflétées. Cette non-robustesse de la notion de variation au sens de Hardy et Krause montre bien qu'en

---

[2] Un processus de Wiener en dimension s (voir figure 3.2) est une fonction aléatoire gaussienne de s variables, de moyenne nulle et de noyau de covariance $K(y,z) = \prod_{j=1}^{s} \min\{y_j, z_j\}$. Nous renvoyons le lecteur à l'ouvrage de Bouleau et Lépingle [BL94] pour une présentation plus détaillée.

général l'inégalité de Koksma-Hlawka fournit une mesure peu représentative (le plus souvent une large surestimation) de l'erreur effective (18). En revanche, comme l'illustre la remarque 3.5, la borne obtenue est la meilleure possible dans le pire des cas.

Précisons également que des fonctions non continues très simples telles que

$$f(t_1, t_2) = \begin{cases} 1 & \text{pour } t_2 \leq t_1, \\ 0 & \text{sinon}, \end{cases}$$

ne sont pas à variation bornée au sens de Hardy et Krause. Il en est d'ailleurs de même pour la fonction caractéristique de n'importe quelle région non rectangulaire du cube unité $\bar{I}^s$. Malgré tout, bien que l'inégalité de Koksma-Hlawka ne s'applique pas pour une telle fonction, la méthode de quasi-Monte-Carlo converge facilement, même dans ces cas-là.

Plus généralement, une condition nécessaire pour qu'une fonction soit à variation bornée au sens de Hardy et Krause est que ses éventuelles discontinuités soient concentrées sur un ensemble dénombrable d'hyperplans orthogonaux aux axes de coordonnées. En ce sens, le concept est donc assez restrictif. Par contre, il est à noter que, bien que le théorème de Koksma-Hlawka ne permette pas de borner l'erreur (18) pour certaines fonctions simples présentant des discontinuités, les formules de quadrature classiques (comme celle du trapèze) ne s'appliquent que pour des fonctions plusieurs fois continûment différentiables. De ce point de vue, la méthode de quasi-Monte-Carlo représente un progrès. En fin de compte, l'approche la plus permissive au niveau des discontinuités reste encore celle de Monte-Carlo, car elle est utilisable pour toute fonction intégrable au sens de Riemann dont la variance (11) est finie.

Dans la plupart des cas, le calcul de la variation au sens de Hardy et Krause est une tâche très difficile. Néanmoins, on connaît des méthodes permettant parfois d'obtenir des majorations (voir Hua et Wang [HW81]). D'autre part, de manière analogue aux techniques de réduction de variance évoquées au chapitre 2, il est envisageable de reformuler le problème de manière à réduire la variation de la fonction (voir Spanier et Maize [SM94] et Tuffin [Tuf97]). Finalement, mentionnons le fait que pour certaines applications, la variation au sens de Hardy et Krause peut être majorée à l'aide de techniques ad hoc (voir par exemple Lécot et Coulibaly [LC98]).

D'un point de vue pratique, en se basant sur de nombreuses expériences numériques, Morokoff et Caflisch [MC95] sont arrivés à la conclusion que ni la variation au sens de Hardy et Krause, ni la variance (11) d'une fonction ne semblent intervenir dans la valeur réelle de l'erreur d'approximation (18). Leur analyse est fondée sur des simulations de quasi-Monte-Carlo effectuées sur un ensemble hétérogène de fonctions-tests maîtrisables analytiquement (*i.e.* pour lesquelles il est possible de calculer ou de borner de manière suffisamment précise l'intégrale, la variation et la variance). Leurs expériences suggèrent également qu'il est nécessaire de considérer des échantillons de taille croissante avec la dimension du problème pour que l'utilisation des séquences déterministes à discrépance faible du chapitre 4 se révèle clairement préférable à l'emploi d'un générateur pseudo-aléatoire.

En théorie, la méthode de quasi-Monte-Carlo possède deux atouts majeurs : une convergence asymptotiquement plus rapide que la méthode de Monte-Carlo et la possibilité de calculer des bornes déterministes sur l'erreur (18). Malheureusement, en l'état actuel de nos connaissances, l'exploitation de la seconde propriété est encore un vœu pieux, car la discrépance et la variation au sens de Hardy et Krause sont très difficiles à calculer. Par contre, sa rapidité de convergence a d'ores et déjà séduit bon nombre d'anciens adeptes de la méthode de Monte-Carlo.

Jusqu'ici les applications sont essentiellement apparues en finance[3] (voir Joy, Boyle et Tan [JBT96], Paskov [Pas97] et Tezuka [Tez98]), en physique[4] (voir Morokoff et Caflisch [MC93], Moskowitz [Mos95] et Lécot et Coulibaly [LC98]), en statistique[5] (voir Shaw [Sha88] et Do [Do91]) et en recherche opérationnelle[6] (voir Fishman [Fis85] et Adlakha [Adl87][Adl92]). Ces diverses expériences illustrent le fait qu'une simulation de quasi-Monte-Carlo peut converger nettement plus rapidement[7] que la méthode de Monte-Carlo correspondante. D'ailleurs, la quête d'une explication théorique satisfaisante à certains succès particulièrement étonnants et, plus généralement, la compréhension du comportement empirique de la méthode occupe de nombreux chercheurs.

Par exemple, on a du mal à expliquer le fait que Paskov [Pas97] obtienne d'excellents résultats pour le calcul d'intégrales en dimension $s = 360$, alors que certains résultats théoriques indiquent qu'en dimension $s > 20$, à moins d'utiliser une séquence de très grande taille, il est peu probable que la méthode de quasi-Monte-Carlo se montre réellement plus performante que son homologue de Monte-Carlo. Afin de mieux comprendre de tels phénomènes, Sloan et Woźniakowski [SW98] (par exemple) suggèrent des concepts alternatifs comme celui de dimension effective (typiquement, on pense que sur les 360 dimensions des intégrales considérées par Paskov, une trentaine de variables suffiraient à décrire convenablement la structure de la fonction) ou d'autres grandeurs que la variation au sens de Hardy et Krause pour mesurer l'irrégularité de la fonction à intégrer. De telles approches sont actuellement en plein développement et on peut espérer qu'un résultat théorique fort permettant d'établir solidement la méthode de quasi-Monte-Carlo ne tarde pas à émerger.

Parallèlement, s'appuyant sur certains acquis de la méthode de quasi-Monte-Carlo, de nouvelles techniques de Monte-Carlo ont vu le jour. L'idée de base de ces approches hybrides est d'appliquer une perturbation aléatoire à une suite à discrépance faible, de manière à pouvoir à la fois calculer des bornes probabilistes sur l'erreur (18) et bénéficier d'un taux de convergence supérieur à celui de la méthode de Monte-Carlo classique. Plus précisément, si la séquence utilisée est un (t, m, s)-réseau en base b (*i.e.* un ensemble de $n = b^m$ points dans I^s possédant certaines propriétés d'équirépartition présentées dans la section 4.6) auquel on a appliqué une transformation aléatoire (préservant sa structure) proposée par Owen [Owe97a][Owe97b][Owe98], on obtient un estimateur de Monte-Carlo M_n (voir page 14) dont la variance (12) décroît suivant $O(n^{-2}(\log n)^{2(s-1)})$ si la fonction à intégrer est à variation bornée au sens de Hardy et Krause, et même suivant $O(n^{-3}(\log n)^{(s-1)})$ si elle satisfait en plus une certaine condition lipschitzienne de régularité. Lorsqu'on les compare à la variance en $O(n^{-1})$ d'une simulation de Monte-Carlo standard ces taux sont impressionnants mais, en dehors des hypothèses fortes faites sur la fonction, le nombre de points nécessaires avant d'atteindre le régime asymptotique peut être particulièrement grand, spécialement en dimension élevée (un (t, m, s)-réseau en base b n'existe pas pour n'importe quelles valeurs de m et b). En vue des applications, une approche pragmatique susceptible de réduire la dimension effective du problème consiste à répartir les différentes composantes en deux groupes : les variables importantes (qui sont traitées à l'aide d'une méthode hybride) et secondaires (pour lesquelles une méthode de Monte-Carlo standard est utilisée). Cette technique est détaillée dans un récent ouvrage de Fox [Fox99] (voir également Caflisch, Morokoff et Owen [CMO97] et Tuffin [Tuf97]).

En conclusion, la méthode de quasi-Monte-Carlo a sur celle de Monte-Carlo certains avantages :

---

[3] Il s'agit de problèmes d'évaluation de produits dérivés habituellement abordés à l'aide de la méthode de Monte-Carlo.

[4] Depuis fort longtemps, la simulation de certains processus physiques (généralement donnés sous forme de solution d'une équation différentielle) est abordée à l'aide de la méthode de Monte-Carlo. Pour de telles applications, le passage à une approche purement déterministe est moins évident, mais la convergence du processus a pu être démontrée dans de nombreux cas.

[5] En statistique bayesienne, la distribution a posteriori peut parfois se ramener au calcul d'une intégrale sur le cube unité.

[6] Typiquement, il s'agit du calcul de l'espérance ou de l'estimation de la fonction de distribution d'une grandeur (un plus court chemin par exemple) associée à un graphe orienté dont les arcs ont pour longueurs des variables aléatoires indépendantes.

[7] Sur certains problèmes d'évaluation d'options européennes, Tezuka [Tez98] parle d'un facteur dépassant la centaine.

- ▷ Elle converge plus rapidement. D'un point de vue théorique, ce phénomène est apparent au niveau des discrépances asymptotiques respectives des séquences en question. En pratique, ce fait est confirmé dans les diverses applications mentionnées plus haut.
- ▷ Elle fournit des bornes déterministes sur l'erreur et non des intervalles de confiance probabilistes. Hélas, au vu de nos connaissances actuelles, cette propriété prometteuse paraît inexploitable. C'est néanmoins à ce niveau que réside tout le potentiel[8] de la méthode.
- ▷ Au niveau des séquences de points à utiliser, la remarque 3.5 et les théorèmes 3.1, 3.6 et 3.7 désignent un but précis pour la méthode de quasi-Monte-Carlo : minimiser la discrépance afin de réduire l'erreur dans le pire des cas (de plus, la génération de bonnes suites à discrépance faible est un sujet relativement bien maîtrisé). Cet objectif a le mérite d'être limpide et permet d'échapper à toute la problématique liée à la génération de nombres aléatoires nécessaire à la simulation de Monte-Carlo (voir chapitre 2).

Dans les applications où une borne probabiliste sur l'erreur est suffisante, il serait surprenant qu'à terme, les méthodes hybrides (lorsqu'elles s'appliquent) ne finissent pas par détrôner la simulation de Monte-Carlo standard. Par contre, dans les situations plus exigeantes où une borne déterministe sur l'erreur est requise, l'approche de quasi-Monte-Carlo reste la seule envisageable. Pour l'instant, son applicabilité reste suspendue aux progrès qui restent à accomplir à différents niveaux :

- ▷ l'établissement de nouveaux résultats théoriques permettant de majorer plus facilement l'erreur d'approximation (18) ;
- ▷ l'élaboration de techniques efficaces pour le calcul de bornes supérieures pour la variation au sens de Hardy et Krause (ou une simplification de cette notion utilisable dans le théorème de Koksma-Hlawka) ;
- ▷ des algorithmes pour le calcul ou la majoration de la discrépance des séquences utilisées.

---

[8] Bien que le théorème de Koksma-Hlawka soit relativement ancien, la question de son applicabilité est une préoccupation très récente. En fait, le développement de la méthode de quasi-Monte-Carlo est resté pendant longtemps confiné à un cercle de spécialistes qui se sont essentiellement intéressés à ses aspects théoriques. Ce n'est que depuis une petite dizaine d'années (et la considération de certaines applications en finance) que le domaine s'est réellement ouvert aux praticiens et à leurs attentes.

CHAPITRE 4

Quelques suites à discrépance faible

Nous savons (voir théorème 1.19) qu'il existe une constante B_s' ne dépendant que de s telle que, pour toute suite $x \subset \bar{I}^s$, $D_n^*(x) \geq B_s' n^{-1} (\log n)^{s/2}$ pour une infinité de valeurs de n. Malheureusement, on ne connaît pas de suite pour laquelle $D_n^*(x) = O(n^{-1}(\log n)^{s/2})$ et il est communément admis qu'il n'en existe vraisemblablement aucune. En revanche, on sait construire des suites, que l'on appelle *suites à discrépance faible*, pour lesquelles $D_n^*(x) = O(n^{-1}(\log n)^s)$. On conjecture que $O(n^{-1}(\log n)^s)$ est l'ordre exact de leur discrépance et qu'il n'existe aucune suite présentant une décroissance plus rapide. Ce chapitre se limite à la présentation de quelques suites à discrépance faible n'exigeant que peu de connaissances spécialisées. Il en existe d'autres, parfois jugées meilleures sur certains critères, mais leur définition reposant sur des disciplines avancées (théorie des nombres, théorie des corps finis, géométrie algébrique, etc.), leur étude n'est pas abordée dans ce travail. Toutefois, quelques-unes de leurs caractéristiques importantes sont discutées dans les sections 4.6 et 4.7.

En vertu du théorème 1.10, les suites à discrépance faible sont équiréparties (au sens de la définition 1.3). Elles ont la particularité de remplir le cube unité de manière extrêmement régulière. Précisons qu'elles le font si bien qu'elles échouent lamentablement à la plupart des tests destinés à l'évaluation des générateurs pseudo-aléatoires (voir section 2.2.3) : typiquement, elles sont jugées suspectes, car anormalement bien distribuées par un test d'uniformité et fortement corrélées par un test d'indépendance. D'autre part, il est clair que ces suites ne satisfont pas non plus la définition d'« aléatoire » discutée dans la section 2.2.1. Ainsi, dans le cadre général de la simulation, elles ne sont pas destinées aux applications dans lesquelles l'indépendance entre les points est importante. En revanche, elles s'avèrent très performantes dans le cadre de la méthode de quasi-Monte-Carlo (voir chapitre 3).

4.1 Les suites de van der Corput

Les suites de van der Corput sont des suites à discrépance faible dans l'intervalle unité $I = [0, 1)$. Tout entier $b \geq 2$ (un tel nombre est appelé une *base*) peut être utilisé pour représenter n'importe quel $i \in \mathbb{N}$ de manière unique à l'aide d'une suite de coefficients de $\mathbb{Z}_b = \{0, 1, \ldots, b-1\}$:

$$i = \sum_{k=1}^{\infty} c_k(i)\, b^{k-1}, \text{ avec } c_k(i) \in \mathbb{Z}_b \text{ pour tout } k \in \mathbb{N}^*.$$

On remarque que les coefficients $c_k(i)$ sont nuls pour tout $k > 1 + \lfloor \log_b i \rfloor$), si bien que la somme ci-dessus est en fait finie. On se sert de cette représentation pour définir la *fonction radicale inverse* ϕ_b :

$$\phi_b(i) = \sum_{k=1}^{\infty} c_k(i)\, b^{-k}, \text{ pour tout } i \in \mathbb{N}. \tag{19}$$

On s'aperçoit facilement que $\phi_b(i) \in [0, 1)$ pour tout $i \in \mathbb{N}$.

DÉFINITION 4.1 (van der Corput [vdC35]) Soit un entier $b \geq 2$. La suite[1] $C_b = \{x^0, x^1, \ldots\} \subset I$ donnée par $x^i = \phi_b(i)$ est appelée *suite de van der Corput* en base b.

---

[1] Comme la notation s'en trouve simplifiée, les points sont numérotés à partir de 0 au lieu de 1 dans ce chapitre.

EXEMPLE 4.2 En base $b = 3$, les premiers points de la suite de van der Corput s'obtiennent de la manière suivante (voir figure 4.1) :

$$\begin{array}{llrll}
i = 0 & \text{s'écrit} & 0 & \text{en base } 3 & \Longrightarrow x^0 = \phi_3(0) = 0 \cdot 3^{-1} = 0, \\
i = 1 & \text{s'écrit} & 1 & \text{en base } 3 & \Longrightarrow x^1 = \phi_3(1) = 1 \cdot 3^{-1} = 1/3, \\
i = 2 & \text{s'écrit} & 2 & \text{en base } 3 & \Longrightarrow x^2 = \phi_3(2) = 2 \cdot 3^{-1} = 2/3, \\
i = 3 & \text{s'écrit} & 10 & \text{en base } 3 & \Longrightarrow x^3 = \phi_3(3) = 0 \cdot 3^{-1} + 1 \cdot 3^{-2} = 1/9, \\
i = 4 & \text{s'écrit} & 11 & \text{en base } 3 & \Longrightarrow x^4 = \phi_3(4) = 1 \cdot 3^{-1} + 1 \cdot 3^{-2} = 4/9, \\
i = 5 & \text{s'écrit} & 12 & \text{en base } 3 & \Longrightarrow x^5 = \phi_3(5) = 2 \cdot 3^{-1} + 1 \cdot 3^{-2} = 7/9, \\
i = 6 & \text{s'écrit} & 20 & \text{en base } 3 & \Longrightarrow x^6 = \phi_3(6) = 0 \cdot 3^{-1} + 2 \cdot 3^{-2} = 2/9.
\end{array}$$

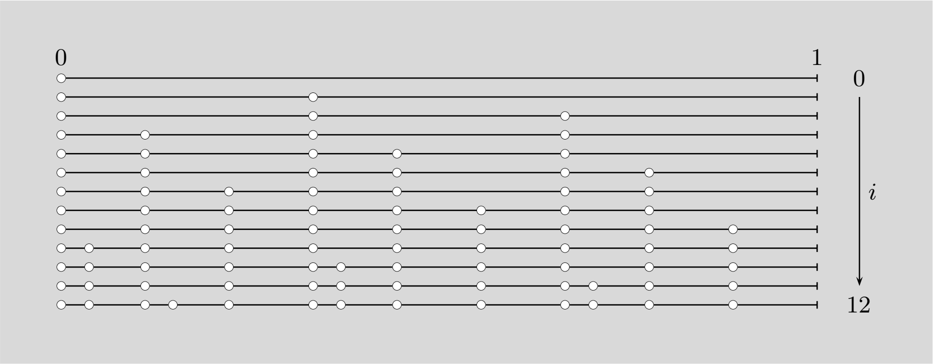


FIG. 4.1. Les 13 premiers points de la suite de van der Corput en base 3.

Certaines majorations et le fait que $D_n^*(C_b) = O(n^{-1} \log n)$ sont connus depuis longtemps, mais Faure a également obtenu des résultats asymptotiques précis :

THÉORÈME 4.3 (Faure [Fau81]) *Soit C_b la suite de van der Corput en base b. On a*

$$\limsup_{n \to \infty} \frac{nD_n^*(C_b)}{\log n} = \begin{cases} \dfrac{b^2}{4(b+1)\log b} & \textit{pour } b \textit{ pair,} \\ \dfrac{b-1}{4 \log b} & \textit{pour } b \textit{ impair.} \end{cases}$$

Ainsi, asymptotiquement, C_3 est la meilleure suite de van der Corput. En d'autres termes, pour la suite $x = C_3$ et pour tout intervalle $P \in \mathcal{I}^*$, on a

$$n\lambda(P) - \frac{1}{2\log 3} \log n \leq A(P, x(n)) \leq n\lambda(P) + \frac{1}{2\log 3} \log n,$$

sauf, éventuellement, pour un nombre fini de valeurs de n. Béjian et Faure ont également obtenu une majoration pour la discrépance des n premiers points de la suite de van der Corput en base 2.

THÉORÈME 4.4 (Béjian et Faure [BF77]) *Pour C_2, la suite de van der Corput en base 2, on a*

$$nD_n^*(C_2) \leq \frac{\log n}{3\log 2} + 1, \textit{ pour tout } n \geq 1.$$

En introduisant dans leur définition une permutation des éléments de $\mathbb{Z}_b$, il est possible d'obtenir des suites en dimension 1 présentant une discrépance asymptotique plus faible que celle des suites de van der Corput standard.

DÉFINITION 4.5 Pour une base b et une permutation σ de $\mathbb{Z}_b$, la suite $x = \{x^0, x^1, \dots\}$ définie par

$$x^i = \sum_{k=1}^{\infty} \sigma(c_k(i))\, b^{-k},$$

est appelée *suite de van der Corput généralisée* en base b.

La suite présentant la plus faible discrépance asymptotique connue en dimension 1 est de ce type. Elle est donnée dans le théorème suivant :

THÉORÈME 4.6 (Faure [Fau81]) *Si x est la suite de van der Corput généralisée en base $b = 12$ engendrée par la permutation*

$$\sigma_{12} = (0\ 5\ 9\ 3\ 7\ 1\ 10\ 4\ 8\ 2\ 6\ 11),$$

on obtient

$$\limsup_{n\to\infty} \frac{nD_n^*(x)}{\log n} = \frac{1919}{3454 \log 12} \approx 0.224.$$

Le fait que la constante obtenue soit deux fois plus petite que celle de la meilleure suite de van der Corput C_3 montre le potentiel que l'on peut espérer tirer de l'utilisation de permutations dans la construction de suites à discrépance faible.

4.2 Les suites de Halton

Les suites de Halton (voir figure 4.2) sont des généralisations en dimension quelconque $s \geq 1$ des suites de van der Corput. L'idée consiste à considérer la fonction radicale inverse (19) dans différentes bases simultanément. Halton [Hal60] a démontré que ses suites sont à discrépance faible.

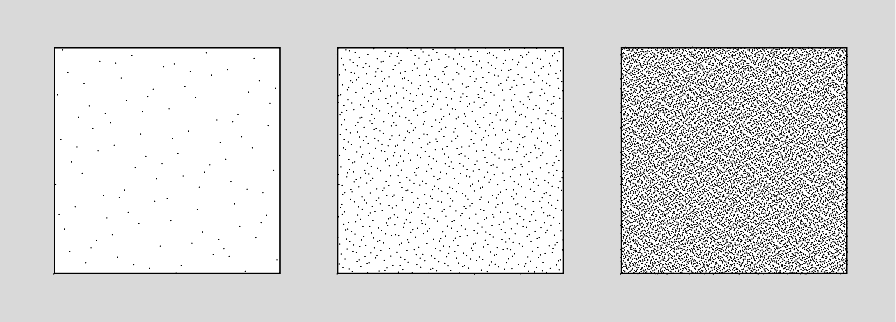


FIG. 4.2. Les 100, 1 000 et 10 000 premiers points d'une suite de Halton en bases 2 et 3.

DÉFINITION 4.7 (Halton [Hal60]) Soit $b_1, \dots, b_s$ des entiers positifs premiers entre eux deux à deux. La suite $H_{b_1,\dots,b_s} = \{x^0, x^1, \dots\}$ donnée par

$$x^i = (\phi_{b_1}(i), \dots, \phi_{b_s}(i)) \in I^s$$

est appelée *suite de Halton* en bases $b_1, \dots, b_s$.

Ainsi, pour tout $j \in \{1, \dots, s\}$, la suite constituée des composantes $\{x_j^0, x_j^1, \dots\}$ des points d'une suite de Halton $H_{b_1,\dots,b_s} = \{x^0, x^1, \dots\}$ n'est autre que la suite de van der Corput en base b_j. Le théorème suivant fournit la meilleure majoration connue sur la discrépance de ces suites.

THÉORÈME 4.8 (Faure [Fau80], repris dans [Nie92]) *Pour une suite de Halton* $H_{b_1,\ldots,b_s}$, *on a*

$$D_n^*(H_{b_1,\ldots,b_s}) \leq \frac{s}{n} + \frac{1}{n}\prod_{j=1}^{s}\left(\frac{b_j-1}{2\log b_j}\log n + \frac{b_j+1}{2}\right), \textit{ pour tout } n \in \mathbb{N}^*.$$

Il est clair que le fait de choisir comme bases $b_1, \ldots, b_s$ les s premiers nombres premiers $\{2, 3, 5, 7, \ldots\}$ permet de minimiser la constante

$$\prod_{j=1}^{s}\frac{b_j-1}{2\log b_j}$$

du terme dominant de la majoration ci-dessus. Hélas, même pour ce choix, la valeur de la constante tend vers l'infini avec la dimension.

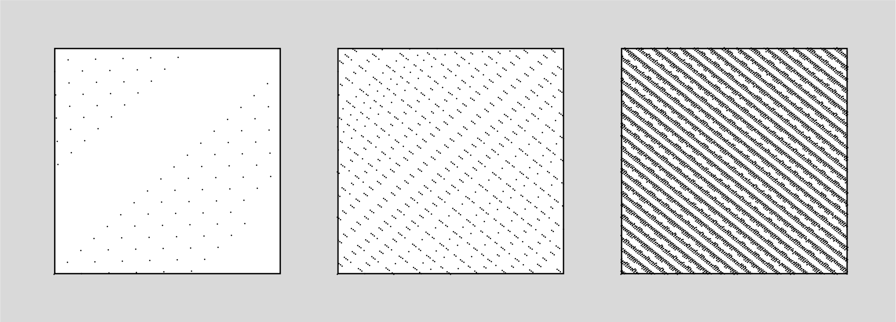


FIG. 4.3. Les 100, $1\,000$ et $10\,000$ premiers points d'une suite de Halton en bases 17 et 19.

Ainsi, en dimension $s = 8$, il convient de choisir les bases $2, 3, 5, 7, 11, 13, 17$ et 19. En examinant la projection des points de cette suite dans le plan correspondant aux bases 17 et 19 (voir figure 4.3), on ne peut être que déçu par l'allure de la structure obtenue. Ce comportement s'explique par le fait qu'une suite de van der Corput en base b consiste en une succession de cycles de longueur b constitués de séquences croissantes de nombres régulièrement espacés. Pour une suite de Halton en bases 17 et 19, cette régularité engendre un balayage du carré unité par une succession de diagonales progressivement décalées au fur et à mesure des itérations. Ainsi, mis à part pour le dernier point d'une telle diagonale, le successeur d'un point donné s'obtient par une translation de vecteur $(1/17, 1/19)$.

Il s'agit bien sûr d'une question d'échelle, mais visuellement (sur la figure 4.3), même après $10\,000$ itérations, un nombre restreint de régions représentant à peu près la moitié de la surface du carré unité ne contiennent encore aucun point de la suite. Après environ $25\,000$ itérations, le carré paraît entièrement recouvert, mais un comportement similaire (légèrement décalé) se reproduit pour les points suivants. D'autre part, en changeant d'ordre de grandeur (en considérant l'intervalle $[0, 1/17) \times [0, 1/19)$ par exemple), le même phénomène se déroule à plus petite échelle et beaucoup plus lentement. Le problème ne se manifeste pas pour n'importe quelle paire de bases (comme l'illustre la figure 4.2, tout se passe bien en bases 2 et 3). Néanmoins et d'une manière générale, il s'intensifie en dimension élevée. Nous renvoyons le lecteur à Morokoff et Caflisch [MC94], ainsi qu'à Kocis et Whiten [KW97], pour une discussion plus détaillée des pathologies relatives aux suites de Halton.

Une manière de conjurer l'apparition de ce malheureux phénomène consiste à généraliser les suites de Halton. Il s'agit d'une simple adaptation de la transformation déjà appliquée aux suites de van der Corput. Plus explicitement, les composantes des suites de Halton généralisées sont des suites de van der Corput généralisées engendrées par différentes permutations. Par exemple, l'utilisation des permutations

$$\sigma_{17} = (0\ 8\ 13\ 3\ 11\ 5\ 16\ 1\ 10\ 7\ 14\ 4\ 12\ 2\ 15\ 6\ 9)$$

et

$$\sigma_{19} = (0\ 9\ 14\ 3\ 17\ 6\ 11\ 1\ 15\ 7\ 12\ 4\ 18\ 8\ 2\ 16\ 10\ 5\ 13)$$

proposées par Braaten et Weller [BW79] pour les bases 17 et 19 semble conduire à des résultats plus satisfaisants au niveau de la distribution à l'intérieur du carré unité (voir figure 4.4).

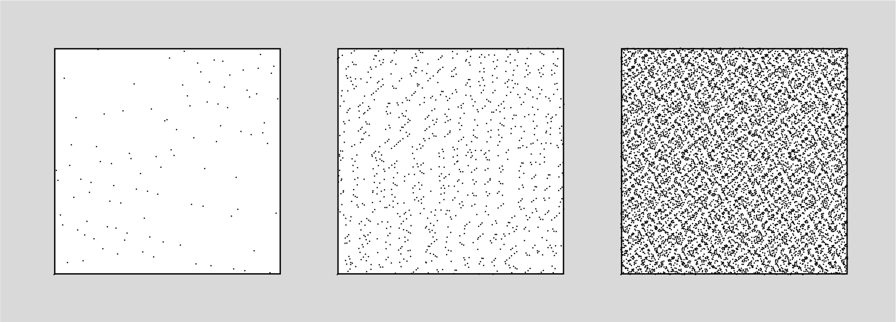


FIG. 4.4. Les 100, 1 000 et 10 000 premiers points d'une suite de Halton généralisée en bases 17 et 19 engendrée par les permutations σ_{17} et σ_{19} suggérées par Braaten et Weller.

Hormis le fait qu'elles restent des suites à discrépance faible, les caractéristiques spécifiques des suites de Halton généralisées sont encore mal connues. En particulier, on ne dispose pas d'une définition précise et communément admise des bénéfices qu'un bon ensemble de permutations devrait permettre d'obtenir et encore moins d'une méthode capable de générer un tel ensemble. Braaten et Weller [BW79] sont les premiers à avoir considéré des suites de Halton généralisées, mais, depuis, d'autres auteurs se sont intéressés à la question du choix des permutations (voir Faure [Fau86][Fau93] et Tuffin [Tuf98]).

4.3 Les séquences de Hammersley

Une séquence de Hammersley en dimension s est constituée d'un ensemble fini de points directement issus d'une suite de Halton en dimension $s - 1$.

DÉFINITION 4.9 (Hammersley [Ham60]) Soit $b_1, \ldots, b_{s-1}$ des entiers positifs premiers entre eux deux à deux et $n \in \mathbb{N}^*$. La séquence $H^n_{b_1,\ldots,b_{s-1}} = \{x^0, \ldots, x^{n-1}\}$ donnée par

$$x^i = \left(\frac{i}{n}, \phi_{b_1}(i), \ldots, \phi_{b_{s-1}}(i)\right) \in I^s,$$

est appelée *séquence de Hammersley* en bases $b_1, \ldots, b_{s-1}$.

THÉORÈME 4.10 (Faure [Fau80], repris dans [Nie92]) *Pour une séquence de Hammersley de n points $H^n_{b_1,\ldots,b_{s-1}}$, on a*

$$D^*_n(H^n_{b_1,\ldots,b_{s-1}}) \leq \frac{s}{n} + \frac{1}{n}\prod_{i=1}^{s-1}\left(\frac{b_i - 1}{2\log b_i}\log n + \frac{b_i + 1}{2}\right).$$

À nouveau, le fait de choisir comme bases $b_1, \ldots, b_{s-1}$ les $s - 1$ premiers nombres premiers permet de minimiser la constante du terme dominant de la majoration ci-dessus. Même pour ce choix, la valeur de la constante tend vers l'infini avec la dimension.

Le passage d'une suite de Halton à une séquence de Hammersley est un principe général. En utilisant la même technique, il est possible de passer, pour toute valeur de $n \in \mathbb{N}^*$, d'une suite à discrépance faible dans $\bar{I}^{s-1}$ à une séquence de n points dans $\bar{I}^s$ dont la discrépance est en $O(n^{-1}(\log n)^{s-1})$.

Bien que ce taux soit asymptotiquement meilleur, les séquences obtenues ont pour inconvénient de ne pas pouvoir être facilement étendues (par ajout de points supplémentaires) sans mettre en péril cette propriété. L'utilisation d'une séquence de Hammersley est donc à proscrire si le nombre de points à générer n'est pas connu à l'avance. D'autre part, pour les séquences de Hammersley en dimension $s = 2$, en base b et de longueur $n = b^m$, des expressions explicites ont été établies pour la discrépance.

THÉORÈME 4.11 (De Clerck [Cle86]) *Pour les séquences de Hammersley en dimension $s = 2$, en base b et de longueur b^m, on a :*

- ▷ *pour b impair et $m \geq 2$ entier*

$$D^*_{b^m}(H_b^{b^m}) = \frac{b-1}{4b^m}m + \frac{1}{b^m}\left(\frac{5}{4} + \frac{1}{b}\right) - \frac{1}{4b^{2m}};$$

- ▷ *pour b pair et $m \geq 2$ pair*

$$D^*_{b^m}(H_b^{b^m}) = \frac{b^2 m}{4b^m(b+1)} + \frac{1}{b^m}\left(\frac{5}{4} + \frac{2b+3}{4(b+1)^2}\right) - \frac{1}{4b^{2m}}\left(1 + \frac{2b+3}{(b+1)^2}\right);$$

- ▷ *pour b pair et $m \geq 3$ impair*

$$D^*_{b^m}(H_b^{b^m}) = \frac{b^2 m}{4b^m(b+1)} + \frac{1}{b^m}\left(\frac{5}{4} + \frac{5b+4}{4b(b+1)^2}\right) - \frac{1}{b^{2m}}\left(\frac{b}{2} - \frac{1}{4} - \frac{1}{b} + \frac{5}{4b^2} - \frac{6b+5}{4b^2(b+1)^2}\right).$$

Considérant le terme dominant de ces expressions, on voit clairement que $D_n^*(x) = \Theta(n^{-1}\log n)$ pour les séquences de Hammersley en dimension 2 (et n de la forme b^n). Cette propriété est apparente dans la table 4.1 et pouvait être directement déduite des théorèmes 1.17 et 4.10 (cette fois pour toute valeur de n).

	base 2		base 3		base 4		base 5	
m	$n = 2^m$	$D_n^*(H_2^n)$	$n = 3^m$	$D_n^*(H_3^n)$	$n = 4^m$	$D_n^*(H_4^n)$	$n = 5^m$	$D_n^*(H_5^n)$
2	4	0.5	9	0.284	16	0.184	25	0.138
3	8	0.316	27	0.114	64	0.0584	125	0.0356
4	16	0.172	81	0.0442	256	0.0178	625	0.00872
5	32	0.0979	243	0.0168	1 024	0.00519	3 125	0.00206
6	64	0.0537	729	0.00629	4 096	0.00150	15 625	0.000477
7	128	0.0296	2 187	0.00232	16 384	0.000422	78 125	0.000108
8	256	0.0161	6 561	0.000851	65 536	0.000118	390 625	0.0000242
9	512	0.00868	19 683	0.000309	262 144	0.0000325	1 953 125	$5.35 \cdot 10^{-6}$
10	1 024	0.00467	59 049	0.000111	1 048 576	$8.93 \cdot 10^{-6}$	9 765 625	$1.17 \cdot 10^{-6}$
11	2 048	0.00250	177 147	0.0000400	4 194 304	$2.41 \cdot 10^{-6}$	48 828 125	$2.55 \cdot 10^{-7}$
12	4 096	0.00133	531 441	0.0000143	16 777 216	$6.53 \cdot 10^{-7}$	244 140 625	$5.51 \cdot 10^{-8}$
13	8 192	0.000705	1 594 323	$5.07 \cdot 10^{-6}$	67 108 864	$1.74 \cdot 10^{-7}$	$1.22 \cdot 10^{9}$	$1.18 \cdot 10^{-8}$
14	16 384	0.000373	4 782 969	$1.79 \cdot 10^{-6}$	268 435 456	$4.68 \cdot 10^{-8}$	$6.10 \cdot 10^{9}$	$2.53 \cdot 10^{-9}$
15	32 768	0.000197	14 348 907	$6.33 \cdot 10^{-7}$	$1.07 \cdot 10^{9}$	$1.24 \cdot 10^{-8}$	$3.05 \cdot 10^{10}$	$5.39 \cdot 10^{-10}$
16	65 536	0.000103	43 046 721	$2.23 \cdot 10^{-7}$	$4.29 \cdot 10^{9}$	$3.30 \cdot 10^{-9}$	$1.53 \cdot 10^{11}$	$1.14 \cdot 10^{-10}$
17	131 072	0.0000543	129 140 163	$7.81 \cdot 10^{-8}$	$1.72 \cdot 10^{10}$	$8.68 \cdot 10^{-10}$	$7.63 \cdot 10^{11}$	$2.42 \cdot 10^{-11}$
18	262 144	0.0000284	387 420 489	$2.73 \cdot 10^{-8}$	$6.87 \cdot 10^{10}$	$2.29 \cdot 10^{-10}$	$3.81 \cdot 10^{12}$	$5.10 \cdot 10^{-12}$
19	524 288	0.0000148	$1.16 \cdot 10^{9}$	$9.54 \cdot 10^{-9}$	$2.75 \cdot 10^{11}$	$6.01 \cdot 10^{-11}$	$1.91 \cdot 10^{13}$	$1.07 \cdot 10^{-12}$
20	1 048 576	$7.74 \cdot 10^{-6}$	$3.49 \cdot 10^{9}$	$3.32 \cdot 10^{-9}$	$1.10 \cdot 10^{12}$	$1.58 \cdot 10^{-11}$	$9.54 \cdot 10^{13}$	$2.25 \cdot 10^{-13}$

TAB. 4.1. Discrépance de quelques séquences de Hammersley en dimension 2 pour $b \in \{2, \ldots, 5\}$.

4.4 Les suites de Sobol

Les suites de Sobol (voir figure 4.5) sont des suites à discrépance faible dans I^s. Elles sont définies à partir de récurrences linéaires en arithmétique modulo 2 sur le corps fini $\mathbb{Z}_2 = \{0, 1\}$ et de polynômes primitifs[2] sur $\mathbb{Z}_2 = \{0, 1\}$. Sans entrer dans les détails, un tel polynôme peut être mis en bijection avec un opérateur monocyclique binaire de période maximale. On ne connaît pas d'algorithme efficace capable de générer systématiquement ces polynômes. Cependant, pour leur utilisation dans la construction des suites de Sobol, les tables existant dans la littérature s'avèrent amplement suffisantes (voir Lidl et Niederreiter [LN86] par exemple). Les polynômes primitifs de degré inférieur ou égal à 5 sont

$$\begin{array}{ccc}
t+1, & t^4+t+1, & t^5+t^3+t^2+t+1, \\
t^2+t+1, & t^4+t^3+1, & t^5+t^4+t^2+t+1, \\
t^3+t+1, & t^5+t^2+1, & t^5+t^4+t^3+t+1, \\
t^3+t^2+1, & t^5+t^3+1, & t^5+t^4+t^3+t^2+1.
\end{array}$$

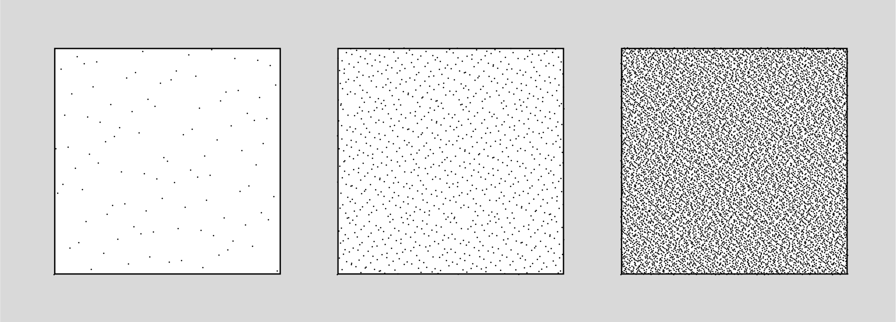


FIG. 4.5. Les 100, $1\,000$ et $10\,000$ premiers points d'une suite de Sobol engendrée par les polynômes primitifs $t+1$ (avec $l_1 = 1$) et t^2+t+1 (avec $l_1 = l_2 = 1$).

Partant d'un polynôme $t^d + u_1 t^{d-1} + \cdots + u_{d-1} t + 1$ primitif de degré d sur $\mathbb{Z}_2$ et d'un ensemble d'entiers impairs $\{l_1, \ldots, l_d\}$ tels que $1 \leq l_i < 2^i$ pour tout $i \in \{1, \ldots, d\}$, on définit $\{l_{d+1}, l_{d+2}, \ldots\}$ en utilisant la relation de récurrence suivante[3] :

$$l_i = 2\, u_1\, l_{i-1} \oplus 4\, u_2\, l_{i-2} \oplus \cdots \oplus 2^{d-1}\, u_{d-1}\, l_{i-d+1} \oplus (2^d\, l_{i-d} \oplus l_{i-d}).$$

Ces éléments suffisent à la construction d'une suite de Sobol en dimension 1.

DÉFINITION 4.12 (Sobol [Sob67]) Une suite de *Sobol* $S = \{x^0, x^1, \ldots\} \subset I$ est donnée par

$$x^i = \frac{1}{2^m}\left(\bigoplus_{k=1}^{m} c_k(i)\, l_k\right),$$

où $(c_1(i), \ldots, c_m(i))$ est la représentation binaire de i

$$i = \sum_{k=1}^{m} c_k(i)\, 2^{k-1}, \text{ avec } m = \begin{cases} 1 & \text{pour } i = 0, \\ 1 + \lfloor \log_2 i \rfloor & \text{sinon.} \end{cases}$$

Pour obtenir une suite de Sobol en dimension s, il suffit de choisir s polynômes primitifs distincts et de juxtaposer les suites correspondantes en dimension 1.

---

[2] Un polynôme de degré d de la forme $t^d + u_1 t^{d-1} + \cdots + u_{d-1} t + u_d$ est dit *primitif* sur le corps $\mathbb{Z}_2$ s'il est irréductible sur $\mathbb{Z}_2$ et si le plus petit entier i pour lequel il divise $t^i + 1$ est égal à $2^d - 1$.

[3] Le symbole $\oplus$ désigne l'opérateur « ou exclusif » en arithmétique binaire.

EXEMPLE 4.13 Soit t^3+t+1 un polynôme primitif de degré $d=3$. Les nombres $l_1,\ldots,l_d$ doivent être impairs et satisfaire la contrainte $l_i < 2^i$. On choisit par exemple $l_1 = 1, l_2 = 3$ et $l_3 = 7$. La relation de récurrence imposée sur les l_i est

$$l_i = 4l_{i-2} \oplus (8l_{i-3} \oplus l_{i-3}), \text{ pour tout } i > 3.$$

On obtient donc l_4 de la manière suivante :

$$\begin{aligned} l_4 &= 4l_2 \oplus 8l_1 \oplus l_1 \\ &= 12 \oplus 8 \oplus 1 \\ &= 1100 \oplus 1000 \oplus 0001 && \text{en binaire} \\ &= 0101 && \text{en binaire} \\ &= 5 \end{aligned}$$

Calculons par exemple le 13e $(1101$ en binaire$)$ point de cette suite de Sobol en dimension 1 :

$$\begin{aligned} x^{13} &= 1/16\,(l_1 \oplus l_3 \oplus l_4) \\ &= 1/16\,(0001 \oplus 0111 \oplus 0101) \\ &= 3/16 \end{aligned}$$

THÉORÈME 4.14 (Sobol [Sob67]) *Pour une suite de Sobol S en dimension s, on a*

$$D_n^*(S) \leq \frac{2^{t_s}}{s!(\log 2)^s}\, n^{-1}(\log n)^s + O\left(n^{-1}(\log n)^{s-1}\right),$$

où t_s ne peut être majoré par une fonction linéaire en la dimension :

$$k\,\frac{s\log s}{\log\log s} \leq t_s \leq \frac{s\log s}{\log 2} + O(s\log\log s), \textit{ avec } k > 0.$$

Les premières valeurs de t_s sont

s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
t_s	0	0	1	3	5	8	11	15	19	23	27	31	35	40	45	50	55	60	65	71

La constante du terme dominant de cette majoration croît nettement plus lentement avec s que pour une suite de Halton (voir théorème 4.8). Néanmoins, elle tend toujours vers l'infini avec la dimension. Par ailleurs, hormis le fait d'avoir proposé ses suites, Sobol [Sob67][Sob76] a effectué un travail de pionnier fructueux qui a conduit, après les contributions subséquentes de Faure et de Niederreiter, à l'émergence des notions de (t,s)-suite et de (t,m,s)-réseau (voir section 4.6).

Sobol a également remarqué que ses suites possèdent des propriétés d'uniformité supplémentaires lorsque $n = 2^m$ avec $m \geq \max(2s, t_s + s - 1)$. D'autre part, observant l'influence des valeurs initiales $\{l_1,\ldots,l_d\}$ (associées à un polynôme primitif de degré d) sur le segment initial de ses suites, il a proposé une table de valeurs assurant de bonnes propriétés jusqu'en dimension 16.

Une méthode de génération rapide des suites de Sobol a été proposée par Antonov et Saleev [AS79] (voir section 4.8). Elle a été implémentée par Bratley et Fox [BF88] pour $s \leq 40$.

4.5 Les suites de Faure

Les suites de Faure (voir figure 4.6) sont des suites à discrépance faible dans I^s. Elles sont définies à partir d'une base b unique, où $b \geq s$ est un nombre premier (on choisit généralement le plus petit nombre premier supérieur ou égal à s). Soit $A = (a_{pq})$, la matrice de Pascal (d'ordre infini) donnée par

$$a_{pq} = \binom{q-1}{p-1} = \begin{cases} \frac{(q-1)!}{(p-1)!\,(q-p)!} & \text{si } p \leq q, \\ 0 & \text{sinon,} \end{cases} \qquad \forall p,q \in \mathbb{N}^*,$$

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdots \\ 0 & 1 & 2 & 3 & 4 & 5 & \cdots \\ 0 & 0 & 1 & 3 & 6 & 10 & \cdots \\ 0 & 0 & 0 & 1 & 4 & 10 & \cdots \\ 0 & 0 & 0 & 0 & 1 & 5 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

On peut montrer que pour tout entier $d \geq 1$, la d^e puissance de A est la matrice $A^d = (a^d_{pq})$, où

$$a^d_{pq} = d^{q-p}\, a_{pq} \qquad \forall d, p, q \in \mathbb{N}^*.$$

On pose également $A^0 = I$, la matrice identité.

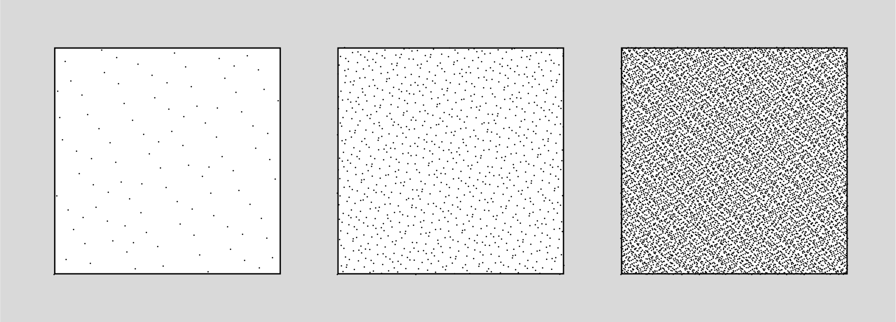


FIG. 4.6. Les 100, $1\,000$ et $10\,000$ premiers points d'une suite de Faure en dimension 2 et en base 3 donnée par $D = (1, 2)$.

DÉFINITION 4.15 (Faure [Fau82]) Soit $b \geq s$ un nombre premier et $D = (d_1, \ldots, d_s)$ un vecteur d'entiers distincts extraits de l'ensemble $\{0, \ldots, b-1\}$. La suite de Faure $F_{b,D} = \{x^0, x^1, \ldots\} \subset I^s$ engendrée par ces paramètres est définie comme suit :

$$x^i = (x^i_1, \ldots, x^i_s) \in I^s,$$

où

$$x^i_j = \sum_{p=1}^{\infty} x^i_{jp}\, b^{-p} \in [0, 1)$$

avec

$$x^i_{jp} = \sum_{q=p}^{\infty} a^{d_j}_{pq}\, c_q(i) \mod b \in \mathbb{Z}_b$$

et la représentation de i en base b

$$i = \sum_{k=1}^{\infty} c_k(i)\, b^{k-1}, \text{ avec } c_k(i) \in \mathbb{Z}_b \text{ pour tout } k \in \mathbb{N}^*.$$

REMARQUE 4.16 À nouveau, les coefficients $c_k(i)$ étant nuls pour tout $k > 1 + \lfloor \log_b i \rfloor$, les sommes ci-dessus sont en fait finies. Plus précisément, pour tout entier $m \geq 1$ et tout $i \in \{0, \ldots, b^m - 1\}$, on a $c_k(i) = 0$ pour tout $k > m$. Il en découle que $x^i_{jp} = 0$ pour tout $p > m$ et tout $j \in \{1, \ldots, s\}$. Ainsi, $x^i_j \in [0, 1)$ est un rationnel de la forme r/b^m. Pour tout $j \in \{1, \ldots, s\}$, la matrice $A^{d_j} \bmod b$ étant régulière, elle induit (par l'application décrite ci-dessus) pour tout $m \in \mathbb{N}^*$ une bijection entre l'ensemble des entiers $\{0, \ldots, b^m - 1\}$ et l'ensemble des rationnels de la forme $\{r/b^m : 0 \leq r < b^m\}$.

THÉORÈME 4.17 (Faure [Fau82]) *Pour une suite de Faure $F_{b,D}$ en dimension s, on a*

$$D_n^*(F_{b,D}) \le \frac{1}{s!}\left(\frac{b-1}{2\log b}\right)^s n^{-1}(\log n)^s + O\left(n^{-1}(\log n)^{s-1}\right), \textit{ pour } b \ge 3$$

et

$$D_n^*(F_{b,D}) \le \frac{3}{16\,(\log 2)^2}\, n^{-1}(\log n)^s + O\left(n^{-1}(\log n)^{s-1}\right), \textit{ pour } b = 2.$$

Il est clair que le fait de choisir comme base b le plus petit nombre premier supérieur ou égal à s permet de minimiser la constante du terme dominant de la majoration ci-dessus. Pour ce choix, cette constante tend vers 0 lorsque la dimension tend vers l'infini. Parmi les suites possédant cette propriété, les suites de Faure sont les plus anciennes. Une implémentation de ces suites a été proposée par Fox [Fox86] (pour $s \le 40$), mais une méthode nettement plus rapide est présentée dans la section 4.8.

4.6 Les (t,s)-suites et les (t,m,s)-réseaux

Un (t,m,s)-réseau en base b est un ensemble de b^m points dans le cube unité I^s, pour lequel la discrépance locale (4) est nulle pour une certaine famille d'intervalles de $\mathcal{I}_s$. Une (t,s)-suite en base b est une suite dont certains segments de longueur b^m (avec $m \ge t$) sont des (t,m,s)-réseaux en base b. Ces concepts ont été introduits de manière générale par Niederreiter [Nie87], bien qu'ils aient été préalablement posés par Sobol [Sob67] dans le cas particulier de la base $b = 2$ et par Faure [Fau82] pour $t = 0$ (voir la figure 4.7 pour une illustration de la notion de $(0,m,s)$-réseau en base b). Comme précédemment, dans les définitions suivantes, la dimension s et la base b sont des entiers fixés.

DÉFINITION 4.18 Un intervalle *élémentaire* en base b est un intervalle de la forme

$$\prod_{j=1}^{s}\left[\frac{a_j}{b^{d_j}}, \frac{a_j+1}{b^{d_j}}\right), \text{ où } a_j, d_j \in \mathbb{N} \text{ et } a_j < b^{d_j} \text{ pour tout } j \in \{1, \dots, s\}.$$

DÉFINITION 4.19 Soit une paire d'entiers $0 \le t \le m$. Un (t,m,s)-*réseau* en base b est une séquence x de b^m points de I^s telle que $A(P,x) = b^t$ pour tout intervalle élémentaire P en base b de volume $\lambda(P) = b^{t-m}$.

DÉFINITION 4.20 Soit un entier $t \ge 0$. Une (t,s)-*suite* en base b est une suite de points $\{x^0, x^1, \dots\}$ telle que pour toute paire d'entiers $k \ge 0$ et $m \ge t$, la séquence $\left\{x^{kb^m}, \dots, x^{(k+1)b^m-1}\right\}$ est un (t,m,s)-réseau en base b.

D'un point de vue pratique, la notion de réseau est importante, car elle fournit des garanties d'équirépartition pour un nombre fini de points. Clairement, pour m, s et b fixés, plus la valeur de t est petite, meilleures sont les propriétés d'un (t,m,s)-réseau en base b. Ce fait est confirmé par le théorème suivant. Ce dernier montre également que toute (t,s)-suite en base b est à discrépance faible :

THÉORÈME 4.21 (Niederreiter [Nie87]) *Pour une (t,s)-suite x en base b, on a*

$$D_n^*(x) \le C_{b,s,t}\, n^{-1}(\log n)^s + O\left(n^{-1}(\log n)^{s-1}\right)$$

où la constante du terme dominant est

$$C_{b,s,t} = \begin{cases} \dfrac{b^t}{s}\left(\dfrac{b-1}{2\log b}\right)^s & \textit{si } s = 2, \textit{ ou si } b = 2 \textit{ et } s \in \{3,4\}, \\[2ex] \dfrac{b^t}{s!}\,\dfrac{b-1}{2\lfloor b/2\rfloor}\left(\dfrac{\lfloor b/2\rfloor}{\log b}\right)^s & \textit{dans les autres cas.} \end{cases}$$

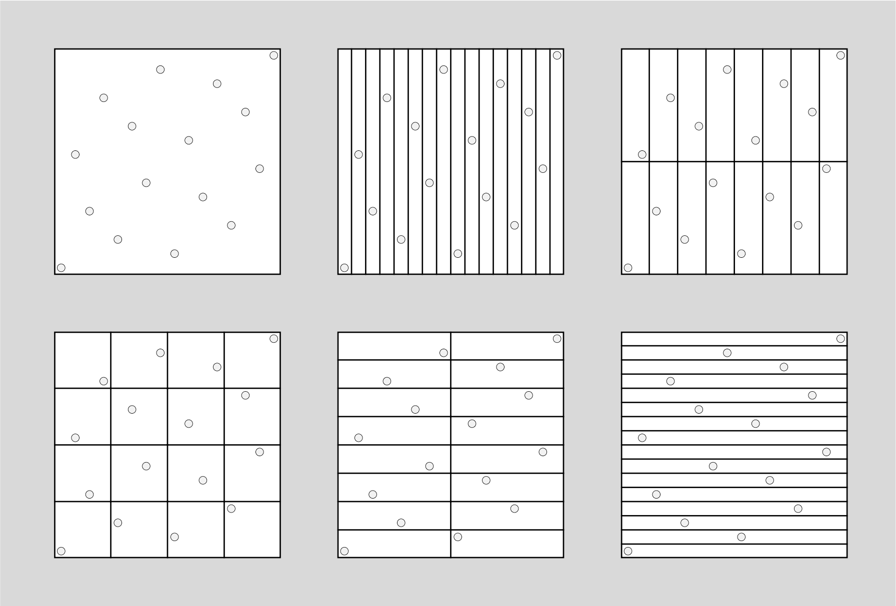


FIG. 4.7. On a représenté dans chacun de ces carrés les mêmes 16 points d'un $(0, 4, 2)$-réseau en base 2. On remarque que tout intervalle élémentaire de volume 2^{-4} contient un et un seul point de la séquence.

En utilisant cette nouvelle terminologie, les suites de van der Corput en base b (définition 4.1) sont des $(0, 1)$-suites en base b, les suites de Sobol (définition 4.12) sont des (t, s)-suites en base 2 (pour la même valeur de t que dans le théorème 4.14) et les suites de Faure en base b (définition 4.15) sont des $(0, s)$-suites en base b. Par contre, les suites de Halton ne sont pas des (t, s)-suites (par le simple fait qu'elles ne sont pas construites à partir d'une base unique), mais elles possèdent des propriétés passablement similaires.

On remarque que dans certains cas, les majorations du théorème 4.21 sont meilleures que celles des théorèmes 4.14 et 4.17. Par ailleurs, Niederreiter [Nie87] ne s'est pas contenté d'améliorer ces constantes ; il a également établi des bornes supérieures explicites sur la discrépance de tout (t, m, s)-réseau en base b et de tout segment initial d'une (t, s)-suite en base b (ces majorations sont présentées et discutées au chapitre 6).

Hormis celles de van der Corput, Sobol et Faure, d'autres (t, s)-suites ont été développées. Les plus connues sont sans doute celles de Niederreiter [Nie87][Nie88] et de Niederreiter et Xing [XN95][NX96] [NX98]. En guise de résumé, rapportons quelques caractéristiques de ces différentes constructions :

- ▷ Les suites de Sobol [Sob67] sont des (t, s)-suites en base 2, où $t = O(s \log s)$. Il en découle que $\lim_{s\to\infty} C_{b,s,t}$ tend vers l'infini.
- ▷ Les suites de Faure [Fau82] sont des $(0, s)$-suites définies pour toute base $b \geq s$, où b est un nombre premier. Si b est systématiquement choisi comme le plus petit nombre premier supérieur ou égal à s, on obtient $\lim_{s\to\infty} C_{b,s,t} = 0$.
- ▷ Les suites de Niederreiter [Nie87] sont des $(0, s)$-suites définies pour toute base $b \geq s$, où b est une puissance d'un nombre premier. Au niveau de la constante $C_{b,s,t}$, elles apportent une légère amélioration sur les suites de Faure dans certaines dimensions seulement.
- ▷ Les suites de Niederreiter [Nie88] sont des généralisations de toutes les précédentes. Il s'agit de la première construction à être définie pour toute base $b \geq 2$. Cependant, pour une base b fixée, on a encore $t = O(s \log s)$. Ces suites ont été implémentées par Bratley, Fox et Niederreiter [BFN92].

- ▷ Les suites de Niederreiter et Xing [XN95][NX96][NX98] sont des (t,s)-suites définies pour toute base $b \geq 2$, où b est une puissance d'un nombre premier. Ce sont les premières constructions pour lesquelles $t = O(s)$ (ce taux est le meilleur possible). Il en découle que $\lim_{s\to\infty} C_{b,s,t} = 0$ pour tout choix de la base. Malheureusement, bien que leur existence soit assurée, l'implémentation de ces suites pose des problèmes de géométrie algébrique calculatoire encore irrésolus. Les plus petites valeurs connues du paramètre t pour lesquelles il existe une (t,s)-suite de Niederreiter et Xing en base $2, 3$ ou 5 sont données dans le tableau suivant pour $s \leq 20$:

s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$b=2$	0	0	1	1	2	3	4	5	6	8	9	10	11	13	15	15	18	19	19	21
$b=3$	0	0	0	1	1	1	2	3	3	4	4	5	6	7	7	9	9	9	11	12
$b=5$	0	0	0	0	0	1	1	1	1	2	2	3	3	3	3	4	4	5	5	6

On constate qu'en base 2, ces paramètres sont bien meilleurs que les valeurs correspondantes des suites de Sobol. Des tables contenant les meilleurs paramètres connus pour les (t,s)-suites et les (t,m,s)-réseaux en base b ont été construites par Mullen, Mahalanabis et Niederreiter [MMN95], puis réactualisées par Clayman, Lawrence, Mullen, Niederreiter et Sloane [CLM+99].

4.7 Discussion

Ainsi, de nombreuses suites à discrépance faible ont été proposées ces dernières décennies. Aussi, est-il parfaitement légitime de soulever la question suivante : quelle séquence faut-il utiliser dans une approximation de quasi-Monte-Carlo ? Au sens du théorème 3.1 et de la remarque 3.5, la réponse est : pour réduire l'erreur (18) dans le pire des cas, la meilleure séquence x de n points est celle qui minimise la discrépance $D_n^*(x)$. Malheureusement, on ne sait calculer la discrépance que pour de petites valeurs de s et de n (voir chapitre 5) et l'étude des échantillons finis à discrépance minimale n'en est qu'à ses prémices (voir théorème 1.23 et section 8.5.4). Cette approche semblant impraticable, certains auteurs ont choisi la discrépance carrée moyenne ou des expériences sur des fonctions-tests comme critère de sélection, alors que d'autres ont préféré glaner des éléments de réponse dans les aspects théoriques des suites à discrépance faible (faisant parfois allégrement l'amalgame entre les caractéristiques asymptotiques de ces suites et les propriétés de leurs segments initiaux).

4.7.1 La discrépance carrée moyenne. Tezuka [Tez95], Warnock [War95] et James, Hoogland et Kleiss [JHK97] ont calculé (respectivement pour des tailles d'échantillon de $10\,000$, $50\,000$ et $100\,000$ points) la discrépance carrée moyenne $T_n^*(x)$ de différentes séquences dont certaines sont extraites des suites de Halton, Sobol, Faure et Niederreiter (en base 2). Les résultats de ces expériences peuvent être résumés de la manière suivante :

- ▷ En dimension $s \leq 10$, les séquences testées présentent toutes des valeurs similaires de $T_n^*(x)$. Ces mesures s'avèrent largement meilleures que l'espérance (7) de la discrépance carrée moyenne d'un échantillon de variables aléatoires i.i.d. $U(I^s)$ de même taille.
- ▷ Dans la zone transitoire $10 < s \leq 15$, les suites de Halton et de Faure présentent déjà des valeurs de $T_n^*(x)$ légèrement moins bonnes que les deux autres types de séquences.
- ▷ En dimension $s > 15$, aucune séquence ne présente une discrépance carrée moyenne clairement meilleure que l'espérance (7) et même, pour les échantillons des suites de Halton (et dans une moindre mesure de Faure), les mesures s'avèrent parfois supérieures à cette valeur.

Ces expériences confortent deux idées assez répandues :

- ▷ Dans une base élevée, une suite à discrépance faible peut présenter certaines pathologies. Pour les suites de Halton, la question a été discutée en page 32 et nous renvoyons le lecteur à Morokoff et Caflisch [MC94] et à Kocis et Whiten [KW97] pour le cas, moins flagrant, des suites de Faure ;

- ▷ la taille minimale pour qu'un échantillon d'une suite à discrépance faible présente des propriétés d'équirépartition réellement meilleures qu'une séquence aléatoire croît exponentiellement avec la dimension (notons que cette opinion, quoique sans doute fondée pour les suites à discrépance faible, ne va pas dans la même direction que le théorème 1.23).

Morokoff et Caflisch [MC94] et surtout Matoušek [Mat98] critiquent la discrépance carrée moyenne en tant que mesure de la non-uniformité d'une séquence x de n points. Ils lui reprochent de totalement biaiser la mesure en donnant un poids exagérément important aux points situés près de l'origine. Bien sûr, le phénomène est asymptotiquement négligeable, mais pour des tailles d'échantillon rencontrées en pratique, il s'avère être dominant, tout particulièrement en dimension élevée.

Ayant pris soin de retirer certains points proches de l'origine avant de calculer la discrépance carrée moyenne de différentes séquences en dimension $s = 4, 6, 10, 15$ et 20 pour $n = 2^{10}, 2^{12}, 2^{14}$ et 2^{16} points, Matoušek aboutit à des conclusions sensiblement différentes de celles des auteurs susmentionnés. Il apparaît notamment que pour une séquence extraite d'une suite de Faure ou de Halton, la discrépance carrée moyenne n'est certainement pas plus élevée que l'espérance de celle d'une séquence aléatoire.

Matoušek [Mat98] va même plus loin, exhibant le cas d'une séquence manifestement non uniforme (n copies du point $(1, \dots, 1)$ où n peut même croître exponentiellement avec la dimension) dont la discrépance carrée moyenne est proche de la meilleure possible. Ces éléments ne peuvent que jeter le discrédit sur la discrépance carrée moyenne et sur le rôle d'arbitre que certains voudraient lui conférer dans la comparaison des suites à discrépance faible.

4.7.2 Les fonctions-tests. De nombreux auteurs (voir par exemple Fox [Fox86], Bratley et Fox [BF88], Bratley, Fox et Niederreiter [BFN92], Morokoff et Caflisch [MC95], Radović, Sobol et Tichy [RST96], Kocis et Whiten [KW97]) ont comparé les performances empiriques de différentes séquences sur la base d'approximations de quasi-Monte-Carlo (17) effectuées sur des fonctions-tests intégrables analytiquement. En fin de compte, ces expériences s'avèrent globalement peu concluantes. En effet, il semble qu'en moyenne les séquences issues des suites de Halton, Sobol, Faure et Niederreiter mènent à des résultats relativement similaires. Bien sûr, dans certains cas particuliers, des différences notables ont été constatées, mais il semble hasardeux d'en tirer une quelconque conclusion définitive. En effet, les résultats sont tels qu'il paraît plus raisonnable d'attribuer les fluctuations observées aux particularités des fonctions choisies plutôt qu'aux qualités intrinsèques des séquences testées.

4.7.3 La discrépance asymptotique, la base et le paramètre de qualité. Considérant l'historique de l'apparition des suites à discrépance faible, on s'aperçoit que chaque nouvelle construction a permis de réduire la constante $C_{b,s,t}$ (du théorème 4.21) pour une dimension donnée ou de diminuer la valeur du paramètre t pour b et s fixés (les deux aspects étant bien évidemment intimement liés).

La table 4.2 illustre les progrès enregistrés au niveau de la réduction de cette constante. On y constate également sa divergence (lorsque $s \to \infty$) pour les suites de Halton et de Sobol, ainsi que sa convergence vers 0 dans le cas des suites de Faure, Niederreiter, et Niederreiter et Xing. Cette propriété pourrait laisser supposer que ces trois suites sont intrinsèquement meilleures que celles de Halton et de Sobol, du moins en dimension élevée. Cependant, il convient de souligner le fait que les constantes en question ne sont que des majorations et qu'elles ne concernent que le régime asymptotique (pour $n \to \infty$). Elles ne constituent donc pas un critère de comparaison fiable de la qualité réelle de ces suites (et encore moins d'une sous-séquence finie utilisée dans une simulation de quasi-Monte-Carlo).

Comme nous l'avons déjà mentionné au chapitre 3, la présence du facteur n^{-1} dans la majoration $D_n^*(x) \leq C\, n^{-1}(\log n)^s + O\left(n^{-1}(\log n)^{s-1}\right)$ est particulièrement réjouissante. Asymptotiquement, il s'agit clairement du terme dominant de l'expression, mais pour de petites valeurs de n, le facteur $(\log n)^s$ est loin d'être négligeable, tout particulièrement en dimension élevée. En effet, on vérifie facilement que

s	2	3	4	5	6	7	8	9	10	11	12
Halton	0.65	0.81	1.25	2.62	6.13	17.3	52.9	185.5	771.5	3 370	16 801
Sobol	0.26	0.25	0.541	0.833	1.6	2.6	7.6	19.6	45.0	94.7	182.2
Faure	0.26	0.13	0.099	0.025	0.019	0.0041	0.0089	$2.1 \cdot 10^{-3}$	$4.2 \cdot 10^{-4}$	$8.1 \cdot 10^{-5}$	$5.6 \cdot 10^{-5}$
Niederreiter	0.26	0.13	0.086	0.025	0.019	0.0041	0.0030	$6.1 \cdot 10^{-4}$	$4.2 \cdot 10^{-4}$	$8.1 \cdot 10^{-5}$	$5.6 \cdot 10^{-5}$
Nr-Xg$_{2,3,5}$	0.26	0.13	0.086	0.016	0.002	0.0009	0.0003	$3.2 \cdot 10^{-5}$	$8.7 \cdot 10^{-6}$	$7.2 \cdot 10^{-7}$	$1.6 \cdot 10^{-7}$

TAB. 4.2. Valeur de la constante C dans la majoration $D_n^*(x) \le C\, n^{-1}(\log n)^s + O\left(n^{-1}(\log n)^{s-1}\right)$ pour différentes suites (uniquement en base $2, 3$ ou 5 pour celles de Niederreiter et Xing). Dans chaque cas, le jeu de paramètres donnant lieu à la plus petite constante possible a été choisi.

la fonction $n^{-1}(\log n)^s$ est croissante pour tout $n < e^s$. D'autre part, comme le font remarquer Morokoff et Caflisch [MC94], la tendance générale de la discrépance d'une suite équirépartie étant de décroître lorsque n augmente, cette borne ne peut constituer une mesure représentative que lorsque ce maximum a été atteint (voir également la section 6.1).

Comme cela a déjà été mentionné en page 38, les meilleures propriétés d'équirépartition s'obtiennent pour des (t, s)-suites en base b présentant une valeur de t minimale (ce nombre est d'ailleurs couramment qualifié de *paramètre de qualité* de la suite). On observe que, de ce point de vue, les suites de Faure constituent une famille optimale. Hélas, si l'on tient également compte de la valeur de la base b, les choses ne sont pas si simples. En effet, le théorème suivant montre qu'une $(0, s)$-suite en base b ne peut exister que si $b \geq s$:

THÉORÈME 4.22 (Niederreiter [Nie87]) *Il n'existe aucune $(0, s)$-suite en base b avec $b < s$.*

Considérons plus en détail la structure d'un (t, m, s)-réseau en base b. On remarque que si $m = t$, la définition 4.19 confirme, si besoin est, que les b^m points sont bien dans le cube unité I^s. Pour $m = t + 1$, l'information n'est guère plus utile, sachant que les suites à discrépance faible classiques sont telles que les j^{e} composantes des points dont elles sont constituées sont distinctes par construction (voir remarque 4.16 pour le cas particulier des suites de Faure). En fin de compte, comme l'ont relevé Caflisch, Morokoff et Owen [CMO97], le fait d'être un (t, m, s)-réseau ne prend tout son sens que lorsqu'il implique une contrainte sur des intervalles élémentaires de pleine dimension (*i.e.* avec $d_j \geq 1$ pour tout $j \in \{1, \ldots, s\}$ dans la définition 4.18), ce qui n'est possible que pour $m \geq t + s$. D'ailleurs, ces auteurs conjecturent que $n_{b,s,t} = b^{t+s}$ pourraît en fait être le seuil du régime asymptotique, c'est-à-dire le nombre minimal de points à partir duquel la discrépance présenterait une décroissance quasi linéaire avec la taille de l'échantillon. Notons que cette hypothèse est corroborée par certains graphiques de Morokoff et Caflisch [MC94] sur la discrépance carrée moyenne des suites de Sobol.

Ainsi, suivant cet argument, le seuil critique se situe à b^s points pour les $(0, s)$-suites en base b. Considérant le théorème 4.22, on obtient $n_{b,s,t} = b^s \geq s^s$, ce qui peut être considéré comme le talon d'Achille des $(0, s)$-suites en dimension élevée. Par exemple, pour $s = 10$, on a $n_{b,s,t} = 11^{10} \approx 25 \cdot 10^9$ points pour les $(0, 10)$-suites de Faure et de Niederreiter en base 11. On comprend donc mieux l'intérêt porté aux constructions dont le paramètre t est non nul, mais où la base b est un petit entier indépendant de la dimension. Par exemple, pour une $(23, 10)$-suite de Sobol, on obtient $n_{b,s,t} = 2^{33} \approx 8 \cdot 10^9$ points. Finalement, pour une $(8, 10)$-suite de Niederreiter et Xing en base 2, le seuil se situe à seulement $2^{18} = 262\,144$ points. De telles considérations justifient l'élan d'enthousiasme soulevé par l'apparition de ces nouvelles suites, ainsi que les espoirs placés dans le développement d'une mise en œuvre.

Un autre point plaidant en faveur des suites en base 2 tient à l'utilisation de la méthode de génération rapide d'Antonov et Saleev [AS79]. Cette affirmation était pleinement justifiée avant que la généralisation en base quelconque présentée ci-dessous ne vienne relativiser l'argument.

4.8 Génération efficace à l'aide d'un code de Gray

Pour la génération d'un (t, m, s)-réseau de Sobol, de Faure ou de Niederreiter, la complexité d'une mise en œuvre directe est de $O(m^2 s)$ opérations élémentaires par point. Néanmoins, pour les suites en base 2, Antonov et Saleev [AS79] ont proposé une méthode, reposant sur l'utilisation d'un code de Gray, qui permet de réduire l'effort d'un facteur m. Cette approche rapide a été implémentée par Bratley et Fox [BF88] pour les suites de Sobol et adaptée par Bratley, Fox et Niederreiter [BFN92] pour les suites de Niederreiter en base 2. En revanche, pour les suites de Faure et de Niederreiter en base quelconque, seules les implémentations en $O(m^2 s)$, respectivement développées par Fox [Fox86] et Bratley, Fox et Niederreiter [BFN92], sont actuellement utilisées.

Une généralisation de la méthode d'Antonov et Saleev, permettant d'obtenir une complexité en $O(ms)$ quelle que soit la base a été proposée par Tezuka [Tez95]. Cependant, indépendamment, nous avons réinventé cette technique. Notre présentation [Thi98] s'avère plus détaillée et comprend la preuve (donnée ci-dessous) que son utilisation préserve les propriétés d'équirépartition des suites en question.

4.8.1 La méthode d'Antonov et Saleev. Cette technique a pour origine une patente déposée par Gray [Gra53] pour un appareillage électromécanique de communication destiné à renforcer la fiabilité de la transmission d'une suite de bits par une certain dispositif de codage (voir Press, Flannery et Teukolsky [PFT88] pour plus de détails). D'un point de vue mathématique, un code de Gray est une bijection définie sur certains ensembles d'entiers, telle que les images respectives de deux nombres consécutifs ne diffèrent que sur un seul bit.

DÉFINITION 4.23 Un *code de Gray* est une fonction $G : \mathbb{N} \to \mathbb{N}$ telle que

1° Pour tout $m \in \mathbb{N}^*$, la restriction de G aux entiers i tels que $0 \leq i < 2^m$ est une bijection.

2° Pour tout $i \in \mathbb{N}$, les représentations binaires de $G(i)$ et de $G(i+1)$ ne diffèrent que sur un seul bit.

Par exemple, la fonction G qui associe à tout $i \in \mathbb{N}$ le nombre

$$G(i) = i \oplus \lfloor i/2 \rfloor, \tag{20}$$

où $\oplus$ désigne l'opérateur binaire « ou exclusif », est un code de Gray. Pour $i \in \{0, \ldots, 15\}$, on obtient

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
i en base 2	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111
$\lfloor i/2 \rfloor$ en base 2	0	0	01	01	010	010	011	011	0100	0100	0101	0101	0110	0110	0111	0111
$G(i)$ en base 2	0	1	11	10	110	111	101	100	1100	1101	1111	1110	1010	1011	1001	1000
$G(i)$	0	1	3	2	6	7	5	4	12	13	15	14	10	11	9	8

De plus, si l'on note q_i l'index du bit nul le plus à droite dans la représentation binaire de i (en ajoutant un 0 à gauche si i est de la forme $2^m - 1$), alors $G(i)$ et $G(i+1)$ ne diffèrent qu'au niveau de ce q_i^e bit. Plus formellement, q_i est le plus petit index $k \in \{1, \ldots, m+1\}$ tel que $c_k(i) = 0$ dans la représentation binaire de i

$$i = \sum_{k=1}^{m+1} c_k(i)\, 2^{k-1}, \text{ où } m = \begin{cases} 1 & \text{pour } i = 0\,; \\ 1 + \lfloor \log_2 i \rfloor & \text{sinon.} \end{cases}$$

Ainsi, pour le code de Gray (20), les coefficients de la représentation binaire de $G(i+1)$ sont

$$c_k(G(i+1)) = \begin{cases} c_k(G(i)) + 1 \mod 2 & \text{si } k = q_i \\ c_k(G(i)) & \text{sinon} \end{cases} \quad \text{pour tout } k \in \{1, \ldots, m+1\}.$$

Antonov et Saleev [AS79] ont remarqué que ce fait pouvait être exploité pour accélérer la génération d'un (t, m, s)-réseau de Sobol. Rappelons que les composantes des points en question sont de la forme

$$x^i = \frac{1}{2^m} \left(\overset{m}{\underset{k=1}{\oplus}} c_k(i)\, l_k \right).$$

Un code de Gray étant une bijection sur les entiers i tels que $0 \le i < 2^m$, il est clair que la séquence $\bar{S} = \{\bar{x}^0, \ldots, \bar{x}^{2^m-1}\}$ donnée par

$$\bar{x}^i = x^{G(i)}$$

est une permutation du même (t, m, s)-réseau de Sobol. Appliquée à une (t, s)-suite, cette transformation préserve ses propriétés d'équirépartition. Pour n'importe quel segment initial de 2^m points, seul l'ordre de génération des points change. L'intérêt de cette approche tient au fait que

$$\bar{x}^{i+1} = \frac{1}{2^m} \left(\overset{m}{\underset{k=1}{\oplus}} c_k(G(i+1))\, l_k \right) = \frac{1}{2^m} \left(l_{q_i} \overset{m}{\underset{k=1}{\oplus}} c_k(G(i))\, l_k \right) = \frac{1}{2^m} \left(l_{q_i} \oplus (2^m\, \bar{x}^i) \right).$$

Il est ainsi possible de générer chaque nouveau point à partir du précédent en n'appliquant qu'une seule fois (au lieu de m) l'opérateur $\oplus$. Par conséquent, lorsque le « ou exclusif » est défini pour les entiers (de taille raisonnable) sur la machine utilisée (ce qui est généralement le cas), la complexité initiale $O(ms)$ s'en trouve réduite à $O(s)$ pour chaque point de la séquence. En revanche, dans le cas où le calcul doit être effectué bit après bit, on passe de $O(m^2 s)$ à $O(ms)$ opérations élémentaires par point.

4.8.2 Un code de Gray en base quelconque. Le cas binaire de la définition 4.23 se laisse généraliser facilement. En effet, un code de Gray en base b est une bijection définie sur certains ensembles d'entiers, telle que les représentations en base b des images respectives de deux nombres consécutifs ne diffèrent qu'au niveau d'un seul coefficient.

DÉFINITION 4.24 Un code de Gray en base b est une fonction $G : \mathbb{N} \to \mathbb{N}$ telle que

1° Pour tout $m \in \mathbb{N}^*$, la restriction de G aux entiers i tels que $0 \le i < b^m$ est une bijection.

2° Pour tout $i \in \mathbb{N}$, les représentations en base b de $G(i)$ et de $G(i+1)$ ne diffèrent qu'au niveau d'un seul coefficient ; l'écart en question est d'exactement $1 \mod b$.

On introduit la notation vectorielle $c(i) = (c_1(i), c_2(i), \ldots)^t$ (où t désigne la transposition) pour l'unique représentation d'un entier non négatif i

$$i = \sum_{k=1}^{\infty} c_k(i)\, b^{k-1}, \text{ avec } c_k(i) \in \mathbb{Z}_b \text{ pour tout } k \in \mathbb{N}^*.$$

THÉORÈME 4.25 *La fonction G qui associe à tout entier non négatif i le nombre $G(i)$ donné par sa représentation en base b*

$$c(G(i)) = B\, c(i) \mod b,$$

où B est la matrice

$$B = \begin{pmatrix} 1 & -1 & 0 & 0 & \cdots \\ 0 & 1 & -1 & 0 & \cdots \\ 0 & 0 & 1 & -1 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

est un code de Gray en base b. De plus, l'index q_i du coefficient qui diffère entre les représentations en base b de $G(i)$ et de $G(i+1)$ n'est autre que le plus petit $k \in \mathbb{N}^*$ tel que $c_k(i) \neq b-1$.

PREUVE. La matrice B étant triangulaire supérieure, il est clair que pour toute paire d'entiers $m \geq 1$ et $i \in \{0, \ldots, b^m - 1\}$, seules les composantes d'index 1 à m des vecteurs $c(i)$ et $c(G(i))$ peuvent comporter des coefficients non nuls. Plus précisément, on a

$$c(i) = (c_1(i), \ldots, c_{m-1}(i), c_m(i), 0, 0, \ldots)^t$$

et

$$c(G(i)) = (c_1(i) - c_2(i) \mod b, \ldots, c_{m-1}(i) - c_m(i) \mod b, c_m(i), 0, 0, \ldots)^t.$$

Soit une paire d'entiers distincts $i, j \in \{0, \ldots, b^m - 1\}$. En notant k le plus grand index tel que $c_k(i) \neq c_k(j)$, on obtient directement $c_k(G(i)) \neq c_k(G(j))$ et donc $G(i) \neq G(j)$. On en conclut que la restriction de G aux entiers i tels que $0 \leq i < b^m$ est une bijection pour tout $m \in \mathbb{N}^*$ (on note au passage que l'application réciproque de G peut être exprimée à l'aide de la matrice triangulaire supérieure dont tous les éléments sur et au-dessus de la diagonale sont égaux à 1).

Soit q_i l'index du premier coefficient différent de $b - 1$ dans la représentation de i en base b

$$i = \sum_{k=1}^{\infty} c_k(i)\, b^{k-1}, \text{ où } c_k(i) \in \begin{cases} \{b-1\} & \text{pour } k < q_i \\ \{0, \ldots, b-2\} & \text{pour } k = q_i \\ \mathbb{Z}_b & \text{pour } k > q_i \end{cases}$$

On en déduit la représentation de $i + 1$ en base b

$$i + 1 = \sum_{k=1}^{\infty} c_k(i+1)\, b^{k-1}, \text{ avec } c_k(i+1) = \begin{cases} c_k(i) + 1 \mod b & \text{pour } k \leq q_i \\ c_k(i) & \text{pour } k > q_i \end{cases}$$

ainsi que les coefficients de celle de $G(i + 1)$:

$$c_k(G(i+1)) = (B\, c(i+1))_k \mod b = c_k(i+1) - c_{k+1}(i+1) \mod b$$

$$= \begin{cases} c_k(i) + 1 - (c_{k+1}(i) + 1) \mod b & = c_k(G(i)) & \text{pour } k < q_i \\ c_k(i) + 1 - c_{k+1}(i) \mod b & = c_k(G(i)) + 1 \mod b & \text{pour } k = q_i \\ c_k(i) - c_{k+1}(i) \mod b & = c_k(G(i)) & \text{pour } k > q_i \end{cases}$$

□

Connaissant l'index q_i, la mise à jour de $c(G(i + 1))$ à partir de $c(G(i))$ est réalisable en temps constant. Pour déterminer q_i, il suffit de parcourir les composantes du vecteur $c(i)$ jusqu'à rencontrer un nombre différent de $b - 1$. L'algorithme suivant se charge d'effectuer ces différentes tâches :

Algorithme 4.1 Mise à jour des représentations en base b lors du passage de i à $i + 1$

Donnée : représentation en base b de i et de $G(i)$ dans les vecteurs $(c_1, c_2, \ldots)$ et $(g_1, g_2, \ldots)$

Résultat : représentation en base b de $i + 1$ et de $G(i + 1)$ dans les vecteurs $(c_1, c_2, \ldots)$ et $(g_1, g_2, \ldots)$

```
$q \leftarrow 1$
tant que $c_q = b - 1$ faire
    $c_q \leftarrow 0$
    $q \leftarrow q + 1$
$c_q \leftarrow c_q + 1$
$g_q \leftarrow g_q + 1 \mod b$
```

On observe que le nombre de composantes parcourues par l'algorithme avant de trouver q dépend de la valeur de i. Afin d'évaluer l'effort à fournir, il paraît plus intéressant de considérer le temps moyen nécessaire au calcul de q_i pour tout i compris entre 0 et $b^m - 1$. Il est clair que sur les b^m itérations, le coefficient d'index k est examiné à b^{m-k+1} reprises. En tout, le test de la ligne 2 est donc exécuté

$$\sum_{k=1}^{m} b^{m-k+1} = \sum_{k=1}^{m} b^k = \frac{b^{m+1}-1}{b-1} - 1 = b^m\left(1 + \frac{1}{b-1}\right) - \frac{b}{b-1}$$

fois. Ainsi, la complexité moyenne de l'algorithme 4.1 pour la mise à jour des représentations de i et de $G(i)$ en base b pour toutes les valeurs successives de i comprises entre 0 et $b^m - 1$ est en $O(b^{-1})$. Comme on pouvait s'y attendre, l'opération est en moyenne moins coûteuse lorsque la base est élevée.

4.8.3 Génération rapide en base quelconque. La généralisation directe de la méthode d'Antonov et Saleev consistant à remplacer le code de Gray (20) par celui du théorème 4.25 permet d'accélérer la génération des points tout en préservant ses propriétés d'équirépartition.

THÉORÈME 4.26 *Si $x = \{x^0, x^1, \dots\}$ est une (t, s)-suite en base b et G est le code de Gray du théorème 4.25, alors la suite $\bar{x} = \{\bar{x}^0, \bar{x}^1, \dots\}$ donnée par $\bar{x}^i = x^{G(i)}$ est une (t, s)-suite en base b.*

PREUVE. Il suffit de montrer que pour toute paire d'entiers $k \geq 0$ et $m \geq t$, la séquence

$$\left\{\bar{x}^{kb^m}, \dots, \bar{x}^{(k+1)b^m-1}\right\}$$

est un (t, m, s)-réseau en base b. L'ensemble de vecteurs

$$\{(d_1, d_2, \dots)^t : d_r \in \mathbb{Z}_b,\ \forall r \leq m \text{ et } d_r = c_{r-m}(k),\ \forall r > m\}$$

contient la représentation en base b de tous les entiers i tels que $kb^m \leq i < (k+1)b^m$. À nouveau, considérant la structure de la matrice B, il est clair qu'en multipliant chaque élément de cet ensemble par B et en prenant le résultat modulo b, on obtient les vecteurs

$$\{(d_1, d_2, \dots)^t : d_r \in \mathbb{Z}_b,\ \forall r \leq m \text{ et } d_r = c_{r-m}(G(k)),\ \forall r > m\},$$

c'est-à-dire la représentation en base b de tous les entiers i tels que $G(k)b^m \leq i < (G(k)+1)b^m$. □

De plus, pour tout $m \in \mathbb{N}^*$, il découle directement de la définition 4.24 que la séquence constituée des b^m premiers points de la suite $\bar{x}$ est une simple permutation de l'ensemble $\{x^0, \dots, x^{b^m-1}\}$.

REMARQUE 4.27 Le théorème 4.26 n'est pas correct pour tout code de Gray en base b. Par exemple, pour une $(0, 2)$-suite de Faure en base 2 et n'importe quel code de Gray tel que $G(0) = 0$ et $G(1) = 2$, les deux premiers points de la suite obtenue ne forment pas un $(0, 1, 2)$-réseau en base 2.

Au niveau de la taille des séquences, le passage d'un (t, m, s) à un $(t, m+1, s)$-réseau en base b peut représenter un saut important, particulièrement lorsque b est grand. Il est alors intéressant de considérer des subdivisions plus fines (d'une (t, s)-suite) possédant de bonnes propriétés d'équirépartition. C'est dans cet esprit qu'Owen [Owe97a] a introduit la notion de (ω, t, m, s)-réseau en base b.À nouveau, on montre que l'utilisation du code de Gray du théorème 4.25 ne perturbe pas la qualité de ces séquences.

DÉFINITION 4.28 Soit des entiers $0 \leq t \leq m$ et $1 \leq \omega < b$. Un (ω, t, m, s)-réseau en base b est une séquence x de ωb^m points dans I^s telle que $A(P, x) = \omega b^t$ pour tout intervalle élémentaire P en base b de volume b^{t-m} et $A(P, x) \leq b^t$ pour tout intervalle élémentaire P en base b de volume b^{t-m-1}.

REMARQUE 4.29 Par définition, si $x = \{x^0, x^1, \dots\}$ est une (t,s)-suite en base b et $m \geq t, a \geq 0$ et $1 \leq \omega < b$ sont des entiers, alors la séquence $\{x^i : ab^{m+1} \leq i < ab^{m+1} + \omega b^m\}$ est un (ω, t, m, s)-réseau en base b.

THÉORÈME 4.30 *Si $x = \{x^0, x^1, \dots\}$ est une (t,s)-suite en base b et $m \geq t, a \geq 0$ et $1 \leq \omega < b$ sont des entiers, alors la séquence $\bar{x} = \{x^{G(i)} : ab^{m+1} \leq i < ab^{m+1} + \omega b^m\}$, où G est le code de Gray du théorème 4.25, est un (ω, t, m, s)-réseau en base b.*

PREUVE. On a

$$\begin{aligned} \bar{x} &= \bigcup_{k=0}^{\omega-1} \left\{ x^{G(i)} : ab^{m+1} + kb^m \leq i < ab^{m+1} + (k+1)b^m \right\} \\ &= \bigcup_{k=0}^{\omega-1} \left\{ x^{G(i)} : (ab+k)\, b^m \leq i < (ab+k+1)\, b^m \right\}. \end{aligned}$$

Par l'argument de la preuve du théorème 4.26, on voit que $\bar{x}$ est l'union de (t,m,s)-réseaux en base b

$$\bar{x} = \bigcup_{k=0}^{\omega-1} \left\{ x^i : G(ab+k)\, b^m \leq i < (G(ab+k)+1)\, b^m \right\},$$

ce qui montre que $A(P, \bar{x}) = \omega b^t$ pour tout intervalle élémentaire P en base b de volume b^{t-m}. D'autre part, on obtient $A(P, \bar{x}) \leq b^t$ pour tout intervalle élémentaire P en base b de volume b^{t-m-1}, car tous les points de $\bar{x}$ font partie du même $(t, m+1, s)$-réseau en base b

$$\left\{ x^i : G(a)\, b^{m+1} \leq i < (G(a)+1)\, b^{m+1} \right\}.$$

□

4.8.4 Le cas particulier des suites de Faure. Rappelons qu'un $(0,m,s)$-réseau de Faure en base b $\{x^0, \dots, x^{b^m-1}\} \subset I^s$ se laisse construire de la manière suivante (voir définition 4.15) :

$$x^i = (x_1^i, \dots, x_s^i),$$

$$x_j^i = \sum_{p=1}^{m} x_{jp}^i\, b^{-p},$$

$$x_{jp}^i = \sum_{q=p}^{m} a_{pq}^{d_j}\, c_q(i) \mod b,$$

où $(c_1(i), \dots, c_m(i))$ désigne la représentation de $i \in \{0, \dots, b^m - 1\}$ en base b

$$i = \sum_{k=1}^{m} c_k(i)\, b^{k-1}, \text{ avec } c_k(i) \in \mathbb{Z}_b \text{ pour tout } k = 1, \dots, m.$$

On remarque qu'une mise en œuvre basée sur l'utilisation directe de ces expressions implique un effort en $O(m^2 s)$ par point. En revanche, en considérant le code de Gray G du théorème 4.25, il est possible de générer une permutation du même réseau nettement plus rapidement. En effet, en posant

$$\bar{x}^i = x^{G(i)} = \left(x_1^{G(i)}, \dots, x_s^{G(i)} \right)$$

pour tout $i \in \{0, \dots, b^m - 1\}$, on a toujours

$$x_j^{G(i)} = \sum_{p=1}^{m} x_{jp}^{G(i)}\, b^{-p}.$$

De plus, comme les représentations en base b de $G(i)$ et $G(i+1)$ ne diffèrent (de $1 \mod b$) qu’au niveau du coefficient d’index q_i, on obtient

$$x_{jp}^{G(i+1)} = \sum_{q=p}^{m} a_{pq}^{d_j}\, c_q(G(i+1)) \mod b = x_{jp}^{G(i)} + a_{pq_i}^{d_j} \mod b.$$

On ramène ainsi la complexité à $O(ms)$ par point. De plus, on observe que $\bar{x}^0 = x^0 = (0, \dots, 0)$.

Une implémentation de cette méthode, baptisée `GrayFaure`, est disponible en langage C (voir Thiémard [Thi98] pour une description détaillée). Ses performances ont été confrontées à celles de la mise en œuvre correspondante de Fox [Fox86] en Fortran77 (`FoxFaureF`). Nous l’avons toutefois traduite en C (`FoxFaureC`), de manière à pouvoir comparer les méthodes dans un même langage. La table 4.3 illustre les performances de ces générateurs sur une station de travail SGI R10000 pour des séquences de différentes tailles extraites d’une suite de Faure en dimension 10.

n	`FoxFaureF`	`FoxFaureC`	`GrayFaure`	`FoxFaureF` / `GrayFaure`	`FoxFaureC` / `GrayFaure`
$100\,000$	3.554 s	3.065 s	0.204 s	17.4	15.0
$500\,000$	21.12 s	18.24 s	1.040 s	20.3	17.5
$1\,000\,000$	43.60 s	37.70 s	2.082 s	20.9	18.1
$5\,000\,000$	256.1 s	222.8 s	10.39 s	24.6	21.4
$10\,000\,000$	531.2 s	462.9 s	20.77 s	25.6	22.3
$50\,000\,000$	$3\,064$ s	$2\,694$ s	103.9 s	29.5	25.9
$100\,000\,000$	$6\,358$ s	$5\,605$ s	207.9 s	30.6	27.0

TAB. 4.3. Comparaison du temps (en secondes) nécessaire à la génération de séquences de différentes tailles pour l’implémentation originale de Fox, sa traduction en langage C et notre approche par un code de Gray.

Ces résultats mettent en lumière les bénéfices que l’on peut tirer de l’utilisation d’un code de Gray. Ils confirment également que la supériorité de notre approche s’accentue lorsque la taille de la séquence augmente (ce fait est clair au vu des complexités respectives des deux méthodes). De plus, notre implémentation effectue tous les calculs intermédiaires en arithmétique entière (ce qui lui confère une plus grande précision) et fonctionne pour n’importe quelle valeur de s (alors que la mise en œuvre de Fox ne dépasse pas la dimension 40).

Partie 2

Calcul et majoration de la discrépance

CHAPITRE 5

Approches calculatoires classiques

Bien que la question du calcul de la discrépance soit centrale en simulation de quasi-Monte-Carlo (voir chapitre 3), la littérature sur le sujet s'avère particulièrement mince. La définition usuelle de la discrépance (3) peut laisser penser que le problème est continu, mais Niederreiter [Nie72] a montré qu'il est en fait discrétisable. Son approche de calcul, ainsi que les résultats de quelques expériences numériques associées, sont présentés dans la première partie de ce chapitre.

Il n'existe à ce jour (si l'on exclut la nouvelle approche présentée au chapitre 8) qu'une seule autre méthode permettant de calculer la discrépance. Il s'agit d'un algorithme proposé par Dobkin et Eppstein [DE93]. Cette technique est sommairement présentée dans la seconde partie de ce chapitre. Elle repose essentiellement sur une structure de données complexe permettant de déterminer les intervalles de plus petit et de plus grand volume contenant exactement k parmi les n points de la séquence. Ces deux approches mènent à des algorithmes dont la complexité est exponentielle en la taille du problème.

5.1 La discrétisation de Niederreiter

Soit une séquence $x = \{x^1, \dots, x^n\} \subset \bar{I}^s$. Pour tout $j \in \{1, \dots, s\}$, on note

$$0 = \delta_j^0 < \delta_j^1 < \cdots < \delta_j^{n_j} = 1$$

les coordonnées distinctes apparaissant dans l'ensemble

$$\{x_j^1, \dots, x_j^n\} \cup \{0, 1\}.$$

On pose ensuite

$$\mathcal{Q} = \{(q_1, \dots, q_s) : 0 \leq q_j < n_j,\ \forall j \in \{1, \dots, s\}\},$$

ainsi que

$$P^-(q) = \prod_{j=1}^{s} \left[0, \delta_j^{q_j}\right) \quad \text{et} \quad P^+(q) = \prod_{j=1}^{s} \left[0, \delta_j^{1+q_j}\right) \quad \text{pour tout } q \in \mathcal{Q}.$$

Niederreiter [Nie72] a montré que

$$D_n^*(x) = \max_{q \in \mathcal{Q}} \max \left\{ \left| \frac{A\left(P^+(q), x\right)}{n} - \lambda\left(P^+(q)\right) \right|, \left| \frac{A\left(P^+(q), x\right)}{n} - \lambda\left(P^-(q)\right) \right| \right\}. \tag{21}$$

En d'autres termes, le problème du calcul de la discrépance peut se ramener à la considération de la grille de taille $O(n^s)$ engendrée par les bords du cube et les coordonnées des n points de la séquence. En tenant compte du temps nécessaire au décompte des points situés dans chaque intervalle $P^{\pm}(q)$, on obtient $O(n^{s+1})$ comme complexité d'une mise en œuvre naïve de cette discrétisation. A priori, cette approche est donc impraticable dans le cas d'une séquence en dimension élevée.

Malgré tout, afin d'avoir une idée concrète de la difficulté du problème abordé, nous avons décidé de tester cette méthode sur quelques $(0, m, s)$-réseaux de Faure en base b. Seules les dimensions pour lesquelles il a été possible de calculer la discrépance d'un tel $(0, 2, s)$-réseau (*i.e.* b^2 points) en moins d'une semaine sur une station de travail SGI R10000 ont été considérées. Sous ces conditions, il n'a pas été possible de dépasser $s = 6$. Les résultats obtenus sont donnés dans les tables 5.1 à 5.4.À notre connaissance, le calcul de la discrépance de ces séquences (ou d'autres réseaux similaires) n'a jamais été présenté jusque-là. On remarque au passage que les valeurs de la table 5.1 sont très proches de celles obtenues par les formules de De Clerck pour les séquences de Hammersley (voir table 4.1).

m	1	2	3	4	5	6	7	8	9	10	11	12	13
$n = 2^m$	2	4	8	16	32	64	128	256	512	$1\,024$	$2\,048$	$4\,096$	$8\,192$
$D_n^*(x)$	0.75	0.4375	0.3125	0.171875	0.089844	0.053711	0.025146	0.014587	0.008408	0.004299	0.002448	0.001329	0.000691

TAB. 5.1. Discrépance de quelques $(0, m, 2)$-réseaux de Faure en base 2.

m	1	2	3	4	5	6
$n = 3^m$	3	9	27	81	243	729
$D_n^*(x)$	0.703704	0.297668	0.196159	0.071280	0.029364	0.013329

TAB. 5.2. Discrépance de quelques $(0, m, 3)$-réseaux de Faure en base 3.

m	1	2	3
$n = 5^m$	5	25	125
$D_n^*(x)$	0.6704	0.170455	0.089387

TAB. 5.3. Discrépance de quelques $(0, m, 4)$-réseaux de Faure en base 5.

	$s = 5$ $b = 5$		$s = 6$ $b = 7$	
m	1	2	1	2
$n = b^m$	5	25	7	49
$D_n^*(x)$	0.722240	0.238297	0.724333	0.210972

TAB. 5.4. Discrépance de quelques $(0, m, s)$-réseaux de Faure en base b.

Les mesures effectuées dans le cadre de ces expériences ont indiqué que le temps nécessaire à la détermination de la discrépance d'une séquence de n points en dimension s sur une station de travail SGI R10000 à l'aide de la discrétisation (21) est d'environ $10^{-11} n^{s+1}$ jours. En dimension supérieure (pour $s \geq 7$ avec $m \geq 2$) ou pour des réseaux plus importants (pour $s \leq 6$ et de plus grandes valeurs de m que celles données dans les tables), le calcul aurait pris plus d'une semaine. On en conclut que l'efficacité de cette approche est particulièrement limitée en dimension élevée.

5.2 L'algorithme de Dobkin et Eppstein

En 1977, Klee [Kle77] s'est intéressé au problème suivant : calculer la mesure de l'union $\bigcup_{i=1}^{n} P_i$ de n intervalles $P_1, \ldots, P_n$ de la forme $\prod_{j=1}^{s} [\alpha_j, \beta_j]$. En dimension $s = 1$, il a donné un algorithme en $O(n \log n)$ et soulevé la question de son optimalité.

Cette question a éveillé l'intérêt de nombreux chercheurs (voir Preparata et Shamos [PS85] pour un compte rendu) et il a été démontré depuis que cette complexité est optimale pour $s = 1$ et $s = 2$. En dimension supérieure, le meilleur algorithme connu a été proposé par Overmars et Yap [OY91]. Il repose sur une subdivision particulière de $\mathbb{R}^s$ en $O(n^{s/2})$ régions (dépendant des intervalles $P_1, \ldots, P_n$), sur une technique de balayage (par des plans parallèles aux axes), ainsi que sur une structure de données spécifique permettant de maintenir certaines informations. Sa complexité est en $O(n^{s/2} \log n)$.

L'algorithme de Dobkin et Eppstein pour le calcul de la discrépance [DE93][DEM96] utilise ces éléments de manière intensive. Leur approche est la suivante : au lieu de chercher l'intervalle $P \in \mathcal{I}_s^*$ qui maximise la discrépance locale (4), on commence par déterminer pour tout $k \in \{0, \ldots, n\}$ quels sont les intervalles de plus petit et de plus grand volume qui contiennent exactement k parmi les n points de la séquence, puis on prend le maximum des discrépances locales obtenues. Cependant, on n'aborde pas ce nouveau problème frontalement. On commence par le dualiser en remplaçant tout point x^i de la séquence par un nouvel intervalle de la forme $\prod_{j=1}^{s} [x_j^i, \infty)$ et tout intervalle considéré dans la discrétisation (21) par le point qui lui est associé (son « coin supérieur-droit »). Maintenant, le problème revient à chercher le point dual contenu dans exactement k intervalles duaux qui minimise ou maximise le produit de ses coordonnées. Il se trouve qu'il est possible d'utiliser la subdivision de $\mathbb{R}^s$ et la structure de données de Overmars et Yap pour déterminer ces points en $O(n^{s/2})$. Répétant l'opération pour tout $k \in \{0, \ldots, n\}$, on obtient un algorithme en $O(n^{1+s/2})$.

Cette complexité en fait la meilleure méthode connue pour le calcul de la discrépance. Cependant, les structures de données impliquées étant très sophistiquées, on peut craindre que la constante cachée dans ce $O(n^{1+s/2})$ soit très élevée. Ainsi, en fin de compte, il n'est pas évident que cet algorithme (qui reste exponentiel) se montre réellement plus performant que l'approche directe par discrétisation (21) sur des problèmes solubles en un temps acceptable par l'une des deux méthodes. Cette conclusion n'est d'ailleurs pas récusée par Dobkin (communication personnelle, 2000). Il est clair que cette question mérite d'être testée numériquement, mais l'algorithme de Dobkin et Eppstein n'a apparemment jamais été implémenté en dimension supérieure à 2.

CHAPITRE 6

Quelques bornes pour la discrépance

Au premier abord, la difficulté du problème du calcul de la discrépance (voir chapitre 5) semble contrebalancée par l'existence de bornes supérieures explicites pour certaines séquences que l'on sait asymptotiquement efficaces dans une application de la méthode de quasi-Monte-Carlo (17). Cependant, les majorations en question ne s'avèrent non triviales (*i.e.* inférieures à 1) que pour des séquences dont la taille est supérieure à un seuil critique qui croît de manière exponentielle avec la dimension.

En revanche, le calcul de bornes inférieures pour la discrépance est une tâche relativement aisée. Les diverses techniques existantes sont présentées dans la section 6.2.

6.1 Le cas des (t, s)-suites et des (t, m, s)-réseaux

Généralisant les travaux de Sobol [Sob67] et de Faure [Fau82], Niederreiter [Nie87] a établi des bornes supérieures pour la discrépance des (t, m, s)-réseaux et des (t, s)-suites en base b. La convention

$$i > j \text{ ou } i < 0 \quad \Longrightarrow \quad \binom{j}{i} = 0$$

est utilisée pour l'écriture des coefficients binomiaux apparaissant dans les expressions ci-dessous.

Théorème 6.1 (Niederreiter [Nie87]) *Si x est un (t, m, s)-réseau en base b, on a*

$$\begin{aligned}
D^*_{b^m}(x) &\leq b^{t-m} \sum_{i=0}^{s-1} \binom{s-1}{i} \binom{m-t}{i} \left\lfloor \frac{b}{2} \right\rfloor^i, \textit{ pour } b \geq 3 \\
D^*_{b^m}(x) &\leq b^{t-m} \left[\sum_{i=0}^{s-1} \binom{m-t}{i} \left(\frac{b}{2}\right)^i + \left(\frac{b}{2} - 1\right) \sum_{i=0}^{s-2} \binom{m-t+i+1}{i} \left(\frac{b}{2}\right)^i \right], \textit{ pour } b \textit{ pair} \\
D^*_{b^m}(x) &\leq b^{t-m} \left\lfloor \frac{b-1}{2}(m-t) + \frac{3}{2} \right\rfloor, \textit{ pour } s = 2 \\
D^*_{b^m}(x) &\leq b^{t-m} \left\lfloor \left(\frac{b-1}{2}\right)^2 (m-t)^2 + \frac{b-1}{2}(m-t) + \frac{9}{4} \right\rfloor, \textit{ pour } s = 3 \\
D^*_{b^m}(x) &\leq b^{t-m} \left\lfloor \left(\frac{b-1}{2}\right)^3 (m-t)^3 + \frac{3}{8}(b-1)^2(m-t)^2 + \frac{3}{8}(b-1)(m-t) + \frac{15}{4} \right\rfloor, \textit{ pour } s = 4
\end{aligned}$$

Théorème 6.2 (Niederreiter [Nie87]) *Pour une (t, s)-suite x en base b et un entier $n \geq b^t$, on a*

$$\begin{aligned}
D^*_n(x) \leq \frac{b^t}{n} \Bigg[& \frac{b-1}{2} \sum_{i=1}^{s} \binom{s-1}{i-1} \binom{k+1-t}{i} \left\lfloor \frac{b}{2} \right\rfloor^{i-1} \\
& + \frac{1}{2} \sum_{i=0}^{s-1} \binom{s-1}{i} \left(\binom{k+1-t}{i} + \binom{k-t}{i} \right) \left\lfloor \frac{b}{2} \right\rfloor^i \Bigg], \textit{ pour } b \geq 3
\end{aligned}$$

$$\begin{aligned}
D_n^*(x) &\leq \frac{b^t}{n}\left[\frac{b-1}{b}\sum_{i=1}^{s}\binom{k+1-t}{i}\left(\frac{b}{2}\right)^i + \frac{(b-1)(b-2)}{2b}\sum_{i=1}^{s-1}\binom{k+i+1-t}{i}\left(\frac{b}{2}\right)^i\right.\\
&\quad + \frac{1}{2}\sum_{i=0}^{s-1}\left(\binom{k+1-t}{i}+\binom{k-t}{i}\right)\left(\frac{b}{2}\right)^i\\
&\quad \left. + \frac{b-2}{4}\sum_{i=0}^{s-2}\left(\binom{k+i+2-t}{i}+\binom{k+i+1-t}{i}\right)\left(\frac{b}{2}\right)^i\right], \textit{ pour } b \textit{ pair}\\
D_n^*(x) &\leq \frac{b^t}{n}\left[\frac{1}{8}(b-1)^2(k-t)^2 + \frac{1}{8}(b-1)(b+9)(k-t) + \frac{3}{4}(b+1)\right], \textit{ pour } s=2\\
D_n^*(x) &\leq \frac{b^t}{n}\left[\frac{1}{24}(b-1)^3(k-t)^3 + \frac{1}{16}(b-1)^2(b+5)(k-t)^2\right.\\
&\quad \left. + \frac{1}{48}(b-1)(b^2+16b+61)(k-t) + \frac{1}{8}(b^2+4b+13)\right], \textit{ pour } s=3\\
D_n^*(x) &\leq \frac{b^t}{n}\left[\frac{1}{64}(b-1)^4(k-t)^4 + \frac{1}{32}(b-1)^3(b+5)(k-t)^3\right.\\
&\quad + \frac{1}{64}(b-1)^2(b^2+16b+13)(k-t)^2\\
&\quad \left. + \frac{1}{32}(b-1)(7b^2+b+64)(k-t) + \frac{1}{16}(b^3+8b+51)\right], \textit{ pour } s=4
\end{aligned}$$

où k est le plus grand entier tel que $b^k \leq n$.

Lorsque la dimension est très petite, les majorations fournies par les théorèmes 6.1 et 6.2 semblent relativement bonnes. Par exemple, en dimension 2, ces bornes supérieures sont très proches des valeurs exactes calculées dans la section 5.1 pour certains $(0,m,2)$-réseaux de Faure en base 2 (voir figure 6.1). Retournant l'argument, on peut en conclure que les réseaux de Faure en question sont d'excellente qualité. D'autre part, comme l'illustre également cette figure, les valeurs obtenues sont bien plus petites pour un réseau (lorsque $n = b^m$) que pour une séquence de taille voisine arbitraire.

Malheureusement, en dimension plus élevée, la situation se détériore très nettement. Par exemple, la figure 6.2 révèle que pour un $(0,m,13)$-réseau en base 13, la borne supérieure fournie par le théorème 6.1 est plus grande que 1 pour tout $m < 10$, c'est-à-dire pour tout réseau comprenant moins de $13^{10} \approx 1.3 \cdot 10^{11}$ points. De même, pour des séquences plus courtes, les valeurs obtenues à l'aide du théorème 6.2 s'avèrent totalement inutiles. Ce phénomène semble même s'accentuer à vitesse exponentielle lorsque la dimension augmente. Cette conclusion s'impose d'elle-même en considérant les valeurs de la table 6.1. Pour $s \in \{2, \ldots, 20\}$, on y trouve la taille minimale d'un $(0,m,s)$-réseau de Faure en base b (où b est le plus petit nombre premier supérieur ou égal à s) pour lequel le théorème 6.1 fournit une borne pour la discrépance plus petite que 1. Approximativement, ces valeurs sont de l'ordre de 10^s points. D'autre part, comme l'illustre la table 6.2, la situation n'est guère plus réjouissante pour les suites de Niederreiter et Xing en base $b \in \{2,3,5\}$. En effet, bien que les réseaux correspondants possèdent d'excellentes propriétés théoriques, la taille minimale nécessaire à l'obtention d'une borne non triviale est encore de l'ordre de 4^s points.

Poursuivant la discussion amorcée dans la section 4.7.3, il paraît raisonnable d'avancer que, le théorème 6.1 étant valide pour un réseau de taille quelconque, la borne qu'il fournit ne peut décrire le comportement de la discrépance qu'une fois le régime asymptotique de la suite sous-jacente atteint. Bien que cette notion ne soit pas clairement établie, on dispose néanmoins de deux candidats intéressants (et relativement concordants) pour en marquer le commencement : le seuil $n_{b,s,t}$ donné en page 42 et les

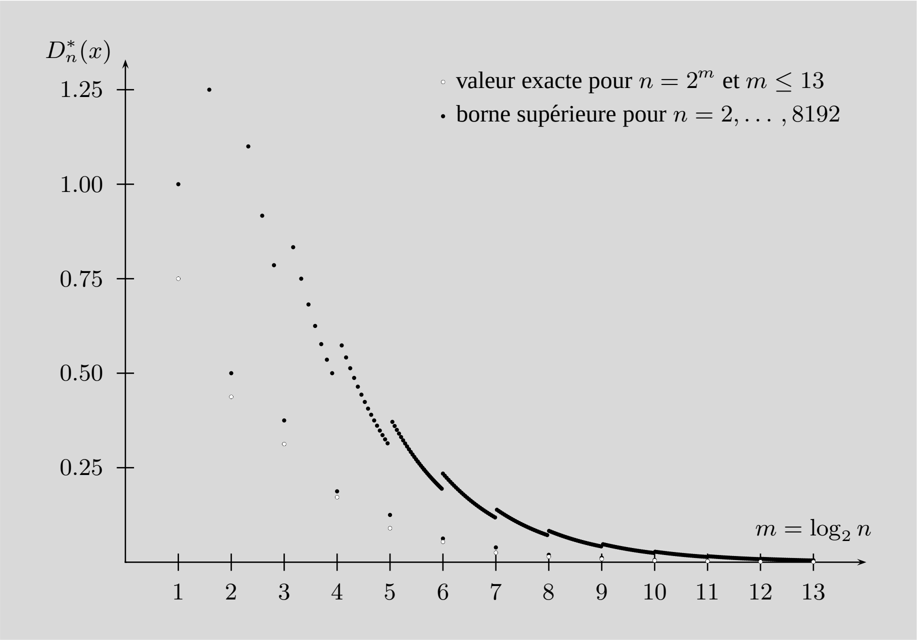


FIG. 6.1. Comparaison entre les majorations fournies par les théorèmes 6.1 et 6.2 et la valeur exacte de la discrépance des quelques $(0, m, 2)$-réseaux de Faure en base 2 de la table 5.1).

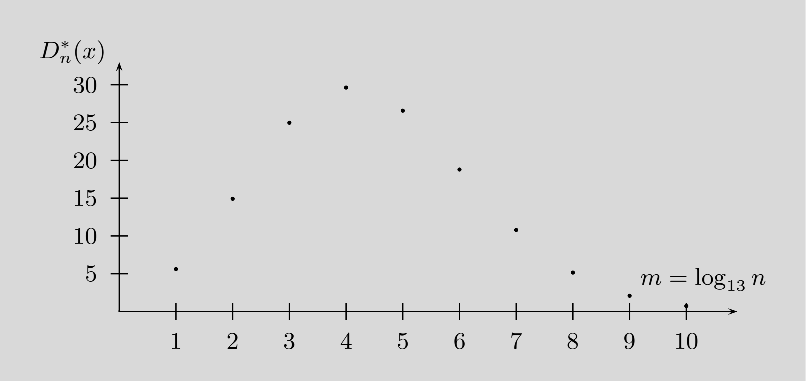


FIG. 6.2. Majoration du théorème 6.1 pour la discrépance de quelques $(0, m, 13)$-réseaux en base 13.

tailles minimales définies à la manière de celles des tables 6.1 et 6.2. De plus, on observe que, une fois cette valeur atteinte, les bornes fournies par les théorèmes 6.1 et 6.2 diminuent très rapidement avec la taille de l'échantillon.

Ainsi, dans le cas d'une application de la méthode de quasi-Monte-Carlo (17) utilisant une séquence de taille limitée en dimension élevée, les majorations explicites fournies par ces théorèmes ne fournissent aucune information utile pour évaluer la qualité de l'ensemble de points retenu. Il n'en reste pas moins qu'un bon candidat doit nécessairement présenter une discrépance réduite (voir chapitre 3). Aussi, ne disposant d'aucun résultat théorique capable de caractériser la discrépance d'une séquence de longueur raisonnable en dimension élevée, on voit l'intérêt pratique d'algorithmes permettant de calculer ou de majorer suffisamment précisément cette grandeur.

s	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
b	2	3	5	5	7	7	11	11	11	11	13	13	17	17	17	17	19	19	23
$m_{\min}$	2	2	3	4	4	5	6	7	8	9	9	10	11	12	13	13	14	15	16
$n_{\min}$	4	9	125	625	$2\,401$	$16\,807$	$1.7 \cdot 10^{6}$	$1.9 \cdot 10^{7}$	$2.1 \cdot 10^{8}$	$2.3 \cdot 10^{9}$	$1.0 \cdot 10^{10}$	$1.3 \cdot 10^{11}$	$3.4 \cdot 10^{13}$	$5.8 \cdot 10^{14}$	$9.9 \cdot 10^{15}$	$9.9 \cdot 10^{15}$	$7.9 \cdot 10^{17}$	$1.5 \cdot 10^{19}$	$6.1 \cdot 10^{21}$

TAB. 6.1. Taille minimale $n_{\min} = b^{m_{\min}}$ d'un $(0, m, s)$-réseau de Faure en base b (où b est le plus petit nombre premier supérieur ou égal à s) pour laquelle le théorème 6.1 fournit une majoration pour la discrépance de valeur inférieure à 1.

s	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
b	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
$t_{s,b}$	0	0	1	2	3	4	5	6	8	9	10	11	13	15	15	18	19	19	21
$m_{\min}$	1	2	5	7	9	11	13	15	18	20	22	24	27	30	31	35	37	38	41
$n_{\min}$	3	9	32	128	512	$2\,048$	$8\,192$	$32\,768$	$262\,144$	$1.0 \cdot 10^{6}$	$4.2 \cdot 10^{6}$	$1.7 \cdot 10^{7}$	$1.3 \cdot 10^{8}$	$1.0 \cdot 10^{9}$	$2.1 \cdot 10^{9}$	$3.4 \cdot 10^{10}$	$1.3 \cdot 10^{11}$	$2.7 \cdot 10^{11}$	$2.1 \cdot 10^{12}$

TAB. 6.2. Taille minimale $n_{\min} = b^{m_{\min}}$ d'un $(t_{s,b}, m, s)$-réseau de Niederreiter et Xing en base b pour laquelle le théorème 6.1 fournit une majoration pour la discrépance de valeur inférieure à 1.

6.2 Les bornes inférieures

Contrairement au cas des majorations, le calcul de bornes inférieures pour la discrépance est une tâche relativement facile. En effet, pour une séquence x de n points dans $\bar{I}^s$ et un intervalle quelconque $P \in \mathcal{I}_s^*$, on a la minoration

$$\left| \frac{A(P, x)}{n} - \lambda(P) \right| \leq D_n^*(x).$$

Pour une séquence x de n points provenant d'un générateur pseudo-aléatoire ou d'une suite à discrépance faible, l'heuristique consistant à prendre le maximum des valeurs $|A(P, x)/n - \lambda(P)|$ pour quelques centaines d'intervalles P choisis aléatoirement fournit généralement une excellente borne inférieure sur la discrépance $D_n^*(x)$. Notons cependant que lorsque x comprend très peu de points, le facteur chance peut devenir prépondérant et donc conduire à des résultats de qualité très variable.

Par définition, la discrépance carrée moyenne $T_n^*(x)$ est une borne inférieure sur la discrépance. Elle a l'avantage d'être facile à calculer (voir section 1.3.3), mais quelques expériences numériques suffisent à montrer qu'elle est de qualité très médiocre. Notons que dans certains cas, une borne inférieure explicite peut être obtenue par considération de l'erreur de discrétisation :

THÉORÈME 6.3 (Niederreiter [Nie92]) *Si* $x \subset I^s$ *est une séquence pour laquelle les coordonnées de chaque point sont des nombres rationnels de la forme* c/d *avec* $c \in \{0, \ldots, d-1\}$*, alors on a*

$$D_n^*(x) \geq 1 - \left(\frac{d-1}{d}\right)^s .$$

Cette minoration concerne toutes les suites à discrépance faible décrites dans le chapitre 4, mais malheureusement, elle s'avère particulièrement mauvaise. D'autre part, une adaptation de la méthode du recuit simulé (voir Aarts et Korst [AK89]) a été proposée par Winker et Fang [WF97] pour le calcul de bornes inférieures pour la discrépance. Fondamentalement, cette technique consiste à n'explorer qu'une partie de la grille correspondant à l'ensemble $\mathcal{Q}$ de la discrétisation de Niederreiter (21). Avec la notation de la section 5.1, le recuit en question utilise

$$V(q) = \{p \in \mathcal{Q} : |p_j - q_j| \leq k, \forall j \in \{1, \ldots, s\}\} \subset \mathcal{Q}$$

comme définition du voisinage d'un point $q \in \mathcal{Q}$, où $k \in \mathbb{N}^*$ est un paramètre fixé de manière arbitraire. Cette approche semble fournir d'excellents résultats. Pour terminer, signalons qu'une nouvelle technique de minoration particulièrement efficace est proposée dans la section 8.4 (elle repose sur des heuristiques présentées dans la section 8.3.2).

CHAPITRE 7

Une approche par décomposition du cube unité

Une méthode permettant de construire des intervalles arbitrairement petits contenant la valeur exacte de la discrépance d'une séquence quelconque dans le cube unité$\bar{I}^s$ est proposée dans la section 7.1. Cette approche repose sur la considération d'une partition de I^s en intervalles. Deux méthodes particulières de décomposition et des techniques de mise en œuvre spécifiques sont présentées dans les sections 7.2 et 7.4. Bien que les algorithmes obtenus ne soient pas polynomiaux, les expériences numériques des sections 7.3 et 7.5 montrent que cette approche permet d'établir des bornes pour la discrépance de séquences dans des cas totalement inaccessibles jusqu'alors. Pour l'essentiel, le contenu de ce chapitre est une version remaniée de nos articles [Thi00] et [Thi01].

7.1 Un intervalle pour la discrépance

Soit x une séquence de n points dans le cube unité $\bar{I}^s$. On veut construire un intervalle pour la discrépance $D_n^*(x)$. Rappelons que l'on associe à tout $P = \prod_{i=1}^s [\alpha_i, \beta_i) \in \mathcal{I}_s$ les deux intervalles $P^- = \prod_{i=1}^s [0, \alpha_i)$ et $P^+ = \prod_{i=1}^s [0, \beta_i)$. Pour toute partition finie $\mathcal{P}$ de I^s en intervalles, on définit

$$B(\mathcal{P}, x) = \max_{P \in \mathcal{P}} \max \left\{ \frac{A(P^+, x)}{n} - \lambda(P^-),\, \lambda(P^+) - \frac{A(P^-, x)}{n} \right\}. \tag{22}$$

Il est clair que $B(\mathcal{P}, x) \leq 1$ pour toute séquence x et tout choix de $\mathcal{P}$. On montre que cette grandeur est une borne supérieure pour la discrépance $D_n^*(x)$:

THÉORÈME 7.1 *Pour toute séquence $x = \{x^1, \ldots, x^n\} \subset \bar{I}^s$ et toute partition finie $\mathcal{P}$ de I^s en intervalles de la forme $\prod_{i=1}^s [\alpha_i, \beta_i) \in \mathcal{I}_s$, on a*

$$D_n^*(x) \leq B(\mathcal{P}, x).$$

PREUVE.

$$\begin{aligned} D_n^*(x) &= \sup_{P \in \mathcal{I}_s^*} \left| \frac{A(P, x)}{n} - \lambda(P) \right| \\ &= \max_{P \in \mathcal{P}} \sup_{\substack{P^- \subset Q \subset P^+ \\ Q \in \mathcal{I}_s^*}} \left\{ \frac{A(Q, x)}{n} - \lambda(Q),\, \lambda(Q) - \frac{A(Q, x)}{n} \right\} \\ &\leq \max_{P \in \mathcal{P}} \max \left\{ \frac{A(P^+, x)}{n} - \lambda(P^-),\, \lambda(P^+) - \frac{A(P^-, x)}{n} \right\}. \end{aligned}$$

$\square$

De même, pour toute partition finie $\mathcal{P}$ de I^s en intervalles,

$$C(\mathcal{P}, x) = \max_{P \in \mathcal{P}} \max \left\{ \left| \frac{A(P^-, x)}{n} - \lambda(P^-) \right|, \left| \frac{A(P^+, x)}{n} - \lambda(P^+) \right| \right\} \tag{23}$$

est une borne inférieure pour la discrépance de x. On vient donc d'établir l'intervalle

$$C(\mathcal{P}, x) \leq D_n^*(x) \leq B(\mathcal{P}, x). \tag{24}$$

DÉFINITION 7.2 On définit le *poids* $W(P)$ d'un intervalle $P = \prod_{i=1}^{s}[\alpha_i^P, \beta_i^P) \in \mathcal{I}_s$ comme la différence entre les volumes de $P^+ = \prod_{i=1}^{s}[0, \beta_i^P)$ et de $P^- = \prod_{i=1}^{s}[0, \alpha_i^P)$:

$$W(P) = \lambda(P^+) - \lambda(P^-) = \prod_{i=1}^{s} \beta_i^P - \prod_{i=1}^{s} \alpha_i^P.$$

De même, pour une partition finie $\mathcal{P}$ de I^s en intervalles, on pose

$$W(\mathcal{P}) = \max_{P \in \mathcal{P}} W(P). \tag{25}$$

Bien évidemment, on aimerait que l'intervalle (24) associé à une partition $\mathcal{P}$ de $\bar{I}^s$ soit aussi étroit que possible. Le résultat suivant montre que sa taille ne peut pas dépasser $W(\mathcal{P})$.

THÉORÈME 7.3 *Pour toute séquence $x \subset \bar{I}^s$ et toute partition finie $\mathcal{P}$ de I^s en intervalles, on a*

$$B(\mathcal{P}, x) - C(\mathcal{P}, x) \leq W(\mathcal{P}).$$

PREUVE. On observe tout d'abord que pour tout intervalle $P \in \mathcal{P}$, on a

$$\begin{aligned}
\frac{A(P^+, x)}{n} - \lambda(P^-) &= \frac{A(P^+, x)}{n} - \lambda(P^+) + [\lambda(P^+) - \lambda(P^-)] \\
&\leq \frac{A(P^+, x)}{n} - \lambda(P^+) + W(\mathcal{P})
\end{aligned}$$

et

$$\begin{aligned}
\lambda(P^+) - \frac{A(P^-, x)}{n} &= [\lambda(P^+) - \lambda(P^-)] + \lambda(P^-) - \frac{A(P^-, x)}{n} \\
&\leq \lambda(P^-) - \frac{A(P^-, x)}{n} + W(\mathcal{P}).
\end{aligned}$$

On en déduit que

$$\begin{aligned}
B(\mathcal{P}, x) &= \max_{P \in \mathcal{P}} \max \left\{ \frac{A(P^+, x)}{n} - \lambda(P^-),\, \lambda(P^+) - \frac{A(P^-, x)}{n} \right\} \\
&\leq \max_{P \in \mathcal{P}} \max \left\{ \frac{A(P^+, x)}{n} - \lambda(P^+),\, \lambda(P^-) - \frac{A(P^-, x)}{n} \right\} + W(\mathcal{P}) \\
&\leq C(\mathcal{P}, x) + W(\mathcal{P}).
\end{aligned}$$

□

Il est clair que la valeur de $W(\mathcal{P})$ est indépendante de la séquence x considérée. Ainsi, pour obtenir un intervalle (24) de petite taille, deux options sont envisageables :

- ▷ choisir un paramètre de précision $\varepsilon \in (0, 1)$ et construire une partition $\mathcal{P}$ telle que $W(\mathcal{P}) = \varepsilon$ (cette approche est envisagée dans la section 7.4) ;
- ▷ chercher la partition $\mathcal{P}$ minimisant $W(\mathcal{P})$ sous certaines contraintes (le cas particulier des grilles extensibles de cardinalité fixée est considéré ci-dessous).

7.2 Les grilles extensibles

Soit un entier $k \geq 2$ et un ensemble de s séquences croissantes $z_{s,k} = \{z^1, \ldots , z^s\}$ donné par

$$z^i = (z^i_0, \ldots , z^i_k)$$

et

$$0 = z^i_0 < z^i_1 < \cdots < z^i_k = 1, \text{ pour tout } i \in \{1, \ldots , s\}.$$

De manière naturelle, on associe à tout vecteur $a = (a_1, \ldots , a_s) \in \{1, \ldots , k\}^s$ l'intervalle

$$P_a = \prod_{i=1}^{s} \left[z^i_{a_i-1}, z^i_{a_i}\right) .$$

Ainsi, $z_{s,k}$ induit une partition du cube unité I^s ayant la forme d'une *grille extensible* de cardinalité k^s

$$\mathcal{P}_{z_{s,k}} = \{P_a : a \in \{1, \ldots , k\}^s\} .$$

Dans ce cas particulier, les bornes (22) et (23) s'écrivent

$$B(\mathcal{P}_{z_{s,k}}, x) = \max_{a\in\{1,\ldots,k\}^s} \max \left\{ \frac{A(P^+_a, x)}{n} - \lambda(P^-_a),\ \lambda(P^+_a) - \frac{A(P^-_a, x)}{n} \right\}$$

et

$$\begin{aligned} C(\mathcal{P}_{z_{s,k}}, x) &= \max_{a\in\{1,\ldots,k\}^s} \max \left\{ \left| \frac{A(P^-_a, x)}{n} - \lambda(P^-_a) \right| , \left| \frac{A(P^+_a, x)}{n} - \lambda(P^+_a) \right| \right\} \\ &= \max_{a\in\{1,\ldots,k\}^s} \left| \frac{A(P^+_a, x)}{n} - \lambda(P^+_a) \right| . \end{aligned}$$

Comme nous l'avons déjà mentionné au chapitre 6, dans le cadre de la construction d'un intervalle pour la discrépance, le plus difficile est en général d'obtenir une bonne borne supérieure. Il se trouve que, pour un choix donné de $z_{s,k}$, il est facile d'établir des bornes inférieures sur la valeur de la borne supérieure $B(\mathcal{P}_{z_{s,k}}, x)$:

▷ Pour $a = (1, k, \ldots , k)$, on a $A(P^-_a, x) = 0$. On en déduit que

$$B(\mathcal{P}_{z_{s,k}}, x) \geq \lambda(P^+_a) - \frac{A(P^-_a, x)}{n} = z^1_1.$$

De la même manière, on obtient

$$B(\mathcal{P}_{z_{s,k}}, x) \geq z^i_1, \text{ pour tout } i \in \{1, \ldots , s\}. \tag{26}$$

▷ Pour $a = (k, k, \ldots , k)$, on a $A(P^+_a, x) = n$ et par conséquent

$$B(\mathcal{P}_{z_{s,k}}, x) \geq \frac{A(P^+_a, x)}{n} - \lambda(P^-_a) = 1 - \prod_{i=1}^{s} z^i_{k-1}.$$

On note que ces bornes inférieures sont indépendantes de x. En d'autres termes, bien que la discrépance d'une séquence puisse être arbitrairement petite, ces valeurs constituent un seuil minimal incompressible pour la valeur de $B(\mathcal{P}_{z_{s,k}}, x)$. En guise d'introduction, examinons la qualité de cette majoration pour quelques séquences dont la discrépance est connue :

EXEMPLE 7.4 On considère le cas des quelques $(0, m, 2)$-réseaux de Faure en base 2 dont la discrépance a été calculée au chapitre 5 (voir table 5.1). Pour ces séquences, la partition définie par

$$k = 100 \quad \text{et} \quad z_j^1 = z_j^2 = 1 - \left(1 - \frac{j}{100}\right)^2, \text{ pour tout } j \in \{0, \dots, 100\}$$

mène aux bornes supérieures données dans la table 7.1. Un intervalle ayant un poids inférieur lorsqu'il est translaté près de l'origine, cette grille (voir figure 7.1) semble constituer un choix raisonnable pour aboutir à une partition présentant une petite valeur de $W(\mathcal{P}_{z_{2,100}})$. Pour $m \in \{1, \dots, 7\}$, les bornes obtenues sont proches des valeurs exactes de la table 5.1 et meilleures que les majorations fournies par le théorème 6.1 (voir figure 6.1). Par contre, pour $m \geq 8$, la valeur de $B(\mathcal{P}_{z_{2,100}}, x)$ s'écarte de la discrépance et semble tendre vers 0.020020. Ce fait n'est guère surprenant au vu de la relation (26). En effet, cette dernière implique que $B(\mathcal{P}_{z_{2,100}}, x) \geq z_1^1 = 0.0199$ pour toute séquence x. Ce phénomène est symptomatique de cette méthode. Il se produit pour un seuil qui grandit avec s et décroît avec k.

m	1	2	3	4	5	6	7	8	9	10	11	12	13
$n = 2^m$	2	4	8	16	32	64	128	256	512	1 024	2 048	4 096	8 192
$B(\mathcal{P}_{z_{2,100}}, x)$	0.754083	0.452548	0.327548	0.183537	0.102539	0.060025	0.036471	0.026929	0.022328	0.020508	0.020237	0.020020	0.020020

TAB. 7.1. Borne $B(\mathcal{P}_{z_{2,100}}, x)$ pour la discrépance de quelques $(0, m, 2)$-réseaux de Faure en base 2.

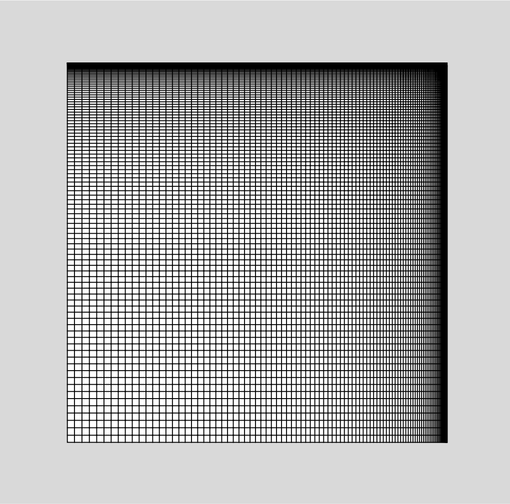


FIG. 7.1. Une partition du carré unité en $10\,000$ intervalles.

7.2.1 Construction de la grille. Intuitivement, plus la discrétisation est fine (*i.e.* plus k est grand), meilleures sont les perspectives d'obtention de bornes de bonne qualité. En fin de compte, le choix de k revient à trouver un compromis entre la précision désirée et le temps de calcul à disposition. Pour l'instant, on suppose k fixé et l'on s'intéresse au problème de la construction d'une grille minimisant la largeur maximale de l'intervalle pour la discrépance (24). Par le théorème 7.3, il s'agit donc de trouver une partition $\mathcal{P}_{z_{s,k}}$ de poids minimum. Sachant que

$$W(\mathcal{P}_{z_{s,k}}) = \max_{P \in \mathcal{P}_{z_{s,k}}} W(P) = \max_{a \in \{1,\dots,k\}^s} W(P_a),$$

où $P_a = \prod_{i=1}^s \left[z^i_{a_i-1}, z^i_{a_i}\right)$, le problème revient à résoudre le programme mathématique non-linéaire

$$\begin{aligned} (27) \qquad G^s_k &= \min_{z_{s,k}} \max_{a \in \{1,\dots,k\}^s} \left\{ \prod_{i=1}^s z^i_{a_i} - \prod_{i=1}^s z^i_{a_i-1} \right\} \\ \text{s.c.} \quad & 0 = z^i_0 < \dots < z^i_k = 1, \text{ pour tout } i \in \{1,\dots,s\}. \end{aligned}$$

Les cas particuliers suivants sont évidents :

▷ En dimension 1, la solution optimale de (27) est donnée par

$$G^1_k = \frac{1}{k}, \text{ où } z^1_j = \frac{j}{k} \text{ pour tout } j \in \{0,\dots,k\}.$$

▷ Pour $k = 2$, il est facile de remplacer le maximum sur 2^s termes par un maximum sur $s+1$ termes dans le programme (27). En effet, il suffit d'observer que pour tout choix de $d \in \{1,\dots,s\}$ et de $a = (a_1,\dots,a_s) \in \{1,2\}^s$ avec $a_d = 1$, on a

$$\prod_{i=1}^s z^i_{a_i} - \prod_{i=1}^s z^i_{a_i-1} = \prod_{i=1}^s z^i_{a_i} \leq z^d_1.$$

Cette inégalité est serrée pour tout $a = (a_1,\dots,a_s)$ avec $a_d = 1$ et $a_i = 2$, pour tout $i \neq d$. Finalement, le dernier candidat s'obtient en prenant $a = (2,\dots,2)$. En fin de compte, on a

$$\begin{aligned} G^s_2 &= \min_{z^1_1,\dots,z^s_1} \max \left\{ z^1_1,\dots,z^s_1, 1 - \prod_{i=1}^s z^i_1 \right\} \\ \text{s.c.} \quad & 0 < z^i_1 < 1, \text{ pour tout } i \in \{1,\dots,s\}. \end{aligned}$$

La solution de ce programme est

$$G^s_2 = z^i_1 = z, \text{ pour tout } i \in \{1,\dots,s\},$$

où z est l'unique solution de l'équation $z = 1 - z^s$ sur l'intervalle $[0,1]$.

En dehors de ces cas particuliers, le programme (27) semble défier toute approche analytique. Il a toutefois été possible d'établir une condition de concavité caractérisant une de ses solutions optimales :

THÉORÈME 7.5 *Le programme mathématique* (27) *possède une solution* $z_{s,k}$ *pour laquelle*

$$(28) \qquad z^i_j - z^i_{j-1} \geq z^i_{j+1} - z^i_j, \textit{ pour tout } i \in \{1,\dots,s\} \textit{ et } j \in \{1,\dots,k-1\}.$$

PREUVE. Soit $z_{s,k}$ une solution admissible du programme (27) telle qu'il existe $d \in \{1,\dots,s\}$ et $p \in \{1,\dots,k-1\}$ pour lesquels la condition (28) n'est pas satisfaite. On construit alors une nouvelle solution admissible $\bar{z}_{s,k}$ de la manière suivante :

$$\bar{z}^i_j = \begin{cases} z^i_{j-1} + z^i_{j+1} - z^i_j & \text{si } (i,j) = (d,p), \\ z^i_j & \text{sinon.} \end{cases}$$

En itérant ce processus, on finit par obtenir une solution admissible pour laquelle la condition (28) est satisfaite. On achève la preuve en démontrant que cette transformation ne détériore pas la qualité de la solution, c'est-à-dire que

$$\max_{a\in\{1,\dots,k\}^s}\left\{\prod_{i=1}^{s}\bar{z}^i_{a_i}-\prod_{i=1}^{s}\bar{z}^i_{a_i-1}\right\}\leq\max_{a\in\{1,\dots,k\}^s}\left\{\prod_{i=1}^{s}z^i_{a_i}-\prod_{i=1}^{s}z^i_{a_i-1}\right\}.$$

Il suffit de prouver que pour tout $a\in\{1,\dots,k\}^s$,

$$\prod_{i=1}^{s}\bar{z}^i_{a_i}-\prod_{i=1}^{s}\bar{z}^i_{a_i-1}\leq\max_{c\in\{1,\dots,k\}^s}\left\{\prod_{i=1}^{s}z^i_{c_i}-\prod_{i=1}^{s}z^i_{c_i-1}\right\}.$$

On distingue trois cas :

▷ Si $a_d\notin\{p,p+1\}$, on a

$$\prod_{i=1}^{s}\bar{z}^i_{a_i}-\prod_{i=1}^{s}\bar{z}^i_{a_i-1}=\prod_{i=1}^{s}z^i_{a_i}-\prod_{i=1}^{s}z^i_{a_i-1}.$$

▷ Si $a_d=p+1$, on a

$$\begin{aligned}\prod_{i=1}^{s}\bar{z}^i_{a_i}-\prod_{i=1}^{s}\bar{z}^i_{a_i-1}&=\prod_{i=1}^{s}z^i_{a_i}-\left(z^d_{p-1}+z^d_{p+1}-z^d_p\right)\prod_{i\neq d}z^i_{a_i-1}\\&<\prod_{i=1}^{s}z^i_{a_i}-\prod_{i=1}^{s}z^i_{a_i-1}.\end{aligned}$$

▷ Si $a_d=p$, on a

$$\begin{aligned}\prod_{i=1}^{s}\bar{z}^i_{a_i}-\prod_{i=1}^{s}\bar{z}^i_{a_i-1}&=\left(z^d_{p-1}+z^d_{p+1}-z^d_p\right)\prod_{i\neq d}z^i_{a_i}-\prod_{i=1}^{s}z^i_{a_i-1}\\&=\left(z^d_{p+1}-z^d_p\right)\prod_{i\neq d}z^i_{a_i}+z^d_{p-1}\left(\prod_{i\neq d}z^i_{a_i}-\prod_{i\neq d}z^i_{a_i-1}\right)\\&<\left(z^d_{p+1}-z^d_p\right)\prod_{i\neq d}z^i_{a_i}+z^d_{p}\left(\prod_{i\neq d}z^i_{a_i}-\prod_{i\neq d}z^i_{a_i-1}\right)\\&=z^d_{p+1}\prod_{i\neq d}z^i_{a_i}-z^d_p\prod_{i\neq d}z^i_{a_i-1}\\&=\prod_{i=1}^{s}z^i_{c_i}-\prod_{i=1}^{s}z^i_{c_i-1},\text{ où }c_i=\begin{cases}p+1&\text{si }i=d,\\a_i&\text{sinon.}\end{cases}\end{aligned}$$

□

Le programme mathématique (27) étant difficilement abordable, nous restreignons notre étude au cas symétrique où les s séquences $\{z^1,\dots,z^s\}$ sont identiques. On obtient donc le nouveau problème

$$\begin{aligned}M^s_k&=\min\max_{a\in\{1,\dots,k\}^s}\left\{\prod_{i=1}^{s}z_{a_i}-\prod_{i=1}^{s}z_{a_i-1}\right\}\\&\text{s.c.}\quad 0=z_0<\cdots<z_k=1.\end{aligned}\tag{29}$$

On remarque que la fonction objectif

$$f(z) = \max_{a\in\{1,\dots,k\}^s} \left\{ \prod_{i=1}^{s} z_{a_i} - \prod_{i=1}^{s} z_{a_i-1} \right\} \tag{30}$$

de ce programme n'est pas convexe. En effet, pour $s = k = 2$, on a

$$f(z) = \max\left\{z_1, 1 - z_1^2\right\},$$

mais en posant $z_1 = 0.5$ et $\bar{z}_1 = 0.6$, on obtient

$$f(1/2\, z_1 + 1/2\, \bar{z}_1) = 0.6975 > 0.695 = 1/2\,(0.75 + 0.64) = 1/2\, f(z_1) + 1/2\, f(\bar{z}_1).$$

Mentionnons au passage que, dans ce cas, la solution de (29) est unique et qu'il s'agit du nombre d'or

$$M_2^2 = \frac{\sqrt{5}-1}{2}.$$

On conjecture que, pour toutes valeurs de s et k, chacun des programmes (27) et (29) possède une unique solution et que celles-ci coïncident (les résultats de toutes nos expérimentations numériques abondent dans ce sens). Par ailleurs, considérant la symétrie de la fonction objectif (30), le problème (29) est équivalent au programme mathématique simplifié

$$\begin{aligned} M_k^s \quad &= \quad \min \max_{\substack{a\in\{1,\dots,k\}^s \\ a_1\le\cdots\le a_s}} \left\{ \prod_{i=1}^{s} z_{a_i} - \prod_{i=1}^{s} z_{a_i-1} \right\} \\ &\text{s.c.} \quad 0 = z_0 < \cdots < z_k = 1. \end{aligned} \tag{31}$$

Que ce problème possède une ou plusieurs solutions, le théorème 7.5 nous assure qu'il en existe au moins une pour laquelle les conditions suivantes sont satisfaites :

$$\begin{array}{ccccc} \dfrac{1}{k} & \le & z_1 & \le & 1, \\ \dfrac{1-z_1}{k-1} & \le & z_2 - z_1 & \le & \min\{z_1, 1-z_1\}, \\ \dfrac{1-z_2}{k-2} & \le & z_3 - z_2 & \le & \min\{z_2 - z_1, 1-z_2\}, \\ \vdots & & \vdots & & \vdots \\ \dfrac{1-z_{k-2}}{2} & \le & z_{k-1} - z_{k-2} & \le & \min\{z_{k-2} - z_{k-3}, 1-z_{k-2}\}. \end{array} \tag{32}$$

De plus, si l'on dispose d'une borne supérieure $M < 1$ pour la valeur d'une solution optimale du programme (31), alors la relation (26) implique que la première contrainte ci-dessus peut être renforcée en la remplaçant par

$$\frac{1}{k} \le z_1 \le M. \tag{33}$$

Ajoutées au programme mathématique (31), les contraintes (32) et (33) permettent de réduire considérablement la taille du domaine de minimisation. Une méthode de recherche locale ou aléatoire comme l'algorithme 7.1 peut alors être utilisée afin d'obtenir une solution approchée. Il est clair que chaque amélioration induit, par le biais de la contrainte (33), une nouvelle contraction du domaine.

Bien évidemment, la partie délicate de l'algorithme 7.1 est la potentielle énumération de l'ensemble $\mathcal{A}$ dans la boucle constituée des lignes 10 à 12. Ce cas pathologique peut être évité pour la quasi-totalité des solutions admissibles $(y_0, \dots, y_k)$ envisagées par une politique adéquate de sélection de l'élément $a \in \mathcal{A}$ à la ligne 11. Expérimentalement, pour une partition donnée, l'intervalle P_a de poids maximum

Algorithme 7.1 Approximation de M_k^s

```
Donnée : $s, k$
Résultat : $\hat{M}_k^s, \{z_0, \dots, z_k\}$
 1: $y_0 \leftarrow 0.0$
 2: $y_k \leftarrow 1.0$
 3: $\hat{M}_k^s \leftarrow 1.0$
 4: tant que condition d'arrêt pas satisfaite faire
 5:   $y_1 \leftarrow U\left(\frac{1}{k}, \hat{M}_k^s\right)$
 6:   pour $i$ de $2$ à $k-1$ faire
 7:     $y_i \leftarrow y_{i-1} + U\left(\frac{1-y_{i-1}}{k-i+1}, \min\{y_{i-1}-y_{i-2}, 1-y_{i-1}\}\right)$
 8:   $m \leftarrow 0.0$
 9:   $\mathcal{A} \leftarrow \{a \in \{1, \dots, k\}^s : a_1 \leq \cdots \leq a_s\}$
10:   tant que $\mathcal{A} \neq \varnothing$ et $m < \hat{M}_k^s$ faire
11:     choisir et retirer un $s$-uple $a = (a_1, \dots, a_s)$ de $\mathcal{A}$
12:     $m \leftarrow \max\left\{m, \prod_{i=1}^{s} y_{a_i} - \prod_{i=1}^{s} y_{a_i-1}\right\}$
13:   si $m < \hat{M}_k^s$ alors
14:     $z_i \leftarrow y_i$ pour tout $i \in \{0, \dots, k\}$
15:     $\hat{M}_k^s \leftarrow m$
```

se trouve presque toujours sur la diagonale centrale (pour $a = (i, \dots, i)$ avec $1 \leq i \leq k$), sur une arête éloignée de l'origine (pour $a = (i, k, \dots, k)$ avec $1 \leq i \leq k$) ou près du « coin supérieur-droit » $(k, \dots, k)$. Empiriquement, en commençant l'énumération de $\mathcal{A}$ par ces régions, la quasi-totalité des « mauvais » candidats $(y_0, \dots, y_k)$ sont éliminés (par le biais du test $m < \hat{M}_k^s$ de la ligne 10) au cours des toutes premières itérations de la boucle en question.

(s,k)	$(7,15)$	$(8,11)$	$(9,8)$	$(10,6)$	$(11,5)$	$(12,4)$	$(13,4)$	$(14,3)$	$(15,3)$	$(16,3)$	$(17,2)$	$(18,2)$	$(19,2)$	$(20,2)$
$\hat{M}_k^s$	0.13198	0.18456	0.25796	0.34733	0.41885	0.51901	0.52634	0.67236	0.67942	0.68599	0.88191	0.88625	0.89023	0.89390

TAB. 7.2. Valeur approchée de la solution de (31) pour quelques valeurs de s et k.

Bien que cet algorithme soit rudimentaire, ses performances sont tout à fait honorables en pratique. La table 7.2 contient la meilleure approximation $\hat{M}_k^s$ de M_k^s obtenue pour quelques valeurs de s et k. Malheureusement, la figure 7.2 montre que $\hat{M}_k^s$ décroît très lentement avec k. Notons qu'une approche de décomposition plus efficace du cube unité menant à des partitions de cardinalité inférieure pour un poids donné est présentée dans la section 7.4.

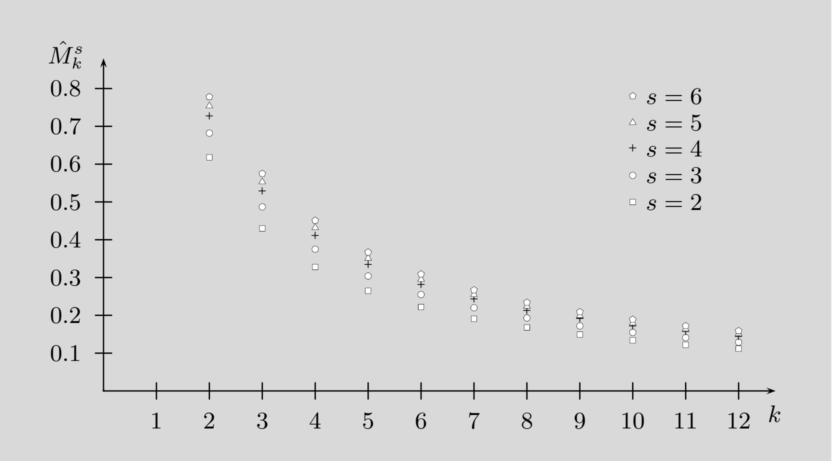


FIG. 7.2. Valeur approchée de la solution de (31) pour quelques valeurs de s et k.

7.2.2 Décompte des points. Le calcul effectif des bornes

$$C(\mathcal{P}_{z_{s,k}}, x) = \max_{a \in \{1,\dots,k\}^s} \left| \frac{A(P_a^+, x)}{n} - \lambda(P_a^+) \right|$$

et

$$B(\mathcal{P}_{z_{s,k}}, x) = \max_{a \in \{1,\dots,k\}^s} \max \left\{ \frac{A(P_a^+, x)}{n} - \lambda(P_a^-),\ \lambda(P_a^+) - \frac{A(P_a^-, x)}{n} \right\}$$

pour la discrépance nécessite la détermination de $A(P_a^+, x)$ pour tout $a \in \{1, \dots, k\}^s$. La méthode directe consistant à compter les points de la séquence contenus dans chaque intervalle P_a^+ possède une complexité $O(nk^s)$. Il est également possible d'effectuer cette tâche en $O(ns \log k + k^{2s})$ en exploitant le fait que $\mathcal{P}_{z_{s,k}}$ est une grille. En effet, pour tout point de la séquence, une quête dichotomique sur chaque composante suffit à déterminer, en un temps $O(s \log k)$, dans quel intervalle P_a il se trouve. Ayant ainsi obtenu les valeurs de $A(P_a, x)$ pour tout $a \in \{1, \dots, k\}^s$, il reste à calculer

$$A(P_a^+, x) = \sum_{\substack{c \in \{1,\dots,k\}^s \\ c_i \le a_i,\, \forall i \in \{1,\dots,s\}}} A(P_c, x), \quad \text{pour tout } a \in \{1, \dots, k\}^s. \tag{34}$$

On note que la complexité $O(k^{2s})$ de ce dernier pas est indépendante de la taille n de la séquence. Il est cependant possible d'accélérer le processus en utilisant de manière adéquate le théorème 7.6. On ramène alors l'effort total à $O(ns \log k + 2^s k^s)$. Il convient de souligner le fait que cette complexité présuppose la disponibilité d'un espace-mémoire important pour le stockage des valeurs de $A(P_a, x)$.

THÉORÈME 7.6 *Pour tout* $a = (a_1, \dots, a_s) \in \{1, \dots, k\}^s$*, on a*

$$A(P_a^+, x) = A(P_a, x) + \sum_{\substack{v \in \{0,1\}^s \\ v \neq (0,\dots,0)}} (-1)^{1+\sum_{i=1}^s v_i} A\left(P_{a-v}^+, x\right). \tag{35}$$

PREUVE. On remarque tout d'abord que pour tout triplet d'ensembles finis E_1, E_2 et E_3, on a

$$\begin{aligned} |E_1 \cup E_2| &= |E_1| + |E_2| \\ &\quad - |E_1 \cap E_2| \end{aligned}$$

et

$$\begin{aligned} |E_1 \cup E_2 \cup E_3| &= |E_1| + |E_2| + |E_3| \\ &\quad -|E_1 \cap E_2| - |E_1 \cap E_3| - |E_2 \cap E_3| \\ &\quad +|E_1 \cap E_2 \cap E_3|. \end{aligned}$$

De même, si $E_1, \ldots, E_s$ sont s ensembles finis, on obtient la généralisation

$$\begin{aligned} |E_1 \cup \cdots \cup E_s| &= \sum_{i=1}^{s} |E_i| \\ &\quad - \sum_{i \neq j} |E_i \cap E_j| \\ &\qquad \vdots \\ &\quad +(-1)^s \sum_{i=1}^{s} \left| \bigcap_{j \neq i} E_j \right| \\ &\quad -(-1)^s \left| \bigcap_{i=1}^{s} E_i \right|. \end{aligned} \tag{36}$$

Cette formule se démontre facilement par induction (voir Hall [Hal86]). D'autre part, on a

$$\begin{aligned} A(P_a^+, x) &= A(P_a, x) + A(P_a^+ \setminus P_a, x) \\ &= A(P_a, x) + A\left(\bigcup_{i=1}^{s} P_{a-e_d}^+, x \right), \end{aligned}$$

où e_d désigne le d^e vecteur unité de la base canonique de $\mathbb{R}^s$. Pour prouver le théorème, il suffit d'utiliser l'expression (36) avec

$$E_d = x \cap P_{a-e_d}^+, \text{ pour tout } d \in \{1, \ldots, s\}.$$

□

Pour un vecteur $a \in \{1, \ldots, k\}^s$ donné, le calcul de $A(P_a^+, x)$ à l'aide de l'expression (34) nécessite la considération de $O(k^s)$ termes, alors que dans (35), la somme ne porte que sur 2^s éléments. Cette amélioration implique toutefois une difficulté supplémentaire. En effet, avec la formule (35), les k^s différentes valeurs de $A(P_a^+, x)$ ne sont pas calculables dans n'importe quel ordre. Plus précisément, la détermination de $A(P_a^+, x)$ requiert que $A(P_{a-v}^+, x)$ soit connu pour tout vecteur $v \in \{0, 1\}^s$ non nul.

L'algorithme 7.2 résout ce problème en énumérant l'ensemble des vecteurs $a \in \{1, \ldots, k\}^s$ dans un ordre non décroissant de la somme de leurs composantes. En effet, un appel à la procédure récursive `distribuer`(c, r) génère l'ensemble des c-uples $(a_{s-c+1}, \ldots, a_s)$ sommant à r et constitués d'entiers compris entre 1 et k. Chaque fois qu'un nouveau vecteur a est obtenu (lorsque $c = 1$ à la ligne 7), la valeur correspondante de $A(P_a^+, x)$ est calculée et les bornes sur la discrépance sont mises à jour en conséquence (vu l'ordre de génération, on a l'assurance qu'en ligne 11, $A(P_a^-, x)$ a été préalablement établi). Tel qu'il est présenté, cet algorithme implique une occupation-mémoire $O(k^s)$. Il est sans doute possible d'imaginer une mise en œuvre plus économique (et sophistiquée) dans laquelle les seules valeurs de $A(P_a, x)$ et de $A(P_a^+, x)$ stockées à un instant donné constituent un petit sous-ensemble de celles impliquées dans les applications ultérieures de l'expression (35).

Notons que, à la ligne 9 de l'algorithme 7.2, le calcul direct de $A(P_a^+, x)$ à l'aide de la formule (35) implique un effort en $O(2^s s)$. L'algorithme 7.3 permet, par l'utilisation d'un code de Gray en base 2 (voir section 4.8.1), de réduire cette complexité à $O(2^s)$. En effet, cet artifice permet d'énumérer l'ensemble

Algorithme 7.2 Calcul de l'intervalle pour la discrépance

```
Donnée : $s, k, x, z_{s,k}$
Résultat : $C(\mathcal{P}_{z_{s,k}}, x), B(\mathcal{P}_{z_{s,k}}, x)$
1:  calculer $A(P_a, x)$ pour tout $a \in \{1, \dots, k\}^s$
2:  $C(\mathcal{P}_{z_{s,k}}, x) \leftarrow 0$
3:  $B(\mathcal{P}_{z_{s,k}}, x) \leftarrow 0$
4:  pour $t$ de $s$ à $ks$ faire
5:     distribuer$(s, t)$

6:  Procédure distribuer$(c, r)$
7:  si $c = 1$ alors
8:     $a_s \leftarrow r$
9:     calculer $A(P_a^+, x)$ à l'aide de l'expression (35)
10:    $C(\mathcal{P}_{z_{s,k}}, x) \leftarrow \max\left\{C(\mathcal{P}_{z_{s,k}}, x), \left|\frac{A(P_a^+, x)}{n} - \lambda(P_a^+)\right|\right\}$
11:    $B(\mathcal{P}_{z_{s,k}}, x) \leftarrow \max\left\{B(\mathcal{P}_{z_{s,k}}, x), \frac{A(P_a^+, x)}{n} - \lambda(P_a^-), \lambda(P_a^+) - \frac{A(P_a^-, x)}{n}\right\}$
12: sinon
13:    pour $i$ de $\max\{1, r-(c-1)k\}$ à $\min\{k, r-(c-1)\}$ faire
14:       $a_{s-c+1} \leftarrow i$
15:       distribuer$(c-1, r-i)$
```

des vecteurs $v = (v_1, \dots, v_s) \in \{0,1\}^s$ non nuls dans un ordre tel que, à chaque itération, une seule composante (et donc la parité de leur somme) change.

Algorithme 7.3 à substituer à la ligne 9 de l'algorithme 7.2

```
$A(P_a^+, x) \leftarrow A(P_a, x)$
$v_i \leftarrow 0$ pour tout $i \in \{1, \dots, s\}$
$c_i \leftarrow a_i$ pour tout $i \in \{1, \dots, s\}$
pour $i$ de $1$ à $2^s - 1$ faire
   $g \leftarrow 1 + \log_2\left[\left((i-1) \oplus \lfloor\frac{i-1}{2}\rfloor\right) \oplus \left(i \oplus \lfloor\frac{i}{2}\rfloor\right)\right]$
   $v_g \leftarrow 1 - v_g$
   $c_g \leftarrow a_g - v_g$
   si $g = 1$ alors
      $A(P_a^+, x) \leftarrow A(P_a^+, x) + A(P_c^+, x)$
   sinon
      $A(P_a^+, x) \leftarrow A(P_a^+, x) - A(P_c^+, x)$
```

En fin de compte, pour tout entier $k \geq 2$, l'algorithme obtenu permet de calculer des bornes pour la discrépance d'une séquence de n points en dimension s en un temps $O(ns \log k + 2^s k^s)$ et pour une occupation $O(k^s)$ de l'espace-mémoire. La complexité de cette méthode est donc exponentielle en la dimension, mais linéaire en la taille de la séquence. D'autre part, il est clair que plus la valeur de k est élevée, meilleures sont les perspectives d'obtenir un intervalle pour la discrépance (24) de bonne qualité. Il s'agit donc de trouver un compromis entre la précision désirée et les ressources à disposition.

7.3 Expériences numériques

La méthode de la section 7.2 a été utilisée pour établir une borne supérieure pour la discrépance de quelques $(0, m, s)$-réseaux de Faure en base b. Chaque majoration $B(\mathcal{P}_{z_{s,k}}, x)$ présentée dans la table 7.3 est meilleure que la valeur correspondante fournie par le théorème 6.1. D'autre part, on remarque que, lorsque m grandit, la borne $B(\mathcal{P}_{z_{s,k}}, x)$ semble tendre vers la valeur correspondante de $\hat{M}_k^s$ donnée dans la table 7.2. Notons que ces majorations ne sont valables que pour les réseaux en question (obtenus à l'aide du générateur `GrayFaure` présenté dans la section 4.8.4), mais que la méthode peut être utilisée pour calculer une borne similaire pour toute séquence présentant les mêmes valeurs de s et n.

	m						
(s, b, k)	2	3	4	5	6	7	8
$(7, 7, 15)$	0.29179	0.24050	0.16100	0.13318	0.13266	$*$	$*$
$(8, 11, 11)$	0.29059	0.21414	0.18881	0.18547	0.18462	$*$	$*$
$(9, 11, 8)$	0.38061	0.27197	0.26268	0.25840	0.25805	0.25799	$*$
$(10, 11, 6)$	0.57184	0.36248	0.35632	0.34751	0.34739	0.34735	0.34734
$(11, 11, 5)$	0.60090	0.43559	0.42816	0.41915	0.41891	0.41887	0.41886
$(12, 13, 4)$	0.58701	0.52095	0.52530	0.51919	0.51912	0.51902	0.51902
$(13, 13, 4)$	0.59532	0.53410	0.53284	0.52663	0.52642	0.52635	0.52635
$(14, 17, 3)$	0.78284	0.67318	0.67567	0.67237	0.67240	0.67236	$\circ$
$(15, 17, 3)$	0.83738	0.67976	0.68475	0.67942	0.67949	0.67942	$\circ$
$(16, 17, 3)$	0.88928	0.68616	0.69338	0.68600	0.68610	0.68600	$\circ$
$(17, 17, 2)$	0.94810	0.88195	0.89215	0.88192	0.88206	0.88192	$\circ$
$(18, 19, 2)$	0.95291	0.88629	0.89388	0.88625	0.88640	0.88625	$\circ$
$(19, 19, 2)$	0.95291	0.89037	0.90073	0.89023	0.89048	0.89023	$\circ$
$(20, 23, 2)$	0.95086	0.89398	0.89991	0.89390	0.89400	$\circ$	$\circ$

TAB. 7.3. Borne supérieure $B(\mathcal{P}_{z_{s,k}}, x)$ pour la discrépance de quelques $(0, m, s)$-réseaux de Faure en base b. Les valeurs manquantes correspondent aux cas suivants :

$*$ La majoration fournie par le théorème 6.1 est meilleure.
$\circ$ La taille de la séquence $n = b^m$ est supérieure à 10^9 points.

La table 7.3 ne concerne que des réseaux en dimension $s \geq 7$. En effet, comme le laisse supposer la table 6.1, le théorème 6.1 fournit déjà des bornes efficaces pour des séquences de taille raisonnable lorsque $s \leq 6$. Cependant, le temps de calcul devenant prohibitif en dimension plus élevée, nous n'avons pas été plus loin que $s = 20$. D'autre part, nous n'avons pas jugé utile de considérer des réseaux de plus de 10^9 points. Les valeurs de k ont été choisies en fonction de s, de manière à ne pas dépasser la capacité-mémoire de la machine ($k^s < 3 \cdot 10^8$) et la grille $z_{s,k}$ utilisée a été, dans chaque cas, construite à l'aide de l'algorithme 7.1. La détermination de chacune de ces bornes a nécessité de quelques heures à quelques jours de calcul sur une station de travail SGI R10000.

7.4 Une partition plus générale

L'utilisation de la méthode présentée dans la section 7.2 est susceptible d'entraîner un gaspillage de ressources. En effet, bien que le théorème 7.3 garantisse l'obtention d'une précision égale au poids de la partition considérée, il existe des approches de décomposition du cube unité autres que celle basée sur les grilles extensibles qui parviennent à un résultat similaire pour un effort nettement moindre. En d'autres termes, il apparaît que dans une solution optimale du programme mathématique (31) et par conséquent dans l'expression (25), une écrasante majorité des k^s intervalles contenus dans une partition $\mathcal{P}_{z_{s,k}}$ donnée ont un poids largement inférieur à M_k^s.

Par exemple, en dimension $s = 8$ et pour $k = 10$, moins de 1% des 10^8 intervalles de la grille obtenue par l'application de l'algorithme 7.1 au programme mathématique M_{10}^8 ont un poids proche de la solution approchée $\hat{M}_{10}^8 = 0.2023$ (voir figure 7.3). Dans ce cas particulier, la méthode plus générale présentée dans cette section mène à une partition de poids 0.2023 comprenant moins de $4 \cdot 10^6$ intervalles, dont 63% atteignent exactement cette valeur.

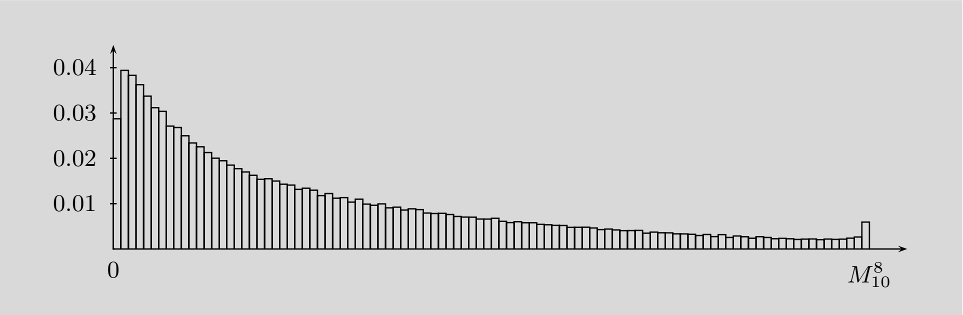


FIG. 7.3. Distribution (discrétisée sur 100 intervalles de même largeur) entre 0 et M_{10}^8 du poids des 10^8 intervalles de la partition correspondant à une solution approchée du programme mathématique M_{10}^8.

Ainsi, d'un certain point de vue, la restriction au cas des grilles extensibles est pénalisante quant à l'efficacité de la méthode proposée dans la section 7.1. Par contre, l'algorithme de décomposition présenté ci-dessous est affranchi de cette contrainte et permet de construire une partition $\mathcal{P}$ de cardinalité plus raisonnable et de poids ε, où $\varepsilon \in (0, 1)$ est un paramètre de précision arbitrairement choisi.

D'après le théorème 7.3, l'intervalle pour la discrépance obtenu $[C(\mathcal{P}, x), B(\mathcal{P}, x)]$ est de largeur inférieure ou égale à ε. De ce point de vue, plus ε est petit, meilleures sont les garanties sur la qualité des bornes en question. En revanche, la cardinalité de la partition et l'effort de calcul associé suivant un mouvement inverse, un compromis doit donc être trouvé.

Au niveau du choix de ε, signalons qu'un autre argument pourrait éventuellement être utilisé. En effet, nous suspectons que le poids $W(\mathcal{P})$ de la partition $\mathcal{P}$ considérée constitue une borne inférieure sur la borne supérieure $B(\mathcal{P}, x)$[1]. Cette assertion n'a pas pu être prouvée en toute généralité, mais la remarque 7.7 ci-dessous, ainsi que toutes nos expériences numériques (dont celles de la section 7.3), abondent dans ce sens. Cette conjecture implique que, si l'on estime la discrépance d'une séquence proche d'une certaine valeur, il est sans doute judicieux de choisir ε inférieure à celle-ci.

Il n'a pas été possible d'obtenir un algorithme capable de générer une partition $\mathcal{P}$ de poids ε qui soit de cardinalité minimale. Toutefois, hormis l'assurance que $W(\mathcal{P}) = \varepsilon$, la construction proposée ci-dessous fournit les garanties supplémentaires suivantes :

---

[1] Par exemple, nous pensons que la situation décrite par $C(\mathcal{P}, x) = 0.1$, $D_n^*(x) = 0.2$, $B(\mathcal{P}, x) = 0.3$ et $W(\mathcal{P}) = 0.7$ ne peut pas se produire, bien qu'elle soit en accord avec tous les résultats établis jusqu'ici.

- ▷ quelle que soit la dimension s, la partition obtenue est finie pour tout choix de $\varepsilon \in (0,1)$;
- ▷ on a $W(P) = \varepsilon$ pour une proportion importante des intervalles $P \in \mathcal{P}$ (ce fait pourrait suggérer que la cardinalité de la partition n'est pas loin d'être minimale) ;
- ▷ l'algorithme de génération obtenu est de complexité optimale (relativement à la cardinalité de la partition), à savoir $O(s|\mathcal{P}|)$;
- ▷ la structure même de la partition $\mathcal{P}$ peut être exploitée de manière à faciliter le calcul des bornes $C(\mathcal{P}, x)$ et $B(\mathcal{P}, x)$. Plus précisément, la complexité de la procédure de décompte des points appartenant à P^- et P^+ pour tout $P \in \mathcal{P}$ est sous-linéaire en la taille de la séquence.

REMARQUE 7.7 On conjecture que pour toute séquence $x \subset \bar{I}^s$ et toute partition $\mathcal{P}$ de I^s,

$$B(\mathcal{P}, x) \geq W(\mathcal{P}).$$

Cette assertion n'a pas pu être prouvée, mais plusieurs arguments indiquent qu'elle est probablement vraie. Tout d'abord, on remarque que, dans le cas où x est une séquence de variables aléatoires i.i.d. $U(I^s)$, on a l'espérance

$$\begin{aligned} E\left[B(\mathcal{P}, x)\right] &= E\left[\max_{P \in \mathcal{P}} \max\left\{\frac{A(P^+, x)}{n} - \lambda(P^-),\, \lambda(P^+) - \frac{A(P^-, x)}{n}\right\}\right] \\ &\geq \max_{P \in \mathcal{P}} \max\left\{E\left[\frac{A(P^+, x)}{n} - \lambda(P^-)\right],\, E\left[\lambda(P^+) - \frac{A(P^-, x)}{n}\right]\right\} \\ &= \max_{P \in \mathcal{P}} \left\{\lambda(P^+) - \lambda(P^-)\right\} = W(\mathcal{P}). \end{aligned}$$

D'autre part, la paire de relations

$$\frac{A(P^+, x)}{n} \geq \lambda(P^+) \quad \Longrightarrow \quad \frac{A(P^+, x)}{n} - \lambda(P^-) \geq \lambda(P^+) - \lambda(P^-) = W(P)$$

$$\frac{A(P^-, x)}{n} \leq \lambda(P^-) \quad \Longrightarrow \quad \lambda(P^+) - \frac{A(P^-, x)}{n} \geq \lambda(P^+) - \lambda(P^-) = W(P)$$

implique que

$$B(\mathcal{P}, x) \geq W(P)$$

pour tout intervalle P appartenant au, généralement très large, sous-ensemble de $\mathcal{P}$ suivant :

$$\left\{P \in \mathcal{P} \;:\; \frac{A(P^+, x)}{n} \geq \lambda(P^+) \quad \text{ou} \quad \frac{A(P^-, x)}{n} \leq \lambda(P^-)\right\}.$$

7.4.1 Le principe de décomposition. Soit un intervalle

$$[\alpha, \beta) = \prod_{i=1}^{s} [\alpha_i, \beta_i) \subset I^s,$$

défini par $\alpha < \beta \in \bar{I}^s$. Alors, pour toute composante $d \in \{1, \ldots, s\}$ et tout choix de $\gamma_d \in [\alpha_d, \beta_d)$, l'intervalle $[\alpha, \beta)$ se laisse partitionner en $[\alpha, \beta')$ et $[\alpha', \beta)$, où $\alpha', \beta' \in \bar{I}^s$ sont deux points donnés par

$$\alpha'_i = \begin{cases} \gamma_d & \text{si } i = d \\ \alpha_i & \text{sinon} \end{cases} \qquad \text{et} \qquad \beta'_i = \begin{cases} \gamma_d & \text{si } i = d \\ \beta_i & \text{sinon.} \end{cases}$$

Un tel pas (voir figure 7.4) est appelé une *décomposition de paramètre γ_d dans la direction d*. Nous allons montrer que, partant du cube unité I^s, ce processus peut être appliqué récursivement jusqu'à engendrer une partition ne contenant que des intervalles de poids inférieur ou égal à ε.

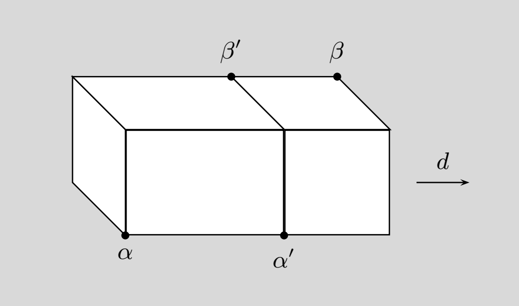


FIG. 7.4. Décomposition dans la direction d d'un intervalle $[\alpha,\beta)$ en $[\alpha,\beta')\cup[\alpha',\beta)$.

Généralisant cette opération pour un point quelconque $\gamma\in[\alpha,\beta)$, on obtient une *décomposition de paramètre* γ. Ce nouveau pas est défini comme l'application successive, pour $d=1,\dots,s$ (dans cet ordre), d'une décomposition de paramètre γ_d dans la direction d à l'intervalle issu de la transformation précédente dont le « coin supérieur-droit » a été préservé. Une telle combinaison de s décompositions dans différentes directions permet de partitionner un intervalle $[\alpha,\beta)\subset I^s$ en au plus $s+1$ parties.

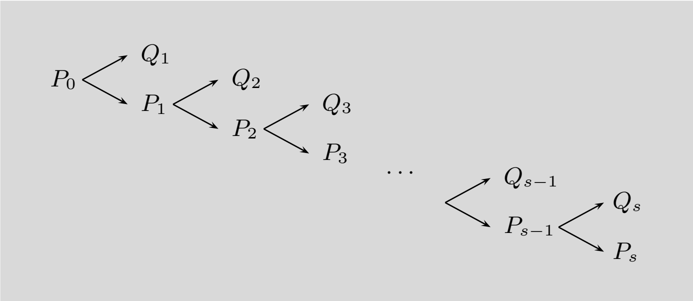


FIG. 7.5. Décomposition d'un intervalle $P_0=[\alpha,\beta)$ en $s+1$ parties $Q_1\cup\dots\cup Q_s\cup P_s$.

Partant de $P_0=[\alpha,\beta)$ et appliquant pour $d=1,\dots,s$ une décomposition de paramètre γ_d dans la direction d à P_{d-1}, on obtient une succession de paires d'intervalles Q_d et P_d tels que

$$Q_d\cup P_d=P_{d-1}.$$

Ce processus est illustré dans les figures 7.5 et 7.6. De manière plus explicite, on a

$$\begin{aligned}
P_0 &= \Big[(\alpha_1,\alpha_2,\dots,\alpha_s),\beta\Big)=[\alpha,\beta)\,,\\
Q_1 &= \Big[(\alpha_1,\alpha_2,\dots,\alpha_s),(\gamma_1,\beta_2,\dots,\beta_s)\Big),\\
P_1 &= \Big[(\gamma_1,\alpha_2,\dots,\alpha_s),\beta\Big),\\
Q_2 &= \Big[(\gamma_1,\alpha_2,\dots,\alpha_s),(\beta_1,\gamma_2,\beta_3,\dots,\beta_s)\Big),\\
P_2 &= \Big[(\gamma_1,\gamma_2,\alpha_3,\dots,\alpha_s),\beta\Big),\\
\vdots & \qquad\vdots\\
Q_s &= \Big[(\gamma_1,\dots,\gamma_{s-1},\alpha_s),(\beta_1,\dots,\beta_{s-1},\gamma_s)\Big),\\
P_s &= \Big[(\gamma_1,\gamma_2,\dots,\gamma_s),\beta\Big)=[\gamma,\beta)\,.
\end{aligned}$$

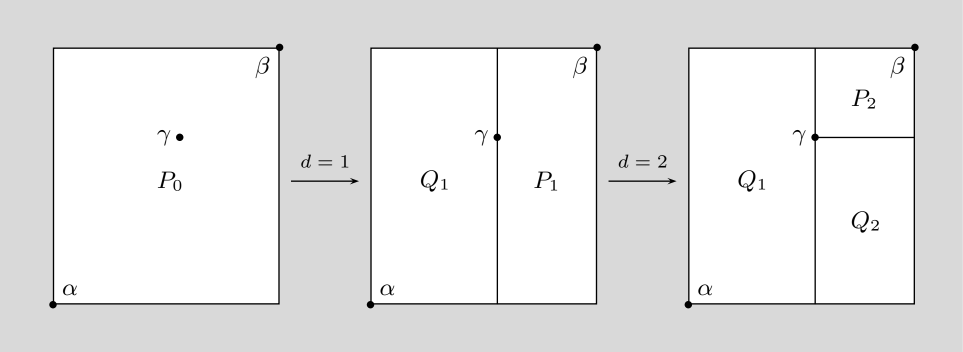


FIG. 7.6. Décomposition de paramètre γ d'un intervalle $[\alpha,\beta) \in I^2$.

En fin de compte, on obtient

$$[\alpha,\beta) = \left(\bigcup_{d=1}^{s} Q_d\right) \cup [\gamma,\beta)\,.$$

Soulignons le fait que, suivant le choix de $\gamma \in [\alpha,\beta)$, la partition de l'intervalle $[\alpha,\beta)$ peut comporter moins de $s+1$ parties. En effet, on observe que la décomposition de paramètre γ_d dans la direction d mène à un intervalle Q_d vide si et seulement si $\gamma_d = \alpha_d$:

$$\gamma_d = \alpha_d \qquad \Longleftrightarrow \qquad Q_d = \varnothing \qquad \Longleftrightarrow \qquad P_d = P_{d-1}.$$

Cette présentation informelle du processus de décomposition étant terminée, introduisons la notation nécessaire à la définition de la partition $\mathcal{P}_\varepsilon^s$ de poids ε du cube unité I^s que nous nous proposons de construire.

DÉFINITION 7.8 Pour un intervalle $P = [\alpha^P,\beta^P) \subset I^s$ et un point $\gamma^P \in [\alpha^P,\beta^P)$ donnés, une *décomposition de paramètre* γ^P est définie de la manière suivante :

$$P = \left(\bigcup_{d=1}^{s} Q_d^P\right) \cup \left[\gamma^P,\beta^P\right),$$

où $Q_d^P = \left[\alpha^{Q_d^P},\beta^{Q_d^P}\right)$ est donné par

$$\alpha_i^{Q_d^P} = \begin{cases} \gamma_i^P & \text{pour } i < d \\ \alpha_i^P & \text{pour } i \geq d \end{cases} \qquad \beta_i^{Q_d^P} = \begin{cases} \gamma_i^P & \text{pour } i = d \\ \beta_i^P & \text{pour } i \neq d. \end{cases} \tag{37}$$

On relève que pour tout $i,d \in \{1,\ldots,s\}$, on a

$$\alpha_i^P \leq \alpha_i^{Q_d^P} \leq \gamma_i^P \leq \beta_i^{Q_d^P} \leq \beta_i^P.$$

Bien évidemment, la considération de ce principe de décomposition est une manière très particulière d'aborder le problème de la partition du cube unité I^s. D'autres approches ont été testées, mais aucune ne semblait mener à des partitions de cardinalité substantiellement inférieure ou possédant une structure aussi agréable que celle qui a été retenue.

7.4.2 Le choix du paramètre γ^P. Partant d'un intervalle $P = [\alpha^P, \beta^P) \subset I^s$ de poids supérieur à ε, plusieurs politiques de choix du paramètre γ^P conduisent, après la décomposition correspondante, à un ensemble d'intervalles de poids inférieur. Cependant, la plupart des stratégies envisageables occasionnent des coûts importants (pour la résolution d'un sous-problème) ou mènent à des partitions de cardinalité infinie ou dont le manque de structure ne facilite par le décompte de $A(P^-, x)$ et $A(P^+, x)$ lors du calcul effectif des bornes $C(\mathcal{P}, x)$ et $B(\mathcal{P}, x)$. Par exemple, il aurait été envisageable de choisir successivement $\gamma_1^P, \ldots, \gamma_s^P$ de manière à ce que chacun des intervalles résultants non vides $Q_1^P, \ldots, Q_s^P$ soit de poids égal à ε et de réappliquer ce processus à l'intervalle restant $[\gamma^P, \beta^P)$ s'il s'avère de poids supérieur à ε. À première vue, cette approche peut sembler judicieuse, mais on s'aperçoit rapidement qu'elle mène parfois à des partitions de cardinalité infinie (par exemple pour $\varepsilon = 0.15$ dans le carré unité).

Une meilleure stratégie consiste à choisir γ^P tel que $W\left([\gamma^P, \beta^P)\right) = \varepsilon$, mais d'une manière qui permette de décomposer les intervalles restants $Q_1^P, \ldots, Q_s^P$ non vides et de poids supérieur à ε en un nombre de plus en plus restreint d'intervalles au cours des phases ultérieures induites par l'application récursive du procédé. Ce but peut être atteint par le biais de la procédure de choix suivante :

DÉFINITION 7.9 À tout intervalle $P = [\alpha^P, \beta^P) \subset I^s$ de poids strictement supérieur à ε, on associe le paramètre de décomposition $\gamma^P = \gamma^P(\delta^P) \in [\alpha^P, \beta^P)$ donné par les s fonctions

$$\gamma_i^P(\delta) = \begin{cases} \alpha_i^P & \text{si } \delta\beta_i^P \leq \alpha_i^P \\ \delta\beta_i^P & \text{sinon} \end{cases}, \text{ pour tout } i \in \{1, \ldots, s\} \tag{38}$$

et le facteur $\delta^P \in (0,1)$ défini comme l'unique solution de l'équation $W\left(\left[\gamma^P(\delta), \beta^P\right)\right) = \varepsilon$.

Intuitivement, γ^P est une transformation linéaire de β^P, tronquée en fonction de α^P selon certaines composantes. Les troncations en question se laissent décrire succinctement comme suit :

$$\delta^P\beta_d^P \leq \alpha_d^P \quad \Longleftrightarrow \quad \gamma_d^P = \alpha_d^P \quad \Longleftrightarrow \quad Q_d^P = \varnothing. \tag{39}$$

On obtient donc une première version de notre algorithme de construction de la partition $\mathcal{P}_\varepsilon^s$:

Algorithme 7.4 Construction de la partition $\mathcal{P}_\varepsilon^s$ du cube unité I^s

```
Donnée : $s, \varepsilon$
Résultat : $\mathcal{P}_\varepsilon^s$
  $\mathcal{P}_\varepsilon^s \leftarrow \varnothing$
  décomposer($I^s$)

  Procédure décomposer($P$)
  calculer $\delta^P$ et $\gamma^P$
  pour $d$ de 1 à $s$ faire
     si $\gamma_d^P \neq \alpha_d^P$ alors
        si $W(Q_d^P) > \varepsilon$ alors
           décomposer($Q_d^P$)
        sinon
           $\mathcal{P}_\varepsilon^s \leftarrow \mathcal{P}_\varepsilon^s \cup Q_d^P$
  $\mathcal{P}_\varepsilon^s \leftarrow \mathcal{P}_\varepsilon^s \cup [\gamma^P, \beta^P)$
```

Bien que des modifications importantes doivent encore être apportées à cette formulation, la structure globale de cet algorithme est quasiment définitive. Précisons notamment qu'une procédure permettant de calculer efficacement δ^P et γ^P reste à formuler.

Considérons tout d'abord un exemple d'application de cette méthode de décomposition. Pour $\varepsilon = 0.6$ et $s = 3$ la partition obtenue est représentée dans la figure 7.7 et les valeurs numériques correspondantes sont données dans la table 7.4. En examinant cette dernière, on remarque que les séquences de points α^P et β^P définissant les intervalles en question semblent avoir été triées dans l'ordre lexicographique.

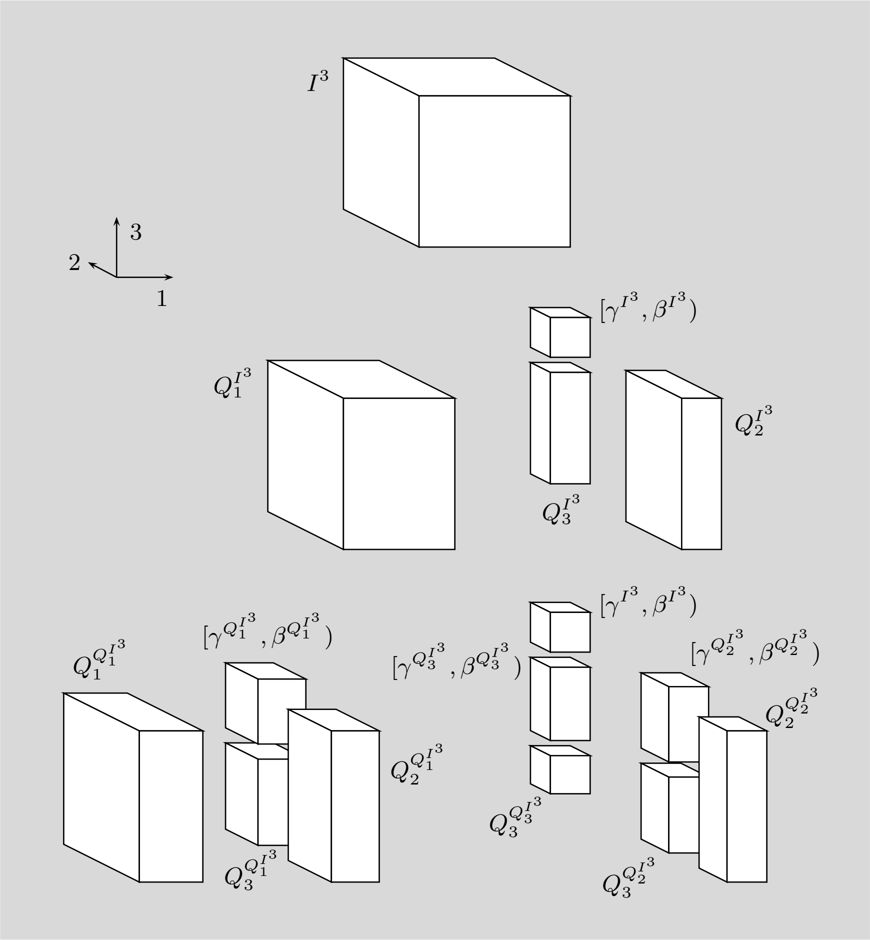


FIG. 7.7. Décomposition du cube unité I^3 pour $\varepsilon = 0.6$.

7.4.3 Structure de la partition $\mathcal{P}_\varepsilon^s$. De manière formelle, $\mathcal{P}_\varepsilon^s$ est l'unique partition de I^s obtenue en appliquant récursivement le principe de décomposition décrit dans les définitions 7.8 et 7.9 à tout intervalle de poids strictement supérieur à ε rencontré au cours du processus en question. Clairement, tout intervalle généré est de poids inférieur ou égal à ε, mais on n'a a priori aucune assurance quant au fait que la partition obtenue soit finie.

Ci-dessous, on prouve dans un premier temps quelques résultats concernant la construction de la partition lorsqu'on examine le processus au cours des « générations » successives de décomposition. On démontre ensuite que la propriété relative à l'ordre lexicographique observée dans l'exemple de la table 7.4 est vraie en toute généralité. Finalement, des formules explicites permettant de calculer efficacement les paramètres δ^P et γ^P sont établies. La question de la cardinalité de la partition $\mathcal{P}_\varepsilon^s$ est abordée dans la section 7.4.4.

$$\begin{aligned}
Q_1^{Q_1^{I^3}} &= [(0.000000, 0.000000, 0.000000), (0.420343, 1.000000, 1.000000))\\
Q_2^{Q_1^{I^3}} &= [(0.420343, 0.000000, 0.000000), (0.736806, 0.570494, 1.000000))\\
Q_3^{Q_1^{I^3}} &= [(0.420343, 0.570494, 0.000000), (0.736806, 1.000000, 0.570494))\\
[\gamma^{Q_1^{I^3}}, \beta^{Q_1^{I^3}}) &= [(0.420343, 0.570494, 0.570494), (0.736806, 1.000000, 1.000000))\\
Q_2^{Q_2^{I^3}} &= [(0.736806, 0.000000, 0.000000), (1.000000, 0.369873, 1.000000))\\
Q_3^{Q_2^{I^3}} &= [(0.736806, 0.369873, 0.000000), (1.000000, 0.736806, 0.501995))\\
[\gamma^{Q_2^{I^3}}, \beta^{Q_2^{I^3}}) &= [(0.736806, 0.369873, 0.501995), (1.000000, 0.736806, 1.000000))\\
Q_3^{Q_3^{I^3}} &= [(0.736806, 0.736806, 0.000000), (1.000000, 1.000000, 0.251999))\\
[\gamma^{Q_3^{I^3}}, \beta^{Q_3^{I^3}}) &= [(0.736806, 0.736806, 0.251999), (1.000000, 1.000000, 0.736806))\\
[\gamma^{I^3}, \beta^{I^3}) &= [(0.736806, 0.736806, 0.736806), (1.000000, 1.000000, 1.000000))
\end{aligned}$$

TAB. 7.4. Décomposition du cube unité I^3 pour $\varepsilon = 0.6$.

LEMME 7.10 *Si $P = [\alpha^P, \beta^P) \subset I^s$ est un intervalle de poids $W(P) > \varepsilon$, alors pour toute direction $j \in \{1, \ldots, s\}$ telle que $Q_j^P \neq \varnothing$ et $W(Q_j^P) > \varepsilon$, on a*

1° $\delta^{Q_j^P} < \delta^P$,

2° $\gamma_i^{Q_j^P} = \alpha_i^{Q_j^P} = \gamma_i^P$, pour toute direction $i < j$.

PREUVE. 1° On commence par déterminer la valeur de $\gamma_i^{Q_j^P}(\delta^P)$ pour tout $i \in \{1, \ldots, s\}$. Reformulant la définition (38), il vient

$$\gamma_i^{Q_j^P}(\delta^P) = \begin{cases} \alpha_i^{Q_j^P} & \text{si } \delta^P \beta_i^{Q_j^P} \leq \alpha_i^{Q_j^P}, \\ \delta^P \beta_i^{Q_j^P} & \text{sinon.} \end{cases} \tag{40}$$

On examine les trois cas suivants :

▷ Cas 1 : $i < j$

$$\begin{aligned}
i < j \quad &\overset{(37)}{\Longrightarrow} \quad \alpha_i^{Q_j^P} = \gamma_i^P \text{ et } \beta_i^{Q_j^P} = \beta_i^P \\
&\overset{(40)}{\Longrightarrow} \quad \gamma_i^{Q_j^P}(\delta^P) = \begin{cases} \gamma_i^P & \text{si } \delta^P \beta_i^P \leq \gamma_i^P, \\ \delta^P \beta_i^P & \text{sinon.} \end{cases}
\end{aligned} \tag{41}$$

Considérant séparément les deux cas de la définition (38), on obtient

$$\left\{ \begin{array}{lll} \delta^P \beta_i^P \leq \alpha_i^P \overset{(38)}{\Longrightarrow} & \gamma_i^P = \alpha_i^P & \overset{(41)}{\Longrightarrow} \gamma_i^{Q_j^P}(\delta^P) = \gamma_i^P \\ \delta^P \beta_i^P > \alpha_i^P \overset{(38)}{\Longrightarrow} & \gamma_i^P = \delta^P \beta_i^P & \overset{(41)}{\Longrightarrow} \gamma_i^{Q_j^P}(\delta^P) = \gamma_i^P \end{array} \right.$$

$$\Longrightarrow \quad \gamma_i^{Q_j^P}(\delta^P) = \gamma_i^P, \text{ pour tout } i < j.$$

▷ Cas 2 : $i = j$

$$\begin{aligned}
i = j \quad &\overset{(37)}{\Longrightarrow} \quad \alpha_j^{Q_j^P} = \alpha_j^P \text{ et } \beta_j^{Q_j^P} = \gamma_j^P \\
&\overset{(40)}{\Longrightarrow} \quad \gamma_j^{Q_j^P}(\delta^P) = \begin{cases} \alpha_j^P & \text{si } \delta^P \gamma_j^P \leq \alpha_j^P \\ \delta^P \gamma_j^P & \text{sinon} \end{cases}
\end{aligned}$$

$$\Longrightarrow \quad \gamma_j^{Q_j^P}(\delta^P) \geq \delta^P \gamma_j^P.$$

▷ Cas 3 : $i > j$

$$i > j \quad \Longrightarrow^{(37)} \quad \alpha_i^{Q_j^P} = \alpha_i^P \text{ et } \beta_i^{Q_j^P} = \beta_i^P$$

$$\Longrightarrow^{(40)} \quad \gamma_i^{Q_j^P}(\delta^P) = \begin{cases} \alpha_i^P & \text{si } \delta^P \beta_i^P \leq \alpha_i^P \\ \delta^P \beta_i^P & \text{sinon} \end{cases} =^{(38)} \gamma_i^P.$$

Il est maintenant possible de majorer le poids de l'intervalle $\left[\gamma^{Q_j^P}(\delta^P), \beta^{Q_j^P}\right)$ par $\delta^P \varepsilon$:

$$\begin{aligned} W\left(\left[\gamma^{Q_j^P}(\delta^P), \beta^{Q_j^P}\right)\right) &= \prod_{i=1}^{s} \beta_i^{Q_j^P} - \prod_{i=1}^{s} \gamma_i^{Q_j^P}(\delta^P) \\ = \delta^P \prod_{i=1}^{s} \beta_i^P - \prod_{i=1}^{s} \gamma_i^{Q_j^P}(\delta^P) &\leq \delta^P \left(\prod_{i=1}^{s} \beta_i^P - \prod_{i=1}^{s} \gamma_i^P\right) = \delta^P \varepsilon < \varepsilon. \end{aligned}$$

Ainsi, l'unique coefficient $\delta^{Q_j^P} \in (0,1)$ tel que $W\left(\left[\gamma^{Q_j^P}(\delta^{Q_j^P}), \beta^{Q_j^P}\right)\right) = \varepsilon$ satisfait $\delta^{Q_j^P} < \delta^P$.

2° Reformulant la définition (38), on a

$$\gamma_i^{Q_j^P} = \begin{cases} \alpha_i^{Q_j^P} & \text{si } \delta^{Q_j^P} \beta_i^{Q_j^P} \leq \alpha_i^{Q_j^P}, \\ \delta^{Q_j^P} \beta_i^{Q_j^P} & \text{sinon.} \end{cases}$$

$$i < j \quad \Longrightarrow^{(37)} \quad \gamma_i^{Q_j^P} = \begin{cases} \gamma_i^P & \text{si } \delta^{Q_j^P} \beta_i^P \leq \gamma_i^P, \\ \delta^{Q_j^P} \beta_i^P & \text{sinon.} \end{cases}$$

Considérant à nouveau les deux cas de la définition (38), on obtient

▷ Cas 1 : $\delta^P \beta_i^P \leq \alpha_i^P$

$$\delta^P \beta_i^P \leq \alpha_i^P \quad \Longrightarrow^{(38)} \quad \gamma_i^P = \alpha_i^P$$

$$\delta^{Q_j^P} < \delta^P \quad \Longrightarrow \quad \delta^{Q_j^P} \beta_i^P < \delta^P \beta_i^P \leq \alpha_i^P = \gamma_i^P \quad \Longrightarrow \quad \gamma_i^{Q_j^P} = \gamma_i^P.$$

▷ Cas 2 : $\delta^P \beta_i^P > \alpha_i^P$

$$\delta^P \beta_i^P > \alpha_i^P \quad \Longrightarrow^{(38)} \quad \gamma_i^P = \delta^P \beta_i^P$$

$$\delta^{Q_j^P} < \delta^P \quad \Longrightarrow \quad \delta^{Q_j^P} \beta_i^P < \delta^P \beta_i^P = \gamma_i^P \quad \Longrightarrow \quad \gamma_i^{Q_j^P} = \gamma_i^P.$$

D'autre part et pour terminer, rappelons que l'égalité entre $\alpha_i^{Q_j^P}$ et γ_i^P pour tout $i < j$ est assurée par la relation (37).

□

Le corollaire fondamental suivant découle de la propriété (39) et de la seconde partie du lemme 7.10.

COROLLAIRE 7.11 *Si $P = [\alpha^P, \beta^P) \subset I^s$ est un intervalle de poids $W(P) > \varepsilon$, alors pour toute direction $j \in \{1, \ldots, s\}$ telle que $Q_j^P \neq \varnothing$ et $W(Q_j^P) > \varepsilon$, on a*

$$Q_i^{Q_j^P} = \varnothing, \text{ pour toute direction } i < j.$$

Par définition, Q_j^P est un intervalle issu de P après décomposition dans les directions 1 à j. Le corollaire 7.11 affirme que, dans le cas où Q_j^P doit être partitionné à son tour, les décompositions dans les directions 1 à $j-1$ mènent systématiquement à des intervalles vides. Plus précisément, on a

$$Q_j^P = \left(\bigcup_{i=j}^{s} Q_i^{Q_j^P} \right) \cup \left[\gamma^{Q_j^P}, \beta^{Q_j^P} \right).$$

On obtient donc un maximum de $s-j+2$ nouveaux intervalles. Ces considérations mènent à la reformulation suivante de notre algorithme de partition :

Algorithme 7.5 Construction de la partition $\mathcal{P}_\varepsilon^s$ du cube unité I^s

```
**Donnée :** $s, \varepsilon$
**Résultat :** $\mathcal{P}_\varepsilon^s$
  $\mathcal{P}_\varepsilon^s \leftarrow \varnothing$
  décomposer($I^s, 1$)

  **Procédure** décomposer($P, j$)
  $\gamma_i^P \leftarrow \alpha_i^P$ pour tout $i < j$
  calculer $\delta^P$ et $\gamma_j^P, \dots, \gamma_s^P$
  **pour** $i$ **de** $j$ à $s$ **faire**
    **si** $\gamma_i^P \neq \alpha_i^P$ **alors**
      **si** $W(Q_i^P) > \varepsilon$ **alors**
        décomposer($Q_i^P, i$)
      **sinon**
        $\mathcal{P}_\varepsilon^s \leftarrow \mathcal{P}_\varepsilon^s \cup Q_i^P$
  $\mathcal{P}_\varepsilon^s \leftarrow \mathcal{P}_\varepsilon^s \cup [\gamma^P, \beta^P)$
```

On en arrive au résultat annoncé plus haut concernant l'ordre particulier dans lequel l'algorithme introduit les intervalles dans la partition $\mathcal{P}_\varepsilon^s$. Cette propriété est exploitée de manière intensive pour accélérer le décompte des points lors du calcul des bornes pour la discrépance (voir section 7.4.5).

THÉORÈME 7.12 *Les deux séquences de points* $\{\alpha^P : P \in \mathcal{P}_\varepsilon^s\}$ *et* $\{\beta^P : P \in \mathcal{P}_\varepsilon^s\}$ *définissant les intervalles de la partition* $\mathcal{P}_\varepsilon^s$ *sont générés par l'algorithme dans l'ordre lexicographique.*

PREUVE. Pour tout intervalle non vide Q_j^P, apparaissant au cours du déroulement de l'algorithme, on note $\mathcal{F}(Q_j^P) \subset \mathcal{P}_\varepsilon^s$ le sous-ensemble de la partition issu de la décomposition de Q_j^P en intervalles de poids inférieur ou égal à ε. Par la nature récursive du processus, il est clair que, pour $j < s$, la génération de $\mathcal{F}(Q_j^P)$ est entièrement terminée avant que la question de la décomposition de Q_{j+1}^P ne soit abordée. D'autre part, considérant l'expression (37), on voit que les $s+1$ intervalles que l'on peut éventuellement obtenir par décomposition directe de P sont dans l'ordre lexicographique désiré :

$$\alpha^P = \alpha^{Q_1^P} \overset{\text{lex}}{<} \cdots \overset{\text{lex}}{<} \alpha^{Q_s^P} \overset{\text{lex}}{<} \gamma^P$$
$$\beta^{Q_1^P} \overset{\text{lex}}{<} \cdots \overset{\text{lex}}{<} \beta^{Q_s^P} \overset{\text{lex}}{<} \beta^P.$$

Il suffit donc de montrer que, pour toute direction $j < s$, cet ordre est préservé entre les intervalles de $\mathcal{F}(Q_j^P)$ et ceux contenus dans $\mathcal{F}(Q_{j+1}^P)$, ainsi qu'entre ceux de $\mathcal{F}(Q_s^P)$ et $[\gamma^P, \beta^P)$.

- ▷ Pour $\{\alpha^P : P \in \mathcal{P}_\varepsilon^s\}$:

  Considérant la relation (37), la seconde partie du lemme 7.10 et le corollaire 7.11, on a
  $$\alpha_i^R = \gamma_i^P, \text{ pour tout } i < j \text{ et tout } R \in \mathcal{F}(Q_j^P).$$
  D'autre part, l'expression (37) implique que $\alpha_j^{Q_j^P} = \alpha_j^P < \gamma_j^P = \beta_j^{Q_j^P}$ lorsque Q_j^P est non vide. On en déduit que
  $$\alpha_j^R \in \left[\alpha_j^P, \gamma_j^P\right), \text{ pour tout } R \in \mathcal{F}(Q_j^P).$$
  En particulier, on a $\alpha_j^R < \gamma_j^P$ pour tout intervalle $R \in \mathcal{F}(Q_j^P)$, si bien qu'en fin de compte
  $$\begin{aligned} \alpha^R = (\gamma_1^P, \dots, \gamma_{j-1}^P, \alpha_j^R, \dots, \alpha_s^R) &\overset{\text{lex}}{<} (\gamma_1^P, \dots, \gamma_j^P, \alpha_{j+1}^P, \dots, \alpha_s^P) = \alpha^{Q_{j+1}^P} && \text{pour } j < s, \\ \alpha^R = (\gamma_1^P, \dots, \gamma_{s-1}^P, \alpha_s^R) &\overset{\text{lex}}{<} (\gamma_1^P, \dots, \gamma_s^P) = \gamma^P && \text{pour } j = s. \end{aligned}$$

- ▷ Pour $\{\beta^P : P \in \mathcal{P}_\varepsilon^s\}$:

  Considérant la relation (37) et le corollaire 7.11, on obtient
  $$\beta_i^R = \beta_i^P, \text{ pour tout } i < j \text{ et tout } R \in \mathcal{F}(Q_j^P).$$
  Or, par construction, $\beta_j^R \leq \beta_j^{Q_j^P} = \gamma_j^P < \beta_j^P$ pour tout $R \in \mathcal{F}(Q_j^P)$. Ainsi, pour $j = s$ et tout intervalle $R \in \mathcal{F}(Q_s^P)$, on a
  $$\beta^R = (\beta_1^P, \dots, \beta_{s-1}^P, \beta_s^R) \overset{\text{lex}}{<} (\beta_1^P, \dots, \beta_s^P) = \beta^P$$
  et finalement, pour toute direction $j < s$ et toute paire d'intervalles $R \in \mathcal{F}(Q_j^P)$ et $T \in \mathcal{F}(Q_{j+1}^P)$, on obtient
  $$\beta^R = (\beta_1^P, \dots, \beta_{j-1}^P, \beta_j^R, \dots, \beta_s^R) \overset{\text{lex}}{<} (\beta_1^P, \dots, \beta_j^P, \beta_{j+1}^T, \dots, \beta_s^T) = \beta^T.$$

$\square$

Il nous reste encore à établir des expressions permettant de calculer les paramètres δ^P et γ^P nécessaires au processus de décomposition.

LEMME 7.13 *Dans l'algorithme 7.5, lors de chaque appel à la procédure* `décomposer`(P, j) *et pour toute direction $i \in \{j, \dots, s\}$ menant à la génération d'un intervalle Q_i^P non vide, on a*
$$W(Q_i^P) = \delta^P W(P).$$

PREUVE. Examinant de plus près l'expression (37), on s'aperçoit que la composante d'index s du « coin inférieur-gauche » des intervalles décomposés est préservée durant tout le processus[2] :
$$\alpha_s^{Q_i^P} = \alpha_s^P = \cdots = \alpha_s^{I^s} = 0.$$

On en déduit facilement que
$$W(Q_i^P) = \prod_{k=1}^{s} \beta_k^{Q_i^P} - \prod_{k=1}^{s} \alpha_k^{Q_i^P} = \prod_{k=1}^{s} \beta_k^{Q_i^P} = \delta^P \prod_{k=1}^{s} \beta_k^P = \delta^P W(P).$$

$\square$

---

[2] Bien sûr, la partition contient des intervalles P avec $\alpha_s^P \neq 0$, mais il s'agit exclusivement d'intervalles du type $[\gamma^*, \beta^*)$.

THÉORÈME 7.14 *Dans l'algorithme 7.5, pour tout appel à* `décomposer`(P, j)*, on a*

$$\delta^P = \left(\frac{\prod_{i=1}^{s} \beta_i^P - \varepsilon}{\prod_{i=1}^{j-1} \alpha_i^P \prod_{i=j}^{s} \beta_i^P} \right)^{\frac{1}{s-j-1}} \tag{42}$$

$$\gamma_i^P = \begin{cases} \alpha_i^P & \textit{pour } i < j \\ \delta^P \beta_i^P & \textit{pour } i \geq j \end{cases} \tag{43}$$

PREUVE. Par définition, $\delta^P \in (0, 1)$ est l'unique solution de l'équation

$$W\left([\gamma^P, \beta^P)\right) = \varepsilon, \quad \text{où} \quad \gamma_i^P = \begin{cases} \alpha_i^P & \text{si } \delta^P \beta_i^P \leq \alpha_i^P \\ \delta^P \beta_i^P & \text{sinon} \end{cases} \text{, pour tout } i \in \{1, \ldots, s\}.$$

Le corollaire 7.11 implique que l'intervalle P est issu de I^s après une séquence de décompositions dans des directions d'index au plus j. En considérant l'expression (37), on voit que $\alpha_i^P = \cdots = \alpha_i^{I^s} = 0$ pour tout $i \geq j$. Ainsi, pour tout $\delta \in (0, 1)$, on a $\delta \, \beta_i^P > \alpha_i^P = 0$ pour tout $i \geq j$. D'autre part, par la relation (39) et le corollaire 7.11, il est clair que $\gamma_i^P = \alpha_i^P$ pour tout $i < j$. L'expression (43) est donc démontrée. En utilisant cette propriété, la valeur (42) du paramètre δ^P s'obtient par simple résolution de l'équation $W\left([\gamma^P, \beta^P)\right) = \varepsilon$. □

Ces résultats permettent de renforcer le corollaire 7.11 :

COROLLAIRE 7.15 *Dans l'algorithme 7.5, pour tout appel à* `décomposer`(P, j)*, l'intervalle* P *est partitionné comme suit en exactement* $s - j + 2$ *parties non vides :*

- ▷ *les* $s - j + 1$ *intervalles* $Q_j^P, \ldots, Q_s^P$ *de poids* $\delta^P W(P)$ *;*
- ▷ *l'intervalle* $[\gamma^P, \beta^P)$ *de poids* ε*.*

On obtient donc la version finale de notre algorithme de partition :

Algorithme 7.6 Construction de la partition $\mathcal{P}_\varepsilon^s$ du cube unité I^s

```
Donnée : $s, \varepsilon$
Résultat : $\mathcal{P}_\varepsilon^s$
  $\mathcal{P}_\varepsilon^s \leftarrow \varnothing$
  décomposer($I^s, 1, 1.0$)

  Procédure décomposer($P, j, v$)
  calculer $\delta^P$ à l'aide de l'expression (42)
  calculer $\gamma^P$ à l'aide de l'expression (43)
  si $\delta^P v > \varepsilon$ alors
    pour $i$ de $j$ à $s$ faire
      décomposer($Q_i^P, i, \delta^P v$)
  sinon
    pour $i$ de $j$ à $s$ faire
      $\mathcal{P}_\varepsilon^s \leftarrow \mathcal{P}_\varepsilon^s \cup Q_i^P$
  $\mathcal{P}_\varepsilon^s \leftarrow \mathcal{P}_\varepsilon^s \cup [\gamma^P, \beta^P)$
```

Cet algorithme est de complexité optimale $O(s|\mathcal{P}_\varepsilon^s|)$, relativement à la cardinalité de $\mathcal{P}_\varepsilon^s$.

7.4.4 Cardinalité de la partition. On montre que la partition $\mathcal{P}_\varepsilon^s$ est finie en exhibant une borne supérieure sur sa cardinalité. Cette preuve utilise la notion d'arbre triangulaire (voir figure 7.8) :

DÉFINITION 7.16 Pour une paire d'entiers $l \geq 1$ et $h \geq 0$, l'*arbre triangulaire* T_h^l (de hauteur h et de largeur l) est défini comme suit :

- ▷ T_0^l est une feuille ;
- ▷ pour une hauteur $h > 0$, l'arbre T_h^l est constitué d'une racine à laquelle sont attachés l arbres triangulaires $T_{h-1}^l, T_{h-1}^{l-1}, \ldots, T_{h-1}^1$.

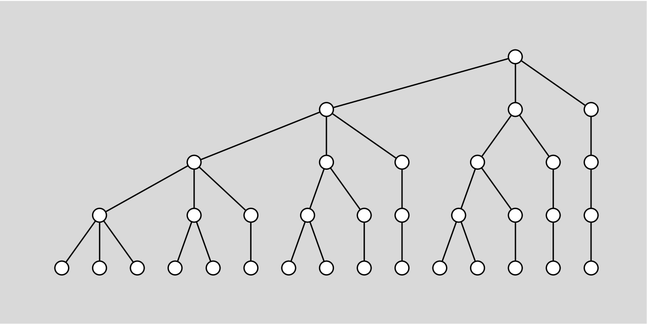


FIG. 7.8. L'arbre triangulaire T_4^3 et ses 15 feuilles.

THÉORÈME 7.17 *L'arbre triangulaire* T_h^l *possède* $N_h^l = \binom{h+l-1}{l-1}$ *feuilles.*

PREUVE. L'affirmation est évidente pour $h = 0$: T_0^l est une feuille et $N_0^l = 1$. On procède par induction sur la hauteur de l'arbre. On suppose le résultat prouvé pour h et on le démontre pour $h+1$:

$$N_{h+1}^l = \sum_{i=1}^{l} N_h^i = \sum_{i=1}^{l} \binom{h+i-1}{i-1} = \sum_{i=0}^{l-1} \binom{h+l-i-1}{l-i-1}.$$

Appliquant récursivement l'identité $\binom{b+1}{a} = \binom{b}{a} + \binom{b}{a-1}$, il vient

$$\binom{b+1}{a} = \sum_{i=0}^{a} \binom{b-i}{a-i}.$$

Utilisant cette expression pour $a = l-1$ et $b = h+l-1$, on obtient le résultat

$$N_{h+1}^l = \binom{h+l}{l-1}.$$

□

On montre maintenant que la partition $\mathcal{P}_\varepsilon^s$ est finie.

THÉORÈME 7.18 *Pour tout choix de* $\varepsilon \in (0,1)$*, on a*

$$|\mathcal{P}_\varepsilon^s| \leq \binom{s+h}{s}, \text{ où } h = \left\lceil \frac{s \log \varepsilon}{\log(1-\varepsilon)} \right\rceil.$$

PREUVE. Dans l'algorithme 7.5, pour tout appel à la procédure `décomposer`(P, j), le corollaire 7.15 stipule que P est partitionné en un intervalle $[\gamma^P, \beta^P)$ de poids ε (directement ajouté à $\mathcal{P}_\varepsilon^s$) et $s-j+1$ intervalles $Q_j^P, \ldots, Q_s^P$ de poids $\delta^P W(P)$. On obtient le résultat en remarquant que la première partie du lemme 7.10 implique que δ^P est inférieur au tout premier facteur utilisé $\delta^{I^s} = (1-\varepsilon)^{1/s}$.

En effet, il est alors possible de majorer la longueur de toute chaîne de décompositions (apparaissant au cours du déroulement de l'algorithme) par un nombre h, où h est le plus petit entier tel que

$$\left((1-\varepsilon)^{1/s}\right)^h \leq \varepsilon.$$

Globalement, le processus de décomposition se laisse donc représenter sous la forme d'un arbre triangulaire de hauteur h et de largeur $s+1$ dont de nombreuses branches ont été tronquées. Les feuilles de cet arbre correspondant aux intervalles de la partition, on obtient

$$|\mathcal{P}^s_\varepsilon| \leq N_h^{s+1} = \binom{s+h}{s}, \text{ où } h = \left\lceil \frac{s\log\varepsilon}{\log(1-\varepsilon)} \right\rceil.$$

En pratique, cette majoration constitue le plus souvent une large surestimation. En revanche, comme l'illustre la table 7.5, l'estimateur empirique $\frac{s}{\varepsilon^s}$ fournit généralement une bonne approximation de la cardinalité en question. Néanmoins, ces deux indicateurs suggèrent que la taille de la partition $\mathcal{P}^s_\varepsilon$ croît de manière exponentielle avec la dimension.

ε	0.3	0.2	0.1	0.05	0.03
$\lvert\mathcal{P}^5_\varepsilon\rvert$	$1\,546$	$12\,566$	$428\,882$	$14\,153\,521$	$184\,095\,539$
$\binom{5+h^5_\varepsilon}{5}$	$26\,334$	$850\,668$	$153\,476\,148$	$18\,934\,442\,604$	$542\,256\,910\,321$
$\frac{5}{\varepsilon^5}$	$2\,057$	$15\,625$	$500\,000$	$16\,000\,000$	$205\,761\,317$

TAB. 7.5. Taille de la partition $\mathcal{P}^s_\varepsilon$ en dimension $s=5$ pour différentes valeurs de ε.

Comme cela a été discuté en page 73, l'utilisation de la méthode décrite dans la section 7.1 est susceptible de s'accompagner d'un certain gaspillage de ressources lorsque la partition $\mathcal{P}$ considérée est une grille extensible. Ce fait semble clair au vu des résultats de la table 7.6. En effet, on observe sur ces quelques exemples que la cardinalité d'une grille paraît nettement plus grande que la taille de la partition $\mathcal{P}^s_\varepsilon$ de même poids. De plus, la différence aurait tendance à s'accentuer lorsque la dimension augmente. Ainsi, pour un même effort de calcul, on peut s'attendre à ce que la nouvelle méthode de décomposition présentée dans cette section permette de construire des partitions de poids inférieur et donc de fournir de meilleures bornes pour la discrépance d'une séquence donnée. Notons toutefois que ce pronostic ne tient pas compte des difficultés supplémentaires occasionnées au niveau du décompte des points.

s	2	3	4	5	6	7	8	10	12	16	20
$\varepsilon = \hat{M}^s_k$	0.090263	0.10514	0.11374	0.12100	0.12737	0.13198	0.18456	0.29999	0.42581	0.68599	0.89390
k^s	225	$3\,375$	$50\,625$	$759\,375$	$11\,390\,625$	$170\,859\,375$	$214\,358\,881$	$282\,475\,249$	$244\,140\,625$	$43\,046\,721$	$1\,048\,576$
$\lvert\mathcal{P}^s_\varepsilon\rvert$	176	$1\,763$	$17\,600$	$163\,171$	$1\,422\,078$	$12\,488\,175$	$8\,419\,312$	$2\,761\,801$	$533\,026$	$4\,734$	21

TAB. 7.6. Comparaison, dans quelques cas donnés par une dimension s et un poids commun $\varepsilon = \hat{M}^s_k$, des cardinalités respectives de la partition $\mathcal{P}^s_\varepsilon$ et de la grille extensible correspondante (voir section 7.2).

7.4.5 Décompte des points. Pour tout paramètre de précision $\varepsilon \in (0,1)$, l'algorithme 7.6 fournit une partition $\mathcal{P}^s_\varepsilon$ du cube unité I^s telle que $W(\mathcal{P}^s_\varepsilon) = \varepsilon$. Pour une séquence donnée x de n points dans le cube unité $\bar{I}^s$, il nous reste donc à calculer les bornes

$$C(\mathcal{P}^s_\varepsilon, x) = \max_{P \in \mathcal{P}^s_\varepsilon} \max \left\{ \left| \frac{A(P^-, x)}{n} - \lambda(P^-) \right|, \left| \frac{A(P^+, x)}{n} - \lambda(P^+) \right| \right\}$$

et

$$B(\mathcal{P}^s_\varepsilon, x) = \max_{P \in \mathcal{P}^s_\varepsilon} \max \left\{ \frac{A(P^+, x)}{n} - \lambda(P^-),\, \lambda(P^+) - \frac{A(P^-, x)}{n} \right\}$$

pour la discrépance $D^*_n(x)$. En utilisant la notation

$$\mathcal{R}^s_\varepsilon = \left\{ P^- : P \in \mathcal{P}^s_\varepsilon \right\} \cup \left\{ P^+ : P \in \mathcal{P}^s_\varepsilon \right\},$$

le problème revient à compter le nombre de points $A(P, x)$ pour tout intervalle $P \in \mathcal{R}^s_\varepsilon$.

Dans la section 7.2.2, la considération de partitions $\mathcal{P}_{z_{s,k}}$ ayant la forme de grilles extensibles a pu être exploitée pour simplifier l'exécution de cette tâche et conduire à un algorithme de complexité $O(n \log(|\mathcal{P}_{z_{s,k}}|) + 2^s |\mathcal{P}_{z_{s,k}}|)$. Soulignons le fait que c'est dans le logarithme de cette expression que réside tout l'intérêt de cette approche, car il permet d'aborder le calcul de bornes pour la discrépance de séquences de grande taille (voir table 7.3).

Pour les partitions plus générales $\mathcal{P}^s_\varepsilon$ considérées ici, il n'a pas été possible de faire aussi bien au niveau de la séparation des difficultés causées d'un côté par la taille de la séquence et de l'autre par celle de la partition. En premier lieu, mentionnons le fait que le problème du décompte des points énoncé ci-dessus peut être résolu directement, par énumération des n points pour chacun des $2|\mathcal{P}^s_\varepsilon|$ intervalles concernés, mais que la complexité de l'algorithme obtenu est $O(n|\mathcal{P}^s_\varepsilon|)$. Il est toutefois possible de réduire cet effort à $O((\log n)^s |\mathcal{P}^s_\varepsilon|)$ en utilisant une technique de comptage basée sur la notion d'arbre d'intervalles. Comme nous le verrons, la structure particulière de la partition $\mathcal{P}^s_\varepsilon$ peut alors être exploitée de manière à accélérer considérablement le processus.

Afin de motiver l'utilisation d'arbres d'intervalles dans la méthode de comptage spécialisée présentée dans cette section, on illustre (à l'avance) son efficacité à travers une expérience numérique dont les résultats sont donnés dans la table 7.7. Il s'agit du calcul des bornes $C(\mathcal{P}^7_{0.1}, x)$ et $B(\mathcal{P}^7_{0.1}, x)$ pour la discrépance d'une séquence de n points dans $\bar{I}^7$. Comme on pouvait s'y attendre, on observe que pour la méthode par énumération directe, le temps de calcul par point est pratiquement constant (la petite tendance vers le haut lorsque n croît étant sans doute due à des effets liés à la mémoire-cache). Par contre, on constate qu'en effectuant le décompte des points à l'aide de la méthode basée sur les arbres d'intervalles, l'effort par point est non seulement moindre, mais qu'il décroît lorsque n augmente.

Signalons au passage que l'utilisation de la méthode de comptage présentée ci-dessous ne requiert pas le stockage de la partition $\mathcal{P}^s_\varepsilon$. En effet, les valeurs de $A(P^-, x)$ et $A(P^+, x)$ peuvent être calculées et les bornes pour la discrépance $C(\mathcal{P}^s_\varepsilon, x)$ et $B(\mathcal{P}^s_\varepsilon, x)$ mises à jour au fur et à mesure de la génération des intervalles $P \in \mathcal{P}^s_\varepsilon$ par l'algorithme 7.6.

n	50	100	500	1 000	5 000
comptage par énumération directe	22.0	22.6	23.7	26.8	26.8
comptage à l'aide d'arbres d'intervalles	15.0	11.0	5.90	4.95	3.46

TAB. 7.7. Comparaison du temps marginal de calcul (en secondes par point) nécessaire à la détermination des bornes $C(\mathcal{P}^7_{0.1}, x)$ et $B(\mathcal{P}^7_{0.1}, x)$ pour la discrépance de n points d'une suite de Faure à l'aide de deux techniques de comptage différentes.

La notion d’arbre d’intervalles a été proposée simultanément par Lucker [Luc78] et Bentley [Ben79]. Des descriptions plus détaillées sont disponibles dans les ouvrages de Mehlhorn [Meh84] et de Preparata et Shamos [PS85]. La définition considérée ci-dessous diffère légèrement de la version classique, mais les principes sous-jacents sont exactement les mêmes.

DÉFINITION 7.19 Pour une séquence de points $x = \{x^1, \dots, x^n\} \subset \bar{I}^s$, un *arbre d’intervalles en dimension* s se définit (par induction) de la manière suivante. Il s’agit d’un arbre binaire ordonné d’après les valeurs de la première composante des points qu’il contient. De plus, pour $s \geq 2$, chaque sommet d’un tel arbre est doté d’un pointeur vers la racine d’un arbre d’intervalles en dimension $s-1$ défini sur les projections des $s-1$ dernières composantes des points se trouvant dans son sous-arbre de gauche, y compris lui-même (voir figure 7.9).

En omettant le terme constant qui croît exponentiellement avec la dimension, $O\left(n(\log n)^s\right)$ est à la fois la complexité d’un algorithme de construction et l’espace-mémoire requis pour le stockage de cette structure de données (voir Lucker [Luc78]). Heureusement, dans notre cas particulier, l’exécution complète de cette tâche n’est pas nécessaire. En effet, il suffit d’une petite partie de l’arbre d’intervalles pour résoudre le problème du décompte des points inclus dans les différents intervalles de $\mathcal{R}^s_\varepsilon$.

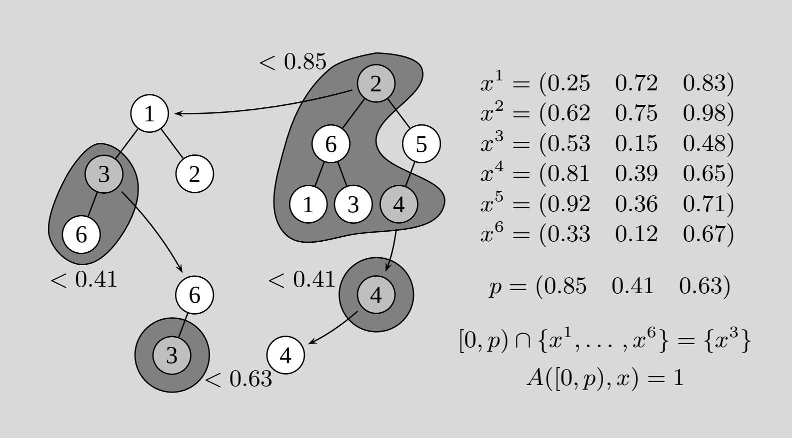


FIG. 7.9. Portion d’un arbre d’intervalles en dimension $s = 3$ correspondant à une séquence de 6 points $x = \{x^1, \dots, x^6\} \subset \bar{I}^3$ et exploration induite par un intervalle $[0,p)$ spécifié.

Le processus menant à la détermination de $A(P,x)$ pour un intervalle $P = \prod_{i=1}^{s} [0,p_i) \in \mathcal{R}^s_\varepsilon$ donné peut être décrit comme suit. Partant de la racine de l’arbre d’intervalles en dimension s, on commence par rechercher p_1. Considérant le chemin parcouru (on note au passage qu’il est de longueur $O(\log n)$ pour un arbre binaire équilibré), on appelle $y(p,1)$ l’ensemble des sommets visités qui sont suivis d’un mouvement sur la droite (ou d’une « intention » de mouvement sur la droite, une fois parvenu au niveau des feuilles). On remarque que l’ensemble des points de la séquence x dont la première composante est inférieure à p_1 est en bijection avec l’ensemble des sommets inclus dans $y(p,1)$ et leurs sous-arbres de gauche. Dans l’exemple de la figure 7.9, on obtient $y(p,1) = \{2,4\}$ et on vérifie que le sous-ensemble des points ayant leur première coordonnée inférieure à $p_1 = 0.85$ est bien $\{1,2,3,4,6\}$ (par abus de langage, on identifie les sommets, les points et leurs index).

Puis, successivement pour chacune des composantes $d \in \{2, \dots, s\}$, il suffit de suivre les pointeurs associés aux différents sommets de $y(p,d-1)$ et de rechercher p_d dans les arbres d’intervalles (en dimension $s-d+1$) correspondants. Comme précédemment, considérant les chemins parcourus, on note

$y(p,d)$ l'ensemble des sommets visités qui sont suivis d'un mouvement sur la droite. Dans l'exemple de la figure 7.9, on obtient $y(p,2)=\{3,4\}$ et $y(p,3)=\{3\}$.

Pour tout $d\in\{1,\ldots,s\}$, l'ensemble des points de la séquence ayant leurs d premières composantes plus petites que $p_1,\ldots,p_d$ (respectivement) est donc en bijection avec les sommets inclus dans $y(p,d)$ et leurs sous-arbres de gauche. Ainsi, si l'on stocke en chaque sommet de tout arbre d'intervalles en dimension 1 la cardinalité de son sous-arbre de gauche (y compris lui-même), on obtient $A(P,x)$ en sommant ces nombres pour tout élément de $y(p,s)$. Tout chemin impliqué dans ce processus étant de longueur $O(\log n)$ (pour des arbres binaires équilibrés), cette valeur se calcule en $O\left((\log n)^s\right)$.

Notons que l'obtention de cette complexité ne repose sur aucune hypothèse particulière quant à la nature des requêtes. Par contre, en exploitant judicieusement la structure spécifique de la partition $\mathcal{P}_\varepsilon^s$, la tâche consistant à déterminer $A(P,x)$ pour tout $P\in\mathcal{P}_\varepsilon^s$ se simplifie radicalement. Notons tout d'abord que, d'après les résultats de la section 7.4.3, le dernier intervalle ajouté dans $\mathcal{P}_\varepsilon^s$ est

$$\left[\gamma^{I^s},\beta^{I^s}\right).$$

Ainsi, si l'on note $S(P)$ l'intervalle inséré dans la partition juste après $P\in\mathcal{P}_\varepsilon^s\setminus\{[\gamma^{I^s},\beta^{I^s})\}$ ($S(P)$ est appelé le *successeur* de P), le théorème 7.12 nous assure que

$$\alpha^P\overset{\text{lex}}{<}\alpha^{S(P)}\quad\text{et}\quad\beta^P\overset{\text{lex}}{<}\beta^{S(P)}.$$

Soit ω^{α^P} (respectivement ω^{β^P}), le plus petit $i\in\{1,\ldots,s\}$ tel que $\alpha_i^{S(P)}>\alpha_i^P$ (respectivement $\beta_i^{S(P)}>\beta_i^P$). Pour tout $P\in\mathcal{P}_\varepsilon^s\setminus\{[\gamma^{I^s},\beta^{I^s})\}$, la relation lexicographique susmentionnée implique que

$$\alpha_i^{S(P)}=\alpha_i^P\text{ pour tout }i<\omega^{\alpha^P}\quad\text{et}\quad\beta_i^{S(P)}=\beta_i^P\text{ pour tout }i<\omega^{\beta^P}.$$

Ce fait va s'avérer précieux par la suite, mais commençons par montrer comment ω^{α^P} et ω^{β^P} peuvent être déterminés (au moment de la génération de $S(P)$) sans le moindre effort de calcul supplémentaire :

THÉORÈME 7.20 *Pour tout $P\in\mathcal{P}_\varepsilon^s\setminus\{[\gamma^{I^s},\beta^{I^s})\}$, on a*

$$\omega^{\alpha^P}=\omega^{\beta^P}=\omega^P,$$

où

$$\omega^P=\begin{cases} j-1 & \textit{si } S(P) \textit{ est directement issu d'une décomposition dans la direction } j,^{3}\\ s & \textit{sinon, c'est-à-dire lorsque } S(P) \textit{ est un intervalle de la forme } [\gamma^*,\beta^*).\end{cases}$$

PREUVE. On remarque tout d'abord que si $\delta^{I^s}\le\varepsilon$, on a

$$\mathcal{P}_\varepsilon^s=\left\{Q_1^{I^s},\ldots,Q_s^{I^s},[\gamma^{I^s},\beta^{I^s})\right\}.$$

Considérant l'expression (37), on en déduit directement que

$$\omega^{\alpha^{Q_j^{I^s}}}=\omega^{\beta^{Q_j^{I^s}}}=j,\text{ pour tout }j\in\{1,\ldots,s\}.$$

Ce cas particulier étant réglé, on suppose désormais que $\delta^{I^s}>\varepsilon$. Il en découle que les intervalles $Q_1^{I^s},\ldots,Q_s^{I^s}$ sont décomposés à leur tour et qu'ainsi, il existe un intervalle R dont P est issu après deux décompositions (en termes informels, P possède un grand-père R, comme indiqué sur la figure 7.10). Fondamentalement, il n'existe que deux possibilités (mutuellement exclusives) :

---

[3] Le cas $j=1$ ne peut pas se produire. En effet, l'unique intervalle de la partition qui soit issu d'une décomposition dans la direction 1 n'est autre que le tout premier. On a donc $j\in\{2,\ldots,s\}$.

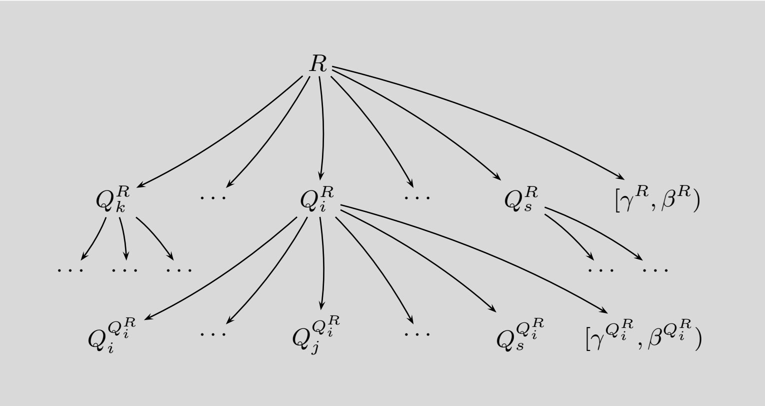


FIG. 7.10. Décomposition d'un intervalle R sur deux générations.

1° Si $P = Q_j^{Q_i^R}$ pour une paire de directions $i \leq j \leq s$, alors son successeur $S(P)$ dans la partition est forcément (voir les dernières lignes de l'algorithme 7.6)

$$S(P) = \begin{cases} Q_{j+1}^{Q_i^R} & \text{si } j < s, \\ \left[\gamma^{Q_i^R}, \beta^{Q_i^R}\right) & \text{si } j = s. \end{cases}$$

Considérant l'expression (37), on en déduit que dans les deux cas

$$\omega^{\alpha^P} = \omega^{\beta^P} = j.$$

2° Si $P = \left[\gamma^{Q_i^R}, \beta^{Q_i^R}\right)$ pour une certaine direction $i \leq s$, alors pour $d \leq i$ on a

$$\alpha_d^P = \gamma_d^{Q_i^R} = \begin{cases} \alpha_d^{Q_i^R} & = \gamma_d^R & \text{si } d < i, \\ \delta^{Q_i^R} \beta_i^{Q_i^R} & = \delta^{Q_i^R} \gamma_i^R & \text{si } d = i \end{cases}$$

et

$$\beta_d^P = \beta_d^{Q_i^R} = \begin{cases} \beta_d^R & \text{si } d < i, \\ \gamma_i^R & \text{si } d = i. \end{cases}$$

On distingue les deux cas suivants :

- ▷ Si $i = s$, alors $S(P) = [\gamma^R, \beta^R)$ et on obtient directement

$$\omega^{\alpha^P} = \omega^{\beta^P} = s.$$

- ▷ Si $i < s$, alors $S(P)$ est soit $Q_{i+1}^{Q_{i+1}^R}$ (comme $W(Q_{i+1}^R) = W(Q_i^R) > \varepsilon$, on sait que Q_{i+1}^R doit forcément être décomposé à son tour) ou l'un de ses descendants $Q_{i+1}^{\cdot^{\cdot^{Q_{i+1}^R}}}$ obtenu après une série de décompositions dans la direction $i + 1$. De toute manière, on a

$$\begin{cases} \alpha_d^{S(P)} = \gamma_d^R, \\ \beta_d^{S(P)} = \beta_d^R, \end{cases} \quad \text{pour tout } d \leq i.$$

On en déduit facilement que

$$\omega^{\alpha^P} = \omega^{\beta^P} = i.$$

□

La majeure partie des bénéfices que l'on peut espérer tirer de l'exploitation de la structure de la partition $\mathcal{P}^s_\varepsilon$ apparaît dans le corollaire suivant :

COROLLAIRE 7.21 *Pour tout intervalle* $P \in \mathcal{P}^s_\varepsilon \setminus \{[\gamma^{I^s}, \beta^{I^s})\}$ *tel que* $\omega^P > 1$*, on a*

$$y\Big(\alpha^{S(P)}, i\Big) = y\left(\alpha^P, i\right) \quad \textit{et} \quad y\Big(\beta^{S(P)}, i\Big) = y\left(\beta^P, i\right), \quad \textit{pour tout } i \in \{1, \ldots, \omega^P - 1\}.$$

En d'autres termes, pour la plupart des intervalles $P \in \mathcal{P}^s_\varepsilon$, une partie importante du travail effectué pour déterminer $A(P^-, x)$ et $A(P^+, x)$ peut être réutilisée pour calculer $A(S(P)^-, x)$ et $A(S(P)^+, x)$. Plus précisément, l'exploration de l'arbre d'intervalles peut être court-circuitée en commençant par la recherche de $\alpha^{S(P)}_{\omega^P}$ et $\beta^{S(P)}_{\omega^P}$ au niveau des arbres d'intervalles de dimension $s - \omega^P - 1$ indiqués par les pointeurs associés aux sommets contenus dans les ensembles $y(\alpha^P, \omega^P - 1)$ et $y(\beta^P, \omega^P - 1)$. On économise ainsi les $\omega^P - 1$ premières phases du processus correspondant à la détermination de la paire de sous-ensembles constitués des points de la séquence x dont les $\omega^P - 1$ premières composantes sont respectivement inférieures à $(\alpha^P_1, \ldots, \alpha^P_{\omega^P-1})$ et à $(\beta^P_1, \ldots, \beta^P_{\omega^P-1})$.

De plus, étant donné que $\alpha^{S(P)}_{\omega^P} > \alpha^P_{\omega^P}$, une partie de l'ensemble $y(\alpha^P, \omega^P)$ peut généralement être réutilisée pour la construction de $y(\alpha^{S(P)}, \omega^P)$. En effet, si la première partie du chemin correspondant à la recherche de $\alpha^P_{\omega^P}$ dans un arbre pointé par un sommet de $y(\alpha^P, \omega^P - 1)$ est exclusivement constituée de mouvements sur la droite, il est bien clair que ce début de chemin est identique pour $\alpha^{S(P)}_{\omega^P}$ et n'a donc pas besoin d'être recalculé[4] (bien entendu, le raisonnement similaire obtenu en remplaçant ci-dessus α par β est également valide). Spécialisant cet argument dans le cas particulier où $\omega^P = s$, on s'aperçoit que la somme partielle des cardinalités des sous-arbres de gauche (y compris leur racine) le long d'un tel chemin vers la droite peut être stockée et n'a donc pas à être réévaluée.

En pratique, ω^P est très fréquemment égal à s (une fois sur deux pour donner un ordre de grandeur) et presque toujours très proche de cette valeur. Ainsi, comparativement à la procédure d'utilisation standard des arbres d'intervalles, on obtient une accélération spectaculaire du processus de comptage. Il n'a pas été possible d'établir un résultat théorique à ce niveau, mais la complexité empirique de l'algorithme obtenu pour le calcul des bornes $C(\mathcal{P}^s_\varepsilon, x)$ et $B(\mathcal{P}^s_\varepsilon, x)$ est bien meilleure que $O((\log n)^s |\mathcal{P}^s_\varepsilon|)$.

Dans notre mise en œuvre de cette méthode, l'arbre d'intervalles associé à une séquence x n'est pas construit complètement. Les parties requises sont simplement ajoutées au fur et à mesure de la génération de la partition $\mathcal{P}^s_\varepsilon$ et de la considération des requêtes correspondantes. Ainsi, l'économie en espace-mémoire et en temps de calcul est fort appréciable. De plus, il n'est généralement pas nécessaire de conserver jusqu'à la fin du déroulement de l'algorithme toutes les portions construites de l'arbre. En effet, dans la plupart des cas, il est possible de libérer l'espace occupé par certaines parties qui, compte tenu de l'ordre lexicographique d'arrivée des requêtes, ne seront de toute manière plus utilisées[5].

REMARQUE 7.22 Dans notre définition d'arbre d'intervalles, nous avons omis de spécifier quel type d'arbre binaire utiliser. Pour des requêtes uniformément distribuées, le choix optimal est donné par les arbres binaires équilibrés[6]. Par contre, celles de l'ensemble $\mathcal{R}^s_\varepsilon$ étant concentrées au voisinage du « coin supérieur-droit » du cube unité I^s (tout particulièrement en dimension élevée), la considération d'arbres binaires qui penchent vers la gauche paraît nettement plus judicieuse.

Par exemple, les résultats de la table 7.7 ont été obtenus à l'aide d'un paramètre de déséquilibre fixé à $4/5$. On entend par là que pour tout sommet d'un des arbres binaires utilisés, on trouve 4 fois plus de

---

[4] Dans l'exemple de la figure 7.9, si $\omega^{[0,p)} = 1$, on peut prédire que l'on va obtenir $\{2\} \subset y(S([0,p)), 1)$.

[5] Il s'agit de branches entières situées sur la partie gauche de l'arbre d'intervalles en dimension s (et de tous les arbres de dimension inférieure qui y sont attachés). Le cas se produit lorsque $\omega^P = 1$ et que le début de chemin exclusivement constitué de mouvements vers la droite est plus long pour la recherche de $\alpha^{S(P)}_1$ que pour celle de α^P_1 (de même pour $\beta^{S(P)}_1$ et β^P_1).

[6] Rappelons qu'un arbre binaire est dit *équilibré* si, pour tout sommet qu'il contient, les cardinalités respectives de son sous-arbre de gauche et de son sous-arbre de droite diffèrent d'au plus 1.

sommets dans son sous-arbre de gauche que dans son sous-arbre de droite. Ce fait permet de raccourcir considérablement la longueur des chemins de recherche pour une très large majorité des requêtes de $\mathcal{R}_\varepsilon^s$ (celles qui sont situées au voisinage du « coin supérieur-droit » du cube) et de la rallonger dans les quelques cas restants. Globalement, le bénéfice est très net au niveau du temps de calcul, mais le prix à payer se situe au niveau de l'espace-mémoire utilisé qui s'avère bien plus conséquent pour une grande valeur du paramètre de déséquilibre.

Du point de vue de la complexité, la considération de tels arbres ne constitue pas un problème. En effet, tant que pour tout sommet le rapport des cardinalités des sous-arbres de droite et de gauche est une constante, on a la garantie que la hauteur totale de l'arbre reste $O(\log n)$.

REMARQUE 7.23 Finalement, mentionnons le fait que la borne inférieure

$$C(\mathcal{P}_\varepsilon^s, x) = \max_{P \in \mathcal{R}_\varepsilon^s} \left| \frac{A(P,x)}{n} - \lambda(P) \right|$$

pour la discrépance de la séquence x peut être facilement améliorée. En effet, il suffit d'observer que toute composante p_j d'un intervalle $P = \prod_{j=1}^s [0, p_j)$ peut être arrondie (vers le haut ou vers le bas suivant le signe de $\lambda(P) - A(P,x)/n$) jusqu'à la coordonnée x_j^i la plus proche (ou éventuellement à 0 ou 1 lorsque P jouxte la facette correspondante du cube unité) sans changer la valeur de $A(P,x)$. Clairement, ces intervalles de volume optimisé conduisent à une meilleure borne inférieure pour $D_n^*(x)$.

7.5 Expériences numériques

La méthode de décomposition décrite dans la section 7.4 a été implémentée et ses performances évaluées à travers quelques expériences numériques. Comme dans la section 7.3, on commence par calculer des bornes pour la discrépance de quelques $(0, m, s)$-réseaux de Faure en base b. On aborde ensuite la question de la comparaison de différents types de séquences dans le cas particulier de la dimension 7. On termine par une expérience illustrant le fait que cette méthode peut éventuellement être utilisée pour se forger une opinion sur des questions théoriques a priori difficilement accessibles.

7.5.1 Bornes pour la discrépance. Notre méthode a permis d'établir des bornes inférieures et supérieures pour la discrépance de quelques $(0, m, s)$-réseaux de Faure en base b (obtenus à l'aide du générateur `GrayFaure` présenté dans la section 4.8.4). Ces résultats sont donnés dans la table 7.8.

s	b	m	$n = b^m$	$\varepsilon_\mathcal{C}$	$\varepsilon_\mathcal{B}$	$D_n^*(x) \in [C(\mathcal{P}_{\varepsilon_\mathcal{C}}^s, x), B(\mathcal{P}_{\varepsilon_\mathcal{B}}^s, x)]$
7	7	2	49	0.05	0.05	$[0.269011, 0.295125]$
		3	343	0.59	0.05	$[0.129832, 0.168598]$
		4	2'401	0.6	0.05	$[0.030518, 0.074176]$
10	11	2	121	0.6	0.125	$[0.248508, 0.337404]$
		3	1'331	0.58	0.15	$[0.093028, 0.220886]$
12	13	2	169	0.61	0.18	$[0.265266, 0.396727]$
		3	2'197	0.58	0.22	$[0.096713, 0.283217]$
15	17	2	289	0.59	0.26	$[0.256021, 0.455008]$
		3	4'913	0.57	0.35	$[0.085855, 0.416446]$
20	23	2	529	0.58	0.5	$[0.259366, 0.722188]$
		3	12'167	0.55	0.5	$[0.080737, 0.509607]$
100	101	1	101	0.99	0.96	$[0.954159, 0.961973]$

TAB. 7.8. Intervalle $[C(\mathcal{P}_{\varepsilon_\mathcal{C}}^s, x), B(\mathcal{P}_{\varepsilon_\mathcal{B}}^s, x)]$ pour la discrépance de quelques réseaux de Faure.

Il s'agit des meilleures bornes connues à ce jour pour la discrépance de ces réseaux[7]. En effet, comme nous l'avons déjà mentionné dans la section 7.3, la majoration fournie par le théorème 6.1 est supérieure à 1 pour les séquences en question et les méthodes de calcul exact (voir chapitre 5) s'avèrent inapplicables dans ces cas-là. D'autre part, hormis pour le $(0, 2, 7)$-réseau en base 7, on remarque que les bornes supérieures obtenues sont meilleures que celles de la table 7.3.

Pour chaque réseau considéré, le paramètre de précision $\varepsilon_{\mathcal{B}}$ associé à la borne supérieure $B(\mathcal{P}^s_{\varepsilon_{\mathcal{B}}}, x)$ a été fixé à la plus petite valeur possible de manière à ce que le temps de calcul ne dépasse pas quelques jours. Cependant, mis à part dans un cas, la borne inférieure $C(\mathcal{P}^s_{\varepsilon_{\mathcal{B}}}, x)$ correspondante n'est pas donnée dans la table 7.8. En effet, de meilleures minorations ont été obtenues (le plus souvent en quelques secondes de calcul seulement) pour des valeurs du paramètre de précision $\varepsilon_{\mathcal{C}}$ plus grandes que $\varepsilon_{\mathcal{B}}$. Bien entendu, cette approche préserve la garantie fournie par le théorème 7.3 d'aboutir à un intervalle de largeur au plus $\varepsilon_{\mathcal{B}}$.

Par ailleurs, signalons qu'en choisissant des valeurs du paramètre $\varepsilon_{\mathcal{B}}$ légèrement plus grandes que celles de la table 7.8, le temps de calcul associé s'avère nettement plus court. Par exemple, pour le $(0, 3, 15)$-réseau en base 17 considéré, la détermination de la borne supérieure 0.416446 avec $\varepsilon = 0.35$ a nécessité 112 heures sur une station de travail dotée d'un Pentium III à 500 MHz, alors qu'avec $\varepsilon = 0.45$ la majoration 0.465755 est obtenue après 2 heures seulement. Ce comportement est lié au fait qu'en première approximation, le temps de calcul est proportionnel à la cardinalité de la partition $\mathcal{P}^s_{\varepsilon}$ qui, comme nous l'avons discuté dans la section 7.4.4, semble croître comme s/ε^s.

Finalement, mentionnons le fait que notre méthode permet également de calculer des bornes assez précises pour la discrépance de séquences en petite dimension. Par exemple, pour un $(0, 20, 2)$-réseau de Faure en base 2 (plus d'un million de points), on obtient l'intervalle $[0.0000071, 0.0000136]$ en utilisant le paramètre de précision $\varepsilon = 0.000007$ (en revanche, dans ce cas particulier, le théorème 7.3 fournit une majoration de meilleure qualité que la nôtre : 0.0000105).

7.5.2 Comparaison de séquences. À notre connaissance, la littérature ne contient aucun résultat sur l'expérience, pourtant naturelle, qui consiste à comparer la discrépance effective de séquences finies extraites de suites classiques. Notre technique de calcul d'intervalles a permis d'effectuer une première tentative en ce sens. Nous avons choisi d'évaluer, en dimension $s = 7$, la discrépance des segments initiaux de trois suites à discrépance faible et de deux générateurs pseudo-aléatoires :

- une suite de Halton en bases $2, 3, 5, 7, 11, 13,$ et 17 (voir section 4.2) ;
- une suite de Sobol permutée à l'aide d'un code de Gray (générée à partir de la mise en œuvre de Bratley et Fox [BF88] décrite dans la section 4.8.1) ;
- une suite de Faure permutée à l'aide d'un code de Gray en base 7 (générée en utilisant l'implémentation `GrayFaure` décrite dans la section 4.8.4) ;
- le générateur (15) combiné à congruences linéaires multiples MRG32k3a de L'Ecuyer [L'E99] ;
- le générateur pseudo-aléatoire `Rand()` de la librairie C standard associée au compilateur `gcc`[8].

Les intervalles obtenus sont représentés dans la figure 7.11 (les paramètres de précision ε utilisés ont été fixés à 0.05 pour $n \in \{30, \ldots, 100\}$ et à 0.04 pour $n \in \{150, 200, 250\}$). En conclusion, pour $n \leq 100$, les séquences de Sobol et Faure présentent les discrépances les plus faibles, mais à partir de 150 points, les échantillons provenant de la suite de Halton semblent tout aussi performants. Notons que cette expérience est reconsidérée dans la section 8.5.3.

---

[7] Certaines de ces discrépances étant calculées au chapitre 8, cette affirmation est partiellement caduque.

[8] Sa période étant de longueur 2^{15}, une telle relique ne devrait plus exister sur un système informatique moderne.

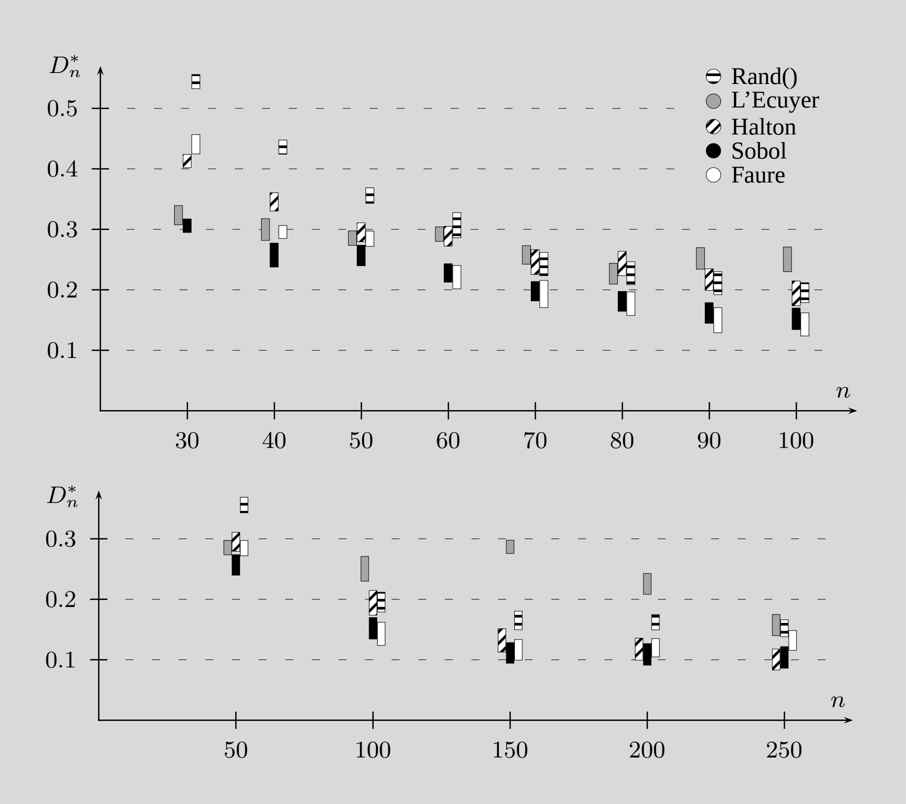


FIG. 7.11. Intervalles pour la discrépance de cinq types de séquences en dimension 7.

Les suites de Halton, Sobol et Faure étant à discrépance faible, elles satisfont

$$D_n^*(x) \leq C\,n^{-1}(\log n)^s + O\left(n^{-1}(\log n)^{s-1}\right). \tag{44}$$

Comme nous l'avons dit dans la section 4.7, on rencontre parfois dans la littérature des affirmations du type : « en vue d'une application de la méthode de quasi-Monte-Carlo, une suite de Faure constitue un meilleur choix qu'une suite de Halton car elle présente une plus petite constante C dans l'expression (44) ». Ce raisonnement est incorrect pour au moins trois raisons :

1° les constantes en question (voir chapitre 4) ne sont que des bornes supérieures ;

2° personne n'a jamais prouvé que $D_n^*(x) = \Omega(n^{-1}(\log n)^s)$ pour les trois suites en question ;

3° même si quelqu'un réussissait à démontrer une telle propriété, la constante impliquée ne décrirait que le régime asymptotique et ne constituerait donc pas une mesure représentative de la discrépance d'une séquence finie susceptible d'être utilisée en pratique.

Pour les suites de Halton, Sobol et Faure en dimension 7, les meilleures constantes C connues pour la majoration asymptotique (44) sont respectivement 17.3, 2.6 et 0.0041 (voir table 4.2, page 42). Les résultats donnés dans la figure 7.11 montrent clairement que, du moins pour des tailles d'échantillon $n \leq 250$, ces constantes ne sont absolument pas représentatives de la discrépance effective des séquences en question.

Au contraire, nous pensons que ces valeurs reflètent surtout la complexité du problème de leur détermination et, d'une manière plus générale, le peu de place que la difficulté intrinsèque du cadre mathématique dans lequel s'inscrit la définition des différentes suites concernées laisse au chercheur pour exercer son talent.

Une séquence pseudo-aléatoire dans I^s est censée imiter un échantillon de variables aléatoires i.i.d. $U(I^s)$. L'expérience considérée ici pour deux générateurs particuliers concerne une seule réalisation du processus et n'a donc rien à voir avec l'estimation de la discrépance moyenne d'un échantillon aléatoire de n points dans I^7. Notre but n'est pas non plus d'homologuer ou d'invalider tel ou tel générateur pseudo-aléatoire, mais d'indiquer que notre méthode de calcul d'intervalles pourrait éventuellement être utilisée dans un tel but si une statistique de test correspondante était développée. D'autre part, pour ne pas laisser planer de doute, précisons que dans les deux cas considérés ici, la séquence de 30 points est un sous-ensemble de celle de 40 et ainsi de suite.

Les ensembles de points provenant du générateur de L'Ecuyer semblent présenter (comparativement aux séquences issues des suites de Halton, Sobol et Faure) une discrépance assez grande, ainsi qu'un comportement globalement décroissant, quoique localement irrégulier. Ces différentes observations sont plutôt rassurantes pour des échantillons censés imiter une séquence aléatoire. Bien entendu, un tel préavis demande confirmation par réplication de l'expérience.

En revanche, pour le générateur `Rand()`, la discrépance décroît régulièrement et presque aussi bien qu'une séquence à discrépance faible. A priori, un tel comportement, quoique vraisemblable pour une réalisation particulière de l'expérience (le germe 12345 a été utilisé), paraît trop « lisse » pour provenir d'un bon générateur pseudo-aléatoire. Des réplications de l'expérience avec d'autres valeurs du germe seraient bien sûr nécessaires avant de pouvoir envisager sérieusement une telle conclusion.

7.5.3 Sur les ensembles minimaux. Le théorème 1.23 stipule que pour toute valeur $d \in (0, 1/2)$, la taille $n(s, d)$ de la plus courte séquence dans $\bar{I}^s$ présentant une discrépance inférieure ou égale à d croît linéairement avec s et au pire de manière quadratique avec d^{-1}. Bien que leur existence soit prouvée, la construction de tels ensembles minimaux est une question ouverte et vraisemblablement très difficile. Au premier abord, des échantillons de points provenant de suites à discrépance faible classiques constituent des candidats naturels pour l'étude de ce problème. En particulier, il serait intéressant de savoir si, pour une famille de suites donnée, la taille des ensembles minimaux correspondants présente également une croissance linéaire avec la dimension.

Nous nous proposons de tester cette question dans le cas particulier des segments initiaux des suites de Faure permutées à l'aide d'un code de Gray (générés en utilisant l'implémentation `GrayFaure` décrite dans la section 4.8.4) pour $d = 0.45$ et les dimensions $s \in \{4, \ldots, 12\}$. Notons que, le théorème 1.23 étant un résultat asymptotique, il n'est pas garanti que $n(s, d)$ présente un comportement linéaire pour d'aussi petites valeurs de s. De manière similaire, même si la croissance de la taille des échantillons minimaux des suites de Faure permutées s'avérait également linéaire, rien n'empêche que, localement, les résultats paraissent très différents (pour $s \in \{4, \ldots, 12\}$ par exemple). Ces réserves préliminaires étant formulées, l'expérience n'en demeure pas moins potentiellement révélatrice et, par conséquent, mérite d'être tentée.

Notre méthode ne fournissant que des bornes inférieures et supérieures pour la discrépance d'une séquence de points donnée, il n'a en général pas été possible d'obtenir des intervalles suffisamment étroits pour déterminer de manière exacte la longueur minimale $n_{\min}$ du segment initial de la suite étudiée présentant une discrépance inférieure ou égale à 0.45. En fin de compte, nous avons dû nous contenter de construire des ensembles de candidats contenant la vraie valeur de la taille en question (voir figure 7.12).

Compte tenu des réserves mentionnées plus haut, la prudence est de mise pour l'analyse des résultats de cette expérience. Toutefois, pour $s \in \{4, \ldots, 12\}$, le comportement obtenu ne paraît pas linéaire en la dimension, mais plutôt de la forme $s \log s$, voire quadratique. Ainsi, pour autant qu'une conclusion puisse être tirée, il semble peu probable que les échantillons minimaux de cette famille de suites mène à la croissance linéaire désirée. Bien entendu, cette assertion ne repose que sur l'intuition et non sur un quelconque argument solide. Cependant, l'objectif de cette expérience était surtout d'illustrer le fait que notre méthode pouvait permettre de se forger une première opinion sur une telle question.

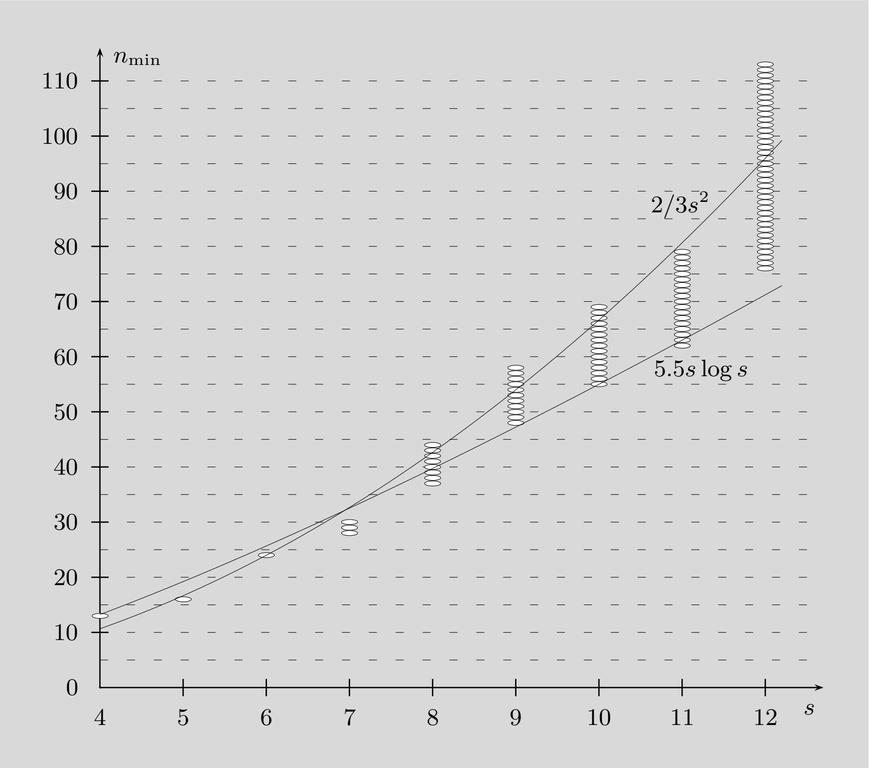


FIG. 7.12. Ensembles de valeurs candidates pour la taille minimale $n_{\min}$ d'une séquence issue d'une suite de Faure permutée à l'aide d'un code de Gray présentant une discrépance inférieure ou égale à 0.45 pour $s \in \{4, \ldots, 12\}$.

CHAPITRE 8

Une approche par programmation linéaire en nombres entiers

Il est possible de ramener la question du calcul de la discrépance d'une séquence x de n points dans $\bar{I}^s$ à la résolution d'une famille de programmes linéaires en nombres entiers. Cette affirmation peut paraître surprenante au premier abord, vu l'évidente non-linéarité de la fonction discrépance locale (4). Cependant, il suffit d'une décomposition en $2n$ sous-problèmes suivie d'une légère transformation pour aboutir à une reformulation de ce type.

Bien que certaines similitudes soient apparentes, cette approche s'avère passablement différente des techniques présentées au chapitre 5. Fondamentalement, on peut la voir comme une manière d'aborder la discrétisation de Niederreiter (21) avec l'ambition d'énumérer implicitement une partie aussi importante que possible de la grille $\mathcal{Q}$ engendrée par les points de la séquence et les bords du cube. D'autre part, elle a en commun avec l'algorithme de Dobkin et Eppstein l'exploitation de la décomposition du problème en $2n+2$ parties, à savoir la question de la détermination, pour tout $k \in \{0, \ldots, n\}$, des intervalles de $\mathcal{I}_s^*$ de plus petit et de plus grand volume contenant exactement k parmi les n points de la séquence.

Cette méthode a la particularité de mener en temps polynomial à un intervalle initial non trivial (mais hélas de qualité médiocre) pour la discrépance de la séquence. Par la suite, les bornes en question sont progressivement améliorées à l'aide de techniques de programmation linéaire en nombres entiers. Il est à noter que le processus peut être stoppé à tout instant (fournissant alors un intervalle dont la qualité reflète l'effort de calcul consenti jusque-là) ou poursuivi jusqu'à l'obtention de la valeur exacte de la discrépance $D_n^*(x)$. Il s'agit de la première approche offrant une telle alternative à l'utilisateur.

Le contenu de ce chapitre est une version revisitée de notre article [Thi]. Dans la section 8.1, on montre comment le calcul de la discrépance se ramène aux $2n+2$ problèmes de géométrie combinatoire mentionnés ci-dessus. Dans la section 8.2, ces derniers sont reformulés comme des programmes linéaires en nombres entiers et diverses techniques de résolution sont présentées dans la section 8.3. Un processus dynamique aboutissant au calcul de la discrépance est proposé dans la section 8.4 et, pour terminer, les résultats de quelques expériences numériques sont donnés dans la section 8.5.

8.1 Décomposition du problème

Soit une séquence de n points $x = \{x^1, \ldots, x^n\}$ dans le cube unité $\bar{I}^s$. La réécriture suivante de la discrépance (3) s'obtient par conditionnement sur le nombre de points k contenus dans l'intervalle P :

$$\begin{aligned}
D_n^*(x) &= \sup_{P \in \mathcal{I}_s^*} \left| \frac{A(P,x)}{n} - \lambda(P) \right| \\
&= \max_{k \in \{0,\ldots,n\}} \max \left\{ \sup_{\substack{A(P,x)=k \\ P \in \mathcal{I}_s^*}} \left(\lambda(P) - \frac{k}{n} \right), \sup_{\substack{A(P,x)=k \\ P \in \mathcal{I}_s^*}} \left(\frac{k}{n} - \lambda(P) \right) \right\} \\
&= \max \left\{ \max_{k \in \{0,\ldots,n\}} \left(\sup_{\substack{A(P,x)=k \\ P \in \mathcal{I}_s^*}} \lambda(P) - \frac{k}{n} \right), \max_{k \in \{0,\ldots,n\}} \left(\frac{k}{n} - \inf_{\substack{A(P,x)=k \\ P \in \mathcal{I}_s^*}} \lambda(P) \right) \right\}.
\end{aligned}$$

C'est donc tout naturellement que l'on voit apparaître, pour tout $k \in \{0, \ldots, n\}$, la question de la détermination des intervalles de $\mathcal{I}_s^*$ (*i.e.* ancrés à l'origine) de volume minimal et maximal contenant exactement k parmi les n points de la séquence. On note ces problèmes

$$(45) \qquad V_{\min}^k = \inf_{\substack{A(P,x)=k \\ P \in \mathcal{I}_s^*}} \lambda(P) \qquad \text{et} \qquad V_{\max}^k = \sup_{\substack{A(P,x)=k \\ P \in \mathcal{I}_s^*}} \lambda(P).$$

Ainsi, on obtient l'expression suivante pour la discrépance :

$$(46) \qquad D_n^*(x) = \max \left\{ \max_{k \in \{0, \ldots, n\}} \left(V_{\max}^k - \frac{k}{n} \right), \max_{k \in \{0, \ldots, n\}} \left(\frac{k}{n} - V_{\min}^k \right) \right\}.$$

8.2 L'intervalle de volume optimal contenant k points

En dimension $s = 2$, Aggarwal, Imai, Katoh et Suri [AIKS91] ont proposé un algorithme permettant de déterminer l'intervalle de $\mathcal{I}_2^*$ de périmètre minimum contenant k parmi les n points de la séquence. Comme nous le verrons dans la remarque 8.4, ce problème entretient des rapports très étroits avec $V_{\min}^k$. Malheureusement, comme pour la plupart des algorithmes de géométrie combinatoire dans le plan, sa généralisation en dimension quelconque n'est pas envisageable.

À notre connaissance, $V_{\min}^k$ et $V_{\max}^k$ n'ont jamais été étudiés. De plus, bien que ce point ne soit pas prouvé, nous pensons qu'il s'agit de problèmes NP-durs. Ci-dessous, ces questions sont abordées à l'aide de la programmation linéaire en nombres entiers (nous renvoyons le lecteur aux ouvrages de Wolsey [Wol98] et de Cook, Cunningham, Pulleyblank et Schrijver [CCPS97] pour une introduction à ce domaine). Pour parvenir à une telle formulation, on définit la paire de points auxiliaires

$$x^0 = (0, \ldots, 0) \qquad \text{et} \qquad x^{n+1} = (1, \ldots, 1)$$

correspondant respectivement au « coin inférieur-gauche » et « supérieur-droit » du cube unité $\bar{I}^s$. On commence par énoncer deux propositions sur la géométrie des solutions optimales.

PROPOSITION 8.1 *Pour un problème $V_{\min}^k$, l'infimum dans l'expression* (45) *correspond au volume d'un intervalle fermé $\bar{P} = [0, p] = \prod_{j=1}^s [0, p_j]$, où $p = (p_1, \ldots, p_s)$ est un point de la grille induite par la séquence x et les bords du cube unité. En d'autres termes, on a*

$$p_j \in \{x_j^0, \ldots, x_j^{n+1}\}, \textit{ pour toute composante } j \in \{1, \ldots, s\}.$$

PREUVE. Supposons par l'absurde qu'il existe un intervalle fermé $\bar{P} = [0, p]$ de volume minimal contenant k points parmi n (dont certains peuvent se trouver sur le bord) tel que $p_d \notin \{x_d^0, \ldots, x_d^{n+1}\}$ pour une composante $d \in \{1, \ldots, s\}$. Il est clair que l'intervalle fermé $\bar{P}' = [0, p']$ donné par

$$p'_j = \begin{cases} \max_{\{i \in \{0, \ldots, n+1\} \,:\, x_j^i < p_j\}} x_j^i & \text{si } j = d \\ p_j & \text{sinon} \end{cases}$$

est de volume strictement inférieur à $\lambda(\bar{P})$ et contient les mêmes k points. □

PROPOSITION 8.2 *Pour un problème $V_{\max}^k$, le supremum dans l'expression* (45) *est atteint pour un intervalle $P = [0, p) = \prod_{j=1}^s [0, p_j) \in \mathcal{I}_s^*$, où $p = (p_1, \ldots, p_s)$ est un point de la grille induite par la séquence x et les bords du cube unité. En d'autres termes, on a*

$$p_j \in \{x_j^0, \ldots, x_j^{n+1}\}, \textit{ pour toute composante } j \in \{1, \ldots, s\}.$$

PREUVE. Supposons par l'absurde qu'il existe un intervalle optimal P tel que $p_d \notin \{x_d^0, \dots, x_d^{n+1}\}$ pour une composante $d \in \{1, \dots, s\}$. Il est clair que l'intervalle $P' = [0, p')$ donné par

$$p'_j = \begin{cases} \min_{\{i \in \{0,\dots,n+1\} \,:\, x_j^i > p_j\}} x_j^i & \text{si } j = d \\ p_j & \text{sinon} \end{cases}$$

est de volume strictement supérieur à $\lambda(P)$ et ne contient aucun point supplémentaire. $\square$

On peut remarquer qu'en combinant ces propositions avec l'expression (46), on retombe facilement sur la discrétisation de Niederreiter (21) pour le calcul de la discrépance.

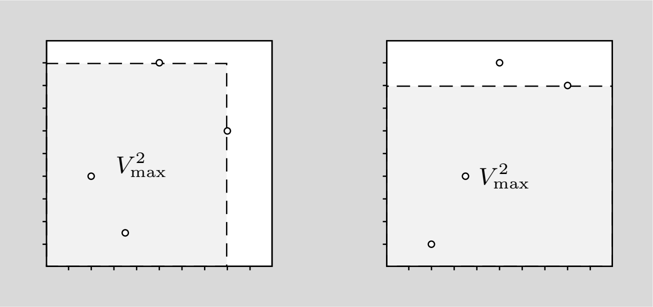


FIG. 8.1. Solution optimale du problème $V^2_{\max}$ pour deux instances données par 4 points du plan.

Pour les deux types de problèmes, la solution optimale est donc à chercher sur la grille induite par la séquence x et les bords du cube unité. Toutefois, les points situés sur le bord ouvert d'un intervalle candidat $[0, p)$ ne sont pas à considérer parmi les k points cherchés pour $V^k_{\max}$ (voir figure 8.1), alors qu'ils doivent être comptés pour $V^k_{\min}$ (voir figure 8.2). De manière équivalente, tout problème $V^k_{\max}$ peut être résolu en recherchant l'intervalle fermé $\bar{P} = [0, p]$ de volume maximal contenant k points dans son « semi-intérieur » $[0, p)$ et $V^k_{\min}$ correspond au volume minimal de la fermeture d'un intervalle semi-ouvert (contenant k points) que l'on peut écrire $P(\varepsilon) = \prod_{j=1}^{s} [0, p_j + \varepsilon)$ avec $\varepsilon \to 0^+$.

Ainsi, en fin de compte, la considération d'intervalles fermés ou semi-ouverts dans la définition des problèmes (45) s'avère tout à fait anecdotique. Cependant, la programmation linéaire étant par nature adaptée aux intervalles fermés, nous travaillerons exclusivement avec ces derniers. Signalons qu'une partie de la difficulté des problèmes (45) tient au fait que, dans les deux cas, le nombre de points situés sur le bord de l'intervalle optimal n'est pas fixe et dépend de x et de k (voir figures 8.1 et 8.2).

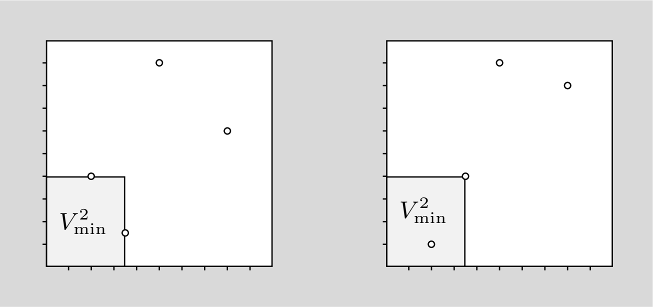


FIG. 8.2. Solution optimale du problème $V^2_{\min}$ pour deux instances données par 4 points du plan.

Une approche brutale envisageable pour la résolution des problèmes $V^k_{\min}$ et $V^k_{\max}$ passe par l'énumération des $\binom{n}{k}$ sous-ensembles de k points de la séquence x. D'autre part, les propositions 8.1 et 8.2 indiquent qu'ils peuvent être résolus en considérant les $(n+2)^s$ points de la grille induite par la séquence x et les bords du cube unité. L'utilisation de la programmation linéaire en nombres entiers permet d'éviter d'aussi importantes énumérations et fournit également, par relaxation, des bornes pour la solution optimale.

8.2.1 Quelques notations et élimination des cas triviaux. Dans ce qui suit, on suppose qu'aucun point de la séquence x ne possède de composante nulle :

$$x^i_j \neq 0, \text{ pour tout point } i \in \{1, \dots, n\} \text{ et toute composante } j \in \{1, \dots, s\}.$$

On pose cette hypothèse dans un souci de simplification des notations. En effet, la présence de coordonnées nulles est susceptible d'engendrer l'apparition de coûts infinis dans la fonction objectif des programmes linéaires énoncés ci-dessous. Bien qu'il soit possible de gérer une telle situation, ce serait au prix d'un alourdissement considérable de la présentation. Nous préférons donc éliminer cette possibilité, afin de ne pas mettre en péril la clarté de l'exposé.

L'origine $(0, \dots, 0)$ est le premier point de bon nombre de suites à discrépance faible classiques (les composantes de tous les autres points étant généralement non nulles). Toutefois, le traitement de ce cas particulier étant réintroduit dans la remarque 8.18, la présence de ce point n'entre pas en conflit avec l'hypothèse ci-dessus. Finalement, signalons qu'en dernier recours, la méthode peut toujours être utilisée telle quelle, en prenant soin de remplacer chaque coordonnée nulle par un petit $\varepsilon > 0$.

Pour chaque composante $j \in \{1, \dots, s\}$, on définit une permutation σ_j de la séquence x, telle que

$$x_j^{\sigma_j(1)} \leq \cdots \leq x_j^{\sigma_j(n)}.$$

De plus, on étend cette permutation au point auxiliaire $x^{n+1} = (1, \dots, 1)$ en posant

$$\sigma_j(n+1) = n+1, \text{ pour tout } j \in \{1, \dots, s\}.$$

L'intervalle nul fournissant la solution optimale du problème $V^0_{\min}$, on a

$$V^0_{\min} = 0$$

quelle que soit la séquence x (n'incluant pas l'origine). Le cas des points à coordonnées nulles ayant été écarté et $V^0_{\min}$ résolu, il est clair que pour les $2n+1$ problèmes restants ($V^1_{\min}, \dots, V^n_{\min}$ et $V^0_{\max}, \dots, V^n_{\max}$), aucun intervalle dégénéré (*i.e.* à intérieur vide) n'est plus à considérer.

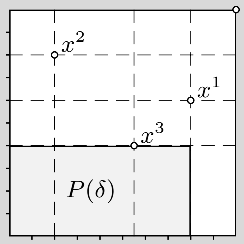
Ainsi, compte tenu des propositions 8.1 et 8.2, nous ne nous intéressons qu'aux intervalles de la forme

$$P(\delta) = \prod_{j=1}^{s} \left[0, x_j^{\sigma_j(\delta_j)}\right) \quad \text{et} \quad \bar{P}(\delta) = \prod_{j=1}^{s} \left[0, x_j^{\sigma_j(\delta_j)}\right]$$

donnés par un paramètre $\delta = (\delta_1, \dots, \delta_s) \in \{1, \dots, n+1\}^s$.

Ces nouvelles notations sont illustrées par un exemple donné dans la figure 8.3. Hormis $V^0_{\min}$, cinq autres cas (pour autant que $n \geq 3$) peuvent être résolus directement. Pour chacun des problèmes en question, on fournit le volume optimal ainsi qu'un intervalle et l'ensemble de k points correspondant :

▷ $V^1_{\min}$
Choisir un point $p \in \{1, \dots, n\}$ minimisant le volume $V(i) = \prod_{j=1}^{s} x^i_j$.
On obtient le volume optimal $V^1_{\min} = V(p)$ et l'intervalle $\bar{P} = \prod_{j=1}^{s}[0, x^p_j]$ correspondant ne contient que le point x^p.



$$\sigma_1 = (2\,3\,1\,4)$$
$$x_1^2 \le x_1^3 \le x_1^1 \le x_1^4$$
$$\sigma_2 = (3\,1\,2\,4)$$
$$x_2^3 \le x_2^1 \le x_2^2 \le x_2^4$$
$$\delta = (\delta_1, \delta_2) = (3, 1)$$
$$P(\delta) = [0, x_1^{\sigma_1(\delta_1)}) \times [0, x_2^{\sigma_2(\delta_2)}) = [0, x_1^1) \times [0, x_2^3)$$

FIG. 8.3. Illustration, pour une séquence de $n = 3$ points dans le plan, des nouvelles notations introduites pour les permutations associées aux différentes composantes et pour les intervalles de la forme $P(\delta)$.

▷ $V_{\min}^{n-1}$
Le point à retirer est $x^{\sigma_d(n)}$, où d est une composante $c \in \{1, \dots, s\}$ minimisant le volume restant

$$V(c) = \prod_{\substack{j=1 \\ \sigma_j(n) \neq \sigma_c(n)}}^{s} x_j^{\sigma_j(n)} \prod_{\substack{j=1 \\ \sigma_j(n) = \sigma_c(n)}}^{s} x_j^{\sigma_j(n-1)}.$$

Le volume optimal est $V_{\min}^{n-1} = V(d)$ et l'intervalle correspondant $\bar{P}(\delta) = \prod_{j=1}^{s} \left[0, x_j^{\sigma_j(\delta_j)}\right]$ donné par

$$\delta_j = \begin{cases} n & \text{si } \sigma_j(n) \neq \sigma_d(n) \\ n-1 & \text{sinon} \end{cases}, \text{ pour tout } j \in \{1, \dots, s\}$$

contient tous les points de la séquence sauf $x^{\sigma_d(n)}$.

▷ $V_{\min}^{n}$
On a le volume optimal $V_{\min}^{n} = \prod_{j=1}^{s} x_j^{\sigma_j(n)}$ et l'intervalle correspondant $\bar{P} = \prod_{j=1}^{s} \left[0, x_j^{\sigma_j(n)}\right]$ contient toute la séquence x.

▷ $V_{\max}^{n-1}$
Le point à retirer est $x^{\sigma_d(n)}$, où d est une composante $c \in \{1, \dots, s\}$ maximisant le volume restant

$$V(c) = x_c^{\sigma_c(n)}.$$

Le volume optimal est $V_{\max}^{n-1} = V(d)$ et l'intervalle correspondant $P(\delta) = \prod_{j=1}^{s} \left[0, x_j^{\sigma_j(\delta_j)}\right)$ donné par

$$\delta_j = \begin{cases} n+1 & \text{si } j \neq d \\ n & \text{sinon} \end{cases}, \text{ pour tout } j \in \{1, \dots, s\}$$

contient tous les points de la séquence sauf $x^{\sigma_d(n)}$.

▷ $V_{\max}^{n}$
Le volume optimal est $V_{\max}^{n} = 1$ et l'intervalle optimal $P = I^s$ contient toute la séquence x.

Ces six cas peuvent être considérés comme réglés et ne nécessitent donc pas l'usage de l'approche par programmation linéaire en nombres entiers présentée ci-dessous.

8.2.2 Reformulation. Tout d'abord, on associe à tout paramètre $\delta \in \{1, \dots, n+1\}^s$ l'unique ensemble de $s(n+1)$ variables binaires donné par

$$z_j^{\sigma_j(i)} = \begin{cases} 1 & \text{si } i \leq \delta_j \\ 0 & \text{sinon} \end{cases}, \text{ pour tout } i \in \{1, \dots, n+1\} \text{ et } j \in \{1, \dots, s\}. \tag{47}$$

En d'autres termes, pour toute composante $j \in \{1, \dots, s\}$, le paramètre $\delta_j \in \{1, \dots, n+1\}$ engendre une séquence non croissante de $n+1$ variables binaires dont la première vaut 1 :

$$1 = z_j^{\sigma_j(1)} \geq z_j^{\sigma_j(2)} \geq \dots \geq z_j^{\sigma_j(n+1)}. \tag{48}$$

Cette transformation est illustrée dans la figure 8.4. Clairement, la définition (47) établit une bijection entre $\{1, \dots, n+1\}^s$ et les ensembles de s séquences non croissantes de $n+1$ variables binaires commençant par un 1.

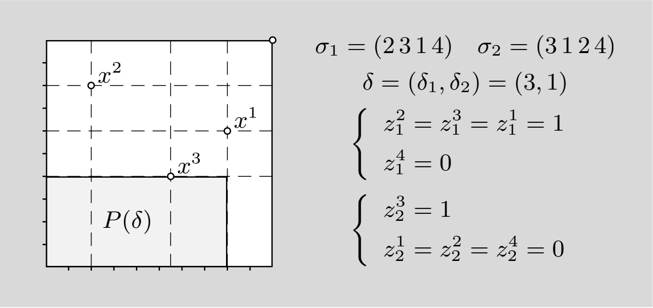


FIG. 8.4. Illustration, pour une séquence de 3 points dans le plan, du lien existant entre un intervalle $P(\delta)$ donné par un paramètre $\delta \in \{1, \dots, n+1\}^s$ et les $s(n+1)$ variables binaires z_j^i qui lui sont associées.

Par définition, $x^{\sigma_j(i)}$ est le point de la séquence présentant la i^{e} plus petite j^{e} composante et $x_j^{\sigma_j(\delta_j)}$ correspond à la longueur du j^{e} côté de l'intervalle $P(\delta)$. L'expression (47) est donc à interpréter comme suit : la variable binaire $z_j^{\sigma_j(i)}$ est égale à 1 si et seulement si la j^{e} coordonnée $x_j^{\sigma_j(i)}$ du point $x^{\sigma_j(i)}$ est inférieure ou égale à la longueur $x_j^{\sigma_j(\delta_j)}$ du j^{e} côté de l'intervalle $P(\delta)$ (voir figure 8.4).

8.2.2.1 *Le problème de l'intervalle de volume minimal contenant k points.* Par définition, un point x^i de la séquence x est donc inclus dans l'intervalle fermé

$$\bar{P}(\delta) = \prod_{j=1}^{s} \left[0, x_j^{\sigma_j(\delta_j)}\right]$$

si et seulement si $z_j^i = 1$ pour toute composante $j \in \{1, \dots, s\}$. Or, pour un problème du type $V_{\min}^k$, la proposition 8.1 indique que l'on cherche justement un intervalle fermé contenant k points de la séquence. Introduisant n variables binaires $y^1, \dots, y^n$ soumises aux contraintes linéaires

$$\begin{cases} y^i \leq z_j^i, \; \forall j \in \{1, \dots, s\}, \; \forall i \in \{1, \dots, n\}, \\ y^i \geq 1 - s + \displaystyle\sum_{j=1}^{s} z_j^i, \; \forall i \in \{1, \dots, n\}, \end{cases} \tag{49}$$

où les variables z_j^i (obtenues par la relation (47) pour un intervalle $P(\delta)$ donné) satisfont la contrainte

$$z_j^{\sigma_j(i)} = z_j^{\sigma_j(i+1)}, \; \forall j \in \{1, \dots, s\}, \; \forall i \in \{1, \dots, n\} \text{ avec } x_j^{\sigma_j(i)} = x_j^{\sigma_j(i+1)}, \tag{50}$$

on a

$$y^i = 1 \quad \Longleftrightarrow \quad z^i_j = 1 \text{ pour tout } j \in \{1, \dots, s\} \quad \Longleftrightarrow \quad x^i \in \bar{P}(\delta) = \prod_{j=1}^{s} \left[0, x_j^{\sigma_j(\delta_j)}\right].$$

La variable binaire y^i vaut donc 1 si et seulement si le point x^i est inclus dans l'intervalle fermé $\bar{P}(\delta)$. Les contraintes (50) sont nécessaires pour préciser que si deux points $x^{\sigma_j(i)}$ et $x^{\sigma_j(i+1)}$ possèdent la même j^{e} coordonnée, on peut envisager de considérer un intervalle donné par $\delta_j = i + 1$, mais en aucun cas par $\delta_j = i$. Finalement, pour obtenir un intervalle contenant k points, il suffit d'imposer

$$\sum_{i=1}^{n} y^i = k. \tag{51}$$

En fin de compte, pour $k \in \{1, \dots, n\}$, les solutions admissibles du problème $V^k_{\min}$ correspondent aux ensembles de variables binaires z^i_j et y^i satisfaisant les contraintes (48)(49)(50) et (51). Par ailleurs, le volume d'un intervalle défini par un paramètre $\delta \in \{1, \dots, n+1\}^s$ s'écrit

$$\lambda(P(\delta)) = \lambda(\bar{P}(\delta)) = \prod_{j=1}^{s} x_j^{\sigma_j(\delta_j)}.$$

Le logarithme de cette grandeur

$$\log\left(\lambda(P(\delta))\right) = \log\left(\lambda(\bar{P}(\delta))\right) = \sum_{j=1}^{s} \log\left(x_j^{\sigma_j(\delta_j)}\right)$$

$$= \sum_{j=1}^{s} \left[z_j^{\sigma_j(1)} \log\left(x_j^{\sigma_j(1)}\right) + \sum_{i=2}^{n+1} z_j^{\sigma_j(i)} \left\{ \log\left(x_j^{\sigma_j(i)}\right) - \log\left(x_j^{\sigma_j(i-1)}\right) \right\} \right] \tag{52}$$

prenant la forme d'une fonction linéaire des variables $z_j^{\sigma_j(i)}$, il suffit de retenir cette expression comme fonction objectif pour aboutir à la formulation désirée :

$$\log(V^k_{\min}) \;=\; \min \sum_{j=1}^{s} \left[\log\left(x_j^{\sigma_j(1)}\right) + \sum_{i=2}^{n+1} z_j^{\sigma_j(i)} \left\{ \log\left(x_j^{\sigma_j(i)}\right) - \log\left(x_j^{\sigma_j(i-1)}\right) \right\} \right]$$

$$\begin{array}{lll}
\text{s.c.} & 1 = z_j^{\sigma_j(1)} \geq \dots \geq z_j^{\sigma_j(n+1)} & \forall j \in \{1, \dots, s\} \\
& z_j^{\sigma_j(i)} = z_j^{\sigma_j(i+1)} & \forall j \in \{1, \dots, s\}, \forall i \in \{1, \dots, n\} \text{ avec } x_j^{\sigma_j(i)} = x_j^{\sigma_j(i+1)} \\
& y^i \leq z^i_j & \forall j \in \{1, \dots, s\}, \forall i \in \{1, \dots, n\} \\
& y^i \geq 1 - s + \sum_{j=1}^{s} z^i_j & \forall i \in \{1, \dots, n\} \\
& \sum_{i=1}^{n} y^i = k & \\
& y^i \in \{0, 1\} & \forall i \in \{1, \dots, n\} \\
& z^i_j \in \{0, 1\} & \forall i \in \{1, \dots, n+1\}, \forall j \in \{1, \dots, s\}
\end{array}$$

La taille d'un tel programme linéaire en nombres entiers est proportionnelle à la dimension s et au nombre de points n de la séquence x considérée.

8.2.2.2 *Le problème de l'intervalle de volume maximal contenant k points.* On obtient facilement la formulation correspondante pour les problèmes du type $V^k_{\max}$ en maximisant le logarithme du volume (52) et en remplaçant l'ensemble de contraintes (49) par

$$\left\{ \begin{array}{l} y^i \leq z_j^{\sigma_j(1+\sigma_j^{-1}(i))}, \; \forall j \in \{1,\dots,s\}, \; \forall i \in \{1,\dots,n\}, \\ y^i \geq 1 - s + \displaystyle\sum_{j=1}^{s} z_j^{\sigma_j(1+\sigma_j^{-1}(i))}, \; \forall i \in \{1,\dots,n\}. \end{array} \right. \tag{53}$$

Comme le stipule la proposition 8.2, les points situés sur le bord ouvert de l'intervalle $P(\delta)$ ne sont pas à comptabiliser pour un problème $V^k_{\max}$. En d'autres termes, alors que les contraintes (49) permettent de fixer à 1 les variables y^i associées aux points x^i situés dans l'intervalle fermé $\bar{P}(\delta)$ (pour des variables z^i_j obtenues par l'intermédiaire de la relation (47) à partir d'un paramètre δ), les restrictions (53) aboutissent au résultat correspondant pour l'intervalle semi-ouvert $P(\delta)$. Pour parvenir à l'énoncé des contraintes (53), il suffit de remarquer que, pour tout $\delta \in \{2,\dots,n+1\}^s$, les intervalles

$$P(\delta) = \prod_{j=1}^{s} \left[0, x_j^{\sigma_j(\delta_j)}\right) \quad \text{et} \quad \bar{P}(\delta - 1) = \prod_{j=1}^{s} \left[0, x_j^{\sigma_j(\delta_j - 1)}\right]$$

contiennent les mêmes points de la séquence x. Remplaçant (49) par (53), on a donc

$$y^i = 1 \iff z_j^{\sigma_j(1+\sigma_j^{-1}(i))} = 1 \text{ pour tout } j \in \{1,\dots,s\} \iff x^i \in P(\delta) = \prod_{j=1}^{s} \left[0, x_j^{\sigma_j(\delta_j)}\right).$$

On obtient la formulation désirée en maximisant le logarithme du volume (52) sous les contraintes (48) (50)(51) et (53) :

$$\log(V^k_{\max}) \; = \; \max \; \sum_{j=1}^{s} \left[\log\left(x_j^{\sigma_j(1)}\right) + \sum_{i=2}^{n+1} z_j^{\sigma_j(i)} \left\{ \log\left(x_j^{\sigma_j(i)}\right) - \log\left(x_j^{\sigma_j(i-1)}\right) \right\} \right]$$

$$\begin{array}{lll} \text{s.c.} & 1 = z_j^{\sigma_j(1)} \geq \cdots \geq z_j^{\sigma_j(n+1)} & \forall j \in \{1,\dots,s\} \\ & z_j^{\sigma_j(i)} = z_j^{\sigma_j(i+1)} & \forall j \in \{1,\dots,s\}, \; \forall i \in \{1,\dots,n\} \text{ avec } x_j^{\sigma_j(i)} = x_j^{\sigma_j(i+1)} \\ & y^i \leq z_j^{\sigma_j(1+\sigma_j^{-1}(i))} & \forall j \in \{1,\dots,s\}, \; \forall i \in \{1,\dots,n\} \\ & y^i \geq 1 - s + \displaystyle\sum_{j=1}^{s} z_j^{\sigma_j(1+\sigma_j^{-1}(i))} & \forall i \in \{1,\dots,n\} \\ & \displaystyle\sum_{i=1}^{n} y^i = k & \\ & y^i \in \{0,1\} & \forall i \in \{1,\dots,n\} \\ & z^i_j \in \{0,1\} & \forall i \in \{1,\dots,n+1\}, \; \forall j \in \{1,\dots,s\} \end{array}$$

8.2.2.3 *Remarques.* Dans le cas où les points de la séquence $x = \{x^1,\dots,x^n\}$ ne sont pas tous distincts, il arrive que, pour certaines valeurs de k, les problèmes (45) (et par conséquent les programmes linéaires correspondants) ne possèdent pas de solution (admissible). C'est par exemple le cas pour $V^1_{\min}$ et $V^1_{\max}$ et la séquence de $n = 2$ points en dimension $s = 2$

$$x = \{(1/2, 1/2), (1/2, 1/2)\}.$$

En revanche, lorsque les n points sont tous différents, les problèmes $V^k_{\min}$ et $V^k_{\max}$ possèdent une solution pour tout $k \in \{0,\dots,n\}$. Ainsi, afin de ne pas alourdir la suite de la présentation, on suppose que la séquence x possède cette propriété. D'ailleurs, à notre connaissance, cette condition est vérifiée pour toutes les suites à discrépance faible classiques.

REMARQUE 8.3 Il est clair que le polyèdre défini comme l'enveloppe convexe de l'ensemble des solutions admissibles d'un programme linéaire du type $\log(V^k_{\mathrm{min}})$ ou $\log(V^k_{\mathrm{max}})$ est un polytope. En revanche, la détermination de sa dimension s'avère difficile. Elle dépend à la fois de n, s et k, mais également de la disposition dans le cube unité (décrite à travers les permutations $\sigma_1, \ldots, \sigma_s$) des n points de la séquence. Pour les deux types de problèmes, la relation entre un tel ensemble de permutations et la dimension du polytope correspondant semble particulièrement complexe. Néanmoins, dans la plupart des cas, la dimension du polytope est supérieure ou égale à $s-1$ (voir corollaire 8.13).

Par exemple, on peut vérifier (après quelques calculs fastidieux) que le polytope (défini à partir des six variables y^1, y^2, z^2_1, z^3_1, z^2_2 et z^3_2) obtenu pour le problème V^1_{min} associé à une séquence décrite par $s = 2$, $n = 2$, $\sigma_1 = (1\,2\,3)$ et $\sigma_2 = (1\,2\,3)$ est de dimension 4. Par contre, pour $s = 2$, $n = 2$, $\sigma_1 = (1\,2\,3)$ et $\sigma_2 = (2\,1\,3)$, on a toujours six variables, mais le polytope V^1_{min} correspondant est cette fois de dimension 3.

REMARQUE 8.4 On appelle *périmètre* d'un intervalle la somme des longueurs de ses arêtes (cette définition correspond à la notion usuelle pour les rectangles dans le plan). Les programmes linéaires ci-dessus peuvent être utilisés pour déterminer les intervalles ancrés à l'origine de périmètre minimal ou maximal contenant k parmi les n points de la séquence x. En effet, si l'on applique préalablement la transformation $f : x^i_j \longmapsto e^{x^i_j}$ à chacune des composantes des points de l'ensemble $\{x^1, \ldots, x^{n+1}\}$, les grandeurs $2^{s-1}\log(V^k_{\mathrm{min}})$ et $2^{s-1}\log(V^k_{\mathrm{max}})$ sont les périmètres optimaux recherchés. Cette propriété découle du fait qu'un intervalle en dimension s possède $s \cdot 2^{s-1}$ arêtes et que, suite à la transformation énoncée, le logarithme du volume d'un intervalle devient

$$\log\left(\prod_{j=1}^{s} f\left(x_j^{\sigma_j(\delta_j)}\right)\right) = \sum_{j=1}^{s} x_j^{\sigma_j(\delta_j)}.$$

8.2.3 Simplification. Avant d'aborder la résolution des programmes linéaires en nombres entiers décrits ci-dessus, on commence par énoncer trois observations permettant de fixer, a priori, la valeur de quelques variables.

OBSERVATION 8.5 Pour toute composante $j \in \{1, \ldots, s\}$, on remarque que le coût

$$\log\left(x_j^{\sigma_j(n+1)}\right) - \log\left(x_j^{\sigma_j(n)}\right) = -\log\left(x_j^{\sigma_j(n)}\right)$$

associé à la variable $z^{n+1}_j = z_j^{\sigma_j(n+1)}$ dans la fonction objectif (52) est positif. Ainsi, on a

$$z^{n+1}_1 = \cdots = z^{n+1}_s = 0$$

pour tout programme linéaire de la forme $\log(V^k_{\mathrm{min}})$ et toute séquence $x \subset I^s$.

OBSERVATION 8.6 Étant donné que, dans toute solution admissible, k variables de l'ensemble $\{y^1, \ldots, y^n\}$ prennent la valeur 1, les contraintes (48)(49) et (53) impliquent que l'on peut poser

$$z_j^{\sigma_j(1)} = \cdots = z_j^{\sigma_j(k)} = 1, \text{ pour tout } j \in \{1, \ldots, s\}$$

pour tout programme du type $\log(V^k_{\mathrm{min}})$ et

$$z_j^{\sigma_j(1)} = \cdots = z_j^{\sigma_j(k+1)} = 1, \text{ pour tout } j \in \{1, \ldots, s\}$$

pour tout programme du type $\log(V^k_{\mathrm{max}})$. De plus, considérant (49) et (53), on obtient dans les deux cas

$$y^i = 1 \text{ pour tout } i \in \{1, \ldots, n\} \text{ tel que } \sigma_j^{-1}(i) \le k \text{ pour toute composante } j \in \{1, \ldots, s\}.$$

OBSERVATION 8.7 Si, pour un point x^i de la séquence, l'intervalle fermé $[0, x^i]$ contient plus de k points, alors x^i lui-même ne peut pas appartenir à l'intervalle optimal. Pour tout problème $V^k_{\min}$ ou $V^k_{\max}$, on peut donc poser

$$y^i = 0, \text{ pour tout } i \in \{1, \dots, n\} \text{ tel que } A([0, x^i], x) > k.$$

8.3 Quelques approches de résolution

Pour un problème donné, on considère tout d'abord sa *relaxation linéaire*, c'est-à-dire le programme obtenu en remplaçant les contraintes d'intégralité

$$\begin{aligned} &y^i \in \{0,1\}, \quad \forall i \in \{1, \dots, n\}, \\ &z^i_j \in \{0,1\}, \quad \forall i \in \{1, \dots, n+1\}, \quad \forall j \in \{1, \dots, s\} \end{aligned}$$

par

$$\begin{aligned} &y^i \in [0,1], \quad \forall i \in \{1, \dots, n\}, \\ &z^i_j \in [0,1], \quad \forall i \in \{1, \dots, n+1\}, \quad \forall j \in \{1, \dots, s\}. \end{aligned}$$

Ce nouveau programme est facile à résoudre et, suivant le type de problème considéré, la valeur de sa solution optimale constitue une borne inférieure pour $\log(V^k_{\min})$ ou supérieure pour $\log(V^k_{\max})$. De plus, lorsque cette solution est entière (*i.e.* toutes les variables y^i et z^i_j prennent leurs valeurs dans l'ensemble $\{0, 1\}$), il s'agit également d'une solution optimale du programme initial non relaxé. Schématiquement, les résultats obtenus en pratique sont de deux formes :

1° La solution optimale de la relaxation d'un programme de la forme $\log(V^k_{\max})$ est fréquemment entière (environ une fois sur deux, généralement pour de grandes valeurs de k). De plus, sa valeur étant le plus souvent très proche de l'optimum du problème non relaxé, elle fournit dans la plupart des cas une borne supérieure d'excellente qualité.

2° En revanche, pour les programmes du type $\log(V^k_{\min})$, les résultats sont nettement moins réjouissants. En effet, en résolvant la relaxation linéaire, on obtient presque systématiquement la solution optimale pathologique où toutes les variables sont égales à k/n. Par ailleurs, la valeur de cette solution étant généralement très éloignée de l'optimum du problème non relaxé, il s'agit le plus souvent d'une mauvaise borne inférieure.

On en conclut que, d'un certain point de vue, la formulation de nos programmes s'avère plus forte pour $\log(V^k_{\max})$ que pour $\log(V^k_{\min})$. Ci-dessous, des contraintes supplémentaires sont ajoutées, dans le but d'améliorer la qualité des bornes obtenues par relaxation linéaire. Un intérêt particulier est donc porté aux problèmes du type $V^k_{\min}$.

8.3.1 Introduction systématique d'inégalités valides. Pour un programme linéaire en nombres entiers donné, une inégalité est dite *valide* si elle est satisfaite pour chacune de ses solutions admissibles. L'ajout d'une telle contrainte dans la formulation du problème peut s'avérér plus ou moins efficace au niveau de la qualité de la solution relaxée correspondante. Suivant les cas, l'opération peut être totalement inutile ou, au contraire, directement mener à une solution optimale entière.

En outre, l'effet de l'introduction d'un ensemble d'inégalités valides n'est pas forcément immédiat. Par exemple, les contraintes disjonctives proposées dans l'observation 8.8 n'ont pas toujours un impact marqué sur la qualité de la relaxation linéaire, mais elles conduisent généralement à une accélération substantielle de la convergence lors de l'utilisation d'une méthode d'énumération par séparation et évaluation (voir section 8.3.3.3).

OBSERVATION 8.8 Si, pour un point x^i de la séquence, l'intervalle $[0, x^i]$ contient exactement k points, alors x^i et tout point x^p n'appartenant pas à $[0, x^i]$ ne peuvent se trouver tous deux dans un même intervalle admissible. Pour tout programme $\log(V^k_{\min})$ et $\log(V^k_{\max})$, on a donc l'inégalité valide

$$y^i + y^p \leq 1, \text{ pour tout } i, p \in \{1, \ldots, n\} \text{ tels que } A([0, x^i], x) = k \text{ et } x^p \notin [0, x^i].$$

On note que, compte tenu des variables fixées à l'aide de l'observation 8.7, cette contrainte devient inutile et n'a donc pas à être introduite si $A([0, x^p], x) > k$.

Intéressons-nous plus particulièrement aux problèmes du type $V^k_{\min}$. Le théorème suivant permet de construire une importante famille d'inégalités valides, certaines d'entre elles améliorant très nettement la qualité de la solution optimale de la relaxation linéaire du programme $\log(V^k_{\min})$.

THÉORÈME 8.9 *Soit un entier $l \in \{2, \ldots, s\}$, un sous-ensemble de composantes $J \subset \{1, \ldots, s\}$ tel que $|J| \geq l$ et des index $\phi_j \in \{k+1, \ldots, n\}$ définis pour tout $j \in J$ tels que*

$$\left| \bigcup_{j \in J'} \{\sigma_j(\phi_j), \ldots, \sigma_j(n)\} \right| > n - k, \text{ pour tout sous-ensemble } J' \subset J \text{ tel que } |J'| = l. \tag{54}$$

Alors, les inégalités suivantes sont valides pour le programme linéaire $\log(V^k_{\min})$:

$$\sum_{j \in J'} z_j^{\sigma_j(\phi_j)} \geq |J'| - l + 1, \text{ pour tout sous-ensemble } J' \subset J \text{ tel que } |J'| \geq l. \tag{55}$$

PREUVE. Commençons par une discussion informelle. L'établissement de ces contraintes repose sur l'idée suivante : si l'on exclut au moins $n-k+1$ points de la séquence, il n'est pas possible de trouver un intervalle en contenant exactement k. Supposons par exemple que, pour un ensemble de l composantes J', l'union pour tout $j \in J'$ de

$$\left\{x^{\sigma_j(\phi_j)}, \ldots, x^{\sigma_j(n)}\right\}$$

contienne plus que $n-k$ points de x (c'est-à-dire la condition (54) pour un ensemble J' particulier). On en déduit qu'au moins une variable de

$$\left\{z_j^{\sigma_j(\phi_j)} : j \in J'\right\}$$

doit nécessairement être égale à 1 dans toute solution admissible du programme $\log(V^k_{\min})$ (c'est-à-dire la conclusion (55) pour le même ensemble J'). Les autres cas s'obtiennent par combinaison.

En fin de compte, l'objectif visé par l'introduction des contraintes (55) est de faire augmenter, dans la solution optimale relaxée, la valeur des variables $z_j^{\sigma_j(1)}, \ldots, z_j^{\sigma_j(\phi_j)}$ associées aux composantes $j \in J$.

Passons maintenant à la preuve formelle. On procède par induction sur la cardinalité c de J' dans (55). Pour $c = l$, on doit donc montrer que

$$\sum_{j \in J'} z_j^{\sigma_j(\phi_j)} \geq 1$$

est une inégalité valide pour tout $J' \subset J$ tel que $|J'| = l$. Pour un tel sous-ensemble J', supposons par l'absurde qu'il existe une solution admissible avec

$$z_j^{\sigma_j(\phi_j)} = 0, \text{ pour tout } j \in J'.$$

Les contraintes $y^i \leq z_j^i$ du programme linéaire impliquent alors que

$$y^i = 0, \text{ pour tout index } i \in \{\sigma_j(\phi_j), \ldots, \sigma_j(n)\} \text{ et toute composante } j \in J'.$$

Considérant la condition (54), cette situation est clairement incompatible avec la contrainte (51). Il s'agit d'une contradiction et le résultat est donc vrai pour $c = l$. Par hypothèse d'induction, on suppose l'assertion prouvée pour $c - 1$ et on la démontre pour c (sans aller plus loin que $c = |J|$). Pour tout $J' \subset J$ avec $|J'| = c$, on note $J'_1, \ldots, J'_c$ les c sous-ensembles distincts de J' de cardinalité $c - 1$. En utilisant l'hypothèse d'induction, on a

$$\sum_{j \in J'_i} z_j^{\sigma_j(\phi_j)} \geq (c-1) - l + 1, \text{ pour tout } i = 1, \ldots, c.$$

Sommant ces inégalités, il vient

$$(c-1) \sum_{j \in J'} z_j^{\sigma_j(\phi_j)} \geq c(c-l) \qquad \Longrightarrow \qquad \sum_{j \in J'} z_j^{\sigma_j(\phi_j)} \geq \frac{c}{c-1}(c-l).$$

Le côté gauche de la dernière inégalité étant un entier, on obtient le résultat

$$\sum_{j \in J'} z_j^{\sigma_j(\phi_j)} \geq c - l + 1.$$

□

Le théorème ci-dessus se limite aux cas où $\phi_j \geq k + 1$ et $l \geq 2$, alors que l'énoncé s'applique plus généralement lorsque $\phi_j \geq 1$ et $l \geq 1$. Par exemple, pour tout $j \in \{1, \ldots, s\}$, cette version étendue du théorème avec $l = 1$, $J = \{j\}$ et $\phi_j = k$ fournit l'inégalité valide

$$z_j^{\sigma_j(k)} \geq 1.$$

Toutefois, un tel effort est inutile compte tenu du fait que l'observation 8.6 implique déjà que

$$z_j^{\sigma_j(i)} = 1, \text{ pour tout } i \in \{1, \ldots, k\} \text{ et } j \in \{1, \ldots, s\}.$$

On peut donc remarquer que cette version étendue ne mène à aucune autre inégalité valide intéressante lorsque $l = 1$ ou $\phi_j \leq k$. Ainsi, la formulation proposée s'avère tout à fait suffisante. Par ailleurs, on observe que le théorème 8.9 permet de générer un très grand nombre de contraintes. Le résultat suivant simplifie la procédure de choix en isolant les plus importantes.

THÉORÈME 8.10 *Soit un entier l, un sous-ensemble de composantes J et des index ϕ_j définis pour tout $j \in J$ pour lesquels les hypothèses du théorème 8.9 sont satisfaites. Si l'on introduit l'inégalité valide* (55) *obtenue pour $J' = J$*

$$\sum_{j \in J} z_j^{\sigma_j(\phi_j)} \geq |J| - l + 1$$

dans la formulation du programme linéaire $\log(V_{\min}^k)$, *alors les* $\sum_{i=l}^{|J|-1} \binom{|J|}{i}$ *autres sont redondantes.*

PREUVE. Soit un sous-ensemble de composantes $J' \subset J$ tel que $l \leq |J'| < |J|$. L'inégalité valide correspondante (55) s'écrit

$$\sum_{j \in J'} z_j^{\sigma_j(\phi_j)} \geq |J'| - l + 1.$$

Pour toute composante $d \in J \setminus J'$, l'inégalité valide (55) associée au sous-ensemble $J'' = J' \cup \{d\}$ est

$$z_d^{\sigma_d(\phi_d)} + \sum_{j \in J'} z_j^{\sigma_j(\phi_j)} = \sum_{j \in J''} z_j^{\sigma_j(\phi_j)} \geq |J''| - l + 1 = |J'| - l + 2.$$

La variable $z_d^{\sigma_d(\phi_d)}$ valant au plus 1, cette contrainte implique la précédente et le résultat est prouvé. $\square$

8.3.1.1 *Les (l, ϕ)-coupes.* On veut renforcer la formulation des programmes du type $\log(V_{\min}^k)$ par l'introduction systématique (*i.e.* préalable à la résolution de la relaxation linéaire) d'un nombre limité d'inégalités valides. En pratique, les (l, ϕ)-coupes équilibrées proposées ci-dessous sont généralement très efficaces.

DÉFINITION 8.11 Pour un vecteur d'index $\phi = (\phi_1, \ldots, \phi_s) \in \{k+1, \ldots, n\}^s$ et un entier $l \in \{2, \ldots, s\}$, une *(l, ϕ)-coupe* est une inégalité valide mise en évidence dans le théorème 8.10 dans le cas particulier où $J = \{1, \ldots, s\}$. Il s'agit donc d'une contrainte de la forme

(56) $$\sum_{j=1}^{s} z_j^{\sigma_j(\phi_j)} \geq s - l + 1.$$

On exige de plus que le vecteur ϕ soit *maximal*, c'est-à-dire qu'il n'existe aucun $\phi' \gneqq \phi$ (*i.e.* $\phi' \geq \phi$ et $\phi' \neq \phi$) tel que ϕ' conduise également à une inégalité valide de la forme de celles du théorème 8.10.

Par exemple, une $(2, \phi)$-coupe permet de préciser que la somme de s variables binaires z_j^j distinctes (une par composante j) est au moins égale à $s - 1$ dans toute solution admissible d'un programme de la forme $\log(V_{\min}^k)$. Dans le cas typique où k/n est la valeur de chaque variable dans la solution optimale de la relaxation linéaire (voir page 106), il est évident que l'introduction d'une seule $(2, \phi)$-coupe, associée à l'action indirecte des contraintes (48), a un impact très net sur la qualité de la solution obtenue.

On remarque que la condition sur la maximalité de ϕ permet de sélectionner les contraintes les plus efficaces. En effet, pour $J = \{1, \ldots, s\}$, $l \in \{2, \ldots, s\}$ et deux vecteurs d'index $\phi' \gneqq \phi$ satisfaisant les conditions du théorème 8.9, l'inégalité valide obtenue est clairement plus forte pour ϕ' que pour ϕ.

Pour que l'inégalité (56) soit valide, le vecteur d'index ϕ utilisé doit satisfaire

(57) $$\left| \bigcup_{j \in J'} \{\sigma_j(\phi_j), \ldots, \sigma_j(n)\} \right| > n - k, \text{ pour tout sous-ensemble } J' \subset \{1, \ldots, s\} \text{ tel que } |J'| = l.$$

Le nombre $\binom{s}{l}$ de conditions à vérifier étant très important pour la plupart des valeurs de $l \in \{2, \ldots, s\}$ (à moins que la dimension s soit petite), on se contente généralement de générer des (l, ϕ)-coupes pour $l \in \{2, 3, s-3, s-2, s-1, s\}$ uniquement.

Pour une valeur de l donnée, il existe généralement de nombreuses (l, ϕ)-coupes, mais en pratique on observe qu'il est le plus souvent inutile d'en introduire plusieurs dans la formulation, l'essentiel du bénéfice pouvant être retiré en en ajoutant une seule. De plus, de meilleurs résultats sont obtenus lorsque le vecteur ϕ utilisé est *équilibré*, c'est-à-dire lorsque $\min\{\phi_1, \ldots, \phi_s\}$ est maximal. En effet, ce choix semble mener à des (l, ϕ)-coupes plus robustes.

Le résultat suivant montre non seulement que les (l, ϕ)-coupes sont des hyperplans d'appui, mais aussi qu'elles induisent des faces de dimension non triviale.

THÉORÈME 8.12 *Pour un programme linéaire en nombres entiers du type $\log(V_{\min}^k)$ associé à une séquence $x \subset \bar{I}^s$, toute (l, ϕ)-coupe induit une face du polytope correspondant (i.e. l'enveloppe convexe de l'ensemble des solutions admissibles) de dimension supérieure ou égale à $s - 1$.*

PREUVE. Il suffit d'exhiber un ensemble de s solutions admissibles affinement indépendantes qui satisfont avec égalité l'inégalité valide en question (56). Le vecteur ϕ étant par définition maximal, on en déduit que pour toute composante $d \in \{1, \ldots, s\}$, il existe un sous-ensemble $J'_d \subset \{1, \ldots, s\}$ de cardinalité l et contenant d tel que

$$\left| \bigcup_{j \in J'_d} \{\sigma_j(\phi_j), \ldots, \sigma_j(n)\} \right| = n - k + 1$$

et

$$\sigma_d(\phi_d) \notin \{\sigma_j(\phi_j), \ldots, \sigma_j(n)\}\text{ , quel que soit } j \in J'_d \setminus \{d\}.$$

On a donc

$$\left| \bigcup_{j \in J'_d \setminus \{d\}} \{\sigma_j(\phi_j), \ldots, \sigma_j(n)\} \cup \{\sigma_d(\phi_d + 1), \ldots, \sigma_d(n)\} \right| = n - k.$$

En d'autres termes, l'intervalle fermé $\bar{P}(\delta^d)$ associé au paramètre $\delta^d = (\delta^d_1, \ldots, \delta^d_s)$ donné par

$$\delta^d_j = \begin{cases} \phi_d & \text{si } j = d \\ \phi_j - 1 & \text{si } j \in J'_d \setminus \{d\} \\ n & \text{si } j \notin J'_d \end{cases}$$

contient k points de la séquence. De plus, par l'expression (47), les variables z^i_j correspondantes

$$z^{\sigma_d(i)}_d = \begin{cases} 1 & \text{si } i \leq \phi_d \\ 0 & \text{sinon} \end{cases} \tag{58}$$

$$z^{\sigma_j(i)}_j = \begin{cases} 1 & \text{si } i \leq \phi_j - 1 \\ 0 & \text{sinon} \end{cases} \text{ , pour } j \in J'_d \setminus \{d\} \tag{59}$$

$$z^{\sigma_j(1)}_j = \cdots = z^{\sigma_j(n)}_j = 1, \text{ pour } j \notin J'_d \tag{60}$$

satisfont avec égalité l'inégalité valide (56). Il reste donc à prouver que les s solutions admissibles ainsi définies sont affinement indépendantes. Il suffit de démontrer cette propriété pour les s vecteurs constitués des valeurs prises, dans chacune de ces solutions, par le sous-ensemble de $2s$ variables

$$\left(z^{\sigma_1(\phi_1)}_1, z^{\sigma_1(\phi_1+1)}_1, z^{\sigma_2(\phi_2)}_2, z^{\sigma_2(\phi_2+1)}_2, \ldots, z^{\sigma_s(\phi_s)}_s, z^{\sigma_s(\phi_s+1)}_s \right).$$

On remarque facilement que ces s vecteurs sont même linéairement indépendants. En effet, pour chaque composante $d \in \{1, \ldots, s\}$, l'expression (58) indique que l'on a $(z^{\sigma_d(\phi_d)}_d, z^{\sigma_d(\phi_d+1)}_d) = (1, 0)$ dans la solution associée au paramètre δ^d, alors que, par (59) et (60), on voit que l'on obtient soit $(0, 0)$, soit $(1, 1)$ dans les $s - 1$ autres cas. □

COROLLAIRE 8.13 *Si, pour un problème du type $V^k_{\min}$ associé à une séquence $x \subset \bar{I}^s$, il existe une (l, ϕ)-coupe, alors la dimension du polytope correspondant est supérieure ou égale à $s - 1$.*

REMARQUE 8.14 En général, une (l, ϕ)-coupe n'induit pas forcément une facette du polytope $V^k_{\min}$. On s'en aperçoit (après quelques calculs assez fastidieux) par exemple pour la $(3, \phi)$-coupe équilibrée correspondant au vecteur d'index $\phi = (3, 3, 2)$ et le problème donné par $s = 3$, $n = 3$, $\sigma_1 = (3\,1\,2\,4)$, $\sigma_2 = (3\,2\,1\,4)$, $\sigma_3 = (2\,3\,1\,4)$ et $k = 1$. En revanche, on peut vérifier que, pour le second exemple de la remarque 8.3, la $(3, \phi)$-coupe équilibrée obtenue pour $\phi = (2, 2)$ induit bien une facette du polytope $V^1_{\min}$ en question.

Pour un entier $l \in \{2, \dots, s\}$, un vecteur maximal et équilibré définissant une (l, ϕ)-coupe peut généralement être construit comme suit. Tout d'abord, on détermine (s'il existe) l'unique vecteur initial ϕ qui soit maximal parmi les s-uples de la forme $(\psi, \dots, \psi)$, avec $\psi \in \{k+1, \dots, n\}$ (cette opération permet de garantir que le vecteur final obtenu est équilibré). Un tel index ψ existe si et seulement si pour tout sous-ensemble $J' \subset \{1, \dots, s\}$ avec $|J'| = l$, il existe $i, j \in J'$ tels que

$$\{\sigma_i(k+1), \dots, \sigma_i(n)\} \neq \{\sigma_j(k+1), \dots, \sigma_j(n)\}.$$

Si cette condition n'est pas vérifiée (ce qui est rarissime en pratique), aucune (l, ϕ)-coupe n'est générée pour la valeur de l en question. Dans le cas contraire, ce vecteur est rendu maximal en lui appliquant aussi longtemps que possible la transformation consistant à augmenter d'une unité une de ses composantes. Cette opération n'est effectuée que si l'expression (57) est encore satisfaite après coup. De plus, afin d'aboutir à un vecteur maximal plus homogène, les composantes candidates à une telle incrémentation sont considérées cycliquement tout au long du processus.

L'algorithme trivial permettant de vérifier si l'augmentation d'une unité de la composante ϕ_d préserve la condition (57) est de complexité $O(nl\binom{s-1}{l-1})$ (il consiste à passer en revue $O(n)$ points de l ensembles pour chacun des $\binom{s-1}{l-1}$ choix possibles de J'). Le même test peut être effectué en seulement $O(l\binom{s-1}{l-1})$ en s'assurant qu'au moins une des l conditions suivantes est satisfaite pour chacun des $\binom{s-1}{l-1}$ sous-ensembles de composantes $J' \supset \{d\}$ en question (auquel cas ϕ_d peut être augmenté) :

1° $\left|\bigcup_{j \in J'} \{\sigma_j(\phi_j), \dots, \sigma_j(n)\}\right| > n - k + 1$;

2° il existe une composante $j \in J' \setminus \{d\}$ telle que $\sigma_j^{-1}(\sigma_d(\phi_d)) \geq \phi_j$.

Bien entendu, cette approche rapide présuppose que la valeur de

$$\left|\bigcup_{j \in J'} \{\sigma_j(\phi_j), \dots, \sigma_j(n)\}\right|$$

soit connue pour tout sous-ensemble $J' \subset \{1, \dots, s\}$ de cardinalité l. Il paraît raisonnable de déterminer ces cardinalités en $O(nl\binom{s}{l})$ au début du processus, puis de les mettre à jour en $O(l\binom{s-1}{l-1})$ lors de chaque incrémentation d'une composante du vecteur ϕ.

REMARQUE 8.15 Dans le cas particulier où x est une séquence de variables aléatoires i.i.d. $U(\bar{I}^s)$, l'analyse probabiliste sommaire présentée ci-dessous fournit une estimation du paramètre ψ recherché dans la première phase de la méthode. En effet, on remarque que pour tout index $i \in \{1, \dots, n\}$, on a

$$P\left\{ i \in \bigcup_{j=1}^{l} \{\sigma_j(\psi), \dots, \sigma_j(n)\} \right\} = 1 - \left(\frac{\psi - 1}{n}\right)^l.$$

On en déduit l'espérance

$$E\left(\left|\bigcup_{j=1}^{l} \{\sigma_j(\psi), \dots, \sigma_j(n)\}\right|\right) = n - n\left(\frac{\psi - 1}{n}\right)^l.$$

Si l'on impose que cette valeur soit supérieure à $n - k$ (une approximation grossière), on obtient

$$\psi < 1 + k^{\frac{1}{l}} n^{\frac{l-1}{l}}.$$

Notons que le raisonnement ci-dessus ne concerne que $J' = \{1, \dots, l\}$, alors que la condition (57) porte sur tous les sous-ensembles de $\{1, \dots, s\}$ de cardinalité l. Toutefois, en pratique, l'estimateur $\lceil k^{\frac{1}{l}} n^{\frac{l-1}{l}} \rceil$ constitue un bon point de départ pour une recherche locale de l'index maximal ψ.

8.3.1.2 *Le cas des problèmes du type* $V^k_{\max}$. Bien que nos expériences numériques préliminaires (voir page 106) suggèrent que la relaxation linéaire d'un programme de la forme $\log(V^k_{\max})$ mène généralement à une borne supérieure de bonne qualité sur l'optimum, il pourrait néanmoins être intéressant d'introduire quelques inégalités valides dans la formulation. Or, il se trouve que les idées présentées ci-dessus pour $V^k_{\min}$ s'adaptent directement au cas des problèmes du type $V^k_{\max}$. On a par exemple l'équivalent du théorème 8.9 :

THÉORÈME 8.16 *Soit un entier* $l \in \{2, \dots, s\}$*, un sous-ensemble de composantes* $J \subset \{1, \dots, s\}$ *tel que* $|J| \geq l$ *et des index* $\phi_j \in \{k+2, \dots, n+1\}$ *définis pour tout* $j \in J$ *tels que*

$$\left| \bigcup_{j \in J'} \{\sigma_j(\phi_j - 1), \dots, \sigma_j(n)\} \right| > n - k, \text{ pour tout sous-ensemble } J' \subset J \text{ tel que } |J'| = l.$$

Alors, les inégalités suivantes sont valides pour le programme linéaire $\log(V^k_{\max})$ *:*

$$\sum_{j \in J'} z_j^{\sigma_j(\phi_j)} \geq |J'| - l + 1, \text{ pour tout sous-ensemble } J' \subset J \text{ tel que } |J'| \geq l.$$

PREUVE. La démonstration est scrupuleusement identique à celle du théorème 8.9 à un détail près : ce sont des contraintes du type

$$y^i \leq z_j^{\sigma_j(1+\sigma_j^{-1}(i))}$$

qui doivent être considérées pour aboutir à l'implication

$$z_j^{\sigma_j(\phi_j)} = 0, \ \forall j \in J' \quad \Longrightarrow \quad y^i = 0, \ \forall i \in \{\sigma_j(\phi_j - 1), \dots, \sigma_j(n)\}, \ \forall j \in J'$$

menant à l'obtention de la contradiction désirée. $\square$

L'adaptation du théorème 8.10 et des résultats concernant les (l, ϕ)-coupes au cas des problèmes $V^k_{\max}$ s'effectue tout aussi facilement. Cependant, ces éléments n'intervenant pas dans l'algorithme de calcul de la discrépance énoncé dans la section 8.4, ils ne sont pas détaillés ici.

8.3.2 Heuristiques. Les heuristiques proposées ci-dessous permettent, suivant le type de problème considéré, de construire des intervalles de k points dont le volume est une borne supérieure pour $V^k_{\min}$ ou inférieure pour $V^k_{\max}$. Les intervalles en question sont généralement d'excellentes solutions admissibles des programmes linéaires correspondants (on observe qu'en pratique les volumes obtenus sont rarement à plus de quelques pour cent de l'optimum).

Pour $V^k_{\min}$, l'inévitable algorithme glouton **MinGlouton** permet d'obtenir un intervalle de volume restreint contenant k points. Partant de $[0, 0]$, ce dernier consiste à ajouter à chaque itération le point qui minimise l'augmentation du volume de l'intervalle résultant. **MinGlouton** utilise la fonction auxiliaire

$$\mathrm{Vol}_\delta(i) = \prod_{j=1}^{s} \max\left\{x_j^{\sigma_j(\delta_j)}, x_j^i\right\}.$$

qui calcule le volume du plus petit intervalle contenant à la fois $\bar{P}(\delta)$ et le point x^i.

Min2opt est une heuristique inspirée de la célèbre technique de Lin [Lin65] pour le problème du voyageur de commerce. Partant de la solution initiale fournie par **MinGlouton**, cet algorithme effectue des échanges aussi longtemps qu'il en existe un qui induise une réduction de volume. Un échange est une transformation qui consiste à remplacer un point situé dans l'intervalle courant (forcément sur le bord) par un point se trouvant à l'extérieur. Clairement, **Min2opt** converge vers un minimum local. Après avoir appliqué **MinGlouton** et **Min2opt**, le volume V obtenu est une borne supérieure pour $V^k_{\min}$.

Algorithme 8.1 MinGlouton

```
Donnée : $x, s, n, k$
Résultat : $\{y_1, \dots, y_n\}, \{\delta_1, \dots, \delta_s\}$
  $y^i \leftarrow 0$ pour tout $i \in \{1, \dots, n\}$
  $\delta_j \leftarrow 1$ pour tout $j \in \{1, \dots, s\}$
  pour $t$ de $1$ à $k$ faire
    choisir $p \in \{1, \dots, n\}$ avec $y^p = 0$ minimisant Vol$_\delta(i)$
    $y^p \leftarrow 1$
    $\delta_j \leftarrow \max\left\{\delta_j, \sigma_j^{-1}(p)\right\}$ pour tout $j \in \{1, \dots, s\}$
```

Algorithme 8.2 Min2opt

```
Donnée : résultat de MinGlouton
Résultat : $V, \{y_1, \dots, y_n\}, \{\delta_1, \dots, \delta_s\}$
  $V \leftarrow \prod_{j=1}^{s} x_j^{\sigma_j(\delta_j)}$
  répéter
    2opt $\leftarrow$ vrai
    pour tout $d \in \{1, \dots, s\}$ faire          $(*$ le point $x^{\sigma_d(\delta_d)}$ est candidat à être retiré $*)$
      $\delta' \leftarrow \delta$
      pour tout $j \in \{1, \dots, s\}$ faire
        si $\sigma_j(\delta'_j) = \sigma_d(\delta_d)$ alors
          répéter
            $\delta'_j \leftarrow \delta'_j - 1$          $(*$ enlever ce point et « tasser » l'intervalle au maximum $*)$
          jusqu'à ce que $y^{\sigma_j(\delta'_j)} \neq 0$
      choisir $p \in \{1, \dots, n\}$ avec $y^p = 0$ minimisant Vol$_{\delta'}(i)$
      si Vol$_{\delta'}(p) < V$ alors          $(*$ le point $x^p$ est candidat à être ajouté $*)$
        2opt $\leftarrow$ faux
        $V \leftarrow$ Vol$_{\delta'}(p)$
        $y^p \leftarrow 1$
        $y^{\sigma_d(\delta_d)} \leftarrow 0$
        $\delta_j \leftarrow \max\left\{\delta'_j, \sigma_j^{-1}(p)\right\}$ pour tout $j \in \{1, \dots, s\}$
  jusqu'à ce que 2opt $=$ vrai
```

Des heuristiques correspondantes sont également proposées pour $V^k_{\max}$. Après une initialisation basée sur l'observation 8.6, **MaxGlouton** construit un intervalle $P(\delta)$ de grande taille contenant k points de la séquence. Le principe de cet algorithme est d'incrémenter à chaque itération l'index δ_d induisant une augmentation de volume maximale. De plus, **MaxGlouton** utilise une procédure auxiliaire **Agrandir**(δ) qui permet d'étendre l'intervalle $P(\delta)$ sans y ajouter de point supplémentaire.

Algorithme 8.3 MaxGlouton

```
Donnée : $x, s, n, k$
Résultat : $\{y_1, \dots, y_n\}, \{\delta_1, \dots, \delta_s\}$
  $\delta_j \leftarrow k + 1$ pour tout $j \in \{1, \dots, s\}$
  $y^i \leftarrow \begin{cases} 1 & \text{si } \sigma_j^{-1}(i) \leq k \text{ pour tout } j \in \{1, \dots, s\} \\ 0 & \text{sinon} \end{cases}$ pour tout $i \in \{1, \dots, n\}$
  $S \leftarrow \sum_{i=1}^{n} y^i$
  tant que $S < k$ faire
    choisir $d \in \{1, \dots, s\}$ avec $\delta_d \leq n$ maximisant $\Delta(j) = \log x_j^{\sigma_j(\delta_j+1)} - \log x_j^{\sigma_j(\delta_j)}$
    si $\sigma_j^{-1}(\sigma_d(\delta_d)) < \delta_j$ pour tout $j \neq d$ alors
      $y^{\sigma_d(\delta_d)} \leftarrow 1$
      $S \leftarrow S + 1$
    $\delta_d \leftarrow \delta_d + 1$
  Agrandir($\delta$)
```

Algorithme 8.4 Max2opt

```
Donnée : résultat de MaxGlouton
Résultat : $V, \{y_1, \dots, y_n\}, \{\delta_1, \dots, \delta_s\}$
  $V \leftarrow \prod_{j=1}^{s} x_j^{\sigma_j(\delta_j)}$
  répéter
    2opt $\leftarrow$ vrai
    pour tout $d \in \{1, \dots, s\}$ faire        ($*$ le point $x^{\sigma_d(\delta_d)}$ est candidat à être ajouté $*$)
      $\delta'_j \leftarrow \begin{cases} \delta_j & \text{si } j \neq d \\ \delta_d + 1 & \text{sinon} \end{cases}$ pour tout $j \in \{1, \dots, s\}$
      Agrandir($\delta'$)
      choisir $c \neq d$ minimisant $V(j) = \min_{i \in \{\sigma_d(\delta_d)\} \cup \{p \in \{1, \dots, n\} \,:\, y^p = 1\}} \left( \log x_j^{\sigma_j(\delta'_j)} - \log x_j^i \right)$
      $i \leftarrow \arg\min V(c)$
      $\delta'_c \leftarrow \sigma_c^{-1}(i)$        ($*$ le point $x^i$ est candidat à être retiré $*$)
      Agrandir($\delta'$)
      si $V < \prod_{j=1}^{s} x_j^{\sigma_j(\delta'_j)}$ alors
        2opt $\leftarrow$ faux
        $y^{\sigma_d(\delta_d)} \leftarrow 1$
        $y^i \leftarrow 0$
        $V \leftarrow \prod_{j=1}^{s} x_j^{\sigma_j(\delta'_j)}$
        $\delta \leftarrow \delta'$
  jusqu'à ce que 2opt $=$ vrai
```

Algorithme 8.5 Permet d'agrandir un intervalle $P(\delta)$ donné sans y ajouter de point supplémentaire.

```
Procédure Agrandir($\delta$)
répéter
    stop $\leftarrow$ vrai
    choisir $d \in \{1,\dots,s\}$ maximisant
        $\Delta(j) = \begin{cases} \log x_j^{\sigma_j(\delta_j+1)} - \log x_j^{\sigma_j(\delta_j)} & \text{si } \delta_j \leq n \text{ et } \exists c \neq j : \sigma_c^{-1}(\sigma_j(\delta_j)) \geq \delta_c \\ 0 & \text{sinon} \end{cases}$
    si $\Delta(d) > 0$ alors
        $\delta_d \leftarrow \delta_d + 1$
        stop $\leftarrow$ faux
jusqu'à ce que stop $=$ vrai
```

Partant de la solution fournie par **MaxGlouton**, l'heuristique correspondante **Max2opt** cherche à améliorer cet intervalle initial en effectuant des échanges aussi longtemps qu'il en existe un qui induise une augmentation de volume. Cet algorithme converge vers un maximum local. Après avoir appliqué successivement **MaxGlouton** et **Max2opt**, le volume V obtenu est une borne inférieure pour $V_{\max}^k$.

En pratique, ces heuristiques conduisent presque systématiquement à des résultats d'excellente qualité, mais il suffit de quelques esquisses dans le carré unité pour se convaincre que, dans le pire des cas, elles peuvent également mener à des résultats arbitrairement mauvais.

8.3.3 Techniques de résolution des programmes linéaires en nombres entiers.

8.3.3.1 *Introduction de plans coupants.* Lorsque la solution optimale de la relaxation linéaire d'un de nos programmes n'est pas entière, il est envisageable de déterminer (s'il en existe) une inégalité valide de la forme de celles proposées dans les théorèmes 8.9 ou 8.16 qui soit violée, de l'introduire dans la formulation et de résoudre le nouveau problème obtenu. Un tel processus pourrait théoriquement être poursuivi jusqu'à l'obtention d'une solution entière ou épuisement de la famille de coupes en question. Malheureusement, on ne sait pas comment résoudre un tel problème de séparation (*i.e.* décider si une solution non entière donnée vérifie toutes les inégalités que l'on pourrait obtenir à l'aide de ces théorèmes ou en trouver une qui soit violée) en un temps raisonnable (*i.e.* polynomial en s et n).

En revanche, la question devient abordable si l'on se limite à la sous-famille d'inégalités valides donnée (par exemple) par $l = |J| = 2$. Dans ce cas particulier, pour un programme du type $\log(V_{\min}^k)$, il s'agit de trouver une paire d'index $(\phi_i, \phi_j) \in \{k+1,\dots,n\}^2$ vérifiant la condition

$$|\{\sigma_i(\phi_i),\dots,\sigma_i(n)\} \cup \{\sigma_j(\phi_j),\dots,\sigma_j(n)\}| > n-k, \tag{61}$$

mais telle que l'inégalité valide résultante (55)

$$z_i^{\sigma_i(\phi_i)} + z_j^{\sigma_j(\phi_j)} \geq 1 \tag{62}$$

soit violée pour la solution optimale de la relaxation linéaire courante. Comme pour les (l,ϕ)-coupes, il suffit de se limiter à la considération de vecteurs (ϕ_i, ϕ_j) maximaux (*i.e.* pour lesquels il n'existe aucune paire d'index $(\phi'_i, \phi'_j) \gneqq (\phi_i, \phi_j)$ telle que la condition (61) soit satisfaite). Sachant que, pour une paire de composantes donnée, il existe au plus $n-k$ vecteurs maximaux, ce processus peut théoriquement mener à la génération d'un maximum de $\binom{s}{2}(n-k)$ plans coupants (62).

Pour une paire de composantes (i, j) fixée ($1 \leq i < j \leq s$), l'ensemble des vecteurs maximaux (ϕ_i, ϕ_j) pour lesquels la condition (61) est vérifiée peut être obtenu en $O(n-k)$ à l'aide de la technique de balayage qui suit. On commence par associer à toute paire d'index (ϕ_i, ϕ_j) la quantité

$$u(\phi_i, \phi_j) = |\{\sigma_i(\phi_i), \ldots, \sigma_i(n)\} \cup \{\sigma_j(\phi_j), \ldots, \sigma_j(n)\}| \, .$$

On observe qu'un vecteur (ϕ_i, ϕ_j) ne peut être maximal que si $u(\phi_i, \phi_j) = n-k+1$ (cette condition n'est pas suffisante). En partant de $\phi_i = k+1$ et $\phi_j = n$, on remarque que $u(\phi_i, \phi_j)$ est égal à $n-k$ si $\sigma_i^{-1}(\sigma_j(n)) \geq k+1$ ou à $n-k+1$ dans le cas contraire. On procède alors comme suit :

Si $u(\phi_i, \phi_j) = n-k$, alors ϕ_j est diminué de 1 jusqu'à ce que $u(\phi_i, \phi_j) = n-k+1$ ou $\phi_j = k$. Si cette première phase aboutit à $\phi_j = k$, la recherche est terminée : il n'existe pas d'autre vecteur maximal pour les composantes (i, j) en question. Dans le cas contraire, on procède à une seconde phase qui consiste à augmenter ϕ_i de 1 tant que $\phi_i < n$ et $u(\phi_i + 1, \phi_j) = n-k+1$. Le vecteur (ϕ_i, ϕ_j) ainsi obtenu est maximal et l'inégalité valide correspondante (62) est ajoutée dans la formulation du programme $\log(V^k_{\min})$ si elle s'avère violée pour la solution optimale de la relaxation linéaire courante.

Puis, si $\phi_i < n$ et $\phi_j > k+1$, il suffit de diminuer la valeur de ϕ_j d'une unité et de recommencer le processus jusqu'à trouver (s'il en existe) une autre paire maximale pour laquelle la condition (61) est satisfaite. À chaque itération, $u(\phi_i, \phi_j)$ peut être mis à jour en $O(1)$ en vérifiant si $\sigma_i^{-1}(\sigma_j(\phi_j)) \geq \phi_i$ (lorsque ϕ_j est diminué) ou $\sigma_j^{-1}(\sigma_i(\phi_i - 1)) \geq \phi_j$ (lorsque ϕ_i est augmenté).

8.3.3.2 *La technique des variables astreintes.* L'approche suivante peut également être utilisée afin d'améliorer la solution relaxée courante lorsqu'elle s'avère non entière. L'idée consiste à choisir une variable que l'on suspecte de prendre une certaine valeur $v \in \{0, 1\}$ dans la solution optimale et de résoudre la nouvelle relaxation obtenue en la fixant à $1-v$. Si ce problème auxiliaire ne possède pas de solution admissible ou si sa valeur est moins bonne que celle de la meilleure solution entière connue (*i.e.* plus grande, respectivement plus petite, que le logarithme du volume de la solution construite à partir des heuristiques de la section 8.3.2 dans le cas d'un problème du type $V^k_{\min}$, respectivement $V^k_{\max}$), alors cette variable est égale à v dans toute solution optimale et peut donc être fixée à cette valeur.

Bien entendu, l'utilisation d'une technique aussi brutale pour toute variable du programme linéaire peut, en fin de compte, coûter plus qu'elle ne rapporte si l'on compare son efficacité à celle d'une méthode d'énumération par séparation et évaluation. Néanmoins, appliquée pour un ensemble raisonnable de variables, l'approche peut s'avérer profitable. Expérimentalement, une stratégie heuristique judicieuse consiste à choisir à chaque itération la variable non entière dont la valeur est la plus proche de 0 ou de 1 et à s'arrêter au premier échec rencontré.

En général, cette méthode permet de fixer quelques variables du problème. Il est alors envisageable d'appliquer à nouveau la technique proposée dans la section 8.3.3.1 au problème simplifié obtenu ou de passer à une énumération par séparation et évaluation.

8.3.3.3 *Énumération par séparation et évaluation.* Il s'agit d'une technique puissante et générale pour la résolution de programmes linéaires en nombres entiers. L'idée de base est de diviser l'ensemble des solutions admissibles en sous-ensembles de plus en plus petits jusqu'à isoler dans l'un d'eux une solution optimale. L'objectif visé est l'énumération implicite, par l'évaluation de bornes, d'une partie importante de ces sous-ensembles. Cette méthode est brièvement décrite ci-dessous dans le cas particulier des programmes à variables binaires. Nous renvoyons le lecteur à l'ouvrage de Wolsey [Wol98] pour une présentation plus détaillée.

Tout programme linéaire binaire peut être séparé en deux sous-problèmes dans lesquels une variable est fixée dans un cas à 0 et dans l'autre à 1. L'application répétée de ce procédé mène à une partition, que l'on représente le plus souvent sous la forme d'un arbre binaire, de l'ensemble des solutions admissibles du programme de départ. Schématiquement, une méthode d'énumération par séparation et évaluation consiste en l'application cyclique de trois phases :

- ▷ La *sélection* : un sous-problème est choisi.
- ▷ L'*évaluation* : une borne pour la solution optimale de ce sous-problème est calculée. Si, pour simplifier, on suppose qu'il s'agit de la solution optimale de la relaxation linéaire associée, trois situations (mutuellement exclusives) sont à envisager.
  - 1° La solution est entière : dans ce cas, sa valeur est comparée à la meilleure solution entière connue, conservée si elle s'avère meilleure et le sous-problème est éliminé ;
  - 2° Il n'existe pas de solution admissible ou sa valeur est moins bonne que la meilleure solution entière connue : le sous-problème est éliminé ;
  - 3° Dans les autres cas, le sous-problème est séparé.
- ▷ La *séparation* : le sous-problème est éliminé et remplacé par une paire de nouveaux programmes dans lesquels une variable (encore libre) est fixée dans un cas à 0 et dans l'autre à 1.

Bien que, dans le pire des cas, la complexité de cette méthode soit exponentielle en la taille du programme à résoudre, elle a déjà largement prouvé sa valeur sur de nombreux problèmes d'optimisation combinatoire réputés difficiles. En pratique, l'efficacité d'une mise en œuvre repose essentiellement sur les choix effectués dans les trois phases décrites ci-dessus. C'est en effet à ce niveau-là qu'il est possible de tirer parti de la structure particulière du problème étudié pour aboutir rapidement à l'obtention d'une solution optimale.

Par ailleurs, il existe aujourd'hui d'excellents logiciels de résolution de programmes linéaires en nombres entiers qui s'avèrent capables, dans une certaine mesure, d'analyser eux-mêmes la structure du problème, mais qui sont également suffisamment souples pour laisser à l'utilisateur une certaine marge de manœuvre pour intervenir. C'est ainsi qu'entrent en scène nos deux atouts les plus précieux : des formulations renforcées de nos programmes linéaires susceptibles de donner lieu à des bornes de bonne qualité lors de l'évaluation des sous-problèmes et des heuristiques capables de fournir d'excellentes solutions entières admissibles.

De plus, dans l'optique de l'utilisation d'une méthode d'énumération par séparation et évaluation pour un programme du type $\log(V^k_{\min})$, la considération du théorème suivant permet de simplifier le problème. En effet, par ce résultat, on passe d'une formulation comportant $n+s(n-k)$ variables binaires (en tenant compte des observations 8.5 et 8.6) à un programme linéaire mixte comprenant $s(n-k)$ variables réelles et seulement n variables binaires.

THÉORÈME 8.17 *Dans un programme de la forme* $\log(V^k_{\min})$*, les contraintes d'intégralité*

$$z^i_j \in \{0,1\},\ \forall i \in \{1,\dots,n+1\},\ \forall j \in \{1,\dots,s\}$$

peuvent être relâchées.

PREUVE. Supposons par l'absurde que certaines variables z^i_j prennent des valeurs non entières dans une solution optimale de ce programme relaxé. Soit $z_j^{\sigma_j(i)}$ une variable de cet ensemble telle que $z_j^{\sigma_j(i+1)} = 0$ (on note que $i \leq n$ par l'observation 8.5) et $p \in \{1,\dots,n\}$ le plus petit index tel que

$$x_j^{\sigma_j(p)} = x_j^{\sigma_j(i)}.$$

Considérant les contraintes (50), on a

$$z_j^{\sigma_j(p)} = \cdots = z_j^{\sigma_j(i)}.$$

De plus, les variables $y^{\sigma_j(p)}, \ldots, y^{\sigma_j(i)}$ étant binaires, les contraintes (49)

$$\begin{aligned} y^{\sigma_j(p)} &\leq z_j^{\sigma_j(p)} \\ &\vdots \\ y^{\sigma_j(i)} &\leq z_j^{\sigma_j(i)} \end{aligned}$$

impliquent que

$$y^{\sigma_j(p)} = \cdots = y^{\sigma_j(i)} = 0.$$

On remarque que la nouvelle solution obtenue en posant

$$z_j^{\sigma_j(p)} = \cdots = z_j^{\sigma_j(i)} = 0$$

est admissible et de valeur strictement inférieure à la solution initiale ($z_j^{\sigma_j(p)}$ est la seule variable modifiée de coût non nul). Il s'agit d'une contradiction et le résultat est démontré. $\square$

Malheureusement, la propriété correspondante pour les programmes du type $\log(V_{\max}^k)$ n'est pas vraie. Il semble cependant que la construction de contre-exemples nécessite la considération de séquences présentant des propriétés tout à fait inhabituelles et donc peu susceptibles d'apparaître en pratique.

8.3.4 Expériences numériques. Les résultats présentés dans cette section ont été obtenus pour des séquences aléatoires de n points i.i.d. $U(\bar{I}^s)$ (en fait, des échantillons du générateur pseudo-aléatoire MRG32k3a de L'Ecuyer (15) ont été utilisés). Naturellement, chaque expérience a été répliquée plusieurs fois (en général entre 5 et 10), afin de garantir une certaine précision aux estimateurs établis.

Dans le cas où x est une séquence de n points i.i.d. $U(\bar{I}^s)$, les volumes optimaux $V_{\min}^k$ et $V_{\max}^k$ sont des variables aléatoires ne dépendant que de s, n et k. Dans la figure 8.5, des estimateurs de l'espérance de $V_{\min}^k$ et $V_{\max}^k$ sont donnés pour k variant entre 0 et n dans le cas particulier où $s = 10$ et $n = 100$. Par ailleurs, dans la figure 8.6, on fournit des estimateurs de l'espérance de $V_{\min}^{n/2}$ et $V_{\max}^{n/2}$ pour $s = 7$ et $n \in \{0, \ldots, 200\}$ (par la loi forte des grands nombres et le fait qu'une suite aléatoire uniforme soit équirépartie, les deux courbes représentées tendent vers 0.5 pour $n \to \infty$). Bien qu'a priori fort intéressante, l'étude théorique de ces variables aléatoires n'est pas entreprise dans ce travail (la question étant d'une part hors sujet et d'autre part vraisemblablement d'une extrême complexité).

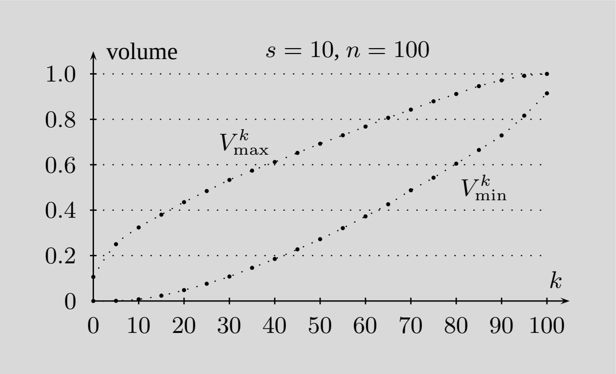


FIG. 8.5. Estimateurs de l'espérance des volumes $V_{\min}^k$ et $V_{\max}^k$ pour k variant de 0 à 100 dans le cas particulier où x est une séquence aléatoire de 100 points uniformément distribués dans le cube unité à 10 dimensions.

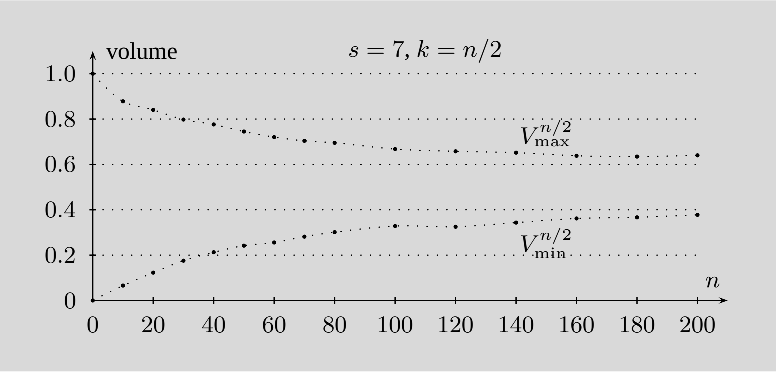


FIG. 8.6. Estimateurs de l'espérance des volumes $V_{\min}^{n/2}$ et $V_{\max}^{n/2}$ pour une séquence aléatoire de n points, avec $n \in \{0, \ldots, 200\}$, uniformément distribués dans le cube unité à 7 dimensions.

En revanche, on fournit ci-dessous quelques résultats sur le temps de calcul de $V_{\min}^k$ et $V_{\max}^k$ (sur une station de travail SGI R10000) en fonction de s, n et k. De nombreuses expériences ont été nécessaires pour la mise au point d'une stratégie efficace d'utilisation des différentes techniques de résolution présentées plus haut. Comme nous l'avons déjà mentionné, la formulation initiale de nos programmes linéaires est bien meilleure pour $V_{\max}^k$ que pour $V_{\min}^k$. Il n'est donc guère surprenant que les différents ingrédients à disposition aient plus d'impact sur les programmes du type $\log(V_{\min}^k)$ que pour ceux de la forme $\log(V_{\max}^k)$. En fin de compte, les éléments suivants ont été retenus[1] :

- ▷ pour les programmes $\log(V_{\min}^k)$: les observations 8.5, 8.6, 8.7 et 8.8, l'ajout d'une (l, ϕ)-coupe équilibrée pour tout $l \in \{2, 3, s-2, s-1, s\}$, la technique des variables astreintes (voir section 8.3.3.2) uniquement pour les variables y^j prenant des valeurs proches de 1 et l'introduction (une seule fois) de la famille de plans coupants proposée dans la section 8.3.3.1 ;
- ▷ pour les programmes $\log(V_{\max}^k)$: uniquement les observations 8.6, 8.7 et 8.8.

La librairie CPLEX [CPL01] a été utilisée de manière intensive pour la résolution des programmes linéaires en question. Dans chaque cas, les heuristiques de la section 8.3.2 ont permis de construire une solution entière initiale de bonne qualité.

Dans une première série d'expériences concernant les problèmes du type $V_{\min}^k$, la dimension a été fixée à 10 et le nombre de points n à 100. Dans ce cas particulier, le temps de calcul a été mesuré pour k variant de 0 à n. Dans la figure 8.7, on remarque que la courbe des durées moyennes obtenues présente une forme de cloche autour des cas les plus difficiles (qui semblent se situer pour k proche de $n/4$). De plus, on observe qu'à partir de $k = n/2$, le temps de calcul décroît très rapidement lorsque k grandit (on peut raisonnablement supposer qu'il s'agit principalement d'une conséquence de l'observation 8.6).

Pour les mêmes séquences de 100 points, les temps de calcul de $V_{\max}^k$ obtenus sont inférieurs à 10 secondes pour toute valeur de $k \in \{0, \ldots, 100\}$. Plus généralement, la résolution d'un programme du type $\log(V_{\max}^k)$ s'avère presque systématiquement plus facile que celle de son homologue $\log(V_{\min}^k)$. L'expérience correspondante pour $V_{\max}^k$ a donc été effectuée pour des séquences de $n = 250$ points, toujours en dimension $s = 10$. On observe sur la figure 8.8 que les cas les plus complexes semblent cette fois se situer pour k proche de $n/10$. On obtient à nouveau une courbe en forme de cloche et on remarque que le problème est déjà difficile pour de très petites valeurs de k. Une autre différence entre les programmes $\log(V_{\min}^k)$ et $\log(V_{\max}^k)$ apparaît au niveau de la variance du temps de calcul. En effet,

---

[1] Notons que ce choix a été optimisé en vue des expériences sur le calcul de la discrépance présentées dans la section 8.5.

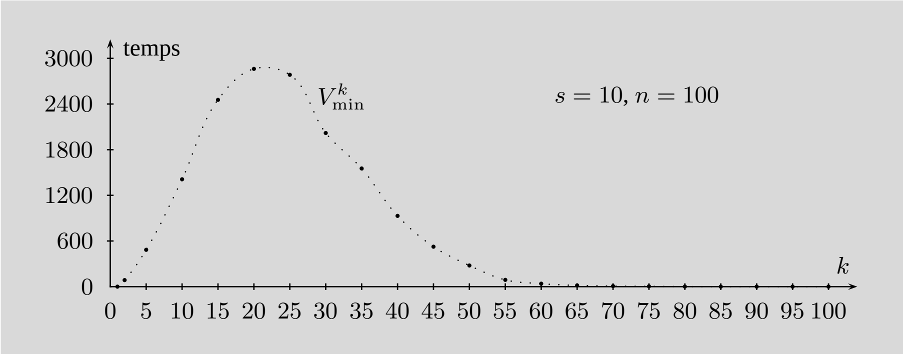


FIG. 8.7. Temps moyen de calcul (en secondes) de $V^k_{\min}$ en fonction de k pour $n = 100$ points en dimension $s = 10$.

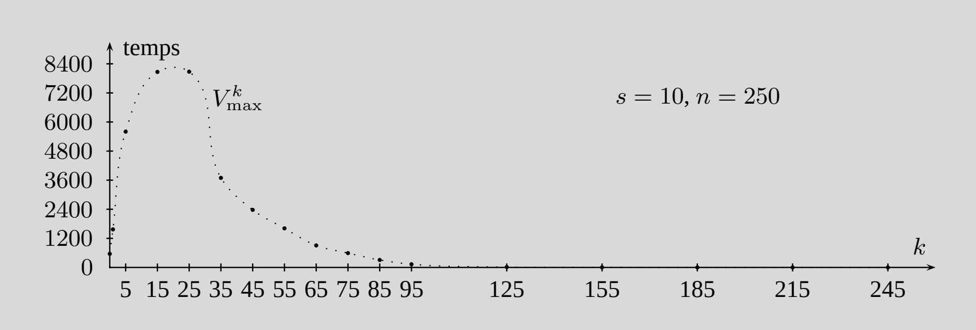


FIG. 8.8. Temps moyen de calcul (en secondes) de $V^k_{\max}$ en fonction de k pour $n = 250$ points en dimension $s = 10$.

alors que (pour des valeurs fixées de s, n et k) cette durée est relativement stable pour $V^k_{\min}$, elle s'avère très variable pour $V^k_{\max}$. Notons que les résultats de cette paire d'expériences sont typiques et qu'un comportement similaire peut être observé pour des valeurs différentes de s et n.

On poursuit cette étude par une expérience sur l'évolution du temps de calcul en fonction de la dimension s (voir figure 8.9), puis de la taille n de la séquence (voir figure 8.10). Les résultats étant qualitativement les mêmes pour les deux types de problèmes, on se limite à leur présentation pour $V^k_{\min}$.

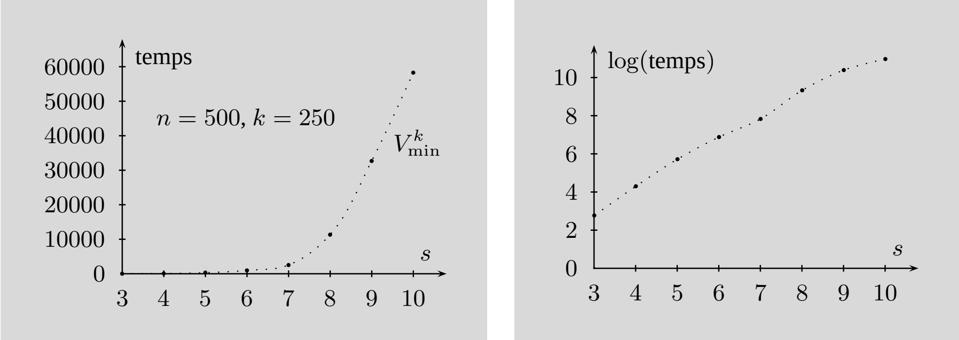


FIG. 8.9. À gauche, temps moyen de calcul (en secondes) de $V^{250}_{\min}$ pour une séquence de $n = 500$ points en dimension s, pour $s \in \{3, \ldots, 10\}$. À droite, le logarithme de cette durée.

Dans la figure 8.9, on a représenté le temps moyen de calcul lorsque $k = n/2 = 250$ points dans le cube unité en dimension s, pour $s \in \{3, \ldots, 10\}$. Sur la partie de droite, on remarque que, en prenant le logarithme de ces durées, le comportement obtenu semble quasiment linéaire. Cette observation suggère que le temps nécessaire à la résolution du problème croît exponentiellement avec la dimension.

Dans une dernière série d'expériences (voir figure 8.10), la dimension s a été fixée à 7 et on a mesuré (en fonction de n) le temps moyen de calcul dans le cas difficile (voir figure 8.7) où $k = n/4$. En considérant le logarithme de ces durées, la courbe obtenue est approximativement concave. Ce fait semble indiquer que le temps de calcul ne croît pas de manière exponentielle avec la taille de la séquence. Cependant, le comportement étant inconnu pour $n > 360$ ou d'autres valeurs de s et k, il serait hasardeux de tirer une conclusion si générale de cette simple observation. D'autre part, en prenant la racine cubique de ces durées, la courbe obtenue paraît également légèrement concave. La prudence est à nouveau de mise, mais cette expérience suggère que la croissance du temps de calcul n'est vraisemblablement pas plus rapide que n^3. Cependant, en effectuant la même analyse pour d'autres valeurs de s, il semble que le degré de ce présumé polynôme en n ne soit pas indépendant de la dimension.

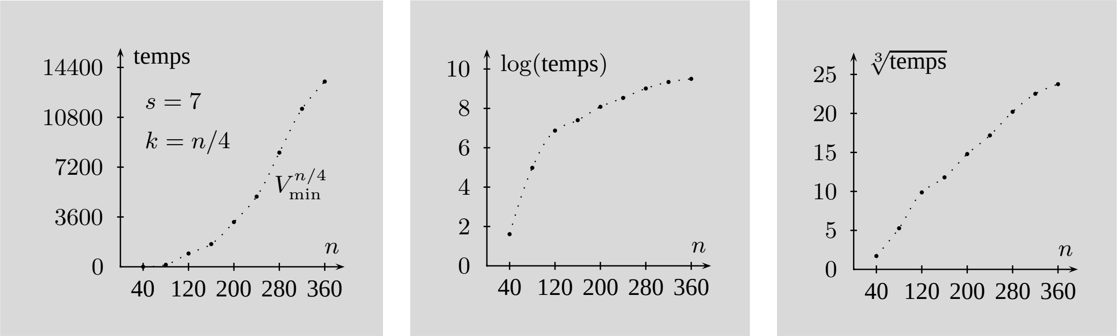


FIG. 8.10. À gauche, temps moyen de calcul (en secondes) de $V_{\min}^{n/4}$ pour une séquence de n points en dimension 7. Au centre, le logarithme de cette durée et à droite, sa racine cubique.

8.4 Un processus dynamique

Dans cette section, la décomposition du calcul de la discrépance introduite dans la section 8.1, les problèmes énoncés dans la section 8.2 et les techniques de résolution correspondantes présentées dans la section 8.3 sont réunies dans un processus dynamique. L'idée de base est de maintenir et d'améliorer des bornes pour les différents éléments apparaissant dans l'expression (46) de manière à ce que la plupart des configurations concernées soient évaluées implicitement. Partant d'un intervalle initial, la méthode consiste en une succession d'améliorations menant au calcul de la discrépance $D_n^*(x)$ d'une séquence x de n points dans $\bar{I}^s$.

Compte tenu du fait que $V_{\min}^0 = 0$ et $V_{\max}^n = 1$, l'expression (46) s'écrit

$$D_n^*(x) = \max\left\{ \max_{k\in\{0,\ldots,n-1\}} \left(V_{\max}^k - \frac{k}{n} \right), \max_{k\in\{1,\ldots,n\}} \left(\frac{k}{n} - V_{\min}^k \right) \right\}. \tag{63}$$

Comme nous l'avons annoncé dans la section 8.2.2, l'hypothèse énoncée sur le fait que la séquence ne contient aucun point présentant une composante nulle peut être assouplie dans le cas de l'origine. Ce point étant le premier de la plupart des (t, s)-suites en base b classiques (ou des suites de Halton), la remarque suivante s'avère précieuse.

REMARQUE 8.18 Le calcul de la discrépance d'une séquence $x \cup \{0\}$ de $n+1$ points comprenant l'origine se ramène au calcul des volumes optimaux $V_{\min}^k(x)$ et $V_{\max}^k(x)$ pour la séquence x de n points. En effet, on observe que

$$V_{\max}^k(x \cup \{0\}) = \begin{cases} 0 & \text{si } k = 0 \\ V_{\max}^{k-1}(x) & \text{sinon} \end{cases}, \text{ pour tout } k \in \{0, \dots, n\}$$

et

$$V_{\min}^k(x \cup \{0\}) = \begin{cases} 0 & \text{si } k = 1 \\ V_{\min}^{k-1}(x) & \text{sinon} \end{cases}, \text{ pour tout } k \in \{1, \dots, n+1\}.$$

En adaptant l'expression (63), on obtient

$$D_{n+1}^*(x \cup \{0\}) = \max \left\{ \max_{k \in \{0, \dots, n-1\}} \left(V_{\max}^k(x) - \frac{k+1}{n+1} \right), \max_{k \in \{1, \dots, n\}} \left(\frac{k+1}{n+1} - V_{\min}^k(x) \right), \frac{1}{n+1} \right\}.$$

Ainsi, pour calculer la discrépance d'une séquence de $n+1$ points comprenant l'origine, il suffit de retirer le point en question et d'utiliser l'expression ci-dessus. On remarque au passage que la discrépance d'une telle séquence est supérieure ou égale à $\frac{1}{n+1}$.

Il est bien évidemment possible de calculer la discrépance d'une séquence à l'aide de l'expression (63) après avoir résolu les $2n$ programmes linéaires en nombres entiers correspondants. Toutefois, cette approche directe est une perte de temps. En effet, le maximum dans (63) étant dans la plupart des cas atteint pour un seul des $2n$ problèmes, il suffit de résoudre le programme en question et d'établir des bornes (inférieures pour $V_{\min}^k$ et supérieures pour $V_{\max}^k$) suffisamment précises pour les autres. Il semble malheureusement très difficile de deviner quel est le problème à résoudre exactement.

Une stratégie permettant de ne résoudre qu'un petit sous-ensemble de ces $2n$ problèmes est proposée ci-dessous, mais il est d'abord nécessaire d'introduire quelques nouvelles notations. En premier lieu, on remarque qu'à l'aide des grandeurs[2]

$$D_{\min}^k = \frac{k}{n} - V_{\min}^k \quad \text{et} \quad D_{\max}^k = V_{\max}^k - \frac{k}{n},$$

l'expression (63) s'écrit

$$D_n^*(x) = \max \left\{ \max_{k \in \{0, \dots, n-1\}} D_{\max}^k \, , \max_{k \in \{1, \dots, n\}} D_{\min}^k \right\}. \tag{64}$$

8.4.1 Initialisation. Pour $D_n^*(x)$, $D_{\min}^k$, $D_{\max}^k$, $V_{\min}^k$ et $V_{\max}^k$, on définit une paire de bornes, l'une inférieure et l'autre supérieure :

$$\begin{array}{ccccc} \underline{\mathrm{D}}_n^*(x) & \leq & D_n^*(x) & \leq & \overline{\mathrm{D}}_n^*(x) \\ \underline{\mathrm{D}}_{\min}^k & \leq & D_{\min}^k & \leq & \overline{\mathrm{D}}_{\min}^k \\ \underline{\mathrm{D}}_{\max}^k & \leq & D_{\max}^k & \leq & \overline{\mathrm{D}}_{\max}^k \\ \underline{\mathrm{V}}_{\min}^k & \leq & V_{\min}^k & \leq & \overline{\mathrm{V}}_{\min}^k \\ \underline{\mathrm{V}}_{\max}^k & \leq & V_{\max}^k & \leq & \overline{\mathrm{V}}_{\max}^k \end{array} \tag{65}$$

---

[2] Il est clair que si l'origine fait partie de la séquence, mais en a été retirée (comme l'indique la remarque 8.18) de manière à vérifier l'hypothèse sur l'absence de composantes nulles, alors la fraction $\frac{k}{n}$ doit être remplacée par $\frac{k+1}{n+1}$ dans toute la section.

On remarque que les différentes bornes sur les volumes optimaux $V^k_{\min}$ et $V^k_{\max}$ ne sont pas indépendantes et que les inégalités triviales

$$V^1_{\min} \leq \cdots \leq V^n_{\min} \qquad \text{et} \qquad V^0_{\max} \leq \cdots \leq V^{n-1}_{\max}$$

peuvent avoir des retombées très utiles. En effet, elles impliquent les affectations suivantes (à utiliser sans modération) qui induisent parfois de longues séquences d'améliorations :

$$\begin{array}{lcll}
\underline{\mathrm{V}}^k_{\min} & \leftarrow & \max\left\{\underline{\mathrm{V}}^k_{\min}, \underline{\mathrm{V}}^{k-1}_{\min}\right\} & \text{pour tout } k \in \{2, \dots, n\}, \\
\underline{\mathrm{V}}^k_{\max} & \leftarrow & \max\left\{\underline{\mathrm{V}}^k_{\max}, \underline{\mathrm{V}}^{k-1}_{\max}\right\} & \text{pour tout } k \in \{1, \dots, n-1\}, \\
\overline{\mathrm{V}}^k_{\min} & \leftarrow & \min\left\{\overline{\mathrm{V}}^k_{\min}, \overline{\mathrm{V}}^{k+1}_{\min}\right\} & \text{pour tout } k \in \{1, \dots, n-1\}, \\
\overline{\mathrm{V}}^k_{\max} & \leftarrow & \min\left\{\overline{\mathrm{V}}^k_{\max}, \overline{\mathrm{V}}^{k+1}_{\max}\right\} & \text{pour tout } k \in \{0, \dots, n-2\}.
\end{array} \tag{66}$$

Pour chacune des bornes $\underline{\mathrm{V}}^k_{\min}$, $\overline{\mathrm{V}}^k_{\min}$, $\underline{\mathrm{V}}^k_{\max}$ et $\overline{\mathrm{V}}^k_{\max}$, une valeur initiale peut être obtenue par au moins un des trois moyens suivants :

1° On détermine une borne supérieure pour $V^k_{\min}$ et une borne inférieure pour $V^k_{\max}$ en appliquant les heuristiques de la section 8.3.2.

2° Les volumes optimaux $V^1_{\min}$, $V^{n-1}_{\min}$ et $V^{n-1}_{\max}$ étant connus (voir section 8.2.1), les relations (66) impliquent que

- ▷ $V^1_{\min}$ est une borne inférieure pour $V^k_{\min}$ quel que soit $k \in \{2, \dots, n-2\}$,
- ▷ $V^{n-1}_{\min}$ est une borne supérieure pour $V^k_{\min}$ quel que soit $k \in \{2, \dots, n-2\}$,
- ▷ $V^{n-1}_{\max}$ est une borne supérieure pour $V^k_{\max}$ quel que soit $k \in \{0, \dots, n-2\}$.

3° Par le biais de l'observation 8.6, des termes constants apparaissent dans la fonction objectif de nos programmes linéaires en nombres entiers. Les coûts associés aux variables non fixées étant non négatifs, on obtient des bornes inférieures pour les volumes optimaux correspondants :

$$\prod_{j=1}^{s} x_j^{\sigma_j(k)} \text{ est une borne inférieure pour } V^k_{\min} \text{ quel que soit } k \in \{2, \dots, n-2\},$$

$$\prod_{j=1}^{s} x_j^{\sigma_j(k+1)} \text{ est une borne inférieure pour } V^k_{\max} \text{ quel que soit } k \in \{0, \dots, n-2\}.$$

Ces valeurs initiales seront progressivement améliorées au cours du processus décrit ci-dessous. Parallèlement, les bornes relatives à $D^k_{\min}$, $D^k_{\max}$ et $D^*_n(x)$ sont systématiquement mises à jour à l'aide des relations[3]

$$\begin{array}{lcl}
\underline{\mathrm{D}}^k_{\min} & = & \frac{k}{n} - \overline{\mathrm{V}}^k_{\min}, \\
\overline{\mathrm{D}}^k_{\min} & = & \frac{k}{n} - \underline{\mathrm{V}}^k_{\min}, \\
\underline{\mathrm{D}}^k_{\max} & = & \underline{\mathrm{V}}^k_{\max} - \frac{k}{n}, \\
\overline{\mathrm{D}}^k_{\max} & = & \overline{\mathrm{V}}^k_{\max} - \frac{k}{n}, \\
\underline{\mathrm{D}}^*_n(x) & = & \max\left\{\max\limits_{k\in\{1,\dots,n\}} \underline{\mathrm{D}}^k_{\min} \ , \max\limits_{k\in\{0,\dots,n-1\}} \underline{\mathrm{D}}^k_{\max}\right\}, \\
\overline{\mathrm{D}}^*_n(x) & = & \max\left\{\max\limits_{k\in\{1,\dots,n\}} \overline{\mathrm{D}}^k_{\min} \ , \max\limits_{k\in\{0,\dots,n-1\}} \overline{\mathrm{D}}^k_{\max}\right\}.
\end{array} \tag{67}$$

---

[3] À nouveau, si l'origine fait partie de la séquence, mais en a été retirée (comme l'indique la remarque 8.18) de manière à satisfaire l'hypothèse sur les composantes nulles, le terme $\frac{1}{n+1}$ doit être ajouté dans les maxima définissant $\underline{\mathrm{D}}^*_n(x)$ et $\overline{\mathrm{D}}^*_n(x)$.

On remarque que la borne inférieure pour la discrépance $\underline{\mathrm{D}}_n^*(x)$ ne dépend que des valeurs de

$$\bar{\mathrm{V}}_{\min}^1, \dots, \bar{\mathrm{V}}_{\min}^n \quad \text{et} \quad \underline{\mathrm{V}}_{\max}^0, \dots, \underline{\mathrm{V}}_{\max}^{n-1}.$$

Ces dernières étant généralement précises (car obtenues à l'aide des heuristiques de la section 8.3.2), la borne inférieure initiale $\underline{\mathrm{D}}_n^*(x)$ pour la discrépance s'avère le plus souvent de bonne qualité. Ce fait est illustré dans l'exemple suivant :

EXEMPLE 8.19 On calcule la borne inférieure initiale $\underline{\mathrm{D}}_n^*(x)$ définie ci-dessus pour une paire de $(0, m, s)$-réseaux de Faure en base b dont la discrépance est donnée dans les tables 5.3 et 5.4 :

1° Pour le $(0, 3, 4)$-réseau de Faure en base 5, on obtient

$$\underline{\mathrm{D}}_n^*(x) = 0.0423478$$

alors que $D_n^*(x) = 0.0893870$.

2° Pour le $(0, 2, 6)$-réseau de Faure en base 7, la borne inférieure initiale

$$\underline{\mathrm{D}}_n^*(x) = 0.2109716$$

est déjà égale à la discrépance cherchée.

L'élément le plus important de la dynamique que nous sommes en train de mettre en place est le suivant : si, au cours du déroulement du processus, la valeur de$\bar{\mathrm{D}}_{\min}^k$ (respectivement $\bar{\mathrm{D}}_{\max}^k$) devient plus petite que la borne inférieure $\underline{\mathrm{D}}_n^*(x)$ pour la discrépance, alors le maximum dans l'expression (64) n'est pas atteint pour $D_{\min}^k$ (respectivement $D_{\max}^k$) et l'étude du problème en question peut donc être avortée :

$$\begin{aligned} &\bar{\mathrm{D}}_{\min}^k < \underline{\mathrm{D}}_n^*(x) \quad \Longrightarrow \quad \text{les problèmes } D_{\min}^k \text{ et } V_{\min}^k \text{ peuvent être écartés} \\ &\bar{\mathrm{D}}_{\max}^k < \underline{\mathrm{D}}_n^*(x) \quad \Longrightarrow \quad \text{les problèmes } D_{\max}^k \text{ et } V_{\max}^k \text{ peuvent être écartés} \end{aligned} \tag{68}$$

Cette paire d'implications permet d'interrompre avant terme la résolution de la majeure partie de nos $2n$ problèmes. De plus, on remarque que le dispositif constitué des expressions (66), (67) et (68) est à même de déclencher une réaction en chaîne. En effet, l'amélioration d'une seule borne pour un volume $V_{\min}^k$ ou $V_{\max}^k$ peut induire une séquence de mises à jour conduisant à l'amélioration de plusieurs bornes dans (65) et à l'abandon de problèmes associés à des programmes linéaires en nombres entiers qui n'auront donc pas à être résolus exactement. En pratique, ce processus dynamique s'avère très efficace et, de manière similaire à une méthode d'exploration par séparation et évaluation, permet l'énumération implicite d'une partie importante des configurations concernées.

EXEMPLE 8.20 Pour les cas envisagés dans l'exemple 8.19, la phase d'initialisation du processus décrite ci-dessus aboutit à la situation suivante :

1° Pour le $(0, 3, 4)$-réseau de Faure en base 5, sur les 248 problèmes concernés, 12 sont déjà éliminés et on obtient l'intervalle initial

$$D_n^*(x) \in [0.0423478, 0.984].$$

2° Pour le $(0, 2, 6)$-réseau de Faure en base 7, sur les 96 problèmes concernés, 22 sont déjà éliminés et on obtient l'intervalle initial

$$D_n^*(x) \in [0.2109716, 0.9591837].$$

Ainsi, plusieurs programmes linéaires en nombres entiers peuvent être écartés d'emblée. En revanche, on observe que les bornes supérieures obtenues sont particulièrement mauvaises.

8.4.2 Résolution. Afin d'améliorer la qualité de cet intervalle initial pour la discrépance, il suffit d'établir des bornes inférieures et supérieures plus précises pour les volumes optimaux $V^k_{\min}$ et $V^k_{\max}$ des problèmes restants. On aborde cette question à l'aide des approches proposées dans la section 8.3. Cependant, d'un point de vue technique (essentiellement pour des questions d'espace-mémoire), il n'est pas envisageable de travailler simultanément sur autant de programmes linéaires en nombres entiers.

Il est donc plus raisonnable de ne traiter qu'un seul problème à la fois et d'essayer d'apporter des améliorations substantielles aux estimateurs qui lui sont associés. De plus, la borne inférieure pour la discrépance $\underline{\mathrm{D}}^*_n(x)$ étant généralement d'excellente qualité, il est préférable de concentrer nos efforts sur $\overline{\mathrm{D}}^*_n(x)$ (laissant les progrès au niveau de $\underline{\mathrm{D}}^*_n(x)$ apparaître de manière indirecte). Pour réduire cette valeur, il convient de choisir le problème pour lequel la borne supérieure $\overline{\mathrm{D}}^*_n(x)$ prend son maximum dans l'expression (67). On propose d'aborder la résolution du programme linéaire associé

- ▷ en résolvant la relaxation linéaire correspondante, renforcée à l'aide des éléments décrits à la page 119, la première fois qu'il est traité ;
- ▷ en appliquant une méthode d'énumération par séparation et évaluation à cette même formulation renforcée lors d'un éventuel second passage.

En fin de compte, chacun des $2n$ problèmes apparaissant dans l'expression (64) est considéré au maximum à deux reprises avant d'être définitivement résolu ou écarté. De plus, le processus peut être stoppé à tout instant, fournissant alors un intervalle pour la discrépance reflétant l'effort de calcul consenti jusque-là, ou poursuivi jusqu'à l'obtention de la valeur exacte de $D^*_n(x)$.

Lors du premier passage, la résolution de la relaxation linéaire renforcée fournit

- ▷ le volume optimal recherché si la solution obtenue est entière ;
- ▷ une nouvelle borne inférieure pour $V^k_{\min}$ dans le cas d'un problème du type $D^k_{\min}$;
- ▷ une nouvelle borne supérieure pour $V^k_{\max}$ dans le cas d'un problème du type $D^k_{\max}$.

Même lorsque la solution optimale de la relaxation linéaire n'est pas entière, il arrive fréquemment que la borne obtenue soit suffisamment bonne pour permettre l'élimination définitive du problème par le biais de l'expression (68).

EXEMPLE 8.21 Reprenons à nouveau les deux cas considérés dans l'exemple 8.19. Si, après la phase d'initialisation du processus, on résout la relaxation linéaire de tous les programmes renforcés restants, on aboutit à la situation suivante :

1° Pour le $(0, 3, 4)$-réseau de Faure en base 5, sur les 248 problèmes concernés, 65 sont éliminés et on obtient l'intervalle

$$D^*_n(x) \in [0.0423478, 0.2336710].$$

2° Pour le $(0, 2, 6)$-réseau de Faure en base 7, sur les 96 problèmes concernés, 72 sont éliminés et on obtient l'intervalle

$$D^*_n(x) \in [0.2109716, 0.3190308].$$

Ces résultats illustrent le fait qu'un intervalle non trivial pour la discrépance peut être calculé en un temps polynomial en la taille du problème.

Lorsqu'un problème requiert une seconde phase de traitement pour améliorer les bornes qui lui sont associées, on lui applique une méthode d'énumération par séparation et évaluation (voir section 8.3.3.3). Dans le cas d'un programme du type $\log(V^k_{\min})$, à un instant donné au cours du déroulement de l'algorithme en question, on note $\omega^k_{\min}$ la plus petite valeur d'un sous-problème actif généré par le processus de séparation de l'ensemble des solutions admissibles de $\log(V^k_{\min})$. Comme $\omega^k_{\min}$ est une borne inférieure pour $\log(V^k_{\min})$, les expressions (67) et (68) mènent au critère d'arrêt suivant :

$$(69)\quad e^{\omega^k_{\min}} > \frac{k}{n} - \mathrm{D}^*_n(x) \quad \Longrightarrow \quad \text{la résolution des problèmes } D^k_{\min} \text{ et } V^k_{\min} \text{ peut être abandonnée.}$$

De manière similaire, pour un programme du type $\log(V^k_{\max})$, on note $\omega^k_{\max}$ la plus grande valeur d'un sous-problème actif généré par le processus de séparation de l'ensemble des solutions admissibles de $\log(V^k_{\max})$ à un instant donné au cours du déroulement de l'algorithme. Comme $\omega^k_{\max}$ est une borne supérieure pour $\log(V^k_{\max})$, les expressions (67) et (68) mènent au critère d'arrêt suivant :

$$(70)\quad e^{\omega^k_{\max}} < \frac{k}{n} + \mathrm{D}^*_n(x) \quad \Longrightarrow \quad \text{la résolution des problèmes } D^k_{\max} \text{ et } V^k_{\max} \text{ peut être abandonnée.}$$

En pratique, cette paire d'implications permet d'interrompre la plupart des énumérations par séparation et évaluation entreprises avant d'avoir atteint l'optimum (la mise en œuvre de ces critères d'arrêt est réalisable à l'aide de la librairie CPLEX [CPL01]). Ce fait est illustré ci-dessous :

EXEMPLE 8.22 En appliquant la méthode décrite dans cette section pour les deux cas considérés dans l'exemple 8.19, on obtient finalement les résultats suivants :

1° Pour le $(0,3,4)$-réseau de Faure en base 5 :

Temps de calcul de la discrépance $D^*_n(x) = 0.0893870$: 364 secondes
Nombre total de problèmes : 248
Relaxations linéaires renforcées résolues : 217
Énumérations par séparation et évaluation avortées : 67
Énumérations par séparation et évaluation menées à terme : 10

L'intervalle pour lequel le supremum dans l'expression (3) est atteint a été obtenu en résolvant le problème $\log(V^{73}_{\min})$ à l'aide d'une énumération par séparation et évaluation.

2° Pour le $(0,2,6)$-réseau de Faure en base 7 :

Temps de calcul de la discrépance $D^*_n(x) = 0.2109716$: 36 secondes
Nombre total de problèmes : 96
Relaxations linéaires renforcées résolues : 71
Énumérations par séparation et évaluation avortées : 22
Énumérations par séparation et évaluation menées à terme : 0

L'intervalle pour lequel le supremum dans l'expression (3) est atteint a été obtenu en résolvant la relaxation linéaire du problème $\log(V^{10}_{\max})$ (la solution optimale est entière).

Rappelons que la discrépance des deux réseaux en question a été déterminée à l'aide de la méthode directe basée sur la discrétisation de Niederreiter (21). Avec cette approche, les temps de calcul obtenus étaient de 8 heures pour le $(0,3,4)$-réseau en base 5 et d'une semaine pour le $(0,2,6)$-réseau en base 7.

8.5 Expériences numériques

Les résultats de quelques expériences effectuées à l'aide de la méthode proposée dans ce chapitre sont présentés ci-dessous. On commence par calculer la discrépance de $(0,m,s)$-réseaux de Faure en base b dans des cas inaccessibles jusqu'alors. On procède ensuite à une analyse sommaire du temps de calcul nécessaire à l'application de cette technique. Pour terminer, on améliore les résultats des sections 7.5.2 (sur la comparaison de différentes séquences) et 7.5.3 (sur les ensembles de points minimaux).

8.5.1 Calcul de la discrépance. L'approche par programmation linéaire en nombres entiers présentée ci-dessus a permis de calculer la discrépance de quelques $(0, m, s)$-réseaux de Faure en base b (obtenus à l'aide du générateur `GrayFaure` de la section 4.8.4). Ces résultats, donnés dans la table 8.1, complètent ceux de la page 52 et améliorent certains intervalles établis à l'aide de la méthode du chapitre 7 (voir page 91). Dans chacun des cas considérés, le temps de calcul nécessaire ne dépasse pas quelques jours sur une station de travail SGI R10000.

Soulignons le fait que l'utilisation de la méthode de la section 5.1 est totalement irréaliste pour de telles séquences. En effet, selon nos estimations, à l'aide de notre mise en œuvre de la discrétisation de Niederreiter (21), le calcul de la discrépance du $(0, 3, 7)$-réseau en base 7 prendrait environ 5 millions d'années, alors que pour le $(0, 2, 12)$-réseau en base 13, il faudrait au moins $2 \cdot 10^{13}$ siècles.

s	4	5	5	6	7	7	8	9	10	11	12
b	5	5	5	7	7	7	11	11	11	11	13
m	4	3	4	3	2	3	2	2	2	2	2
$n = b^m$	625	125	625	343	49	343	121	121	121	121	169
$D_n^*(x)$	0.01772458	0.1417881	0.02666228	0.08988426	0.2690111	0.1298317	0.1701839	0.2121262	0.2574323	0.3010480	0.2718370
Temps de calcul	2.3 jours	650 secondes	15.6 jours	2.65 jours	13 secondes	6.7 jours	7.4 heures	6.7 heures	3.7 heures	1.9 heures	3.85 jours

TAB. 8.1. Discrépance de quelques $(0, m, s)$-réseaux de Faure en base b.

D'autre part, on note que la borne inférieure initiale $\underline{D}_n^*(x)$ définie dans la section 8.4.1 est égale à la discrépance dans 9 des 11 cas considérés dans la table ci-dessus. Cette observation confirme le fait qu'il est possible d'établir d'excellentes bornes inférieures pour la discrépance en quelques secondes de calcul (indépendamment du fait que le processus d'amélioration de la borne supérieure correspondante prenne plusieurs jours).

8.5.2 Temps de calcul. La figure 8.11 montre l'évolution typique de l'intervalle

$$\left[\underline{D}_n^*(x), \overline{D}_n^*(x)\right]$$

pour la discrépance $D_n^*(x)$ au cours du déroulement de l'algorithme dans un des deux cas présentés dans la table 8.1 où la borne inférieure initiale $\underline{D}_n^*(x)$ n'est pas optimale. On remarque que les progrès sont particulièrement irréguliers. Ils s'avèrent tout d'abord très rapides, lorsqu'il suffit de résoudre une relaxation linéaire pour améliorer l'intervalle, avant de progressivement ralentir, quand des bornes de plus en plus précises doivent être établies par le biais d'énumérations par séparation et évaluation.

Une étude sérieuse de la complexité empirique de notre méthode semble difficile, en partie parce que le temps de calcul dépend non seulement de la dimension s et du nombre de points n, mais également de la discrépance elle-même. Ce phénomène est apparent dans les résultats de la table 8.1 : pour les trois séquences de 121 points considérées, l'effort à fournir décroît avec la dimension, ce qui paraît plutôt contre-intuitif. Ce comportement s'explique par le fait que le mécanisme lié aux critères d'arrêt (69) et (70) entre en action plus rapidement lorsque la borne inférieure $\underline{D}_n^*(x)$ est élevée. Notons cependant que

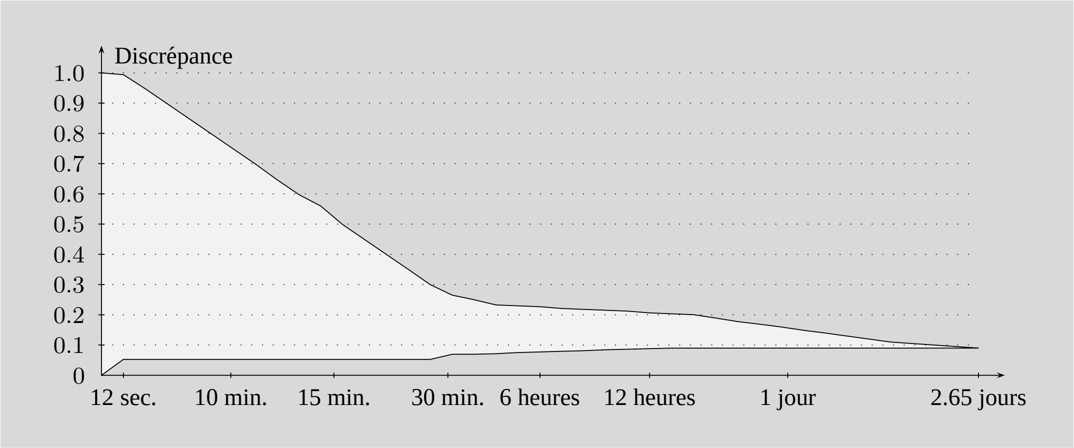


FIG. 8.11. Convergence de l'intervalle pour la discrépance d'un $(0, 3, 6)$-réseau de Faure en base 7 durant les 2.65 jours nécessaires au déroulement de l'algorithme.

cet effet semble devenir marginal pour des séquences présentant une discrépance suffisamment faible (inférieure à 0.15 environ).

On poursuit néanmoins cette étude par une paire d'expériences sur l'évolution du temps de calcul en fonction de la dimension s (voir figure 8.12) et du nombre de points n (voir figure 8.13). Dans chacun des cas ci-dessous, la séquence considérée est le segment initial d'une suite de Faure permutée à l'aide d'un code de Gray (voir section 4.8.4).

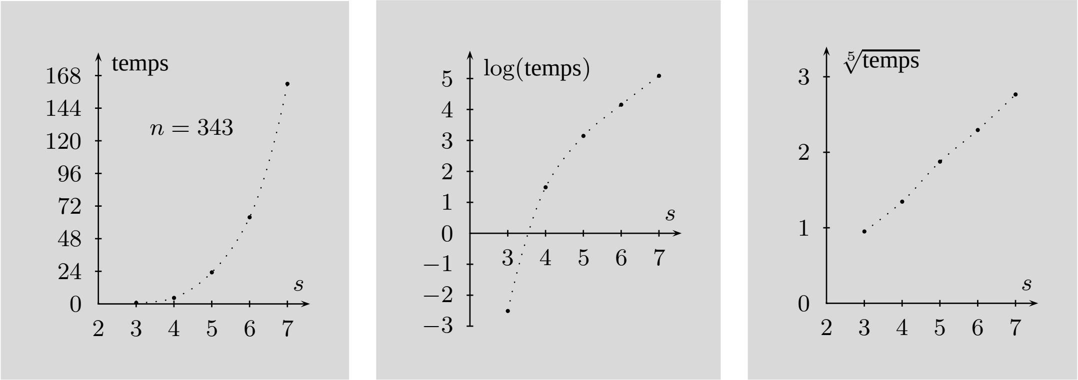


FIG. 8.12. À gauche, temps (en heures) nécessaire au calcul de la discrépance d'une séquence (le segment initial d'une suite de Faure permutée à l'aide d'un code de Gray) de 343 points en dimension s, pour $s \in \{3, \ldots, 7\}$. Au centre, le logarithme de cette durée et à droite sa racine cinquième.

Dans la figure 8.12, on a représenté le temps nécessaire au calcul de la discrépance d'une séquence de 343 points en dimension s, pour $s \in \{3, \ldots, 7\}$. En considérant le logarithme de ces durées, la courbe obtenue semble concave et, en prenant sa racine cinquième, elle paraît quasiment linéaire. Ces observations suggèrent que le temps de calcul ne croît pas de manière exponentielle avec la dimension, mais plutôt comme s^5. Toutefois, le comportement étant inconnu pour d'autres valeurs de s et de n, il serait tout à fait hasardeux de tirer une conclusion si générale des résultats de cette expérience.

Dans la figure 8.13, on a représenté le temps nécessaire au calcul de la discrépance d'une séquence de n points en dimension 7, pour $n \in \{49, \ldots, 343\}$. Le même type de réserves devant à nouveau être formulées, les résultats obtenus doivent être interprétés comme de simples tendances. Néanmoins, dans ce cas particulier, il semble que le temps de calcul ne croît pas de manière exponentielle avec la taille de la séquence, ni même plus vite que n^3 (d'autres expériences suggèrent que le degré de ce présumé polynôme en n n'est pas indépendant de la dimension).

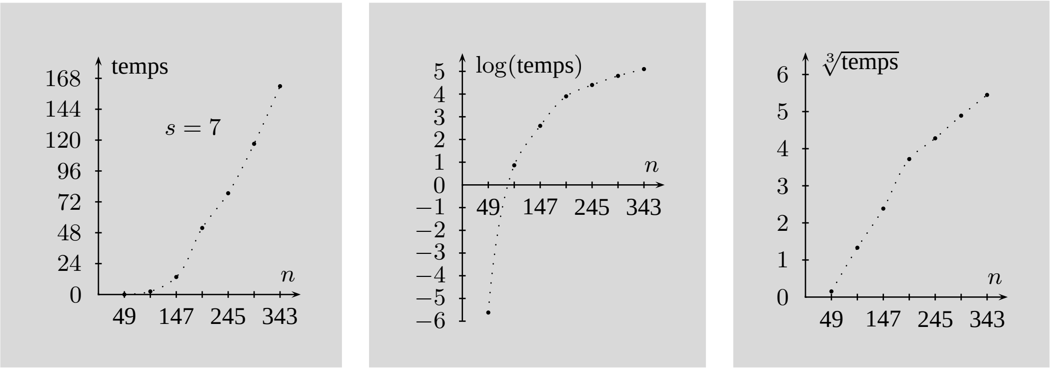


FIG. 8.13. À gauche, temps (en heures) nécessaire au calcul de la discrépance d'une séquence (le segment initial d'une suite de Faure permutée à l'aide d'un code de Gray) de n points en dimension 7, pour $n \in \{49, \ldots, 343\}$. Au centre, le logarithme de cette durée et à droite sa racine cubique.

8.5.3 Comparaison de séquences. La méthode proposée dans ce chapitre a permis d'améliorer les résultats de la section 7.5.2. Il a en effet été possible de calculer exactement les discrépances des segments initiaux des suites de Halton, Sobol et Faure en dimension 7 pour lesquelles des intervalles sont donnés dans la table 7.11. Ces valeurs sont représentées dans la figure 8.14. Notons que de tels résultats sur la discrépance effective de séquences en dimension non triviale n'ont jamais été obtenus auparavant (en effet, 30 points est la taille maximale d'une séquence en dimension 7 pour laquelle nous pouvons calculer la discrépance à l'aide de la discrétisation de Niederreiter (21) en moins d'une semaine).

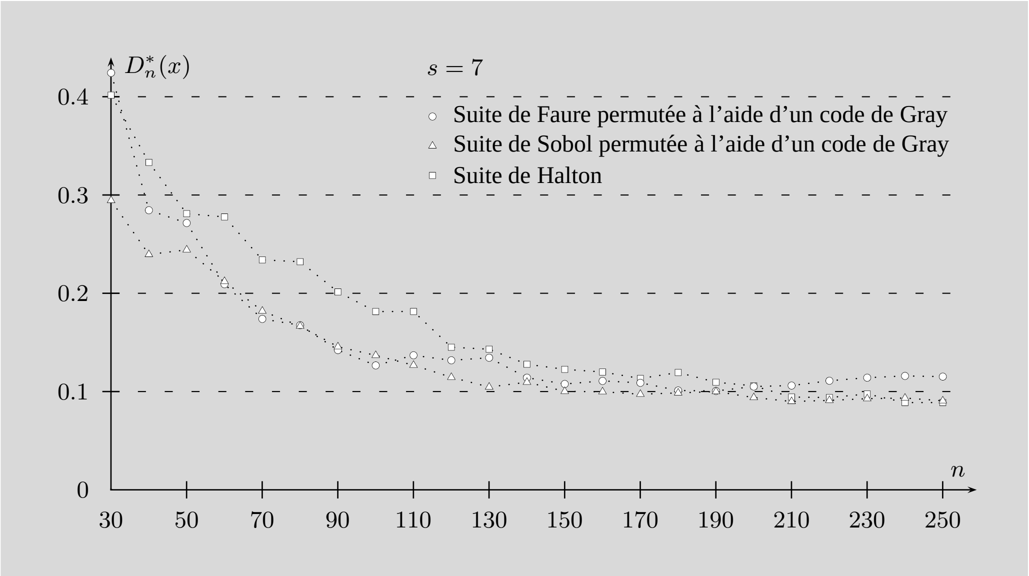


FIG. 8.14. Discrépance des segments initiaux de trois suites classiques.

Quelques remarques peuvent être formulées en complément des conclusions de la section 7.5.2. En premier lieu, on observe que la séquence de valeurs obtenue n'est pas strictement décroissante en fonction de n. Considérant la nature discrète de la discrépance, il n'est guère surprenant de constater l'apparition occasionnelle de tels soubresauts. Deuxièmement, bien que les différences soient vraiment minimes, il semble que les séquences présentant la décroissance la plus régulière et la discrépance la plus faible pour la plupart des tailles $n \in \{30, \ldots, 250\}$ soient celles issues de la suite de Sobol.

8.5.4 Sur les ensembles minimaux. Pour terminer, on revient sur l'expérience considérée dans la section 7.5.3 à propos du théorème 1.23 de Heinrich, Novak, Wasilkowski et Woźniakowski [HNWW][4]. Rappelons qu'il s'agit d'étudier en fonction de la dimension la croissance de la taille minimale $n_{\min}$ du segment initial d'une suite de Faure permutée à l'aide d'un code de Gray présentant une discrépance inférieure ou égale à $d = 0.45$. Alors que dans la section 7.5.3, des intervalles approximatifs contenant la vraie valeur ont été obtenus pour $s \in \{4, \ldots, 12\}$ (voir figure 7.12), la méthode proposée dans ce chapitre a permis de calculer exactement ces nombres pour $s \in \{4, \ldots, 19\}$ (voir figure 8.15).

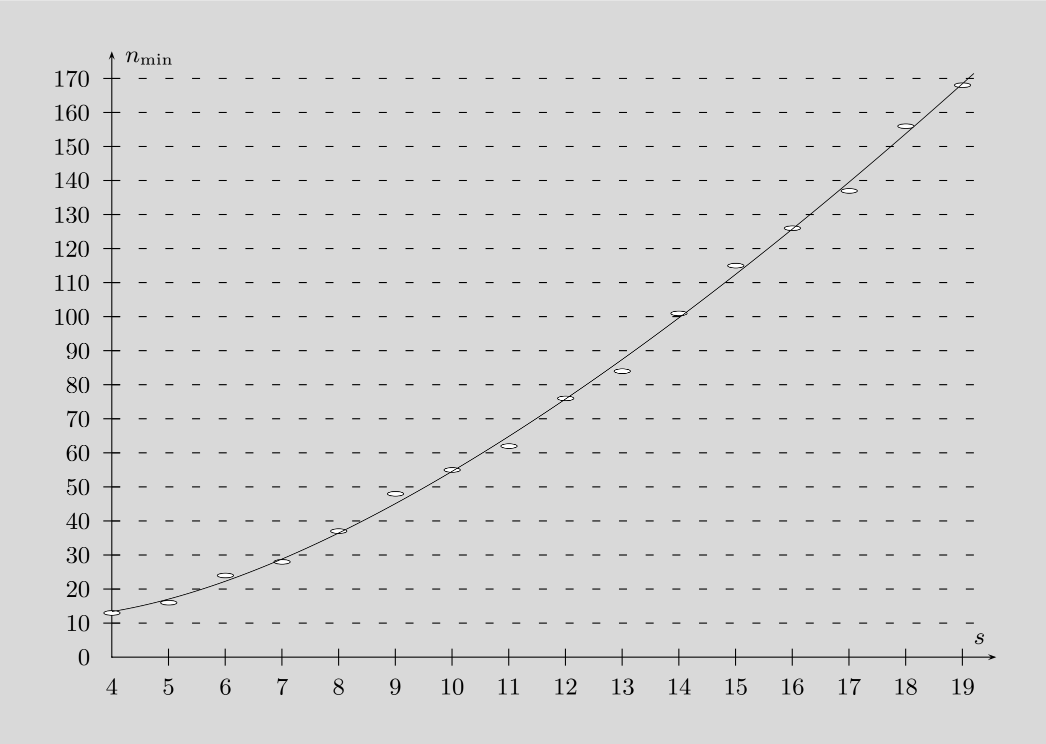


FIG. 8.15. Taille minimale $n_{\min}$ du segment initial d'une suite de Faure permutée à l'aide d'un code de Gray présentant une discrépance inférieure ou égale à 0.45 pour $s \in \{4, \ldots, 19\}$ et représentation de $f(s) = 7.77411\, s \log s - 15.7798 s + 33.4046$.

L'article de Heinrich, Novak, Wasilkowski et Woźniakowski [HNWW] contient également quelques résultats théoriques (fondés sur une analyse probabiliste) permettant de majorer $n(s, d)$. Par exemple, après quelques calculs, on obtient $n(5, 0.45) \leq 656$ et $n(10, 0.45) \leq 1\,540$. On note cependant que les valeurs présentées dans la figure 8.15 fournissent les bornes supérieures plus précises $n(5, 0.45) \leq 16$ et $n(10, 0.45) \leq 55$.

---

[4] Ce dernier stipule que pour toute valeur $d \in (0, 1/2)$, la taille $n(s, d)$ de la plus courte séquence dans $\bar{I}^s$ présentant une discrépance inférieure ou égale à d croît linéairement avec la dimension s.

Cette expérience indique que la taille des échantillons minimaux considérés ne semble pas croître linéairement avec la dimension, mais légèrement plus rapidement. Cependant, le comportement étant inconnu en dimension plus élevée ou pour d'autres valeurs de d, la prudence est à nouveau de mise pour l'analyse des résultats. Par ailleurs, on remarque que la fonction

$$f(s) = 7.77411\, s \log s - 15.7798 s + 33.4046,$$

qui minimise l'erreur carrée moyenne d'approximation pour $f(s)$ du type $a_2 s \log s + a_1 s + a_0$, épouse remarquablement bien les valeurs de la figure 8.15. Ainsi, on conjecture que la taille minimale $n_{\min}$ du segment initial d'une suite de Faure (permutée à l'aide d'un code de Gray) en dimension s présentant une discrépance inférieure ou égale à d est de l'ordre $\Theta(s \log s)$.

Les suites à discrépance faible sont des séquences infinies telles que toute sous-séquence de points consécutifs possède de bonnes propriétés d'équirépartition. En revanche, on remarque que les ensembles minimaux du théorème 1.23 ne sont pas soumis à une telle contrainte de pouvoir être indéfiniment complétés. Il pourrait s'agir d'un argument indiquant pourquoi les échantillons optimaux correspondants extraits de suites à discrépance faible ne présentent pas le comportement linéaire désiré en la dimension. Cette observation suggère également que la structure des ensembles minimaux du théorème 1.23 pourrait être totalement différente de celle des suites que nous connaissons.

Conclusions et perspectives

Une particularité de ce travail est d'aborder une question de théorie des nombres, l'évaluation de la discrépance, comme un problème d'optimisation combinatoire. En effet, les approches proposées dans cette thèse reposent essentiellement sur des principes généralement utilisés en géométrie algorithmique et en recherche opérationnelle.

Commençons par un bref récapitulatif des caractéristiques principales de ces méthodes en prenant soin de mettre en évidence quelques points de comparaison.

▷ Dans la section 7.1, un principe général (basé sur la considération d'une partition finie $\mathcal{P}$ du cube unité I^s en intervalles) permettant de calculer un intervalle (de largeur inférieure ou égale à son poids $W(\mathcal{P})$) pour la discrépance d'une séquence quelconque x de n points dans$\bar{I}^s$ est présenté. Deux techniques particulières de décomposition sont envisagées :

1° Dans la section 7.2, le cas des partitions $\mathcal{P}_{z_{s,k}}$ de I^s ayant la forme de grilles extensibles de cardinalité k^s est considéré. La méthode se déroule en deux temps : il s'agit tout d'abord de minimiser le poids $W(\mathcal{P}_{z_{s,k}})$, puis de calculer efficacement les bornes correspondantes pour la discrépance de x.

2° L'approche proposée dans la section 7.4 permet de construire des intervalles de précision arbitraire. En effet, après avoir choisi un paramètre $\varepsilon \in (0, 1)$ quelconque, il est possible de générer une partition $\mathcal{P}_\varepsilon^s$ finie telle que $W(\mathcal{P}_\varepsilon^s) = \varepsilon$. Bien qu'a priori complexe, la structure de cette décomposition est exploitable lors du calcul des bornes correspondantes pour la discrépance de x.

L'avantage de la première méthode est que la considération de grilles $\mathcal{P}_{z_{s,k}}$ permet de ne compter explicitement qu'une seule fois chacun des n points, même s'ils appartiennent à une multitude d'intervalles à explorer. L'heureuse présence du logarithme dans la complexité $O(n \log(|\mathcal{P}_{z_{s,k}}|) + 2^s |\mathcal{P}_{z_{s,k}}|)$ qui en découle explique pourquoi cette technique permet d'établir des bornes pour la discrépance de séquences de très grande taille. Cependant, pour un même effort de calcul, la précision des majorations obtenues est bien moins bonne que celle qu'il est possible d'atteindre à l'aide de la seconde approche, lorsqu'elle s'applique.

Cette propriété découle du fait que la cardinalité d'une grille $\mathcal{P}_{z_{s,k}}$ est nettement supérieure à celle de la partition $\mathcal{P}_\varepsilon^s$ de même poids. En revanche, comme l'illustre la complexité $O((\log n)^s |\mathcal{P}_\varepsilon^s|)$ de l'algorithme correspondant, les différents points de la séquence doivent être comptés à plusieurs reprises avec la seconde méthode de décomposition. Bien que ce problème soit fortement atténué par l'utilisation d'une technique de comptage spécifique basée sur les arbres d'intervalles, il n'a pas été possible de séparer totalement les difficultés causées d'une part par la taille de la séquence et d'autre part par la cardinalité de la partition. Par ailleurs, nos expériences suggérant que $|\mathcal{P}_\varepsilon^s| = O((1/\varepsilon)^s)$, cette approche conduit également à un algorithme exponentiel en la dimension.

▷ La méthode proposée dans le chapitre 8 est totalement différente. Elle consiste en l'amélioration progressive (en au plus $4n$ itérations) d'un intervalle initial pour la discrépance. En effet, le calcul

> de $D_n^*(x)$ est décomposé en $2n$ sous-problèmes de géométrie combinatoire que l'on traite en deux phases à l'aide de techniques de programmation linéaire en nombres entiers. Un effort particulier a été fourni pour qu'une partie importante des configurations concernées soient énumérées de manière implicite. Cette approche conduit à la valeur exacte de la discrépance ou, éventuellement, à un intervalle pour cette grandeur lorsque le processus est interrompu avant terme.
>
> Les sous-problèmes les plus difficiles étant résolus à l'aide d'une méthode d'énumération par séparation et évaluation, la complexité de cet algorithme est, jusqu'à preuve du contraire, exponentielle en n et s. A priori, cette propriété ne semble pas se confirmer expérimentalement.

La table 7.3 illustre le fait que la méthode basée sur les grilles extensibles permet de calculer des bornes pour la discrépance de séquences de très grande taille, mais montre également que la précision qu'il est possible d'atteindre est assez limitée. Les résultats de la table 7.8 révèlent que les intervalles obtenus sont bien meilleurs à l'aide de l'approche basée sur les partitions de la forme $\mathcal{P}^s_\varepsilon$, mais il s'avère malheureusement difficile de maintenir une telle qualité pour des tailles supérieures à quelques milliers de points (pour $s \leq 20$). Finalement, la table 8.1 et la figure 8.15 indiquent que la technique du chapitre 8 permet de calculer la discrépance de séquences de quelques centaines de points en dimension $s \leq 20$.

Bien évidemment, ces performances peuvent paraître ridicules s'il s'agit d'utiliser nos algorithmes pour comparer la qualité de la distribution d'échantillons destinés à une application de la méthode de quasi-Monte-Carlo. Cependant, d'un autre point de vue, notre travail constitue un premier pas prometteur sur une question négligée, car jugée inabordable, par de nombreux experts. En effet, en examinant nos résultats à la lumière de ce qu'il était préalablement possible d'espérer, les progrès semblent nettement plus substantiels. Par exemple, en dimension $s = 10$ et en une journée de calcul (sur une station de travail SGI R10000), nos expériences numériques montrent que l'on peut calculer

- ▷ la discrépance d'une séquence de 10 points à l'aide de la discrétisation de Niederreiter (21) ;
- ▷ la discrépance d'une séquence de plus de 200 points en utilisant la méthode du chapitre 8 ;
- ▷ un intervalle de bonne qualité (à $\pm 10\%$) pour la discrépance d'une séquence de $1\,000$ points à partir de l'approche basée sur les partitions de la forme $\mathcal{P}^s_\varepsilon$;

- ▷ au niveau des résultats théoriques, le théorème 6.1 fournit instantanément des bornes pour la discrépance de tout $(t, m, 10)$-réseau en base b, mais cette majoration est supérieure à 1 pour des séquences de taille inférieure à 8.5 milliards et 200 millions de points respectivement pour les suites de Sobol et de Faure.

Ces observations illustrent le fait que les techniques algorithmiques présentées dans ce travail ouvrent une nouvelle voie de recherche pour le calcul et la majoration de la discrépance. Notons cependant que les méthodes proposées n'ont pas encore trouvé leur forme définitive et que les améliorations envisageables semblent nombreuses. Par exemple, une spécialisation de ces approches au cas où la séquence considérée est un (t, m, s)-réseau en base b (ou plus simplement où les coordonnées des points sont des rationnels possédant un plus petit dénominateur commun de taille raisonnable) pourrait être intéressante. D'autre part, il est vraisemblable que la formulation des programmes linéaires en nombres entiers du chapitre 8 puisse être améliorée et qu'il soit possible de mettre au point de nouvelles techniques de résolution spécifiques nettement plus efficaces.

Bibliographie

[Adl87] V. G. Adlakha, *A Monte Carlo technique with quasirandom points for the stochastic shortest path problem*, American Journal of Mathematical and Management Sciences **7** (1987), no. 4, 325–358.

[Adl92] V. G. Adlakha, *An empirical evaluation of antithetic variates and quasirandom points for simulating stochastic networks*, Simulation : Journal of the Society for Computer Simulation International **58** (1992), no. 1, 23–31.

[AIKS91] A. Aggarwal, H. Imai, N. Katoh, and S. Suri, *Finding k points with minimum diameter and related problems*, J. Algorithms **12** (1991), 38–56.

[AK89] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, Wiley, 1989.

[AS79] I. A. Antonov and V. M. Saleev, *An economic method of computing LP_τ-sequences*, USSR Comput. Math. Math. Phys. **19** (1979), no. 1, 252–256.

[Bak99] R. C. Baker, *On irregularities of distribution II*, J. London Math. Soc. **59** (1999), 50–64.

[BC87] J. Beck and W. W. L. Chen, *Irregularities of Distribution*, Cambridge University Press, 1987.

[Béj82] R. Béjian, *Minoration de la discrépance d'une suite quelconque sur T*, Acta Arith. **41** (1982), 185–202.

[Ben79] J. L. Bentley, *Decomposable searching problems*, Information Processing Letters **8** (1979), no. 5, 244–251.

[BF77] R. Béjian et H. Faure, *Discrépance de la suite de van der Corput*, C. R. Acad. Sci. Paris Sér A **285** (1977), 313–316.

[BF88] P. Bratley and B. L. Fox, *Algorithm 659 : Implementing Sobol's quasi-random sequence generator*, ACM Trans. Math. Software **14** (1988), no. 1, 88–100.

[BFN92] P. Bratley, B. L. Fox, and H. Niederreiter, *Implementation and tests of low-discrepancy sequences*, ACM Trans. Modeling Comput. Simulation **2** (1992), no. 3, 195–213.

[BFS83] P. Bratley, B. L. Fox, and L. E. Schrage, *A Guide to Simulation*, Springer-Verlag, 1983.

[BL94] N. Bouleau and D. Lépingle, *Numerical Methods for Stochastic Processes*, Wiley, 1994.

[BW79] E. Braaten and G. Weller, *An improved low-discrepancy sequence for multidimensional quasi-Monte Carlo integration*, J. Comput. Phys. **33** (1979), 249–258.

[BZ93] P. Bundschuh and Y. Zhu, *A method for exact calculation of the discrepancy of low-dimensional finite point sets I*, Abh. Math. Sem. Univ. Hamburg **63** (1993), 115–133.

[CCPS97] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*, Wiley, 1997.

[CF93] H. Chaix et H. Faure, *Discrépance et diaphonie en dimension un*, Acta Arith. **63** (1993), no. 2, 103–141.

[Che80] W. W. L. Chen, *On irregularities of distribution*, Mathematika **27** (1980), 153–170.

[Cle81] L. De Clerck, *A proof of Niederreiter's conjecture concerning error bounds for quasi-Monte Carlo integration*, Adv. in Appl. Math. **2** (1981), 1–6.

[Cle86] L. De Clerck, *A method for exact calculation of the star discrepancy of plane sets applied to the sequences of Hammersley*, Monatsh. Math. **101** (1986), 261–278.

[CLM$^+$99] A. T. Clayman, K. M. Lawrence, G. L. Mullen, H. Niederreiter, and N. J. A. Sloane, *Updated tables of parameters of (t, m, s)-nets*, J. Combinatorial Designs **7** (1999), no. 5, 381–393.

[CM67] R. R. Coveyou and R. D. MacPherson, *Fourier analysis of uniform random number generators*, Journal of the ACM **14** (1967), 100–119.

[CMO97] R. E. Caflisch, W. J. Morokoff, and A. B. Owen, *Valuation of mortgage backed securities using brownian bridges to reduce effective dimension*, J. Comput. Finance **1** (1997), 27–46.

[CPL01] CPLEX, *Using the CPLEX Callable Library*, CPLEX Optimization Inc., version 6.0.1.

[CS] W. W. L. Chen and M. M. Skriganov, *Explicit constructions in the classical mean squares problem in irregularities of point distribution*, submitted.

[DE93] D. P. Dobkin and D. Eppstein, *Computing the discrepancy*, in Proceedings of the Ninth Annual Symposium on Computational Geometry, 1993, pp. 47–52.

[DEM96] D. P. Dobkin, D. Eppstein, and D. P. Mitchell, *Computing the discrepancy with applications to supersampling patterns*, ACM Transactions on Graphics **15** (1996), no. 4, 354–376.

[Do91] K.-A. Do, *Quasi-random resampling for the bootstrap*, in Computer Science and Statistics : Proceedings of the Twenty-third Symposium on the Interface (E. M. Keramidas, ed.), 1991, pp. 297–300.

[DR84] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, second ed., Academic Press, 1984.

[DT97] M. Drmota and R. F. Tichy, *Sequences, Discrepancies and Applications, Lecture Notes in Mathematics 1651*, Springer, 1997.

[Fau80] H. Faure, *Suites à faible discrépance dans* T^s, Publ. Dép. Math., Université de Limoges, Limoges, France (1980).

[Fau81] H. Faure, *Discrépance de suites associées à un système de numération (en dimension un)*, Bull. Soc. Math. France **109** (1981), 143–182.

[Fau82] H. Faure, *Discrépance de suites associées à un système de numération (en dimension* s*)*, Acta Arith. **41** (1982), 337–351.

[Fau86] H. Faure, *On the star-discrepancy of generalized Hammersley sequences in two dimensions*, Monatsh. Math. **101** (1986), 291–300.

[Fau93] H. Faure, *Suggestions for quasi-Monte-Carlo users*, in Proc. Internat. Conf. on Finite Fields, Coding Theory, and Advances in Communications and Computing, Las Vegas, August 1991, Lecture Notes in Pure and Applied Math., Vol. 141, Marcel Decker, New York, 1993, pp. 269–278.

[Fau94] H. Faure, *Méthodes quasi-Monte-Carlo multidimensionnelles*, Theoretical Computer Science **123** (1994), 131–137.

[FC96] H. Faure et H. Chaix, *Minoration de discrépance en dimension deux*, Acta Arith. **76** (1996), 149–164.

[Fis85] G. S. Fishman, *Estimating network characteristics in stochastic activity networks*, Management Science **31** (1985), no. 5, 579–593.

[Fis96] G. S. Fishman, *Monte Carlo : Concepts, Algorithms, and Applications*, Springer, 1996.

[Fox86] B. L. Fox, *Algorithm 647 : Implementation and relative efficiency of quasirandom sequence generators*, ACM Trans. Math. Software **12** (1986), no. 4, 362–376.

[Fox99] B. L. Fox, *Strategies for Quasi-Monte Carlo*, Kluwer, 1999.

[Gra53] F. Gray, *Pulse code communication*, U.S. Patent 2632058, 1953.

[Hal60] J. H. Halton, *On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals*, Numer. Math. **2** (1960), 84–90.

[Hal72] J. H. Halton, *Estimating the accuracy of quasi-Monte Carlo integration*, in Applications of Number Theory to Numerical Analysis (S. K. Zaremba, ed.), Academic Press, 1972, pp. 345–360.

[Hal86] M. Hall, *Combinatorial Theory*, Wiley, 1986.

[Ham60] J. M. Hammersley, *Monte Carlo methods for solving multivariate problems*, Annals of the New York Academy of Sciences **86** (1960), 844–874.

[Hei96] S. Heinrich, *Efficient algorithms for computing the* L_2*-discrepancy*, Math. Comput. **65** (1996), 1621–1633.

[Hic98] F. J. Hickernell, *A generalized discrepancy and quadrature error bound*, Math. Comput. **67** (1998), 299–322.

[Hla61] E. Hlawka, *Funktionen von beschränkter Variation in der Theorie der Gleichverteilung*, Ann. Mat. Pura. Appl. **54** (1961), 325–333.

[HNWW] S. Heinrich, E. Novak, G. W. Wasilkowski, and H. Woźniakowski, *The star discrepancy depends linearly on the dimension*, submitted to Acta Arith.

[HW81] L. K. Hua and Y. Wang, *Applications of Number Theory to Numerical Analysis*, Springer, 1981.

[JBT96] C. Joy, P. P. Boyle, and K. S. Tan, *Quasi-Monte Carlo methods in numerical finance*, Management Science **42** (1996), no. 6, 926–938.

[JHK97] F. James, J. Hoogland, and R. Kleiss, *Multidimensional sampling for simulation and integration : measures, discrepancies, and quasi-random numbers*, Computer Physics Communications **99** (1997), 180–220.

[Kie61] J. Kiefer, *On large deviations of the empiric d.f. of vector chance variables and a law of the iterated logarithm*, Pacific J. Math. **11** (1961), 649–660.

[Kle77] V. Klee, *Can the measure of $\cup[a_i, b_i]$ be computed in less than $O(n \log n)$ steps ?*, Amer. Math. Monthly **84** (1977), 284–285.

[KN74] L. Kuipers and H. Niederreiter, *Uniform Distribution of Sequences*, John Wiley, New York, 1974.

[Knu69] D. E. Knuth, *The Art of Computer Progrmming : Seminumerical Algorithms (third ed. 1998)*, vol. 2, Addison-Wesley, 1969.

[KW97] L. Kocis and W. J. Whiten, *Computational investigations of low-discrepancy sequences*, ACM Transactions on Mathematical Software **23** (1997), no. 2, 266–294.

[LC97] P. L'Ecuyer and R. Couture, *An implementation of the lattice and spectral tests for multiple recursive linear random number generators*, INFORMS Journal on Computing **9** (1997), no. 2, 206–217.

[LC98] C. Lécot and I. Coulibaly, *A quasi-Monte Carlo scheme using nets for a linear Boltzmann equation*, SIAM Journal of Numerical Analysis **35** (1998), no. 1, 51–70.

[L'E94] P. L'Ecuyer, *Uniform random number generation*, Annals of Operations Research **53** (1994), 77–120.

[L'E98] P. L'Ecuyer, *Random number generation*, in Handbook on Simulation, Chapter 4 (J. Banks, ed.), Wiley, 1998, pp. 93–137.

[L'E99] P. L'Ecuyer, *Good parameters and implementations for combined multiple recursive random number generators*, Operations Research **47** (1999), no. 1, 159–164.

[Leh51] D. H. Lehmer, *Mathematical methods in large scale computing units*, Annals Comp. Laboratory Harvard University **26** (1951), 141–146.

[LH98] P. L'Ecuyer and P. Hellekalek, *Random number generators : selection criteria and testing*, in Random and Quasi-Random Point Sets, Lecture Notes in Stat. 138 (P. Hellekalek and G. Larcher, eds.), Springer, 1998, pp. 223–265.

[Lin65] S. Lin, *Computer solutions of the traveling salesman problem*, Bell System Technical Journal **44** (1965), 2245–2269.

[LN86] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications*, Cambridge University Press, Cambridge, UK, 1986.

[Luc78] G. S. Lucker, *A data structure for orthogonal range queries*, in Proc. 19th IEEE Sympos. Found. Comput. Sci., 1978, pp. 28–34.

[Mat98] J. Matoušek, *On the L^2-discrepancy for anchored boxes*, Journal of Complexity **14** (1998), 527–556.

[Mat99] J. Matoušek, *Geometric Discrepancy*, Springer-Verlag, 1999.

[MC93] W. J. Morokoff and R. E. Caflisch, *A quasi-Monte Carlo approach to particle simulation of the heat equation*, SIAM Journal of Numerical Analysis **30** (1993), no. 6, 1558–1573.

[MC94] W. J. Morokoff and R. E. Caflisch, *Quasi-random sequences and their discrepancies*, SIAM Journal on Scientific Computing **15** (1994), 1251–1279.

[MC95] W. J. Morokoff and R. E. Caflisch, *Quasi-Monte Carlo integration*, Journal of Computational Physics **122** (1995), 218–230.

[Meh84] K. Mehlhorn, *Multi-dimensional Searching and Computational Geometry, Data Structures and Algorithms 3*, Springer-Verlag, 1984.

[MMN95] G. L. Mullen, A. Mahalanabis, and H. Niederreiter, *Tables of (t, m, s)-net and (t, s)-sequence parameters*, in Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Lecture Notes in Statistics 106 (H. Niederreiter and P.J.-S. Shiue, eds.), Springer, 1995, pp. 58–86.

[Mos95] B. Moskowitz, *Quasirandom diffusion Monte Carlo*, in Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Lecture Notes in Statistics 106 (H. Niederreiter and P.J.-S. Shiue, eds.), Springer, 1995, pp. 278–298.

[MU49] N. Metropolis and S. Ulam, *The Monte Carlo method*, J. Amer. Statist. Assoc. **44** (1949), 335–341.

[Nie72] H. Niederreiter, *Discrepancy and convex programming*, Ann. Mat. Pura Appl. **93** (1972), 89–97.

[Nie77] H. Niederreiter, *Pseudo-random numbers and optimal coefficients*, Adv. Math. **26** (1977), 99–181.

[Nie78] H. Niederreiter, *Quasi-Monte Carlo methods and pseudo-random numbers*, Bull. Amer. Math. Soc. **84** (1978), 957–1041.

[Nie85] H. Niederreiter, *The serial test for pseudorandom numbers generated by the linear congruential method*, Numerische Mathematik **46** (1985), 51–68.

[Nie87] H. Niederreiter, *Point sets and sequences with small discrepancy*, Monatsh. Math. **104** (1987), 273–337.

[Nie88] H. Niederreiter, *Low-discrepancy and low-dispersion sequences*, J. Number Theory **30** (1988), 51–70.

[Nie92] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM-CBMS Lecture Notes No. 63, 1992.

[NW75] H. Niederreiter und J. M. Wills, *Diskrepanz und Distanz von Maßen bezüglich konvexer und Jordanscher Mengen*, Math. Z. **144** (1975), 125–134.

[NX96] H. Niederreiter and C.P. Xing, *Low-discrepancy sequences and global function fields with many rational places*, Finite fields Appl. **2** (1996), 241–273.

[NX98] H. Niederreiter and C.P. Xing, *The algebraic-geometry approach to low-discrepancy sequences*, in Monte Carlo and Quasi-Monte Carlo Methods 1996, Lecture Notes in Statistics 127 (H. Niederreiter, P. Hellekalek, G. Larcher, and P. Zinterhof, eds.), Springer, 1998, pp. 139–160.

[Owe97a] A. B. Owen, *Monte Carlo variance of scrambled equidistribution quadrature*, SIAM J. Numer. Anal. **34** (1997), no. 5, 1884–1910.

[Owe97b] A. B. Owen, *Scrambled net variance for integrals of smooth functions*, Annals of Statistics **25** (1997), no. 4, 1541–1562.

[Owe98] A. B. Owen, *Scrambling Sobol and Niederreiter-Xing points*, Journal of Complexity **14** (1998), 466–489.

[OY91] M. H. Overmars and C.-K. Yap, *New upper bounds in Klee's measure problem*, SIAM J. Comput. **20** (1991), no. 6, 1034–1045.

[Pas97] S. H. Paskov, *New methodologies for valuing derivatives*, in Mathematics of Derivative Securities (S. Pliska and M. Dempster, eds.), Cambridge University Press, 1997, pp. 545–582.

[PFT88] W. H. Press, B. P. Flannery, and S. A. Teukolsky, *Numerical Recipes in C : The Art of Scientific Computing*, Cambridge U. P., New York, 1988.

[PG81] J. W. Pratt and J. D. Gibbons, *Concepts of Nonparametric Theory*, Springer, 1981.

[Pro83] P. D. Proinov, *Estimation of L^2 discrepancy of a class of infinite sequences*, C. R. Acad. Bulg. Sci. **36** (1983), no. 1, 37–40.

[Pro88] P. D. Proinov, *Discrepancy and integration of continuous functions*, J. Approx Theory **52** (1988), 121–131.

[PS85] F. P. Preparata and M. I. Shamos, *Computational Geometry : An Introduction*, Springer-Verlag, 1985.

[Ros97] S. M. Ross, *Simulation*, second ed., Academic Press, 1997.

[Rot54] K. F. Roth, *On irregularities of distribution*, Mathematika **1** (1954), 73–79.

[Rot80] K. F. Roth, *On irregularities of distribution IV*, Acta Arith. **37** (1980), 67–75.

[RST96] I. Radović, I. M. Sobol, and R. F. Tichy, *Quasi-Monte Carlo methods for numerical integration : Comparison of different low discrepancy sequences*, Monte Carlo Methods and Appl. **2** (1996), no. 1, 1–14.

[Sch72] W. M. Schmidt, *Irregularities of distribution VII*, Acta Arith. **21** (1972), 45–50.

[Sch77] W. M. Schmidt, *Irregularities of distribution X*, in Number theory and algebra, Academic Press, 1977, pp. 311–329.

[Sha88] J. E. H. Shaw, *A quasi-random approach to integration in Bayesian statistics*, Annals of Statistics **16** (1988), no. 2, 895–914.

[SM94] J. Spanier and H. Maize, *Quasi-random methods for estimating integrals using relatively small samples*, SIAM Review **36** (1994), no. 1, 18–44.

[Sob67] I. M. Sobol, *On the distribution of points in a cube and the approximate evaluation of integrals*, USSR Comput. Math. Math. Phys. **7** (1967), no. 4, 86–112.

[Sob76] I. M. Sobol, *Uniformly distributed sequences with an additional uniform property*, USSR Comput. Math. Math. Phys. **16** (1976), 236–242.

[Sob82] I. M. Sobol, *On an estimate of the accuracy of a simple multidimensional search*, Soviet Math. Dokl. **26** (1982), no. 2, 398–401.

[Sri78] S. Srinivasan, *On two-dimensional Hammersley's sequences*, Journal of Number Theory **10** (1978), 421–429.

[SW98] I. H. Sloan and H. Woźniakowski, *When are quasi-Monte Carlo algorithms efficient for high dimensional integrals ?*, Journal of Complexity **14** (1998), 1–33.

[Tez95] S. Tezuka, *Uniform Random Numbers : Theory and Practice*, Kluwer Academic Publishers, Boston, 1995.

[Tez98] S. Tezuka, *Financial applications of Monte Carlo and quasi-Monte Carlo methods*, in Random and Quasi-Random Point Sets, Lecture Notes in Stat. 138 (P. Hellekalek and G. Larcher, eds.), Springer, 1998, pp. 303–332.

[Thi] E. Thiémard, *Optimal volume rectangles with k points and star discrepancy*, submitted to Mathematical Methods of Operations Research.

[Thi98] E. Thiémard, *Economic generation of low-discrepancy sequences with a b-ary Gray code*, EPFL-DMA-ROSO, RO981201 (1998), http ://rosowww.epfl.ch/papers/grayfaure/.

[Thi00] E. Thiémard, *Computing bounds for the star discrepancy*, Computing **65** (2000).

[Thi01] E. Thiémard, *An algorithm to compute bounds for the star discrepancy*, Journal of Complexity **17** (2001).

[Tuf97] B. Tuffin, *Simulation accélérée par les méthodes de Monte Carlo et quasi-Monte Carlo : théorie et applications*, Thèse de doctorat : Université de Rennes 1 (1997).

[Tuf98] B. Tuffin, *A new permutation choice in Halton sequences*, in Monte Carlo and Quasi-Monte Carlo Methods 1996, Lecture Notes in Statistics 127 (H. Niederreiter, P. Hellekalek, G. Larcher, and P. Zinterhof, eds.), Springer, 1998, pp. 427–435.

[TW94] J. Traub et H. Woźniakowski, *Les problèmes à grand nombre de variables*, Pour la Science **197** (1994), 52–58.

[vAE45] T. van Aardenne-Ehrenfest, *Proof of the impossibility of a just distribution of an infinite sequence of points over an interval*, Proc. Kon. Ned. Akad. v. Wetensch. **48** (1945), 266–271.

[vdC35] J. G. van der Corput, *Verteilungsfunktionen I and II*, Proc. Kon. Ned. Akad. v. Wetensch. **38** (1935), 813–821 and 1058–1066.

[War72] T. T. Warnock, *Computational investigations of low-discrepancy point sets*, in Applications of Number Theory to Numerical Analysis (S. K. Zaremba, ed.), 1972, pp. 319–343.

[War95] T. T. Warnock, *Computational investigations of low-discrepancy point sets II*, in Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Lecture Notes in Statistics 106 (H. Niederreiter and P.J.-S. Shiue, eds.), Springer, 1995, pp. 354–361.

[Wey16] H. Weyl, *Über die Gleichverteilung von Zahlen mod Eins*, Math. Ann. **77** (1916), 313–352.

[WF97] P. Winker and K.-T. Fang, *Application of threshold accepting to the evaluation of the discrepancy of a set of points*, SIAM J. Numer. Anal. **34** (1997), no. 5, 2028–2042.

[Whi77] B. E. White, *On optimal extreme-discrepancy point sets in the square*, Numer. Math. **27** (1977), 157–164.

[Wol98] L. A. Wolsey, *Integer Programming*, Wiley-Interscience, 1998.

[Woź91] H. Woźniakowski, *Average case complexity of multivariate integration*, Bull. Amer. Math. Soc. **24** (1991), 185–194.

[XN95] C.P. Xing and H. Niederreiter, *A construction of low-discrepancy sequences using global function fields*, Acta Arith. **73** (1995), no. 3, 87–102.

[Zar68a] S. K. Zaremba, *The mathematical basis of Monte Carlo and quasi-Monte Carlo methods*, SIAM Review **10** (1968), no. 3, 303–314.

[Zar68b] S. K. Zaremba, *Some applications of multidimensional integration by parts*, Ann. Polon. Math. **21** (1968), 85–96.

[Zar70] S. K. Zaremba, *La discrépance isotrope et l'intégration numérique*, Ann. Mat. Pura Appl. **87** (1970), 125–136.

Curriculum vitæ

Nom, prénom : Thiémard Eric
Date de naissance : 27 avril 1971
Nationalité : suisse, originaire de Chénens (FR)

▷ Formation :

1994	Ingénieur mathématicien diplômé EPFL
1989	Maturité fédérale et baccalauréat cantonal de type C

▷ Activités professionnelles :

1994 – 2000	Assistant au département de Mathématiques de l'EPFL au sein de la chaire de Recherche Opérationnelle du Professeur Th. M. Liebling.
1996 – 1999	Chargé de cours dans le cadre du cycle postgrade en management de systèmes logistiques de l'IML (Institut International de Management pour la Logistique) à Lausanne et Paris.
1997	Mandaté par une entreprise de distribution d'information numérique pour une étude de simulation sur le dimensionnement d'un réseau.
1997	Participation à l'organisation de la conférence internationale ISMP 97.
1994 – 1995	Mandaté par une entreprise de distribution de produits pharmaceutiques pour un projet de gestion de stocks et de simulation.
1995	Chargé de cours pour la Formation Postgrade en Informatique et Télécommunications, École d'Ingénieurs de l'État de Vaud.