Publications du **La**boratoire de
Combinatoire et d'
Informatique
Mathématique

3

Louise Laforest

Étude des arbres hyperquaternaires

Département de mathématiques et d'informatique

Université du Québec à Montréal

Étude des arbres hyperquaternaires

par

Louise Laforest

School of Computer Science
Université McGill, Montréal

Avril 1990

Une thèse
présentée à la faculté d'études supérieures et de recherche
dans le cadre de l'obtention du diplôme de
Docteur en philosophie

Résumé

Cet ouvrage traite principalement des arbres hyperquaternaires de points ("point quadtree") qui sont une généralisation des arbres binaires de fouille. En premier lieu, un survol des structures de données hiérarchiques est présenté. Sont décrits entre autres les arbres hyperquaternaires de région, les arbres k-d, les arbres pseudo-hyperquaternaires et pseudo-k-d. Les résultats relatifs aux arbres binaires de fouille ainsi que ceux concernant les arbres pseudo-hyperquaternaires de points sont donnés. Une étude plus poussée des arbres hyperquaternaires nous a permis d'obtenir des résultats concernant la profondeur du dernier nœud inséré, la proportion des divers types de nœuds dans un arbre hyperquaternaire de points. Enfin, nous étudions une alternative de ces arbres, soit les arbres pseudo-hyperquaternaires ce qui nous permet de montrer, après étude du cas à deux dimensions, que ceux-ci semblent plus performants au niveau de la fouille, de l'ajout et de la suppression de points que les arbres hyperquaternaires originaux.

Abstract

In this thesis we talk about point quadtrees which are a generalization of the binary search tree. First of all, we present a survey on hierarchical data structures such as pixel quadtrees, k-d trees, pseudo-quadtrees and pseudo k-d trees. Results concerning binary search trees are given as well as for point quadtrees. A study in depth allowed us to get interesting results about the depth of the last node inserted and the proportion of the different kind of nodes in a quad tree. Finally, we study the pseudo-quadtree which can be viewed as a good alternative of the original quadtree in the sense that it is more powerful for search, insert and delete points than its counterpart.

Remerciements

Tout d'abord j'aimerais remercier mon directeur de thèse, Luc Devroye, qui m'a permis d'accomplir ce travail. Je le remercie plus particulièrement pour sa grande disponibilité, ses suggestions, ses critiques me permettant ainsi de profiter de ses compétences incontestées.

En second lieu, j'aimerais remercier de nombreux amis dont mon compagnon de vie, Alain Latour qui fut une aide précieuse, entre autres pour la mise en page de cet ouvrage, et pour ses judicieux conseils en général; Gilbert Labelle qui m'a aidée de son expertise éclairée; mon amie Hélène Décoste pour ses connaissances relatives au Macintosh et tous les membres de ma famille qui sont sûrement contents que cela soit terminé.

Table des matières

Liste des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 Introduction

Comme le mentionne Samet (1984) dans son exposé sur les arbres quaternaires et les structures de données y étant reliées, les structures de données hiérarchiques, comme le sont les arbres quaternaires, deviennent de plus en plus de représentations techniques importantes dans les domaines du graphisme sur ordinateur, la manipulation d'images, la géométrie computationnelle, les systèmes d'information géographiques et la robotique, pour n'en nommer que quelques-unes. Toutes sont basées sur la décomposition hiérarchique, c'est-à-dire récursive, de l'entité à décrire, à stocker, à questionner, à manipuler. La structure hiérarchique sous-jacente en découlant permet une observation locale d'un sous-ensemble, ou partie, de l'entité pouvant alors être considérée comme entité propre. Ceci permet aussi d'observer la structure à un niveau de détail plus ou moins grossier. Les structures de données hiérarchiques sont séduisantes à cause de leur clarté conceptuelle, de la simplicité de leur implantation ainsi que des algorithmes inhérents à leur manipulation. Pour avoir une vue générale assez complète ainsi qu'une bibliographie couvrant tous les aspects concernant le sujet, il faut consulter Samet (1984).

Nous nous proposons d'étudier cette classe spécifique de structures de données

hiérarchiques. On parlera plus spécifiquement d'arbre binaire, d'arbre quaternaire, d'arbre hyperquaternaire. On décrira aussi certains types d'arbres comme les arbres pseudo-hyperquaternaires, les arbres k–d, les arbres pseudo-k–d, les "tries" multidimensionnels, pour n'en nommer que quelques-uns. Le lecteur doit être avisé qu'un effort a été fait pour franciser les termes anglais plus connus. Ainsi, ce que Samet (1984) nomme "quad tree" a été traduit ici par "arbre hyperquaternaire", et lorsqu'on parle d'arbre binaire, il s'agit du cas particulier d'arbre hyperquaternaire à une dimension et que lorsqu'on parle d'arbre quaternaire, il s'agit des arbres hyperquaternaires à deux dimensions. Il ne semble pas exister, à l'heure actuelle, de consensus à ce sujet. Berstel et Abdallah (1989) parlent quant à eux de tétrarbre qui correspond au terme "arbre quaternaire" dans le présent ouvrage.

Le présent chapitre sera consacré aux résultats relatifs aux arbres hyperquaternaires de région appelés en anglais "region quadtree" ou "pixel quadtree". Les arbres hyperquaternaires de points, nommés "point quadtrees" en anglais, qui sont une généralisation de l'arbre binaire de fouille, seront examinés de plus près. On mettra en relief les résultats théoriques connus jusqu'ici, ainsi que des résultats originaux constituant la pierre angulaire de la présente thèse.

Le chapitre 2 concerne l'étude plus spécifique de l'arbre hyperquaternaire de points. La profondeur du dernier nœud ajouté dans un arbre sera la quantité étudiée. On y retrouvera certains résultats connus sur les arbres binaires de recherche, exprimés selon notre nouvelle approche. Des résultats relatifs aux arbres hyperquaternaires de dimension deux, plus spécifiquement l'espérance et la variance de la profondeur du dernier nœud ajouté à un arbre quaternaire seront démontrés. La section 1.5 du présent chapitre fait mention de ces résultats.

Au chapitre 3, on retrouvera une étude sur le comportement asymptotique de la

profondeur du dernier nœud ajouté à un arbre hyperquaternaire de points.

Le chapitre 4 est consacré aux nouveaux résultats relatifs aux nœuds de l'arbre hyperquaternaire de dimension deux. On avait déjà des résultats pour les arbres binaires de fouille (Mahmoud, 1986.) Tous ces résultats sont aussi résumés à la section 1.5 de ce chapitre.

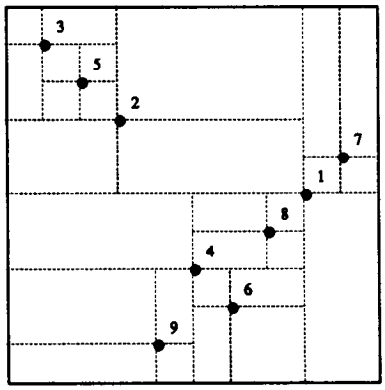
Le chapitre 5 fait l'étude d'une version modifiée des arbres hyperquaternaires, soit les arbres pseudo-hyperquaternaires. On y étudie la profondeur du dernier nœud ajouté dans une telle structure dans le cas à deux dimensions. Nous y retrouvons une étude empirique sur le sujet.

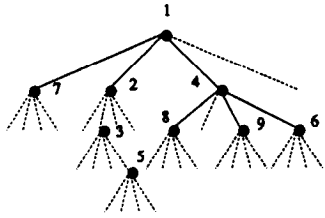
Maintenant, attaquons-nous au sujet qui nous préoccupe. On pourrait donner comme définition d'arbre hyperquaternaire, la définition que Samet (1984) en donne. Celle-ci englobe une grande variété de structures.

Définition: Un *arbre hyperquaternaire* est une structure de données hiérarchique basée sur la décomposition récursive de l'espace.

Ce qui permet de différencier les types d'arbres hyperquaternaires est, premièrement, le type des données qu'ils représentent. Ce peut être un ensemble de points dans l'espace, des données de nature géographique, la représentation d'images, la représentation de volumes, etc... Deuxièmement, le principe guidant le processus de décomposition les caractérise aussi. Nous verrons des exemples plus loin quand nous parlerons d'arbres hyperquaternaires de région et d'arbres hyperquaternaires de points. Finalement, les différents types se caractérisent par le niveau de résolution des données représentées. Cette résolution peut être variable ou non.

La figure suivante nous donne un bon exemple de ce que peut être un arbre hyperquaternaire de points à deux dimensions. Un ensemble de neuf points est représenté sous forme graphique et sous forme d'arbre. Les points sont ajoutés un à un. Le point X_1 sépare la région initiale en quatre quadrants. Le point X_2 se trouve dans le deuxième quadrant induit par X_1 et divise celui-ci en quatre quadrants. Tous les autres points sont ainsi ajoutés à la structure. Le graphique montre bien la hiérarchie entre les points, ainsi que le principe sous-jacent de subdivision de l'espace. Des branches pointillées ont été ajoutées dans l'arbre pour bien montrer où sont les régions exemptes de points. On peut aussi en faire un arbre plein en ajoutant des nœuds externes (voir la définition de nœud externe plus loin) représentant les régions vides.





$X_1 = (8,5), X_2 = (3,7), X_3 = (1,9), X_4 = (5,3), X_5 = (2,8)$
$X_6 = (6,2), X_7 = (9,6), X_8 = (7,4), X_9 = (4,1)$

Fig. 1.1 – Arbre hyperquaternaire à deux dimensions.

Il faut remarquer que la structure de l'arbre représentant les points est moins

fonction des coordonnées des points comme telles mais plutôt fonction de l'ordre d'"arrivée" des points. On remarquera, entre autres, que les points qui sont à une même profondeur dans l'arbre peuvent arriver dans n'importe quel ordre entre eux sans changer la structure de l'arbre. Par exemple, si, au lieu d'insérer les points de X_1 à X_9 dans cet ordre, on avait inséré $X_1, X_4, X_2, X_3, X_7, X_8, X_6, X_9$ puis X_5, on aurait eu exactement la même subdivision du plan ainsi que le même arbre. On peut remarquer aussi que, hormis la racine, tout nœud a un numéro supérieur à celui de son parent.

La figure 1.2 nous donne la subdivision de l'espace obtenue en prenant les mêmes points que ceux utilisés pour la figure 1.1 mais en utilisant la structure d'arbre k–d. L'espace est subdivisé alternativement par la coordonnée x ou y des points considérés. On remarque que l'on obtient un arbre binaire. La structure d'arbre k–d nous donne toujours un arbre binaire quelle que soit la dimension d considérée car à chaque niveau de l'arbre correspond la subdivision selon une des d coordonnées.

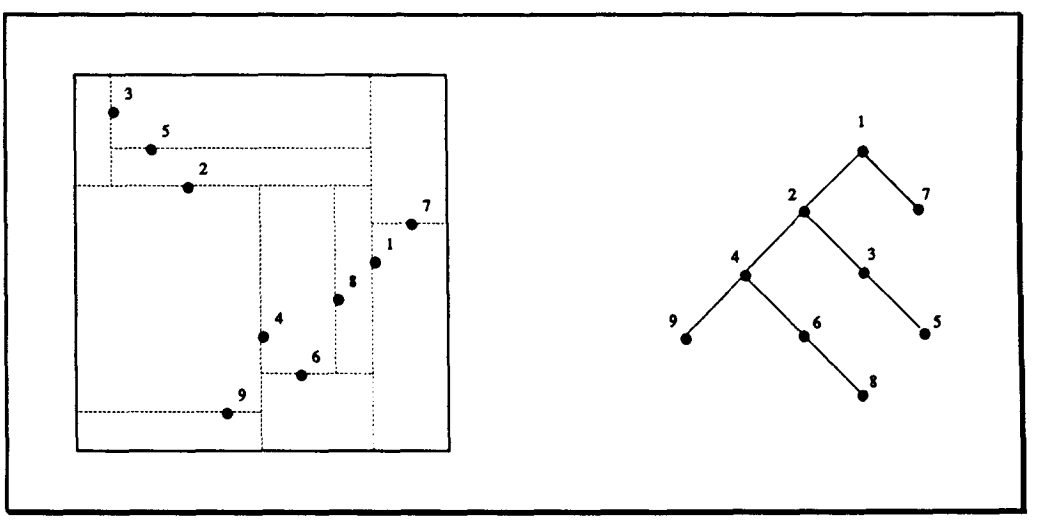


Fig. 1.2 – Arbre k–d à deux dimensions.

La figure 1.3 utilise le même ensemble de points mais les représente selon un arbre pseudo-hyperquaternaire, tandis que la figure 1.4 les représente sous forme d'arbre pseudo-k–d. La caractéristique fondamentale des ces arbres pseudo-hyperquaternaires et pseudo-k–d est le fait que les points de l'ensemble sont des feuilles de l'arbre alors que les nœuds internes sont des points choisis hors de l'ensemble de points pour effectuer les subdivisions.

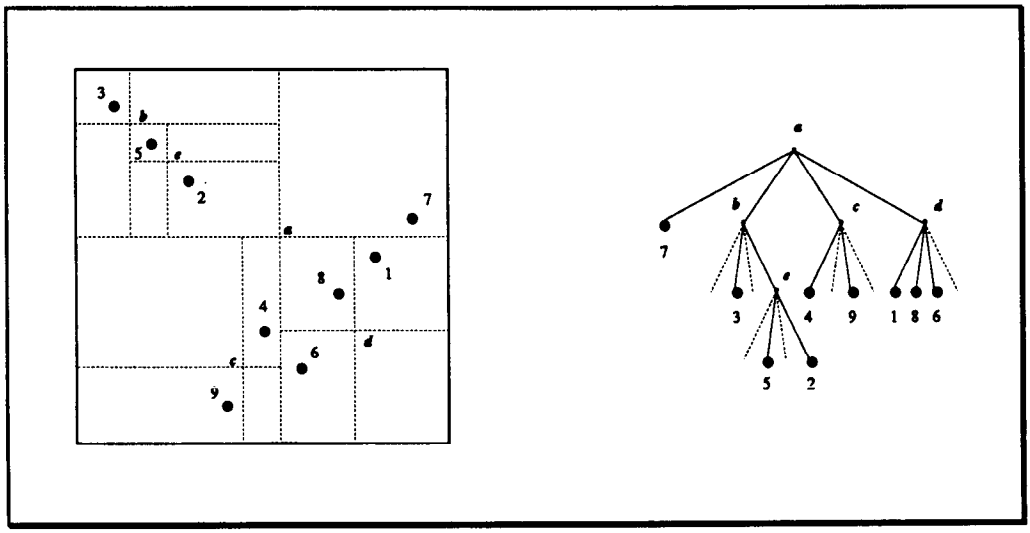


Fig. 1.3 – Arbre pseudo-hyperquaternaire à deux dimensions.

La figure 1.5 nous montre un exemple d'image rangée sous forme d'arbre hyperquaternaire de région à deux dimensions. L'image est d'abord décomposée en "pixels" qui sont représentés par la valeur 1 pour un pixel noir, et 0 pour un pixel blanc. Ensuite, on ajoute une bordure de pixels blancs de façon à avoir un carré de pixels dont chacun des côtés contient un nombre de pixels qui soit une puissance de deux. On divise en quatre carrés égaux le carré de départ et si un carré n'est pas de couleur uniforme, celui-ci est divisé à son tour en quatre carrés.

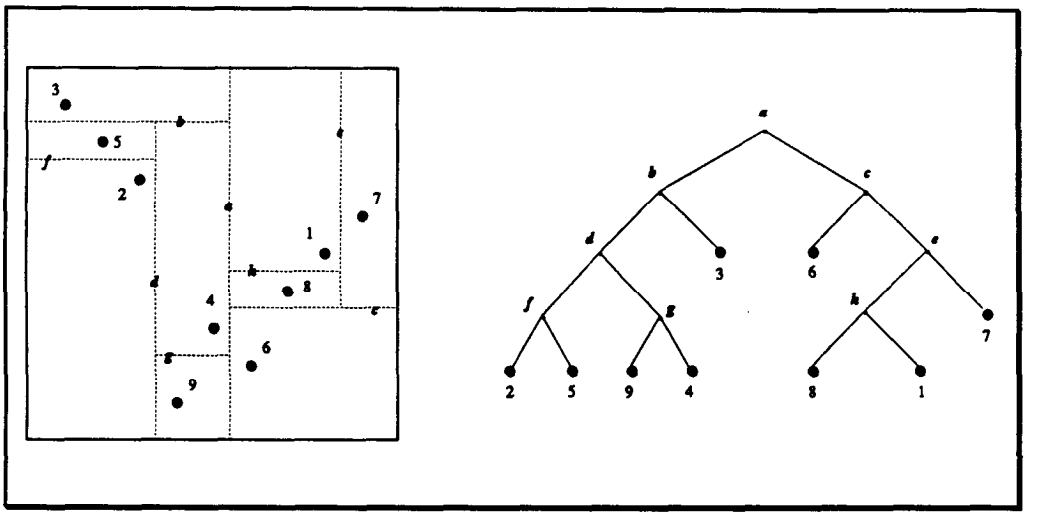


Fig. 1.4 – Arbre pseudo-*k*–d à deux dimensions.

On peut voir ici que le nombre maximal de divisions dépend de la grosseur de l'image en nombre de pixels, le pixel étant la plus petite entité non-morcelable. Il dépend aussi de la forme de l'image. L'arbre quaternaire de région est donc une structure à résolution variable. Chaque nœud a aucun ou quatre enfants, ce qui en fait un arbre plein (voir Standish 1980). Les quatre enfants représentent respectivement les quadrants nord-est, nord-ouest, sud-ouest et sud-est, dans l'ordre habituel. Les feuilles sont noires ou blanches et les nœuds non-terminaux sont considérés comme étant gris, car constitués de composantes noires, blanches et/ou grises. Comme on ne divise pas une région de couleur uniforme, les enfants d'un nœud ne peuvent être tous blancs ou tous noirs. Dans la figure 1.6, le nœud E est une feuille car on n'a pas eu à subdiviser la région y correspondant. Puisque la grille de départ comporte $2^3 \times 2^3$ pixels, le nombre maximal de subdivisions est de trois. Ce nombre aurait pu ne pas être atteint si la région avait été toute noire ou toute blanche. On aurait alors eu un arbre ne contenant qu'une feuille noire ou blanche selon le cas. Une autre caractéristique de cette façon de faire est que, si la grille de départ est un

carré de $2^n \times 2^n$ pixels alors un nœud à une profondeur de i, la racine étant à une profondeur de zéro, représentera un carré de $2^{n-i} \times 2^{n-i}$ pixels. Donc, plus on descend dans l'arbre, plus on a affaire à de petits carrés jusqu'à concurrence d'un pixel.

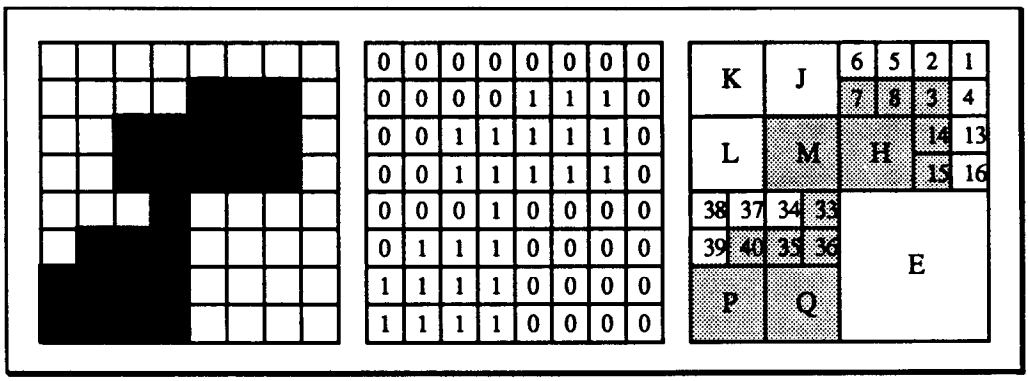


Fig. 1.5 – Image décomposée en pixels.

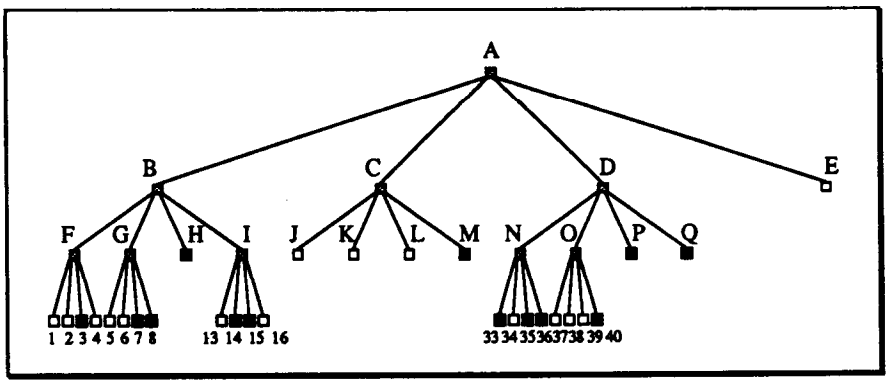


Fig. 1.6 – Arbre quaternaire de région.

Il y a plusieurs façons de représenter les régions. La façon la plus simple est le "*run length code*", décrit par Rutovitz (1968), où les blocs sont restreints à des rectangles de 1 par m, m étant le nombre de pixels représentant la région. Une autre représente la

région en un ensemble maximal de blocs carrés se chevauchant possiblement. Ces carrés sont habituellement décrits par leur centre et leur rayon, méthode appelée "*medial axis transformation*". L'arbre quaternaire décrit plus haut est une variante de cette dernière méthode puisqu'il repose sur un ensemble de carrés, ne se chevauchant pas, ayant comme grandeur et position des valeurs prédéterminées. La longueur des côtés est une puissance de deux. On peut aussi utiliser une méthode de subdivision non régulière qui consisterait, par exemple, à diviser, en rectangles de grandeur arbitraire, la région à décrire. Ceci peut prendre moins d'espace.

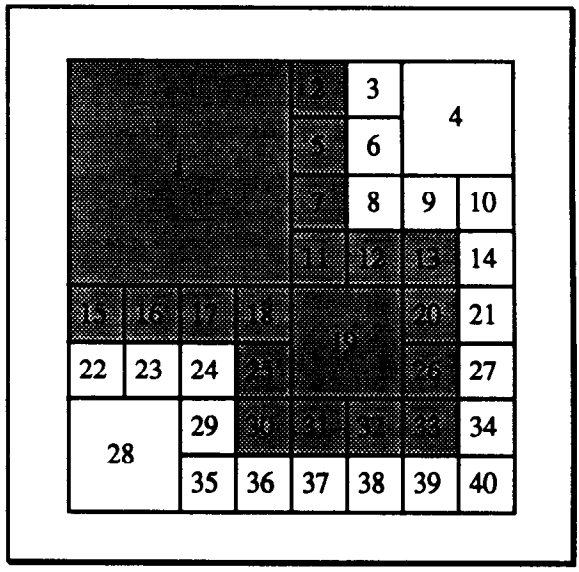


Fig. 1.7 – Image décomposée en pixels.

Considérons la région de la figure 1.7 qui serait représentée par l'arbre quaternaire de la figure 1.8. Elle demande beaucoup de subdivisions, alors qu'on voit très bien que la région est formée essentiellement de deux rectangles superposés. Cependant, cette méthode a le désavantage que, n'étant pas régulière, il faut déterminer où l'on effectuera la partition en rectangles, ce qui suppose une fouille. Les méthodes qui subdivisent selon un

schème régulier sont plus avantageuses. On est amenés à penser que la région régulière qui résulte du processus de subdivision pourrait ne pas être un carré, mais plutôt un triangle équilatéral, un triangle isocèle ou un hexagone (voir figure 1.9).

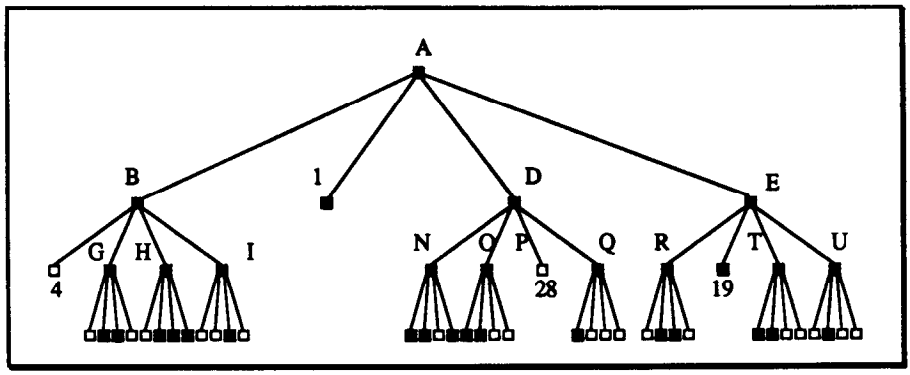


Fig. 1.8 – Arbre quaternaire de région.

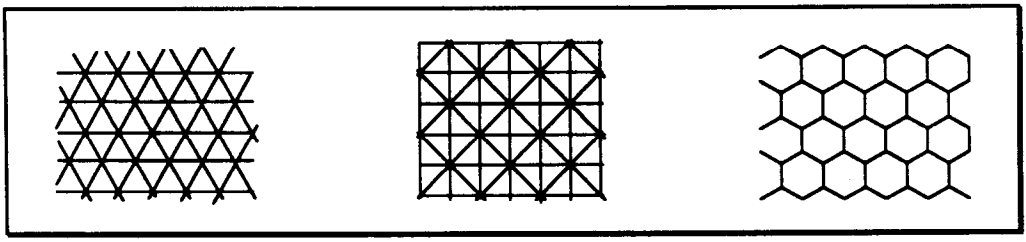


Fig. 1.9 – Différentes subdivisions du plan.

En général, la décomposition de l'image doit avoir les propriétés suivantes:

(1) On doit pouvoir répéter indéfiniment la décomposition de façon à pouvoir décrire des images de n'importe quelles dimensions,

(2) La partition devrait être indéfiniment décomposable en des partitions de plus en plus fines afin d'augmenter la résolution.

Dans le cas des hexagones, la propriété (2) n'est pas tout à fait respectée puisque ceux-ci ne

peuvent se diviser en d'autres hexagones plus petits contrairement aux triangles équilatéraux, par exemple. La méthode la plus facile reste cependant celle produisant des blocs carrés, et nous nous en tiendrons à l'étude de celle-ci.

1.2 Autres structures de données multidimensionnelles

Une structure de données hiérarchique très intéressante est l'arbre *k*–d, introduit par Bentley (1975). C'est une structure particulièrement avantageuse en ce qui a trait aux bases de données puisqu'elle est assez aisée à implanter et permet de pouvoir répondre de façon plutôt efficace à un grand nombre de types différents de requêtes. Cette structure est appelée aussi "arbre binaire multidimensionnel". Cependant, la première appellation est plus connue et usitée. Dans l'expression "arbre *k*–d", *k* désigne la dimension considérée. Le principal avantage est que, justement, le nombre d'enfants de chaque nœud est limité à deux, contrairement à l'arbre hyperquaternaire de dimension *d* qui peut engendrer jusqu'à 2^d enfants par nœud, ce qui est extrêmement coûteux du strict point de vue mémoire utilisée. L'arbre *k*–d est construit de sorte que chaque niveau correspond à l'une des *k* coordonnées. Par exemple, dans le cas à deux dimensions, où il est question d'arbre 2–d, le sous-arbre gauche correspond aux points de coordonnée *x* plus petite que la coordonnée *x* de la racine, et le sous-arbre droit correspond aux points de coordonnée *x* plus grande que la coordonnée *x* de la racine. En fait, les niveaux pairs séparent les points selon la coordonnée *x*, et les niveaux impairs les divisent selon la coordonnée *y*. La figure 1.10 donne une bonne idée de quoi il retourne.

Comme il s'agit fondamentalement d'un arbre binaire, les propriétés de ceux-ci demeurent, quelque soit la dimension considérée.

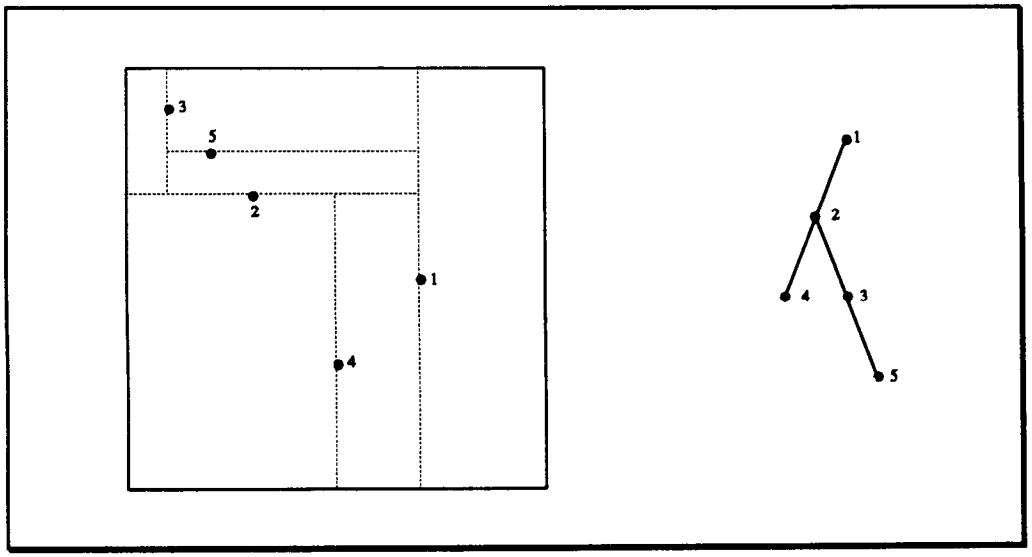


Fig. 1.10 – Arbre k–d.

Willard (1982) a élaboré une structure hiérarchique, appelée arbre polygone ("polygon tree"), qui permet la fouille dans des régions plus complexes comme le sont les polygones. L'idée est de subdiviser $\mathbb{R}^2$ à l'aide de J droites, pas nécessairement orthogonales entre elles, mais ayant d'autres propriétés. Cette subdivision est appelée "J–way division". Le cas où $J = 2$ se ramène à l'arbre hyperquaternaire de dimension deux, mais avec des droites de subdivision non perpendiculaires.

Une autre structure hiérarchique permettant de représenter des ensembles de points, définie par Samet (1984), est l'arbre PR hyperquaternaire (P pour point et R pour région), appelé en anglais, "PR quadtree". Ceux-ci sont structurés comme les arbres hyperquaternaires de région en ce sens que la décomposition hiérarchique est fixe. La différence est que les feuilles sont soit vides (blanc), soit qu'elles contiennent une donnée (noir). La figure 1.11 en donne un exemple. Orenstein (1982) a défini une structure analogue

utilisant des arbres binaires plutôt que des arbres hyperquaternaires et cette structure pourrait être appelée arbre *k*–d PR ou mieux encore, "*k*–d trie".

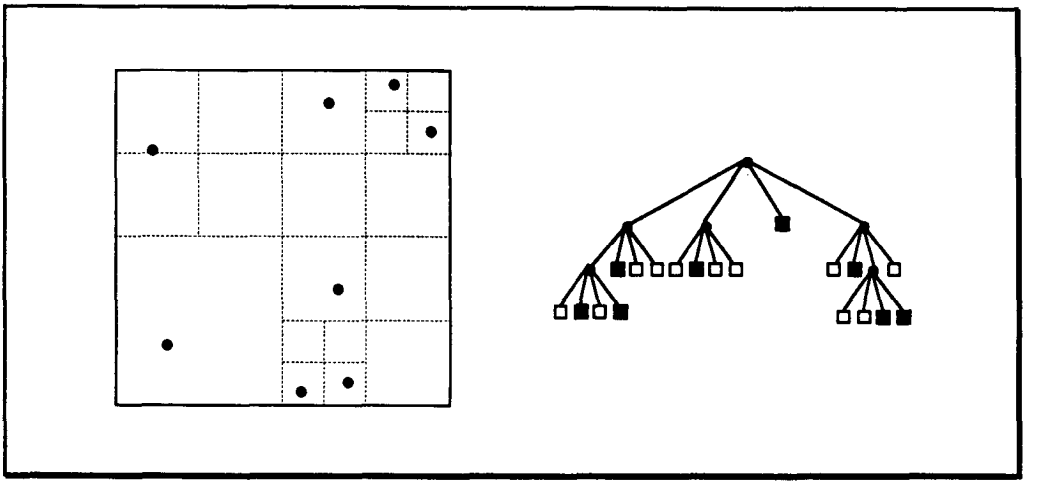


Fig. 1.11 – arbre PR hyperquaternaire.

1.3 Historique des arbres hyperquaternaires de points

Les premiers à parler d'arbre quaternaire furent Finkel & Bentley (1974). Ils découvrirent empiriquement que la longueur moyenne du chemin interne (voir la définition plus loin), pour un arbre hyperquaternaire à deux dimensions est, en gros, proportionnelle à $n \log n$, où n est le nombre de nœuds dans l'arbre. On verra plus loin que la valeur exacte de la longueur moyenne du chemin interne est de $(n+\frac{1}{3})H_n - \frac{7}{6}n - \frac{1}{6}$, où $H_n = \sum_{i=1}^{n} \frac{1}{i}$. Ils donnent un algorithme d'insertion dans une telle structure. Aussi, ils ont essayé d'améliorer la structure de l'arbre en développant un algorithme simple de rebalancement analogue à celui utilisé pour rebalancer les arbres binaires. On peut le voir dans les deux figures qui suivent:

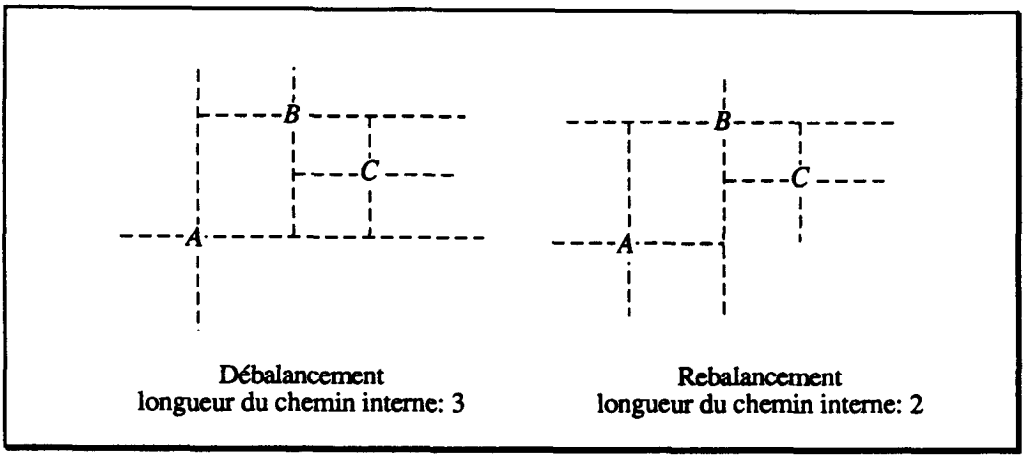


Fig. 1.12 – Simple rebalancement.

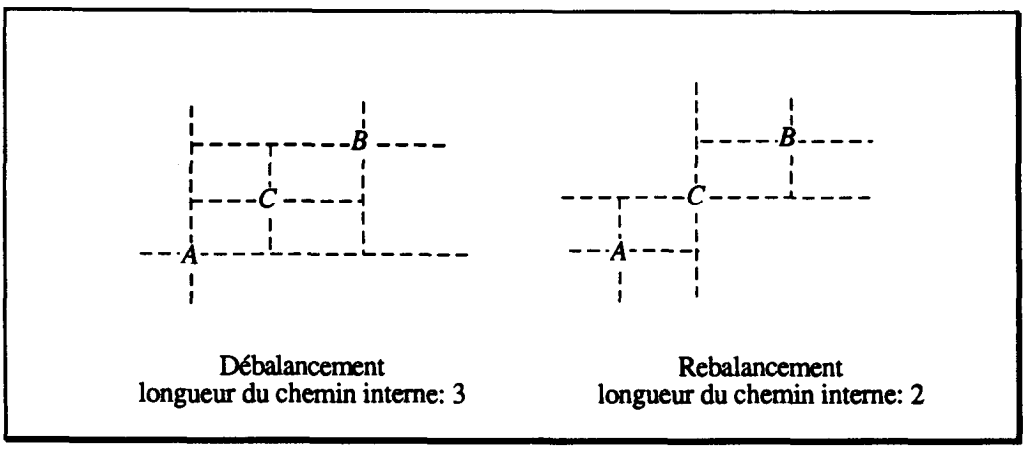


Fig. 1.13 – Double rebalancement.

Cependant, ils remarquèrent que, dans certains cas, un arbre construit avec l'algorithme de rebalancement décrit donne un arbre ayant une longueur de chemin interne plus grande que celle obtenue sans rebalancement, avec les mêmes points, arrivant dans le même ordre. Ils mentionnèrent aussi qu'il y a deux classes de fouilles facilitées par les arbres

hyperquaternaires de points. La première est la recherche de points dans un tel arbre. La deuxième est la fouille de région, par exemple: "Quels sont les points qui sont à l'intérieur d'un cercle de rayon 7 et de centre (5,31)?" Ils donnent un algorithme effectuant de telles fouilles.

1.4 Revue de littérature

En fouillant la littérature sur le sujet, on découvre une abondante bibliographie sur les structures de données hiérarchiques. Cela va des études théoriques sur ces structures aux différents algorithmes inhérents à celles-ci.

La présente section se veut descriptive de ce qui s'est écrit sur le sujet, sans pour autant être limitative.

Tout d'abord, l'article de Samet (1984), dans lequel il expose une étude sur les arbres hyperquaternaires et les structures de données afférentes, se décompose en trois grandes parties, c'est-à-dire: les arbres hyperquaternaires de région, les arbres hyperquaternaires de points et les arbres représentant des données curvilignes. Ballard (1981) introduit la structure hiérarchique d'arbre de bandes ("strip trees") qui permet la représentation de telles données. Ces dernières concernent la description de la bordure d'une région en opposition avec les arbres hyperquaternaires de région et les arbres hyperquaternaires de point qui décrivent l'intérieur ou la région elle-même.

Dans la section des arbres hyperquaternaires de région, on y décrit aussi les différentes façons de subdiviser l'espace dans le but de décrire une région. Des algorithmes permettant de trouver les voisins d'un élément de région, ainsi que des références s'y rapportant sont donnés. Une section est consacrée aux différentes façons de représenter les ar-

bres hyperquaternaires, dont, entres autres, celles utilisant des pointeurs vers les enfants et celles n'utilisant pas de pointeurs. On donne en plus des algorithmes pour passer d'une représentation à l'autre. Il est aussi intéressant de pouvoir effectuer l'union, l'intersection, la transformation d'images, pour ne nommer que quelques-unes des opérations possibles. Des algorithmes permettant d'effectuer de telles opérations sont décrits. Dyer, Rosenfeld et Samet (1980) ont publié des résultats concernant la représentation de régions et d'images. Ils décrivent un algorithme transformant un arbre quaternaire en un code chaîné. Ils en donnent aussi une analyse théorique. Une autre section de l'article de Samet (1984) survole le sujet concernant les données de nature volumétrique. Puech et Yahia (1985), traitent des structures de données hiérarchiques permettant la description d'images, de volumes. Une description formelle et récursive de l'ensemble de tous les arbres hyperquaternaires à d dimensions est donnée. Les auteurs donnent des statistiques sur les arbres hyperquaternaires. Une section est réservée à l'analyse des algorithmes de recherche des voisins dans un arbre hyperquaternaire de dimension deux, un autre à la compression des arbres hyperquaternaires. Enfin, on fait mention des opérations booléennes possibles et de la superposition d'images.

Pour ce qui est des arbres hyperquaternaires de points, dont il est question dans Samet (1984), on en parle comme d'une généralisation de l'arbre binaire de fouille. En fait, le premier article décrivant les arbres hyperquaternaires de points est celui de Finkel et Bentley (1974). Ils les décrivent aussi comme étant une généralisation des arbres binaires de fouille. Les auteurs mentionnent que l'arbre hyperquaternaire de points est une structure bien adaptée pour la recherche d'un point et pour la recherche dans une région. Ils donnent un algorithme pour la fouille dans une région suivi d'une étude empirique. Dans l'ouvrage d'Overmars (1983), on parle de problèmes de fouille, de reconstruction locale, ou rebalancement, de reconstruction partielle, de reconstruction globale, pour ne nommer que quelques items discutés qui peuvent s'appliquer aux arbres hyperquaternaires. L'article de

Lueker (1978) fait mention des structures de données conçues pour manipuler les ensembles de données multidimensionnelles. Il y est aussi question d'opérations plus générales sur ces structures, comme la recherche dans une région ("range query"). L'auteur fait remarquer que Knuth (1973) dit que pour le problème de recherche dans une région orthogonale, ou, autrement dit, dans un hyperrectangle, il n'existe pas jusqu'à maintenant de structure de données permettant efficacement de résoudre ce problème. Faisant référence à l'article de Finkel et Bentley (1974), ainsi qu'à celui de Bentley (1975), Lueker nous rappelle que l'on peut effectuer une recherche dans une région orthogonale en utilisant un arbre hyperquaternaire, en un temps $O(n^{1-1/d})$ dans le pire cas. Ce dernier résultat est dû à Bentley et Stanat (1975) et à Lee et Wong (1977). Un cas particulier de la recherche dans un hyperrectangle est le problème de recherche partielle ("partial match query") où la recherche se fait dans un sous-espace de dimension c, $0 \leq c \leq d$, d étant la dimension de l'espace considéré. Lorsque $c = d$, il s'agit de la recherche d'un point dans l'espace. L'auteur fait référence à Rivest (1976) qui montre dans son article que pour répondre à la recherche partielle ("partial match query"), en utilisant différentes façons de représenter les données, on a besoin d'un temps d'environ $O(n^{\log_2(1+\alpha)})$ ou $O(n^{\alpha})$ où $\alpha = 1 - 1/d$. L'auteur mentionne les travaux de Dobkin et Lipton (1976) sur les problèmes de fouille multidimensionnelle, et ceux de Kung, Luccio et Preparata (1975) concernant les points maxima selon une relation d'ordre partielle, en d dimensions. Fredman (1981) s'est penché quant à lui sur la complexité de la recherche dans un hyperrectangle. Il montre aussi que $\Omega(n(\log n)^d)$ est une borne inférieure du temps requis pour effectuer une séquence de n insertions, suppressions et recherches dans une région. Traitant des structures basées sur les arbres hyperquaternaires de points, l'article d'Overmars et van Leeuwen (1982) se penche sur les cas dynamiques et statiques.

Lueker (1978) discute d'une structure de données pour les fouilles dans un hyperrectangle avec laquelle une séquence de n insertions, suppressions et recherches, en partant

avec une structure vide, peut s'effectuer en un temps moyen de $O(n\log^d n)$. Plus loin, il améliore cette structure de sorte que le pire cas est résolu en un temps de $O(n\log^d n)$. En ce qui concerne l'insertion dans un arbre hyperquaternaire, Finkel et Bentley (1974) donnent un algorithme pour l'insertion dans le cas à deux dimensions. Suit une étude empirique sur cet algorithme. Leurs résultats corroborent les résultats théoriques obtenus ultérieurement. Les auteurs donnent ensuite un algorithme d'insertion plus sophistiqué utilisant une méthode de rebalancement. Une autre étude empirique y faisant suite montre que la longueur du chemin interne est inférieure à celle obtenue avec le premier algorithme. Ils remarquent cependant que, quelquefois, un arbre construit avec l'algorithme utilisant la méthode de rebalancement a une longueur de chemin interne supérieure que s'il avait été construit avec l'algorithme simple.

En ce qui concerne les algorithmes de suppression et d'insertion, plusieurs auteurs s'y sont penché. Tous s'accordent pour dire que la suppression dans un arbre hyperquaternaire de points, même dans le cas à deux dimensions, est une tâche ardue. En effet, Finkel et Bentley (1974) notent que la suppression de points est difficile dans le sens qu'il faut décider quoi faire avec les points situés dans les sous-arbres du point enlevé. Samet (1980) s'y est penché dans un article. Il analyse le cas à deux dimensions. La méthode suggérée par Finkel et Bentley (1974) qui réinsère les nœuds des sous-arbres du nœud enlevé dans la structure peut s'avérer très coûteuse dans le cas, par exemple, où l'on enlève la racine de l'arbre. Samet (1980) propose une méthode analogue à celle utilisée dans le cas à une dimension, soit pour le cas des arbres binaires de fouille, où l'on remplace le nœud enlevé par le nœud le plus "près" de lui. Dans le cas des arbres quaternaires, Samet ajoute d'autres conditions à celle d'être plus près pour choisir le nœud qui remplacera le nœud à enlever. Il en résulte deux procédés qui consistent à choisir le nœud le plus près au hasard dans l'ensemble des candidats, qui sont au nombre d'au plus quatre, ou choisir le nœud, parmi les candidats, qui est le plus près, selon les deux axes, du nœud à enlever. Si ce

nœud n'existe pas, ou qu'il y a plusieurs candidats, il faut choisir celui qui est le plus près selon la métrique L_1. Il en résulte que pour les arbres ayant beaucoup de nœuds, le nœud le plus près ne devrait pas être choisi au hasard. Une étude empirique fait la comparaison entre les deux méthodes mentionnées plus haut et celles décrites dans Finkel et Bentley (1974). Overmars et van Leeuwen (1982) mentionnent aussi qu'il est difficile d'enlever des points dans un arbre hyperquaternaire de points. Ils décrivent alors une méthode permettant d'insérer et d'enlever des points dans la structure tout en conservant l'arbre balancé. Ici, un arbre est considéré comme balancé lorsque pour tous les nœuds internes, chacun des sous-arbres associés possède un nombre de nœuds n'excédant pas une certaine limite fonction d'une constante fixée d'avance. Cet algorithme a un temps moyen de $O(\log^2 N)$ par insertion, où N est le nombre de mises à jour effectuées (insertions avec ou sans rebalancement, suppressions avec ou sans rebalancement.)

Dans Samet (1984), on y discute des arbres k–d introduits par Bentley (1975). Les applications possibles aux bases de données sont discutées dans l'article de Bentley (1979). Overmars et van Leeuwen (1982) s'intéressent aux cas statique et dynamique concernant ces arbres k–d.

Finkel et Bentley (1974) décrivent ce qu'ils appellent l'arbre optimisé. Il s'agit en fait du cas statique où nous disposons de l'ensemble de points avant de construire la structure. Récursivement, on prend le point médian qui servira de racine et qui séparera l'ensemble en quatre quadrants dont chacun ne contiendra pas plus de $\lceil \frac{n}{2} \rceil$ points, n étant le nombre de points considérés. Construire un tel arbre prend un temps de $O(n \log n)$.

Overmars et van Leeuwen (1982) mettent au point une structure analogue à l'arbre hyperquaternaire, appelée arbre pseudo-hyperquaternaire. Celle-ci permet l'ajout et la suppression de points en un temps de $O(\log^2 N)$. Une structure similaire est aussi décrite

pour les arbres *k*–d. Les principaux résultats de cet article font l'objet d'une description plus détaillée plus loin dans la thèse, au chapitre 5.

L'article de van Leeuwen et Wood (1981) fait mention de l'utilisation d'arbres hyperquaternaires dans la résolution du problème de mesure de Klee (1977). Le problème est de trouver un algorithme efficace pour calculer la mesure d'un ensemble de n hyper-rectangles dans un espace à d dimensions. Ils mentionnent que le cas à une dimension est facilement résolu en un temps $O(n \log n)$ et que Bentley a trouvé un algorithme, pour $d \geq 2$, en un temps $O(n^{d-1} \log n)$. Les auteurs présentent un algorithme ayant un temps d'exécution dans le cas le pire de $O(n^{d-1})$ pour $d \geq 3$. Leur méthode est basée sur les arbres hyperquaternaires requérant un espace mémoire quadratique.

Une section de l'article de Samet (1984) fait état de la comparaison entre les arbres hyperquaternaires de région et les arbres hyperquaternaires de points. Une autre section est consacrée à la représentation d'un ensemble de rectangles, les arbres CIF, ayant des applications dans le design des VLSI.

1.5 Résultats sur les arbres hyperquaternaires de points

Quelques définitions

Définition: L'*arbre hyperquaternaire* de points, à d dimensions, est un ensemble fini de nœuds qui est soit vide, ou est constitué d'un nœud appelé *racine* ayant 2^d sous-arbres hyperquaternaires de points, ordonnés, étant disjoints les uns par rapport aux autres et de la racine.

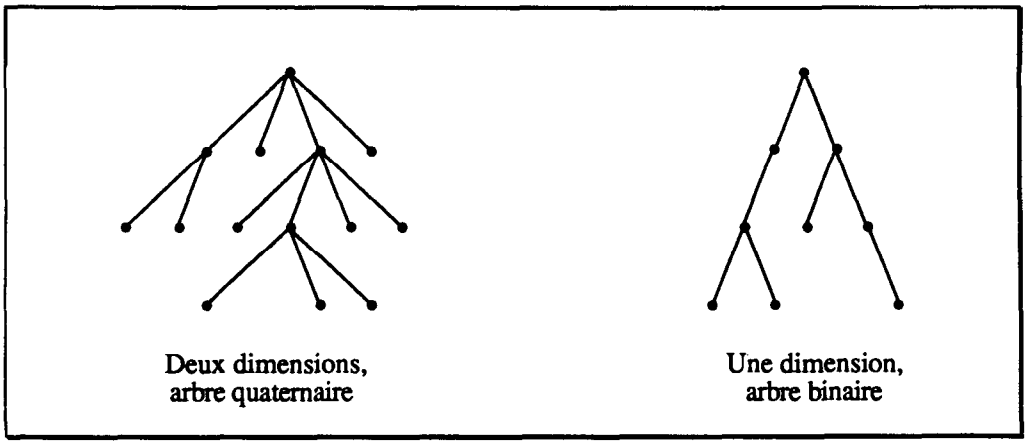


Fig. 1.14 – Exemples d'arbres hyperquaternaires.

Définition: Un *arbre hyperquaternaire* de points *plein*, à d dimensions, est un *arbre hyperquaternaire* de points dont tous les nœuds ont aucun ou 2^d enfants.

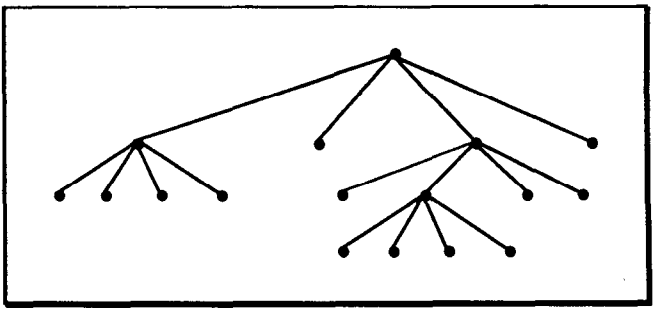


Fig. 1.15 – Arbre plein.

Définition: Un *arbre hyperquaternaire* de points *complet* est un *arbre hyperquaternaire* de points dont les feuilles sont sur au plus deux niveaux adjacents; les feuilles du dernier niveau occupent les positions les plus à gauche.

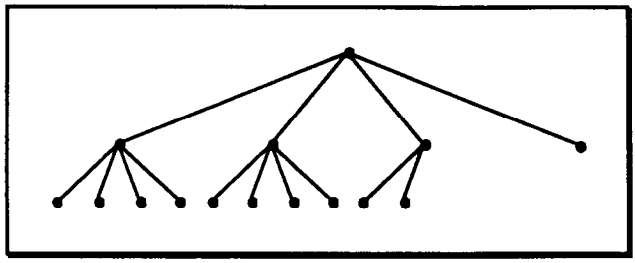


Fig. 1.16 – Arbre complet.

Définition: La *profondeur* d'un nœud est la longueur du chemin de la racine à ce nœud.

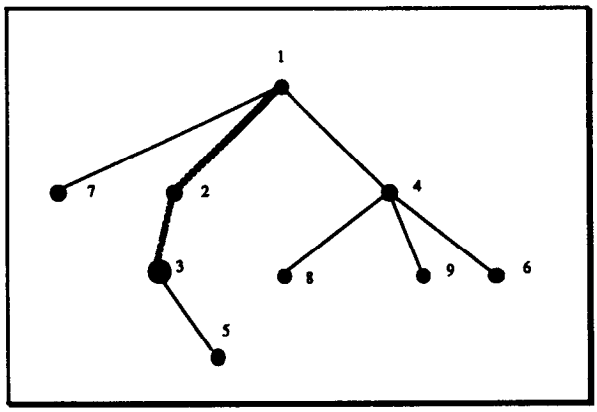


Fig. 1.17 – Profondeur d'un nœud

Dans la figure précédente, la profondeur du nœud 3 est 2.

Définition: La *hauteur* d'un arbre est la longueur du plus long chemin de la racine à chacun de ses nœuds.

Définition: Tous les nœuds d'un arbre hyperquaternaire de points sont appelés *nœuds internes*. Si on ajoute des nœuds partout où il y a un arbre vide, on appelle ces nœuds, *nœuds externes*.

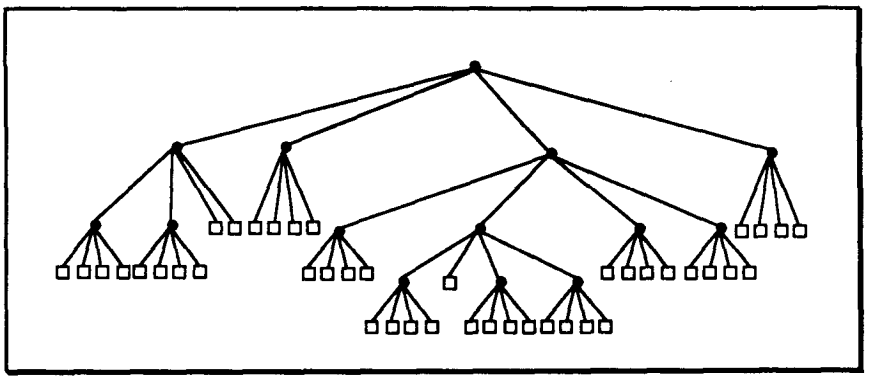


Fig. 1.18 – Arbre quaternaire avec nœuds externes.

Définition: On appelle *longueur du chemin interne* d'un arbre la somme de la profondeur de tous les nœuds internes de l'arbre.

Définition: On appelle *longueur du chemin externe* d'un arbre la somme de la profondeur de tous les nœuds externes de l'arbre.

Exemple: La longueur de chemin interne de l'arbre de la figure 1.18 est de 25 tandis que la longueur de chemin externe est de 131.

Les arbres binaires de fouille

Si on considère l'arbre hyperquaternaire de points à une dimension et que l'on exa-

mine le modèle uniforme, c'est-à-dire que toutes les permutations de n points sont équiprobables, on a les résultats suivants:

Le nombre d'arbres différents que l'on obtient est de $\frac{1}{n+1}\binom{2n}{n}$, le nombre de permutations étant de $n!$, on en conclut que chaque arbre n'a pas le même poids. Le nombre de nœuds externes est de $n+1$. La hauteur minimale d'un arbre binaire est de $\lfloor \log_2 n \rfloor$, tandis que la hauteur maximale est de $n-1$. Si on note par E la longueur de chemin externe et par I, la longueur de chemin interne, alors l'équation qui relie la longueur de chemin externe et la longueur de chemin interne est $E = I + 2n$. La longueur minimale du chemin interne nous est donnée par $(n+1)\,q - 2^{q+1} + 2$, où $q = \lfloor \log_2 n \rfloor$ et la longueur maximale du chemin interne par $n(n-1)/2$, pour $n \geq 1$. La variance de la longueur de chemin interne et de $7n^2 - 4(n+1)^2 H_n^{(2)} - 2(n+1)H_n + 13n$, pour $n \geq 1$ (Knuth 1973), où $H_n^{(j)} = \sum_{i=1}^{n} 1/i^j$, où $H_n = H_n^{(1)}$, pour $n \geq 1$. Le nombre moyen de comparaisons effectuées lors d'une fouille fructueuse est égal à $2(1+1/n)H_n - 3$ et la variance du nombre de comparaisons lors d'une recherche fructueuse est égale à $(2+10/n)H_n - 4(1+1/n)(H_n^{(2)} + H_n^2/n) + 4$, pour $n \geq 1$ (Knuth 1973). Maintenant, le nombre moyen de comparaisons effectuées lors d'une fouille sans succès est de $2(H_{n+1} - 1)$, pour $n \geq 1$. Des dernières expressions, il découle aisément que la profondeur moyenne des nœuds dans un arbre binaire de fouille est de $2(1+1/n)H_n - 4$, que la profondeur moyenne du dernier nœud inséré est de $2(H_n - 1)$ et que la variance de la profondeur du dernier nœud est de $2H_n - 4H_n^{(2)} + 2$, pour $n \geq 1$. L'espérance du nombre de nœuds de degré i, $i = 1, 2, 3$ est égale à $\frac{n}{3} + O(1)$ (Mahmoud, 1986). La quantité $D_n / \log n$, où D_n dénote dans cette section la profondeur du dernier nœud ajouté dans un arbre binaire de n nœuds, tend en probabilité vers 2 lorsque $n \to \infty$ et aussi, $E(D_n) \sim E(A_n) \sim 2\log n$. Ces deux derniers résultats ainsi que les différentes propriétés des arbres binaires de recherche peuvent être retrouvés dans différents papiers dont ceux de Lynch (1965), Knuth (1973), Robson (1979), Sedgewick (1983), Pittel (1984), Mahmoud et Pittel (1984), Brown et Shubert

(1984) et Devroye (1986,1987, 1988).

Les arbres hyperquaternaires

Considérons maintenant la généralisation de l'arbre binaire de fouille à d dimensions, l'arbre hyperquaternaire. Si on examine le modèle uniforme où l'ensemble des $i^{\text{ièmes}}$ coordonnées de chacun des n points est une permutation aléatoire de l'ensemble $\{1, 2, \ldots, n\}$, pour $1 \le i \le d$, où les d permutations sont indépendantes, on a les résultats suivants:

Le nombre d'arbres différents que l'on a est de $\binom{2^d n}{n} / ((2^d - 1)n + 1)$ (Puech et Yahia, 1985, Flajolet (Börger), 1988), tandis que le nombre d'ensemble de points différents est de $(n!)^d$, comme pour les arbres binaires, tous les arbres ne sont pas équiprobables. Le nombre de nœuds externes est de $(2^d - 1)n + 1$ (Gonnet, 1984). La hauteur minimale d'un arbre hyperquaternaire est $\lfloor \log_{2^d} n(2^d - 1) \rfloor$, tandis que la hauteur maximale est de $n - 1$. La hauteur est en probabilité asymptotique à $\frac{c}{d} \log n$, où $c = 4{,}31107\ldots$ est l'unique solution supérieure à 2 de l'équation $c \log(2e/c) = 1$ (Devroye, 1987). Si on note par E la longueur de chemin externe et par I, la longueur de chemin interne, alors l'équation qui relie la longueur de chemin externe et la longueur de chemin interne est $E = n2^d + I(2^d - 1)$. La longueur minimale du chemin interne nous est donnée par l'expression $\left(n + 1/(2^d - 1)\right)q - (2^{d(q+1)} - 2^d)/(2^d - 1)^2$, où $q = \lfloor \log_{2^d} n(2^d - 1) \rfloor$ et la longueur maximale du chemin interne par $n(n-1)/2$. Le nombre moyen de comparaisons effectuées lors d'une fouille fructueuse, dans un arbre hyperquaternaire de dimension deux, est égal à $(n + 1/3)H_n / n - (n+1)/6n$ pour $n \ge 1$. Maintenant, le nombre moyen de comparaisons effectuées lors d'une fouille sans succès, dans un arbre quaternaire, est de $H_n - 1/6 + 1/(3(n+1))$, pour $n \ge 2$. Des dernières expressions, il découle aisément que la profondeur moyenne des nœuds dans un arbre

hyperquaternaire de dimension deux est de $(n+1/3)H_n/n-(7n+1)/6n$, pour $n \geq 1$, que la profondeur moyenne du dernier nœud inséré est de $H_n-1/6-2/3n$, pour $n \geq 2$ et que la variance de la profondeur du dernier nœud est de $H_n^{(2)}+H_n/2+5/9n-4/9n^2-13/6$, pour $n \geq 2$. Le nombre moyen de feuilles dans un arbre hyperquaternaire à deux dimensions est de $8H_n^{(2)}(3n+1)+11-39n-4/n$, pour $n \geq 2$, ce qui implique que la proportion de feuilles dans un arbre hyperquaternaire à deux dimensions est asymptotiquement égale à $4\pi^2-39 \approx 0{,}48$. Le nombre moyen de nœuds à un enfant dans un arbre hyperquaternaire à deux dimensions est de $4/n^2 + 16/n - 1171/27 + 2393n/9 + 4(6 - 1/n)H_n - 52(3n + 1)H_n^{(2)} - 8(3n + 1)H_n^{(3)} + 8(3n + 1)\sum_{i=4}^{n} H_i/i^2$, pour $n \geq 2$, ce qui implique que la proportion de nœuds à un enfant dans un arbre hyperquaternaire à deux dimensions est asymptotiquement égale à 0.239651196.... Les deux derniers résultats sont démontrés au chapitre 4. La quantité $D_n/\log n$ tend en probabilité vers $2/d$ lorsque $n \to \infty$. Aussi, $E(D_n) \sim E(A_n) \sim \frac{2}{d}\log n$ lorsque $n \to \infty$. Ces deux derniers résultats ont été démontrés dans Devroye et Laforest (1989), ce dont fait l'objet le troisième chapitre de la présente thèse.

Chapitre 2

Étude théorique des arbres hyperquaternaires de points

2.1 L'arbre hyperquaternaire de points, aléatoire.

Nous étudierons ici l'allure de la structure de données permettant de garder en mémoire un nuage de points ayant certaines propriétés, l'arbre hyperquaternaire. Ici, nous nous attarderons au cas où les points sont des vecteurs de $\mathbf{R}^d$, identiquement distribués et indépendants, tout comme chaque composante. En fait, ce modèle se ramène au cas où chaque composante est uniforme, indépendante des autres. En effet, puisque la structure dépend de la position relative des points entre eux plutôt que de la valeur comme telle des coordonnées, la fonction F qui régit la coordonnée peut être transformée de sorte à agir comme une loi uniforme, sans que cela ne change l'ordre des points entre eux, en fonction de cette coordonnée. Ce raisonnement s'appliquant à toutes les coordonnées, le problème uniforme sera donc étudié.

Sans perte de généralité, on peut même considérer sur un même pied, à titre équivalent, le modèle discret uniforme où l'ensemble des $i^{\text{ièmes}}$ coordonnées de chacun des n points est une permutation aléatoire de l'ensemble $\{1, 2, \ldots, n\}$ et ce pour $1 \leq i \leq d$, ces

d permutations étant indépendantes entre elles. En effet, pour la même raison invoquée au paragraphe précédent, la position relative des points entre eux étant l'unique aspect influençant la structure de l'arbre, nous n'avons qu'à transformer les points de la façon suivante: remplacer chacune des coordonnées par son rang à l'intérieur de l'ensemble des n coordonnées.

Ce dernier modèle nous permet de pouvoir générer tous les arbres possibles sur ordinateur. Cependant, pour toute l'étude théorique qui fait l'objet de cette thèse, on considérera, sans perte de généralité $[0,1]^d$ plutôt que $\mathbf{R}^d$. Nous disposons donc de n points dans $[0,1]^d$, que l'on notera $X_1, X_2, \ldots, X_n$.

Définition: Un *arbre hyperquaternaire de points, aléatoire*, est l'arbre obtenu d'un nuage de points ordonnés, indépendants, identiquement distribués et uniformes $[0,1]^d$, où la racine est déterminée par le premier point, séparant ainsi le nuage en 2^d quadrants ordonnés, déterminant par le fait même 2^d sous-arbres, étant eux-mêmes des arbres hyperquaternaires de points.

L'arbre hyperquaternaire considéré est en fait une extension de l'arbre binaire de fouille (d=1), largement étudié (voir Knuth (1981)). Il s'agit de l'arbre quaternaire lorsque $d = 2$ et nous utiliserons le terme hyperquaternaire dans le cas général.

On construit l'arbre au fur et à mesure que l'on ajoute des points. Le premier, la racine de l'arbre, sépare $[0,1]^d$ en 2^d régions appelées hyperquadrants et déterminés par les d hyperplans perpendiculaires entre eux ayant X_1 comme intersection et parallèles à chacun des axes. Ces 2^d régions définissent autant de sous-arbres associés à la racine représentée par X_1. On ajoute ensuite X_2 qui appartiendra à l'un des 2^d hyperquadrants et qui sera à son tour découpé en 2^d autres hyperquadrants. Le point X_2 sera la racine du sous-arbre

correspondant à l'hyperquadrant dans lequel X_2 se trouve. On continue de la même manière jusqu'à épuisement des n points. La figure 2.1 en illustre le principe pour $d = 2$ et $n = 5$.

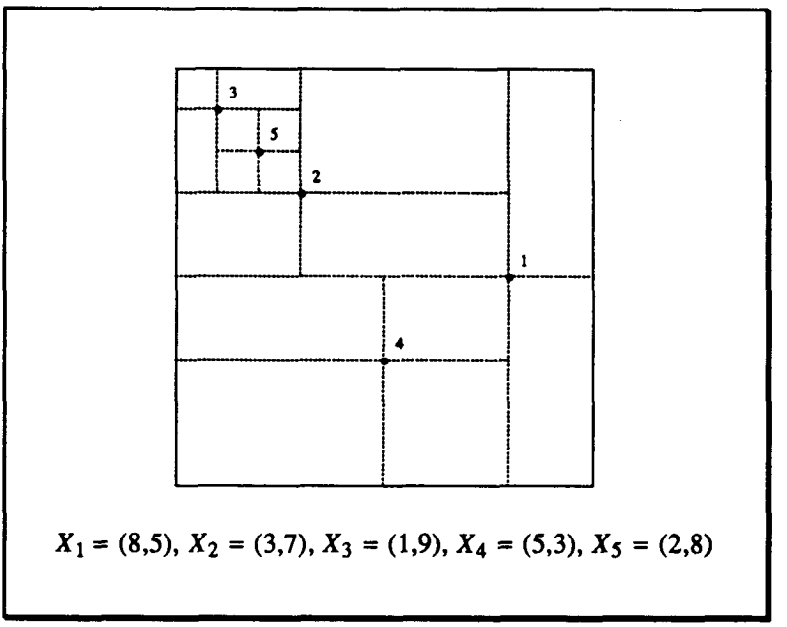


Fig. 2.1 – Arbre quaternaire de points.

Nous nous intéresserons à la profondeur D_n du $n^{\text{ième}}$ nœud inséré dans l'arbre hyperquaternaire. La variable D_n est importante puisqu'elle est reliée au temps de fouille sans succès dans un arbre. De cette quantité, on peut directement avoir la profondeur moyenne d'un arbre ainsi que sa hauteur qui sont données respectivement par les expressions suivantes:

$$A_n = \frac{1}{n}\sum_{i=1}^{n} D_i \quad \text{et} \quad D_n^* = \max_{1\le i\le n} D_i.$$

Devroye (1987) a déjà démontré le théorème suivant.

Théorème 2.1 (Devroye, 1987)

Soit D_n^*, la hauteur d'un arbre hyperquaternaire de points à d dimensions, construit à partir de permutations aléatoires. Alors,

$$\frac{D_n^*}{\log n} \to \frac{c}{d}, \text{ en probabilité,}$$

où c = 4.31107... est l'unique solution plus grande que 2 de l'équation

$$c \log\left(\frac{2e}{c}\right) = 1.$$

Dans la section qui suit, nous étudierons en détail D_n.

2.2 Étude de D_n

Soient $\mu_{n,1}$ et $\mu_{n,2}$, les deux premiers moments non centrés de la variable aléatoire D_n, et $H_n^{(j)} = \sum_{i=1}^{n} \frac{1}{i^j}$, où $H_n^{(1)} \equiv H_n$. Nous allons montrer dans cette section le résultat suivant.

Théorème 2.2

Pour $d = 2$, $\mu_{1,m} = 0$, pour $m > 0$,

$$E(D_n) = \mu_{n,1} = H_n - \frac{1}{6} - \frac{2}{3n},$$

$$E(D_n^2) = \mu_{n,2} = H_n^2 + H_n^{(2)} + \frac{H_n}{6} - \frac{4H_n}{3n} + \frac{7}{9n} - \frac{77}{36},$$

$$\text{et } \mathrm{Var}(D_n) = H_n^{(2)} + \frac{H_n}{2} + \frac{5}{9n} - \frac{4}{9n^2} - \frac{13}{6}, \quad n \geq 2.$$

Définissons $p_{n,l} = P(D_n = l)$. On a que $p_{1,0} = 1$, i.e. la racine est à une profondeur de 0. On a aussi que $p_{i,0} = 0$ pour $i \neq 1$ et que $p_{i,l} = 0$ pour $l \geq i$.

Nous verrons qu'il est difficile d'expliciter les valeurs de $p_{n,l}$, mais que nous n'avons pas besoin de leur valeur explicite pour évaluer les divers moments de D_n. Nous aurons besoin des résultats intermédiaires suivants.

Lemme 2.3

Soient $U_1, U_2, \ldots, U_n$, n variables aléatoires uniformes indépendantes. Soit

$$T = \prod_{i=1}^{n} U_i .$$

Alors la fonction de densité de T est donnée par

$$f_n(t) = \frac{(-\log t)^{n-1}}{(n-1)!}, \quad 0 \leq t \leq 1.$$

Preuve du lemme 2.3.

Voir Devroye (1986, page 24). ■

Lemme 2.4

Soient N_α, $1 \le \alpha \le 2^d$, la cardinalité de chacun des 2^d sous-arbres déterminés par X_1, aléatoire. Alors,

a) $P(N_\alpha = i) = \binom{n-1}{i} \sum_{j=0}^{n-1-i} \frac{(-1)^j \binom{n-1-i}{j}}{(i+j+1)^d}, \quad 0 \le i < n.$

Pour $d = 2$, où les quadrants 1 et 2 partagent un côté, ainsi que les quadrants 1 et 3,

b) $P(N_1 = i, N_2 = j) = \frac{1}{i+j+1} \frac{1}{n}, \quad 0 \le i+j < n, i \ge 0, j \ge 0.$

c) $P(N_1 = i, N_2 = j, N_3 = k, N_4 = n-1-i-j-k) = \frac{\binom{n-1}{i,j,k,n-1-i-j-k}}{n^2 \binom{n-1}{i+j}\binom{n-1}{i+k}},$

$$0 \le i+j+k < n, i \ge 0, j \ge 0, k \ge 0.$$

Preuve du lemme 2.4.

a) Sans perte de généralité, supposons que N_1 correspond à l'hyperquadrant dont un des coins est (0, 0, ..., 0). Nous avons que

$$P(N_1 = i \mid X_1) = \binom{n-1}{i} t^i (1-t)^{n-1-i}$$

où t est l'hypervolume de l'hyperquadrant correspondant à N_1. Cette variable aléatoire est le produit de d uniformes indépendantes dont la fonction de densité nous est donnée par le

lemme précédent. Nous avons alors

$$P(N_1 = i) = \int_0^1 P(N_1 = i \mid X_1) f_d(t) dt$$

$$= \int_0^1 \binom{n-1}{i} t^i (1-t)^{n-1-i} (-1)^{d-1} \frac{(\log t)^{d-1}}{(d-1)!} dt$$

$$= \frac{(-1)^{d-1}\binom{n-1}{i}}{(d-1)!} \sum_{j=0}^{n-1-i} (-1)^j \binom{n-1-i}{j} \int_0^1 t^{i+j} (\log t)^{d-1} dt.$$

On a (Beyer, 1984) que

$$\int_0^1 t^a (\log t)^b dt = \frac{(-1)^b b!}{(a+1)^{b+1}}.$$

Donc,

$$P(N_1 = i) = \frac{(-1)^{d-1}\binom{n-1}{i}}{(d-1)!} \sum_{j=0}^{n-1-i} (-1)^j \binom{n-1-i}{j} \frac{(-1)^{d-1}(d-1)!}{(i+j+1)^d}$$

$$= \binom{n-1}{i} \sum_{j=0}^{n-1-i} \frac{(-1)^j \binom{n-1-i}{j}}{(i+j+1)^d}.$$

Par symétrie, on a le résultat désiré pour chacun des autres hyperquadrants.

b) Pour cette probabilité, la preuve est simple:

$$P(N_1 = i, N_2 = j) = P(N_1 = i, N_1 + N_2 = i + j)$$
$$= P(N_1 = i \mid N_1 + N_2 = i + j) P(N_1 + N_2 = i + j) = \frac{1}{i+j+1} \frac{1}{n}.$$

c) Si on conditionne sur le premier point X_1, constitué de deux uniformes [0,1] indépendantes, la probabilité cherchée est une multinomiale. Sans condition, on obtient

$$P(N_1 = i, N_2 = j, N_3 = k, N_4 = n-1-i-j-k)$$

$$= \binom{n-1}{i,j,k,n-1-i-j-k} \int_0^1 \int_0^1 (uv)^i (u(1-v))^j ((1-u)v)^k ((1-u)(1-v))^{n-1-i-j-k} \, du\, dv$$

$$= \binom{n-1}{i,j,k,n-1-i-j-k} \int_0^1 \int_0^1 u^{i+j} v^{i+k} (1-u)^{n-1-i-j} (1-v)^{n-1-i-k} \, du\, dv.$$

La dernière expression contient deux fonctions bêta. Cette dernière est donc égale à

$$= \binom{n-1}{i,j,k,n-1-i-j-k} \frac{(i+j)!(n-1-i-j)!}{n!} \frac{(i+k)!(n-1-i-k)!}{n!}$$

$$= \frac{\binom{n-1}{i,j,k,n-1-i-j-k}}{n^2 \binom{n-1}{i+j} \binom{n-1}{i+k}}.$$

■

On connaissait déjà le fait que, pour un arbre binaire, le nombre de nœuds dans le sous-arbre gauche est distribué uniformément. Ce résultat se retrouve en utilisant le lemme précédent avec $d = 1$, ce qui donne $P(N_j = i) = 1/n$, $0 \le i < n, j = 1, 2$.

Nous verrons plus loin qu'une formule de récurrence pour calculer $P(N_\alpha = i)$ est plus utile. Puisque cette récurrence est fonction de la dimension d, nous introduisons ici l'indice d dans la notation. Donc, nous noterons $P(N_\alpha = i)$ par $P(N_{d,\alpha} = i)$. Nous utiliserons plus loin le lemme suivant:

Lemme 2.5

$$P(N_{d,\alpha} = i) = \sum_{k=i}^{n-1} \frac{P(N_{d-1,\alpha} = k)}{k+1}, \quad 0 \le i < n, d \ge 1, \ 1 \le \alpha \le 2^d,$$

où $P(N_{0,\alpha} = i) = \delta_{n-1,i}$, $0 \le i < n$.

Preuve du lemme 2.5.

Nous nous préoccuperons de $P(N_{d,1} = i)$, les autres probabilités étant les mêmes, s'obtiennent par symétrie. Si on projette le nuage de points uniformes de $[0,1]^d$ sur $[0,1]^{d-1}$, le nuage obtenu demeure uniforme et se ramène donc au même problème mais avec une dimension en moins. De plus, les points projetés sur l'hyperquadrant correspondant à $N_{d-1,1}$ proviennent de deux hyperquadrants dont celui correspondant à $N_{d,1}$. Donc, si l'on fixe le nombre de points des deux hyperquadrants au total à j, la probabilité que $N_{d,1}$ en contienne i est égale à $1/(j+1)$. On a alors que

$$\begin{aligned} P(N_{d,\alpha} = i) &= \sum_{j=i}^{n-1} P(N_{d,1} = i \,|\, \text{le nombre de points projetés sur } N_{d-1,1} = j) \\ &\qquad \times P(\text{le nombre de points projetés sur } N_{d-1,1} = j) \\ &= \sum_{j=i}^{n-1} \frac{1}{j+1} P\,(N_{d-1,1}=j) \ . \end{aligned}$$

■

Corollaire 2.6

$$P(N_{2,\alpha} = i) = \frac{H_n - H_i}{n}, \ 0 \le i < n, \ 1 \le \alpha \le 4,$$

$$\text{où } H_n = 1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n}.$$

Preuve du corollaire 2.6.

Il suffit d'appliquer le lemme précédent avec $d = 2$. En effet,

$$P(N_{2,\alpha} = i) = \sum_{k=i}^{n-1} \frac{P(N_{1,\alpha} = k)}{k+1} = \sum_{k=i}^{n-1} \frac{1}{n} \frac{1}{k+1} = \frac{1}{n}(H_n - H_i).$$

■

En calculant les probabilités dont il est question ici pour quelques valeurs de d, on se rend compte que celles-ci ont une allure particulière. Le corollaire suivant en exhibe quelques-unes.

Corollaire 2.7

Soit $\Delta_j = H_n^{(j)} - H_i^{(j)}, j \geq 1,\ 0 \leq i < n$, où

$$H_n^{(j)} = \sum_{i=1}^{n} \frac{1}{i^j},\ H_0^{(j)} \triangleq 0,\ H_n \triangleq H_n^{(1)},$$

alors,

$$P(N_{1,\alpha} = i) = \frac{1}{n},$$

$$P(N_{2,\alpha} = i) = \frac{\Delta_1}{n},$$

$$P(N_{3,\alpha} = i) = \frac{\Delta_1^2 + \Delta_2}{2n},$$

$$P(N_{4,\alpha} = i) = \frac{\Delta_1^3 + 3\Delta_1\Delta_2 + 2\Delta_3}{6n},$$

$$P(N_{5,\alpha} = i) = \frac{\Delta_1^4 + 6\Delta_1^2\Delta_2 + 8\Delta_1\Delta_3 + 3\Delta_2^2 + 6\Delta_4}{24n}.$$

Preuve du corollaire 2.7.

Il suffit, pour chacune des probabilités, d'utiliser le lemme 2.5. Le reste est ardu et mécanique. ■

Du dernier corollaire, on remarque l'apparition des nombres de Stirling de première

espèce, notés par $\begin{bmatrix} n \\ i \end{bmatrix}$. Voici les quelques premières lignes. (Voir Knuth (1973))

$n \backslash i$	1	2	3	4	5
1	1				
2	1	1			
3	2	3	1		
4	6	11	6	1	
5	24	50	35	10	1
⋮	⋮	⋮	⋮	⋮	⋮

Si on regarde $P(N_{5,\alpha} = i)$ et qu'on écrit la liste de toutes les façons d'obtenir le nombre 5–1 = 4, on a

1+1+1+1	$\rightarrow \Delta_1^4$	$1 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$
1+1+2	$\rightarrow 6\Delta_1^2\Delta_2$	$6 = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$
1+3 et 2+2	$\rightarrow 8\Delta_1\Delta_3 + 3\Delta_2^2$	$8+3 = 11 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$
4	$\rightarrow 6\Delta_4$	$6 = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$.

Nous allons maintenant nous attarder à $p_{n,l}$, la probabilité que le $n^{\text{ième}}$ nœud inséré dans l'arbre soit à une profondeur de l, comme indiqué au début de la section. Pour ne pas alourdir la notation inutilement, nous n'ajouterons d'indice d ou n que si cela s'avère nécessaire. Dans ce même ordre d'idée, posons $P(N_{d,\alpha} = i) = P_i$.

Lemme 2.8

$$p_{n,l} = \frac{2^d}{n-1} \sum_{i=l}^{n-1} iP_i p_{i,l-1}, \quad 1 \le l < n, \quad n \ge 2.$$

Preuve du lemme 2.8.

Le dernier point inséré dans l'arbre doit nécessairement tomber dans un des 2^d sous-arbres. Si le point est au niveau l dans l'arbre, alors il sera au niveau $l-1$ dans le sous-arbre. Le sous-arbre peut contenir i points, $0 \le i < n$; la probabilité qu'il en contienne i étant de P_i. La probabilité que le dernier point soit dans le sous-arbre est de $i/(n-1)$, les n points étant fixés. On peut donc écrire que:

$$p_{n,l} = \sum_{j=1}^{2^d} \sum_{i=0}^{n-1} \frac{i}{n-1} P_i p_{i,l-1} = \frac{2^d}{n-1} \sum_{i=1}^{n-1} i P_i p_{i,l-1}$$

puisque $p_{i,j} = 0$ pour $j \ge i$. ■

Nous utiliserons ce dernier lemme pour trouver la fonction génératrice des moments de D_n. Définissons

$$\phi_n(t) = E\,(e^{tD_n})$$

et

$$\mu_{n,m} = E\,(D_n^m).$$

Lemme 2.9

$$\phi_n(t) = \frac{2^d e^t}{n-1} \sum_{i=1}^{n-1} i P_i \phi_i(t), \quad n>1, \quad \phi_1(t) = 1,$$

et, pour $m > 0$,

$$\mu_{n,m} = \frac{2^d}{n-1} \sum_{i=1}^{n-1} i P_i \sum_{j=0}^{m} \binom{m}{j} \mu_{i,j}, \quad n > 1,$$

où $\mu_{n,0} = 1$ pour $n \ge 1$ et $\mu_{1,m} = 0$ pour $m > 0$.

Preuve du lemme 2.9.

Pour $\mu_{n,0}$, il suffit de calculer $E(D_n^0)$ qui est égal à 1. Pour $\mu_{1,m}$, on se sert du fait que la racine d'un arbre est toujours au niveau 0. Maintenant, en utilisant le lemme 2.8 on a que

$$\begin{aligned}\phi_n(t) &= \sum_{l=1}^{n-1} p_{n,l} e^{tl} = \sum_{l=1}^{n-1} \frac{2^d}{n-1} \sum_{i=l}^{n-1} iP_i p_{i,l-1} e^{tl} \\ &= \frac{2^d}{n-1} \sum_{i=1}^{n-1} iP_i \sum_{l=1}^{i} p_{i,l-1} e^{tl} \\ &= \frac{2^d e^t}{n-1} \sum_{i=1}^{n-1} iP_i \phi_i(t).\end{aligned}$$

Ceci prouve donc la première récurrence du lemme. Pour la deuxième partie, posons

$$f_n(t) = \sum_{i=1}^{n-1} iP_i \phi_i(t) .$$

On obtient alors

$$\phi_n(t) = \frac{2^d e^t}{n-1} f_n(t) \qquad \text{et,}$$

$$\phi_n^{(m)}(t) = \frac{2^d e^t}{n-1} \sum_{j=0}^{m} \binom{m}{j} f_n^{(m)}(t),$$

donc,

$$\mu_{n,m} = \phi_n^{(m)}(0) = \frac{2^d}{n-1} \sum_{j=0}^{m} \binom{m}{j} \sum_{i=1}^{n-1} iP_i \mu_{i,j}. \qquad \blacksquare$$

La deuxième récurrence du dernier lemme est en fait une double récurrence qui en fait une expression pas très commode à manipuler. En la travaillant un peu, on peut en faire une récurrence simple, en fonction des moments précédents, la récurrence se faisant sur l'indice m. Ceci donne lieu au corollaire suivant.

Corollaire 2.10

$$\mu_{n,m} = \sum_{j=0}^{m-1}\binom{m}{j}\mu_{n,j}(-1)^{m-1-j} + \frac{2^d}{n-1}\sum_{i=1}^{n-1} iP_i\mu_{i,m},\quad m>0,\ n>1.$$

Preuve du corollaire 2.10.

Nous utiliserons une preuve par induction sur m. Pour $m = 1$, en utilisant le lemme 2.9, on obtient la même chose que l'énoncé du corollaire, c'est-à-dire,

$$\mu_{n,1} = 1 + \frac{2^d}{n-1}\sum_{i=1}^{n-1} iP_i\mu_{i,1}\,.$$

L'hypothèse d'induction nous dit que l'expression suivante est vraie:

$$\mu_{n,m-1} = \sum_{j=0}^{m-2}\binom{m-1}{j}\mu_{n,j}(-1)^{m-2-j} + \frac{2^d}{n-1}\sum_{i=1}^{n-1} iP_i\mu_{i,m-1}.$$

Ce qui nous permet d'écrire que

$$\sum_{i=1}^{n-1} iP_i\mu_{i,m-1} = \frac{n-1}{2^d}\left[\mu_{n,m-1} - \sum_{j=0}^{m-2}\binom{m-1}{j}\mu_{n,j}(-1)^{m-2-j}\right] \qquad (2.1)$$

que nous utiliserons plus loin. Maintenant, utilisant le lemme précédent, on a que

$$\mu_{n,m} = \frac{2^d}{n-1}\sum_{j=0}^{m}\binom{m}{j}\sum_{i=1}^{n-1} iP_i\mu_{i,j}$$

$$= \frac{2^d}{n-1}\left[\sum_{j=0}^{m-1}\binom{m}{j}\sum_{i=1}^{n-1} iP_i\mu_{i,j} + \sum_{i=1}^{n-1} iP_i\mu_{i,m}\right].$$

En utilisant l'expression (2.1) ci-haut, on obtient

$$\mu_{n,m} = \sum_{j=0}^{m-1}\binom{m}{j}\left[\mu_{n,j} - \sum_{k=0}^{j-1}\binom{j}{k}\mu_{n,k}(-1)^{j-1-k}\right] + \frac{2^d}{n-1}\sum_{i=1}^{n-1} iP_i\mu_{i,m}$$

$$= \sum_{j=0}^{m-1}\binom{m}{j}\mu_{n,j} - \sum_{k=0}^{m-1}\mu_{n,k}\sum_{j=k+1}^{m-1}\binom{m}{j}\binom{j}{k}(-1)^{j-1-k} + \frac{2^d}{n-1}\sum_{i=1}^{n-1} iP_i\mu_{i,m}. \qquad (2.2)$$

Les deux premiers termes de la dernière équation peuvent s'écrire comme suit.

$$m\mu_{n,m-1} + \sum_{k=0}^{m-2}\mu_{n,k}\left[\binom{m}{k} - \sum_{j=k+1}^{m-1}\binom{m}{j}\binom{j}{k}(-1)^{j-1-k}\right]$$

$$= m\mu_{n,m-1} + \sum_{k=0}^{m-2}\mu_{n,k}\sum_{j=k}^{m-1}\binom{m}{j}\binom{j}{k}(-1)^{j-k}$$

$$= m\mu_{n,m-1} + \sum_{k=0}^{m-2}\binom{m}{k}\mu_{n,k}\sum_{j=k}^{m-1}\binom{m-k}{m-j}(-1)^{j-k}$$

$$= m\mu_{n,m-1} + \sum_{k=0}^{m-2}\binom{m}{k}\mu_{n,k}\sum_{j=0}^{m-1-k}\binom{m-k}{j}(-1)^{j}$$

$$= m\mu_{n,m-1} + \sum_{k=0}^{m-2}\binom{m}{k}\mu_{n,k}(-1)^{m-1-k}$$

$$= \sum_{k=0}^{m-1}\binom{m}{k}\mu_{n,k}(-1)^{m-1-k}.$$

En remplaçant, par cette dernière expression, les deux premiers termes de (2.2) on obtient l'expression du corollaire. ■

Nous allons maintenant nous attarder au cas où $d = 2$, c'est-à-dire lorsque le nuage de points se situe dans $[0,1]^2$. On a déjà calculé (voir corollaire 2.6) que

$$P(N_{2,\alpha} = i) = \frac{H_n - H_i}{n},\ 0 \le i < n,\ 1 \le \alpha \le 4.$$

À partir de ce résultat découle directement, pour $d = 2$, le corollaire suivant.

Corollaire 2.11

$$p_{n,l} = \frac{4}{n(n-1)} \sum_{i=l}^{n-1} i(H_n - H_i) p_{i,l-1}, \quad 1 \le l < n, \ n > 1.$$

Preuve du corollaire 2.11.

Ceci est une conséquence directe du lemme 2.8, lorsque $d = 2$. ■

Bien que cette dernière expression soit élégante et que nous nous en servions plus loin, une tentative de la simplifier à défaut de pouvoir la résoudre donne quelques résultats non dénués d'intérêt. Un premier calcul a donné lieu au corollaire suivant.

Corollaire 2.12

$$p_{n,l} = \frac{4}{n^2} p_{n-1,l-1} + \frac{(2n-1)(n-2)}{n^2} p_{n-1,l} - \frac{(n-2)(n-3)}{n^2} p_{n-2,l}, \quad 1 \le l < n.$$

Preuve du corollaire 2.12.

Il suffit de transformer l'expression du corollaire 2.11 dont nous pouvons conclure que

$$\sum_{i=l}^{n-2} i(H_{n-1} - H_i) p_{i,l-1} = \frac{(n-2)(n-1)}{4} p_{n-1,l}.$$

En utilisant le fait que $H_n = H_{n-1} + 1/n$, on a que

$$\sum_{i=l}^{n-2} i(H_n - H_i)p_{i,l-1} = \frac{1}{n}\sum_{i=l}^{n-2} ip_{i,l-1} + \frac{(n-1)(n-2)}{4} p_{n-1,l}$$

Donc, l'expression du corollaire 2.11 devient

$$\begin{aligned} p_{n,l} &= \frac{4}{n^2} p_{n-1,l-1} + \frac{4}{n^2(n-1)} \sum_{i=l}^{n-2} ip_{i,l-1} + \frac{n-2}{n} p_{n-1,l} \\ &= \frac{4}{n^2(n-1)} \sum_{i=l}^{n-1} ip_{i,l-1} + \frac{n-2}{n} p_{n-1,l} \end{aligned}$$

et de ceci on conclut que

$$\sum_{i=l}^{n-2} iP_{i,l-1} = \left[p_{n-1,l} - \frac{n-3}{n-1} p_{n-2,l}\right] \frac{(n-1)^2(n-2)}{4},$$

d'où

$$p_{n,l} = \frac{4}{n^2} p_{n-1,l-1} + \frac{(n-1)(n-2)}{n^2}\left[p_{n-1,l} - \frac{n-3}{n-1} p_{n-2,l}\right] + \frac{n-2}{n} p_{n-1,l},$$

et on obtient l'expression voulue en regroupant correctement les termes. ■

Attardons-nous maintenant au calcul des divers moments de D_n, dans le cas planaire.

Corollaire 2.13

$$\mu_{n,m} = \sum_{j=0}^{m-1} \binom{m}{j} \mu_{n,j} (-1)^{m-1-j} + \frac{4}{n(n-1)} \sum_{i=1}^{n-1} i(H_n - H_i)\mu_{i,m}, \quad m > 0, n > 1, d = 2.$$

Preuve du corollaire 2.13.

Il suffit de placer $d = 2$ dans l'expression du corollaire 2.10. ■

Comme on l'avait remarqué dans le cas général, nous avons une récurrence pour calculer les $\mu_{n,m}$, en fonction des moments d'ordre inférieur. Nous avons en particulier que:

$$\mu_{n,0} = 1,$$

$$\mu_{n,1} = 1 + \frac{4}{n(n-1)} \sum_{i=1}^{n-1} i(H_n - H_i)\mu_{i,1},$$

$$\mu_{n,2} = 2\mu_{n,1} - 1 + \frac{4}{n(n-1)} \sum_{i=1}^{n-1} i(H_n - H_i)\mu_{i,2}.$$

La récurrence du corollaire 2.13 est de la forme:

$$x_n = a_n + \frac{4}{n(n-1)} \sum_{i=1}^{n-1} i(H_n - H_i)x_i, \quad n > 1, \quad x_1 = 0. \tag{2.3}$$

Ce qui nous amène le lemme suivant:

Lemme 2.14

La solution de la récurrence précédente (2.3) est:

$$x_n = a_n + 4 \sum_{j=3}^{n} \frac{\sum_{i=1}^{j-1} i^2(i-1)a_i}{j^2(j-1)^2(j-2)}, \quad n \geq 3, \quad x_1 = 0, \quad x_2 = a_2.$$

Preuve du lemme 2.14.

De l'expression (2.1), on a que

$$\sum_{i=1}^{n-2} i(H_{n-1}-H_i)x_i = \frac{(n-1)(n-2)}{4}(x_{n-1}-a_{n-1}).$$

En remplaçant H_{n-1} par H_n-1/n, on obtient

$$\sum_{i=1}^{n-2} i(H_n-H_i)x_i = \frac{(n-1)(n-2)}{4}(x_{n-1}-a_{n-1}) + \frac{1}{n}\sum_{i=1}^{n-2} ix_i.$$

Réécrivons l'expression (2.1) à l'aide des dernières expressions obtenues:

$$\begin{aligned} x_n &= a_n + \frac{4}{n^2}x_{n-1} + \frac{n-2}{n}(x_{n-1}-a_{n-1}) + \frac{4}{n^2(n-1)}\sum_{i=1}^{n-2} ix_i \\ &= a_n + \frac{n-2}{n}(x_{n-1}-a_{n-1}) + \frac{4}{n^2(n-1)}\sum_{i=1}^{n-1} ix_i. \end{aligned} \tag{2.4}$$

De cette dernière équation, on peut écrire:

$$\sum_{i=1}^{n-2} ix_i = \frac{(n-1)^2(n-2)}{4}\left[x_{n-1}-a_{n-1}-\frac{n-3}{n-1}(x_{n-2}-a_{n-2})\right].$$

En remplaçant cette dernière expression dans (2.4), on a:

$$\begin{aligned} x_n = a_n + \frac{n-2}{n}(x_{n-1}-a_{n-1}) + \frac{4}{n^2}x_{n-1} + \\ \frac{(n-1)(n-2)}{n^2}(x_{n-1}-a_{n-1}) - \frac{(n-2)(n-3)}{n^2}(x_{n-2}-a_{n-2}). \end{aligned}$$

Si on pose $z_n = x_n-a_n$ pour $n \geq 1$, on obtient:

$$z_n = \frac{(2n-1)(n-2)}{n^2}z_{n-1} - \frac{(n-2)(n-3)}{n^2}z_{n-2} + \frac{4}{n^2}(z_{n-1}-a_{n-1}).$$

Et en posant $y_n = z_n-z_{n-1}$, pour $n \geq 2$, on a:

$$y_n = \frac{4}{n^2}a_{n-1} + \frac{(n-2)(n-3)}{n^2}y_{n-1}$$

$$= \frac{4}{n^2}a_{n-1} + \frac{4(n-2)(n-3)}{n^2(n-1)^2}a_{n-2} + \frac{(n-2)(n-3)^2(n-4)}{n^2(n-1)^2}y_{n-2}$$

$$= \frac{4}{n^2}a_{n-1} + \frac{4(n-2)(n-3)}{n^2(n-1)^2}a_{n-2} + \frac{4(n-3)^2(n-4)}{n^2(n-1)^2(n-2)}a_{n-3}$$

$$+ \frac{(n-3)^2(n-4)^2(n-5)}{n^2(n-1)^2(n-2)}y_{n-2}$$

$$= \sum_{i=1}^{k} \frac{4(n-i)^2(n-1-i)}{n^2(n-1)^2(n-2)}a_{n-i} + \frac{(n-k)^2(n-1-k)(n-2-k)}{n^2(n-1)^2(n-2)}y_{n-k},\ 0 \le k \le n-2.$$

Si on pose $k = n-2$, alors $y_2 = z_2 - z_1 = (x_2 - a_2) - (x_1 - a_1) = a_1$ et,

$$y_n = \frac{4}{n^2(n-1)^2(n-2)}\sum_{i=1}^{n-2}(n-i)^2(n-1-i)a_{n-i},\ n \ge 3$$

$$= \frac{4}{n^2(n-1)^2(n-2)}\sum_{i=2}^{n-1}i^2(i-1)a_i,\ n \ge 3.$$

Nous avions que $y_n = (x_n - a_n) - (x_{n-1} - a_{n-1})$, alors,

$$\sum_{j=3}^{n} y_j = \sum_{j=3}^{n}(x_j - a_j) - (x_{j-1} - a_{j-1}) = (x_n - a_n) - (x_2 - a_2),$$

donc,
$$x_n = \sum_{j=3}^{n} y_j + a_n = a_n + 4\sum_{j=3}^{n}\frac{\sum_{i=2}^{j-1} i^2(i-1)a_i}{j^2(j-1)^2(j-2)}.$$ ■

Ce dernier lemme nous permettra de calculer les formules explicites pour $\mu_{n,1}$ et $\mu_{n,2}$ que l'on retrouve dans le théorème suivant qui avait été énoncé au début de la section 2.2.

Théorème 2.2

$$\mu_{1,m} = 0, \text{ pour } m > 0,$$

$$\mu_{n,1} = H_n - \frac{1}{6} - \frac{2}{3n},$$

$$\mu_{n,2} = H_n^2 + H_n^{(2)} + \frac{H_n}{6} - \frac{4H_n}{3n} + \frac{7}{9n} - \frac{77}{36},$$

$$\text{et } \mathrm{Var}(D_n) = H_n^{(2)} + \frac{H_n}{2} + \frac{5}{9n} - \frac{4}{9n^2} - \frac{13}{6}, \quad n \geq 2.$$

Preuve du théorème 2.2.

Pour $\mu_{n,1}$, on utilise le lemme 2.14 avec $a_n = 1$, ce qui donne

$$\mu_{n,1} = 1 + 4\sum_{j=3}^{n} \frac{\sum_{i=1}^{j-1} i^2(i-1)}{j^2(j-1)^2(j-2)} = 1 + \frac{1}{3}\sum_{j=3}^{n} \frac{3j-1}{j(j-1)} = 1 + \frac{1}{3}\sum_{j=3}^{n}\left(\frac{1}{j} + \frac{2}{j-1}\right)$$

$$= 1 + \frac{1}{3}\left[H_n - 1 - \frac{1}{2} + 2(H_{n-1} - 1)\right] = H_n - \frac{1}{6} - \frac{2}{3n}.$$

De ceci, on peut calculer $\mu_{n,2}$ en utilisant le lemme 2.14 avec $a_n = 2\mu_{n,1} - 1$, ce qui donne:

$$a_n = 2H_n - \frac{4(n+1)}{3n}, \text{ donc}$$

$$\mu_{n,2} = 2H_n - \frac{4(n+1)}{3n} + 4\sum_{j=3}^{n} \frac{b_j}{j^2(j-1)^2(j-2)},$$

où

$$b_j = \sum_{i=1}^{j-1} i^2(i-1)\left[2H_i - \frac{4(i+1)}{3i}\right]$$

$$= 2\left[\sum_{i=1}^{j} i^2(i-1)H_i - j^2(j-1)H_j\right] - \frac{4}{3}\sum_{i=1}^{j-1} i(i^2-1)$$

$$= \frac{j(j-1)(j-2)}{72}\left[12(3j-1)H_j - (33j+29)\right].$$

Donc,

$$\mu_{n,2} = 2H_n - \frac{4(n+1)}{3n} + \frac{1}{18}\sum_{j=3}^{n} \frac{12(3j-1)H_j - (33j+29)}{j(j-1)}$$

$$= 2H_n - \frac{4(n+1)}{3n} + \frac{2}{3}\sum_{j=3}^{n}\left(\frac{1}{j} + \frac{1}{j-1}\right)H_j - \frac{1}{18}\sum_{j=3}^{n}\left(\frac{62}{j-1} - \frac{29}{j}\right)$$

$$= 2H_n - \frac{4(n+1)}{3n} + \frac{2}{3}\left[\sum_{j=3}^{n}\frac{H_j}{j} + 2\sum_{j=3}^{n}\frac{H_{j-1}+1/j}{j-1}\right] - \frac{1}{18}\left[33H_n - \frac{37}{2} - \frac{62}{n}\right]$$

$$= \frac{1}{6}H_n - \frac{11}{36} + \frac{19}{9n} + \frac{2}{3}\left[\sum_{j=1}^{n}\frac{H_j}{j} - 1 - \frac{3}{4} + 2\sum_{j=2}^{n-1}\frac{H_j}{j} + 2\sum_{j=3}^{n}\frac{1}{j(j-1)}\right]$$

$$= \frac{1}{6}H_n - \frac{53}{36} + \frac{19}{9n} - \frac{2}{3}(2H_n+2) + 2\sum_{j=1}^{n}\frac{H_j}{j} + \frac{4}{3}\left[\sum_{j=2}^{n-1}\frac{1}{j} - \sum_{j=3}^{n}\frac{1}{j}\right]$$

$$= \frac{1}{6}H_n - \frac{101}{36} + \frac{19}{9n} - \frac{4H_n}{3n} + H_n^2 + H_n^{(2)} + \frac{4}{3}\left[H_n - \frac{1}{n} - 1 - H_n + 1 + \frac{1}{2}\right]$$

$$= \frac{1}{6}H_n - \frac{77}{36} + \frac{7}{9n} - \frac{4H_n}{3n} + H_n^2 + H_n^{(2)}.$$

Pour effectuer ce dernier développement, certaines identités concernant les nombres

harmoniques $H_n^{(j)}$ ont été utilisées. On en trouvera une liste en annexe. Pour calculer $\mathrm{Var}(D_n)$, on n'a qu'à calculer $\mu_{n,2} - \mu_{n,1}^2$. ■

Chapitre 3

Analyse probabiliste des arbres hyperquaternaires de points

Les résultats de ce chapitre sont issus d'un article de Devroye et Laforest (1990).

3.1 Introduction

Nous allons considérer le modèle d'arbre décrit au chapitre précédent, à savoir, l'arbre hyperquaternaire de points à d dimensions, construit avec une suite de n points aléatoires de l'hypercube $[0,1]^d$. Nous nous intéresserons ici au comportement asymptotique de la variable aléatoire D_n, la profondeur du dernier nœud inséré dans un arbre de n nœuds. Le principal résultat que nous allons démontrer est le suivant.

Théorème 3.1

La quantité $\frac{D_n}{\log n}$ tend en probabilité vers $\frac{2}{d}$ lorsque $n \to \infty$. On a aussi que $E(D_n) \sim E(A_n) \sim (2/d)\log n$ lorsque $n \to \infty$.

Comme mentionné au premier chapitre, lorsque $d = 1$, le problème se ramène aux arbres binaires de fouille, construits à partir de permutations équiprobables. On peut retrouver ces propriétés, incluant la loi des grands nombres énoncée au théorème 3.1, dans une série de papiers écrits par Lynch (1965), Knuth(1973), Robson (1979), Sedgewick (1983), Pittel (1984), Mahmoud et Pittel (1984), Brown et Shubert (1984) et Devroye (1986, 1987,1988).

3.2 Résultats concernant les espacements, les records et les coupes aléatoires.

Nous allons considérer une suite de n variables aléatoires, indépendantes, uniformes sur l'intervalle [0,1], que nous noterons $U_1, U_2, \ldots, U_n$. Soit S_{nx}, la taille de l'intervalle auquel x, un nombre fixé de l'intervalle [0,1], appartient. Nous avons le résultat suivant:

Lemme 3.2

Pour tout $x \in [0,1]$, $S_{nx} \stackrel{L}{=} \min(x, U_1, U_2, \ldots, U_n) + 1 - \max(x, U_1, U_2, \ldots, U_n)$, où L représente l'égalité en distribution. Si $x = U$ et que $U, U_1, U_2, \ldots, U_n$ est une suite de variables aléatoires uniformes [0,1], indépendantes, alors S_{nU} est distribué comme la deuxième plus petite valeur de la suite $U, U_1, U_2, \ldots, U_n$.

Preuve du lemme 3.2.

Nous allons considérer trois cas selon l'appartenance de x à l'un des trois sous-intervalles suivants:

$$[0, \min_i U_i),\ [\min_i U_i, \max_i U_i]\ \text{et}\ (\max_i U_i, 1].$$

Considérons premièrement le cas où $x \in [\min_i U_i, \max_i U_i]$. On définit

$$V_i = \begin{cases} x - U_i & \text{si } U_i < x \\ 1 + x - U_i & \text{si } U_i \geq x \end{cases}, \quad 1 \leq i \leq n.$$

On remarque que les variables V_i sont indépendantes et aussi uniformes [0,1]. De plus, on a que

$$\begin{aligned} S_{nx} &= (x - \max_{i:U_i<x} U_i) + (\min_{i:U_i\geq x} U_i - x) \\ &= \min_{i:U_i<x} (x - U_i) + \min_{i:U_i\geq x} (U_i - x) \\ &= \min_{i:U_i<x} (x - U_i) - \max_{i:U_i\geq x} (x - U_i) \\ &= \min_{i:U_i<x} (x - U_i) - \max_{i:U_i\geq x} (1 + x - U_i) + 1 \\ &= \min_{i:U_i<x} V_i - \max_{i:U_i\geq x} V_i + 1 \\ &= \min_i V_i - \max_i V_i + 1. \end{aligned}$$

La dernière égalité s'obtient en remarquant que $V_i \leq x$ pour les i tels que $U_i < x$ car $V_i = x - U_i$ et que $V_i \geq x$ pour les i tels que $U_i \geq x$ car $V_i = x + (1 - U_i)$, donc $\max_{i:U_i<x} V_i \leq \min_{i:U_i\geq x} V_i$, ce qui entraîne que $\min_{i:U_i<x} V_i = \min_i V_i$ et que $\max_{i:U_i\geq x} V_i = \max_i V_i$.

Maintenant, comme $S_{nx} = \min_i V_i + 1 - \max_i V_i$ et que les V_i sont indépendantes, uniformes [0,1], S_{nx} est bien distribué comme spécifié dans l'énoncé du lemme.

Deuxièmement, si $x \in [0, \min_i U_i)$ alors

$$\begin{aligned} S_{nx} &= x + \min_i (U_i - x) \\ &= x - \max_i (x - U_i) \\ &= x - \max_i (1 + x - U_i) + 1 \\ &= x - \max_i V_i + 1 \\ &= \min(\min_i V_i, x) - \max_i V_i + 1, \end{aligned}$$

puisque $x < U_i$, $1 \le i \le n$ et que $V_i = x + (1 - U_i)$, $1 \le i \le n$, donc $x \le V_i$, $1 \le i \le n$.

Troisièmement, si $x \in (\max_i U_i, 1]$, alors

$$\begin{aligned} S_{nx} &= 1 - x + \min_i (x - U_i) \\ &= 1 - \max(\max_i V_i, x) + \min_i V_i \end{aligned}$$

puisque $x \ge U_i$, $1 \le i \le n$ et que $V_i = x - U_i$, $1 \le i \le n$, alors $x \ge V_i$, $1 \le i \le n$. On a donc démontré la première partie du lemme.

Pour la deuxième partie, posons $U_{(1)} = \min(U, U_1, U_2, \ldots, U_n)$, $U_{(n+1)} = \max(U, U_1, U_2, \ldots, U_n)$, et $U_{(k)}$, le $k^{\text{ième}}$ plus petit de $U, U_1, U_2, \ldots, U_n$. De plus, posons $U_{(0)} = 0$ et $U_{(n+2)} = 1$. Supposons que $U = U_{(k)}$, $1 \le k \le n+1$. Nous avons que

$$S_{nU} = U_{(k+1)} - U_{(k-1)}, \quad 1 \le k \le n+1.$$

Si on pose maintenant $W_{s-r} = U_{(s)} - U_{(r)}$, $0 \le r < s \le n+1$, on a (voir Gibbons, 1971, page 28) que la variable aléatoire W_t a comme fonction de densité

$$f_t(u) = \frac{n!}{(t-1)!(n-t)!} u^{t-1}(1-u)^{n-t}, \quad 0 \le u \le 1, \quad 1 \le t \le n.$$

Cette fonction est indépendante des statistiques d'ordre comme telles, elle est plutôt fonction du délai considéré. Autrement dit, $S_{nU} \stackrel{L}{=} W_2$, et en particulier,

$S_{nU} \stackrel{L}{=} U_{(2)} - U_{(0)} = U_{(2)}$ dont la fonction de densité est donnée par

$$f_t(u) = n(n+1)u(1-u)^{n-1}, \quad 0 \le u \le 1.$$

On peut retrouver le même résultat en utilisant une propriété des espacements uniformes spécifiant que la somme de k espacements quelconques est distribuée comme la somme des k premiers espacements (voir Pyke (1965, 1972) pour une vue générale). Ceci termine donc la démonstration du lemme. ■

Le lemme suivant nous donne des bornes concernant les probabilités de la variable aléatoire S_{nU}.

Lemme 3.3

Pour $t \in (0,1)$,

$$P(S_{nU} < t) = 1 - (1+tn)(1-t)^n \le (tn)^2 \left[\frac{1}{2} + \frac{1}{n(1-t)} \right]$$

et

$$P(S_{nU} > t) = (1+tn)(1-t)^n \le (1+tn)e^{-tn} \le e^{\frac{-(tn)^2}{2(1+tn)}}$$

et pour $tn \ge 1$,

$$e^{\frac{-(tn)^2}{2(1+tn)}} \le e^{\frac{-tn}{4}}.$$

Preuve du lemme 3.3.

Le lemme précédent nous assure que S_{nU} est distribué comme $U_{(2)}$, la deuxième statistique d'ordre de la suite $U, U_1, U_2, \ldots, U_n$ dont la fonction de densité nous est donnée par

$$n(n+1)u(1-u)^{n-1},\ 0 \le u \le 1.$$

La fonction de répartition est exprimée par

$$\int_0^t n(n+1)u(1-u)^{n-1}du \;=\; 1 \;-\; (1-t)^n(1-tn),$$

donc,

$$P\,(S_{nU}<t) \;=\; 1 \;-\; (1-t)^n(1-tn),\quad \text{et}$$

$$P\,(S_{nU}>t) \;=\; (1-t)^n(1-tn).$$

On peut arriver aux mêmes probabilités en remarquant que, si Y est binomiale$(n+1,t)$, alors

$$\begin{aligned} P(S_{nU} < t) &= P(Y \ge 2) \\ &= 1-P(Y=0)-P(Y=1) \\ &= 1-(1-t)^{n+1}-(n+1)t(1-t)^n \\ &= 1-(1-t)^n(1+tn). \end{aligned}$$

Pour les inégalités du lemme, nous allons nous servir du fait que

$$1 - e^{-v} \le v \tag{3.1}$$

$$\log(1+v) \ge v - v^2/2, \text{ pour } v \ge 0, \tag{3.2}$$

$$\log(1-v) \ge -v/(1-v), \text{ pour } 0 \le v < 1, \tag{3.3}$$

$$\log(1+v) - v \le -\tfrac{1}{2}\, v^2/(1+v), \text{ pour } v \ge 0. \tag{3.4}$$

Il suffit d'utiliser les séries de Taylor avec reste pour montrer les trois premières inégalités. En effet, pour $0 < \xi < v$,

$$\log(1+v) \;=\; v - \frac{v^2}{2(1+\xi)^2}\ ,\ 1-e^{-v} \;=\; ve^{-\xi} \text{ et } \log(1-v) \;=\; -\frac{v}{1-\xi} \text{ pour } v < 1.$$

Pour la dernière inégalité, il suffit de remarquer que les deux côtés de celle-ci s'évaluent à 0

lorsque $v = 0$, et que pour $v \geq 0$,

$$\frac{d}{dv}[\log(1+v)-v] = \frac{-v}{1+v} \leq \frac{-v}{1+v}\frac{2+v}{2+2v} = \frac{d}{dv}\left[\frac{-v^2}{2(1+v)}\right].$$

Maintenant, en ce qui a trait aux inégalités du lemme, nous avons, en utilisant les inégalités (3.1), (3.2) et (3.3), que

$$1 - (1-t)^n(1+tn) \leq 1 - \left[\frac{-t}{e^{1-t}}\right]^n\left[e^{tn-\frac{(tn)^2}{2}}\right] = 1 - e^{tn-\frac{(tn)^2}{2}-\frac{tn}{1-t}}$$
$$\leq -tn + \frac{(tn)^2}{2} + \frac{tn}{1-t} = (tn)^2\left[\frac{1}{2}+\frac{1}{n(1-t)}\right].$$

Pour la deuxième probabilité, on se sert de l'inégalité (3.4) et de l'inégalité (3.3). Ceci nous donne

$$(1-t)^n(1+tn) \leq (1+tn)e^{-tn} \leq e^{\frac{-(tn)^2}{2(1+tn)}}.$$

De plus, si $tn \geq 1$, alors $2tn \geq 1+tn$ et,

$$\frac{(tn)^2}{2(1+tn)} \geq \frac{(tn)^2}{2(2tn)} = \frac{tn}{4}.$$

Cette dernière expression nous permet donc de déduire la dernière inégalité du lemme. ■

Considérons encore une fois $U, U_1, U_2, \ldots, U_n$, une suite de $n+1$ variables aléatoires indépendantes, uniformes [0,1] et définissons $[V_i,U]$ et $[U,W_i]$ comme étant les sous-intervalles les plus près de U, après avoir considéré $U, U_1, U_2, \ldots, U_i$, en ayant comme convention que $V_0 = 0$ et que $W_0 = 1$. Considérons maintenant N_n, le nombre d'indices i pour lesquels $(V_i,W_i) \neq (V_{i-1},W_{i-1})$, $1 \leq i \leq n$; autrement dit, le nombre de fois où l'on s'est rapproché de U, soit par la gauche, soit par la droite. Dans Devroye (1988), le rapprochement entre N_n et la théorie des records a été exploré. Entre autres, il a

été montré que N_n est distribué comme la somme d'une suite de n variables aléatoires indépendantes Bernoulli Y_i où $E(Y_i) = \frac{2}{i+1}$, c'est-à-dire que $N_n = \sum_{i=1}^{n} Y_i$ et donc que $E(N_n) = 2(H_{n+1}-1)$.

Nous aurons besoin d'en savoir plus sur N_n dans la mesure où N_n représente le nombre de fois où le sous-intervalle contenant U est "coupé", à mesure où l'on considère les U_i. En particulier, nous avons besoin de bornes solides que nous retrouvons dans le lemme suivant:

Lemme 3.4

Soit $E(N_n) = 2(H_{n+1}-1) = \mu$, alors

pour $k \geq \mu$,

$$P(N_n \geq k) \leq e^{-\frac{(k-\mu)^2}{2k}}, \text{ et}$$

pour $k \leq \mu$,

$$P(N_n \leq k) \leq e^{-\frac{(\mu-k)^2}{2\mu}}.$$

Preuve du lemme 3.4.

Dans cette preuve, nous utiliserons la technique de borne exponentielle de Chernoff. Nous avons, pour $\lambda \geq 0$, que

$$P(N_n \geq k) \leq E(e^{\lambda(N_n-k)}) = e^{-\lambda k} E(e^{\lambda \sum_{i=1}^{n} Y_i})$$

$$= e^{-\lambda k} \prod_{i=1}^{n} \left[1 - \frac{2}{i+1} + \frac{2e^{\lambda}}{i+1} \right].$$

Puisque $1+a \leq e^{a}$, $\forall a$, on a que

$$P(N_n \geq k) \leq e^{-\lambda k} \prod_{i=1}^{n} e^{\frac{2}{i+1}(e^{\lambda}-1)} = e^{-\lambda k} e^{\sum_{i=1}^{n} \frac{2}{i+1}(e^{\lambda}-1)} = e^{-\lambda k + (e^{\lambda}-1)\mu}.$$

L'exposant de la dernière expression est minimal lorsque $e^{\lambda} = k/\mu$. Lorsque l'on remplace la valeur de k, déduite de l'expression précédente pour minimiser l'exposant, on obtient que

$$P(N_n \geq k) \leq e^{\mu(e^{\lambda}-1-\lambda e^{\lambda})}.$$

Maintenant, on peut montrer, en utilisant le développement en série de Taylor avec reste, que pour $y \geq 0$, $y-(1+y)\log(1+y) \leq -\frac{1}{2} y^2/(1+y)$. Si on pose $y = e^{\lambda}-1$, on peut écrire que

$$P(N_n \geq k) \leq e^{\mu(y-(1+y)\log(1+y))} \leq e^{-\frac{\mu y^2}{2(1+y)}}.$$

Si on revient à la notation de départ, en fonction de k, on retrouve la première inégalité du lemme. Pour la deuxième inégalité, en procède de la même façon que pour la première. Pour $\lambda \geq 0$ on a

$$P(N_n \leq k) = E(e^{\lambda(k-N_n)}) = e^{\lambda k} E(e^{-\lambda \sum_{i=1}^{n} Y_i})$$

$$= e^{\lambda k} \prod_{i=1}^{n} \left[1 - \frac{2}{i+1} + \frac{2e^{-\lambda}}{i+1} \right]$$

$$\leq e^{\lambda k} \prod_{i=1}^{n} e^{\frac{2}{i+1}(e^{-\lambda}-1)} = e^{\lambda k} e^{\sum_{i=1}^{n} \frac{2}{i+1}(e^{-\lambda}-1)} = e^{\lambda k - (1-e^{-\lambda})\mu}.$$

On remplace, dans l'expression précédente, la valeur de k, provenant du fait que $e^{-\lambda} = k/\mu$

minimise l'exposant ce celle-ci. On obtient alors que

$$P\,(N_n \geq k) \;\leq\; e^{-\mu(1-e^{-\lambda}-\lambda e^{-\lambda})}.$$

On peut montrer que, pour $0 \leq y < 1$, on a $y+(1-y)\log(1-y) \geq \frac{1}{2}y^2$, en utilisant le développement en série de Taylor avec reste. En posant $y = 1-e^{-\lambda}$, on obtient que

$$P\,(N_n \leq k) \;\leq\; e^{-\mu(y+(1-y)\log(1-y))} \;\leq\; e^{-\frac{\mu y^2}{2}}.$$

Cette dernière expression est bien la dernière inégalité du lemme si on prend la peine de remplacer $y = 1-k/\mu$. Ceci termine donc la démonstration du lemme. ■

Le dernier lemme montre vraiment clairement que la variable aléatoire N_n est proche de son espérance $\mu = 2(H_{n+1}-1)$.

Nous sommes maintenant presque en mesure de présenter le lemme principal concernant les coupes aléatoires. Considérons une séquence infinie de variables aléatoires uniformes indépendantes sur [0,1], U, U_1, U_2, ..., U_n, ..., et soit Z_k, la taille du sous-intervalle auquel U appartient après avoir été "coupé", ou "frappé" k fois par les éléments de la suite U_1, U_2, ... Dans la notation introduite plus haut, $Z_k = W_i - V_i$ où (V_i, W_i) est la $k^{\text{ième}}$ paire n'étant pas égale à la précédente. On peut remarquer que $Z_0 > Z_1 > \ldots > Z_k > \ldots$. Il est intéressant de noter que Z_k, S_{nU} et N_n sont reliées comme l'indique le lemme suivant.

Lemme 3.5

Soit $k > 0$ et $t \in (0,1)$ fixé. Alors, pour tout entier positif n,

$$[Z_k < t] \subseteq [S_{nU} < t] \cup [N_n < k], \text{ et}$$

$$[Z_k \geq t] \subseteq [S_{nU} > t] \cup [N_n \geq k].$$

Preuve du lemme 3.5.

Premièrement, comme $N_n \in \{0, 1, \ldots, n\}$, nous distinguerons le cas où $k > n$ de celui où $k \leq n$. Pour la première inclusion, si $k > n$ et si l'évènement $[Z_k < t]$ s'est réalisé alors les évènements $[S_{nU} < t]^c$ et $[N_n < k]^c$ ne peuvent se réaliser simultanément. Lorsque $k \leq n$, les évènements $[Z_k < t]$, $[S_{nU} < t]^c$ et $[N_n < k]^c$ ne peuvent se produire simultanément car $N_n \geq k \Rightarrow Z_k \geq S_{nU} \geq t$, ce qui est une contradiction. Ceci prouve donc la première inclusion. On prouve la deuxième par un raisonnement analogue. Pour $k > n$, on a que $Z_k < S_{nU}$ et si $[S_{nU} > t]^c$ s'est réalisé, on a que $Z_k < S_{nU} < t$ donc, l'évènement $[Z_k \geq t]$ ne peut se réaliser. Pour $k \leq n$, si $[N_n \geq k]^c$ s'est réalisé, on a que $Z_k < S_{nU}$. Si, de plus, $[Z_k \geq t]$ s'est réalisé, on a que $S_{nU} > Z_k \geq t$, ce qui entraîne que l'évènement $[S_{nU} > t]^c$ ne peut se réaliser. On a donc terminé la démonstration du lemme. ■

Nous pouvons maintenant énoncer le lemme principal se rapportant aux k-coupes uniformes Z_k.

Lemme 3.6

Pour $k \geq 3$ et $\delta > 0$, on a

$$P\left[Z_k < e^{-\frac{k-1}{2}(1+2\delta)} \right] \leq 6e^{-\delta(k-1)} + e^{-\frac{\delta^2(k-1)}{2(1+\delta)}}.$$

Aussi, si $\delta \in (0,\frac{1}{2})$, $\delta \geq 3/k$ et $k \geq 2/(1-\delta)$, on a

$$P\left[Z_k \geq e^{-\frac{k}{2}(1-2\delta)} \right] \leq e^{\frac{1}{2}-\frac{1}{4}e^{\frac{k\delta}{2}}} + (2e)^{\frac{\delta^2}{1-\delta}} e^{-\frac{k\delta^2}{2}}.$$

Preuve du lemme 3.6.

Des lemmes 3.3, 3.4 et 3.5, nous avons que, pour une valeur de n que nous fixerons plus tard,

$$P(Z_k<t) \leq P(S_{nU}<t) + P(N_n<k)$$

$$\leq (tn)^2\left[\frac{1}{2} + \frac{1}{n(1-t)} \right] + e^{-\frac{(\mu-k+1)^2}{2\mu}},$$

ceci étant valide pour $k-1 \leq \mu$. Considérons une constante $\delta > 0$ et définissons

$$n = \left\lfloor 2e^{\frac{k-1}{2}(1+\delta)} \right\rfloor.$$

On note par ailleurs que

$$H_{n+1} - 1 = \sum_{i=2}^{n+1} \frac{1}{i} \geq \int_2^{n+2} \frac{1}{x}dx = \log\left[\frac{n+2}{2}\right] \geq \frac{k-1}{2}(1+\delta).$$

Ceci implique que $(k-1) \leq (k-1)(1+\delta) \leq 2(H_{n+1}-1) = \mu$ comme souhaité. Maintenant, puisque

$$\frac{d}{dy}\left[\frac{(y-a)^2}{y}\right] = \frac{y^2-a^2}{y^2} > 0 \text{ pour } y > a \geq 0,$$

on a que

$$\frac{(\mu-k+1)^2}{2\mu} \geq \frac{((k-1)(1+\delta)-k+1)^2}{2(k-1)(1+\delta)} = \frac{\delta^2(k-1)}{2(1+\delta)}.$$

On obtient alors que

$$P(Z_k<t) \leq (tn)^2\left[\frac{1}{2} + \frac{1}{n(1-t)}\right] + e^{-\frac{\delta^2(k-1)}{2(1+\delta)}}$$

$$\leq 4t^2e^{(k-1)(1+\delta)}\left[\frac{1}{2} + \frac{1}{n(1-t)}\right] + e^{-\frac{\delta^2(k-1)}{2(1+\delta)}}.$$

En utilisant le fait que $n \geq 2$ et en assumant que $t \leq \frac{1}{2}$, on obtient que $\frac{1}{n(1-t)} \leq 1$ et,

$$P(Z_k<t) \leq 6t^2e^{(k-1)(1+\delta)} + e^{-\frac{\delta^2(k-1)}{2(1+\delta)}}.$$

On obtient la première moitié du lemme en posant $t = e^{-\frac{k-1}{2}(1+2\delta)}$. La condition utilisée plus haut, $t \leq \frac{1}{2}$, est bien remplie puisque pour $k \geq 3$, $(k-1)(1+2\delta) > 2$, donc

$$e^{-\frac{k-1}{2}(1+2\delta)} \leq e^{-1} < \frac{1}{2}.$$

Pour ce qui est de la deuxième partie du lemme, assumons que n est tel que $k \geq 2(H_{n+1}-1) = \mu$ soit vérifiée. De plus, supposons que $tn \geq 1$. Nous pouvons donc écrire, à l'aide des lemmes 3.3, 3.4 et 3.5 que

$$P(Z_k \geq t) \leq P(S_{nU}>t) + P(N_n \geq k) \leq e^{-\frac{tn}{4}} + e^{-\frac{(k-\mu)^2}{2k}}.$$

Si on choisit $\delta \in (0,\frac{1}{2})$ et

$$n = \left\lfloor e^{\frac{k}{2}(1-\delta)} \right\rfloor - 1,$$

on a que

$$H_{n+1} - 1 = \sum_{i=2}^{n+1} \frac{1}{i} \leq \int_{1}^{n+1} \frac{1}{x}\,dx = \log(n+1) = \log(\left\lfloor e^{\frac{k}{2}(1-\delta)} \right\rfloor) \leq \frac{k}{2}(1-\delta)$$

et donc que $\mu = 2(H_{n+1}-1) \leq k(1-\delta) \leq k$. On a donc la valeur de k souhaitée. En plus, si $k(1-\delta) \geq 2$, on a que $n \geq 1$. De plus, en se servant du même argument concernant $(y-a)^2/y$, on peut écrire que

$$\frac{(k-\mu)^2}{2\mu} \geq \frac{\left[\frac{\mu}{1-\delta} - \mu\right]^2}{2\mu/(1-\delta)} = \frac{\delta^2\mu}{2(1-\delta)}.$$

On a donc que

$$P(Z_k \geq t) \leq e^{-\frac{tn}{4}} + e^{-\frac{\delta^2\mu}{2(1-\delta)}}.$$

Puisque

$$H_{n+1} - 1 \geq \log\left\lfloor \frac{n+2}{2} \right\rfloor - 1 = \log(n+2) - \log(2e)$$

$$\geq \log\left\lfloor e^{\frac{k}{2}(1-\delta)} \right\rfloor - \log(2e) = \frac{k(1-\delta)}{2} - \log(2e),$$

donc $\mu \geq k(1-\delta) - 2\log(2e)$. On obtient alors

$$P(Z_k \geq t) \leq e^{-\frac{tn}{4}} + e^{\left[-\frac{k(1-\delta)}{2} + \log(2e)\right]\frac{\delta^2}{1-\delta}}$$

$$= e^{-\frac{tn}{4}} + e^{-\frac{k\delta^2}{2}}(2e)^{\frac{\delta^2}{1-\delta}}.$$

Pour obtenir l'expression du lemme, on pose $t = e^{-\frac{k}{2}(1-2\delta)}$, ce qui donne

$$tn \geq e^{-\frac{k}{2}(1-2\delta)}\left[e^{\frac{k}{2}(1-\delta)} - 2\right] = e^{\frac{k\delta}{2}} - 2e^{-\frac{k}{2}(1-2\delta)} \geq e^{\frac{k\delta}{2}} - 2$$

et
$$P(Z_k \geq t) \leq e^{\frac{1}{2}-\frac{1}{4}e^{\frac{k\delta}{2}}} + (2e)^{\frac{\delta^2}{1-\delta}} e^{-\frac{k\delta^2}{2}}.$$

On voit bien aussi que la condition $tn \geq 1$ est bien vérifiée si $k\delta \geq 3$. ■

3.3 Une loi des grands nombres pour les arbres hyperquaternaires.

L'objectif de cette section est de prouver le théorème 3.1. Ceci sera effectué en réduisant le problème à d dimensions en d problèmes à une dimension pour lesquels nous disposons de solutions. Nous considérons l'arbre construit à partir d'insertions consécutives de vecteurs aléatoires indépendants et tous uniformes $[0,1]^d$. Le niveau D_{n+1} de X_{n+1} correspond au nombre de fois où l'hyperrectangle, provenant de la partition de l'arbre et contenant X_{n+1}, est "coupé" par $X_1, X_2, \ldots, X_{n+1}$. Ici, "coupé" signifie que l'extrémité d'un vecteur est arrivée dans l'hyperrectangle en question. À ce moment là, un nouvel hyperrectangle est considéré, celui formé par l'hyperquadrant, déterminé par le point ayant "coupé" l'hyperrectangle, contenant X_{n+1}. Au départ, on commence avec l'hypercube $[0,1]^d$. Le procédé de coupe peut être résumé par une séquence de variables aléatoires (T_k, Z_k), $k \geq 0$, où $T_0 = 0$ et $Z_0 = 1$. La variable T_k est un compteur représentant le nombre de points considérés afin de "couper" l'hyperrectangle k fois, et Z_k est la taille de l'hyperrectangle contenant X_{n+1} après avoir été coupé précisément k fois. Avant d'aller plus loin, voyons un petit exemple simple, à deux dimensions.

Nous allons considérer 8+1 points, donc $n = 8$ et $d = 2$. Ces points sont $X_1 = (4,2)$, $X_2 = (3,5)$, $X_3 = (8,3)$, $X_4 = (5,7)$, $X_5 = (6,1)$, $X_6 = (2,6)$, $X_7 = (1,8)$, $X_8 = (9,4)$ et $X_9 = (7,9)$. Ces points engendrent la partition et l'arbre suivants:

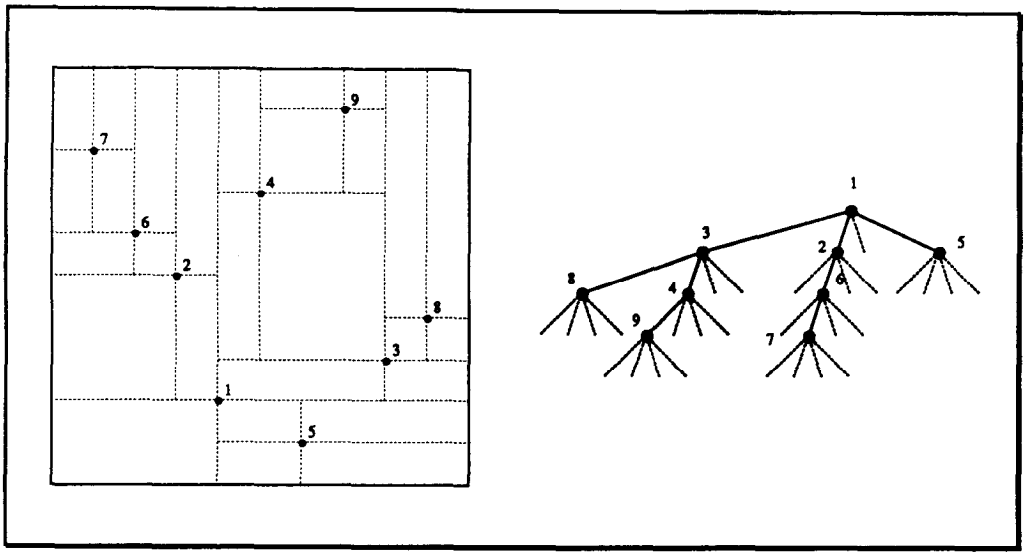


Fig. 3.1 – Arbre hyperquaternaire à deux dimensions.

Pour illustrer le couple de variables aléatoires (T_k, Z_k) nous allons plutôt placer X_9 dans le rectangle au tout début. Ensuite, on ajoute les huit points, un par un, en commençant par X_1. On obtient la série de figures suivantes.

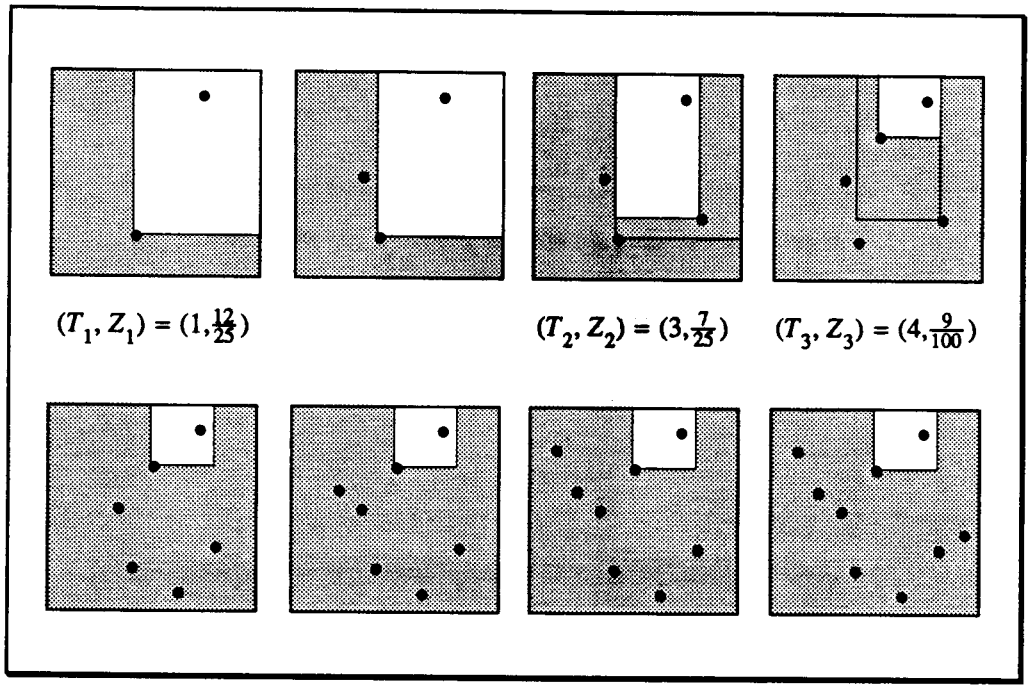


Fig. 3.2 – Coupes aléatoires.

Étant donnés (T_k, Z_k), X_{n+1} et $X_1, \ldots, X_{T_k}$, il est facile de voir que $T_{k+1} - T_k$ est une variable aléatoire géométrique de paramètre Z_k. En effet, $T_{k+1} - T_k$ représente le "temps" d'attente (le nombre de points considérés) avant d'obtenir un "succès" représenté ici par le fait de "couper" l'hyperrectangle dans lequel X_{n+1} se trouve. Alors,

$$P\,(T_{k+1} - T_k = i \mid (T_k, Z_k), X_{n+1}, X_1, \ldots, X_{T_k}) = Z_k\,(1-Z_k)^{i-1},\ i \geq 1,\ k \geq 0.$$

Par ailleurs, Z_{k+1} est distribué comme la taille de l'hyperrectangle contenant X_{n+1}, après avoir été coupé précisément k+1 fois. Il est à noter que D_{n+1} est égal à $\max_{k:T_k \leq n} k$. Nous avons donc

$$P\,(D_{n+1} \geq k) \;=\; P\,(T_k \leq n) \;\leq\; P\,(T_k - T_{k-1} \leq n).$$

On observe aussi que $Z_k = \prod_{i=1}^{d} Z_k(i)$ où les $Z_k(i)$ sont des variables aléatoires indépendantes, identiquement distribuées comme les k-coupes aléatoires dont il a été fait mention dans le lemme 3.6. Nous choisissons une petite constante positive $\delta \in (0,\frac{1}{3})$ et définissons

$$q \;=\; e^{-\frac{k-1}{2}(1-2\delta)}.$$

Soit A, l'événement faisant en sorte que $\max_i Z_{k-1}(i) \leq q$ Nous allons utiliser le fait que si A et B sont deux événements alors $B \subseteq (A \cap B) \cup A^c$, ce qui entraîne que $P\,(B) \leq P\,(B \mid A) + P\,(A^c)$ car $P\,(B \mid A) \geq P\,(A \cap B)$. Aussi, nous utiliserons le fait que si $0 \leq p \leq 1$, alors $1 + p + p^2 + \ldots + p^{n-1} \leq n$. Nous voulons évaluer l'expression suivante, à savoir,

$$P\,(D_{n+1} \geq k) \;\leq\; P\,(T_k - T_{k-1} \leq n) \;\leq\; P\,(T_k - T_{k-1} \leq n \mid A) + P\,(A^c).$$

Si l'événement A^c se réalise, c'est que $\max_i Z_{k-1}(i) > q$ et ce maximum est atteint, soit lorsque $i = 1$ ou $i = 2$ ou $\ldots$ $i = d$. Comme les $Z_{k-1}(i)$ sont identiquement distribués, on a

$$P\,(A^c) \;\leq\; \sum_{i=1}^{d} P\,(Z_{k-1}(i) > q) \;=\; dP\,(Z_{k-1}(1) > q).$$

Maintenant, si A s'est réalisé, c'est que $\max_i Z_{k-1}(i) \leq q$, et on peut voir que

$$Z_{k-1} \;=\; \prod_{i=1}^{d} Z_{k-1}(i) \;\leq\; q^d, \text{ et}$$

$$P\,(T_k - T_{k-1} \leq n \mid A) \;=\; \sum_{i=1}^{n} Z_{k-1}(1-Z_{k-1})^{i-1} \;=\; Z_{k-1} \sum_{i=1}^{n} (1-Z_{k-1})^{i-1} \;\leq\; nq^d$$

donc, en utilisant le lemme 3.6, on obtient que

$$P\,(D_{n+1}\geq k) \;\leq\; nq^d + d\left[\; e^{\frac{1}{2}} - \frac{1}{4}e^{\frac{k-1}{2}\delta} + (2e)^{\frac{\delta^2}{1-\delta}}\, e^{-\frac{k-1}{2}\delta^2} \right] \tag{3.5}$$

si $\delta \geq 3/(k-1)$ et $k-1 \geq 2/(1-\delta)$. Lorsque $k \to \infty$, la borne supérieure de l'expression précédente est $nq^d + o(1)$. Si on prend maintenant

$$k - 1 \;=\; \left\lfloor \frac{2}{d(1-3\delta)} \log n \right\rfloor$$

on a que

$$nq^d \;=\; ne^{-\frac{d}{2}(k-1)(1-2\delta)} \;=\; ne^{-\frac{d}{2}(1-2\delta)\left[\frac{2}{d(1-3\delta)}\log n - \varepsilon\right]} \;=\; n^{\frac{-\delta}{1-3\delta}}\, e^{\frac{d}{2}(1-2\delta)\varepsilon}, \quad 0 \leq \varepsilon < 1.$$

Donc, on peut voir que nq^d est $o(1)$ aussi. En conséquence, $P\,(D_{n+1}\geq k) = o(1)$. En fait, $P\,(D_{n+1}\geq k) = O(\log^{-R} n)$, pour toute constante positive R. Ceci prouve donc la première moitié du théorème 3.1.

La deuxième moitié se prouve de façon similaire. Posons A comme étant l'événement voulant que $\min_i Z_{k-1}(i) \geq q$ où

$$q \;=\; e^{-\frac{k-2}{2}(1+2\delta)}.$$

Soit $k \geq 3$ et assumons que $\delta > 0$ est une petite constante arbitraire. Alors,

$$P\,(D_{n+1} < k) \;=\; P\,(T_k > n) \;=\; P\,(\sum_{j=1}^{k} T_j - T_{j-1} > n).$$

Si $T_j - T_{j-1} \leq n/k$ pour $1 \leq j \leq k$ alors $T_k \leq n$. De façon équivalente, on peut affirmer que si $T_k > n$ alors $\exists j$ tel que $T_j - T_{j-1} > n/k$. Ce qui nous permet d'écrire, en termes probabilistes que

$$P\,(T_k > n) \;\leq\; P\left[\bigcup_{j=1}^{k} (\,T_j - T_{j-1} > \frac{n}{k}\,)\right] \;\leq\; k\,P\,(T_k - T_{k-1} > \frac{n}{k}\,)$$

$$\leq k\left[P\left(T_k - T_{k-1} > \frac{n}{k} \mid A\right) + P\left(A^c\right) \right].$$

Si l'événement A ne se réalise pas, c'est que $\min_i Z_{k-1}(i) < q$, c'est-à-dire que $\exists i$ tel que $Z_{k-1}(i) < q$. On peut alors écrire que

$$P\left(A^c\right) \leq P\left(\bigcup_{i=1}^{d} \left(Z_{k-1}(i) < q \right)\right) \leq dP\left(Z_{k-1}(1) < q\right).$$

Si maintenant A s'est réalisé, c'est que $\min_i Z_{k-1}(i) \geq q$, c'est-à-dire que $Z_{k-1}(i) \geq q$, $\forall i$, $1 \leq i \leq d$, ce qui nous permet d'écrire que $Z_{k-1} \geq q^d$. Nous allons utiliser la fait que si G_1 et G_2 sont deux variables aléatoires géométriques de paramètres p_1 et p_2 respectivement et que $p_1 \geq p_2$ alors $P(G_1 > k) \leq P(G_2 > k)$. Nous avons donc

$$P\left(T_k - T_{k-1} > \frac{n}{k} \mid A\right) = \sum_{i > \frac{n}{k}} Z_{k-1}(1-Z_{k-1})^{i-1} \leq \sum_{i > \frac{n}{k}} q^d (1-q^d)^{i-1}$$

$$= q^d \left[\frac{1}{q^d} - \frac{1-(1-q^d)^{\lfloor n/k \rfloor}}{q^d} \right] \leq (1-q^d)^{n/k-1} \leq e^{-(n/k-1)q^d}.$$

On obtient, pour la probabilité cherchée, que

$$P\left(D_{n+1} < k\right) \leq k\left[e^{-(n/k-1)q^d} + dP\left(Z_{k-1}(1) < q\right) \right].$$

On peut voir que $kdP\left(Z_{k-1}(1) < q\right) = o(1)$ lorsque $k \to \infty$. En effet,

$$kdP\left(Z_{k-1}(1) < q\right) \leq kd\left[6e^{-\delta(k-2)} + e^{-\frac{\delta^2(k-2)}{2(1+\delta)}} \right] = o(1).$$

De plus, $nq^d/k \to \infty$ lorsque $n \to \infty$ et que l'on pose

$$k - 2 = \left\lfloor \frac{2}{d(1+3\delta)} \log n \right\rfloor$$

car

$$\frac{nq^d}{k} = \frac{n}{\left\lfloor \frac{2}{d(1+3\delta)} \log n \right\rfloor + 2} e^{-\frac{d}{2}(1+2\delta)\left\lfloor \frac{2}{d(1+3\delta)} \log n \right\rfloor}$$

$$\leq \frac{n}{\frac{2}{d(1+3\delta)} \log n} e^{-\frac{d}{2}(1+2\delta)\left[\frac{2}{d(1+3\delta)} \log n - \varepsilon \right]} \quad \text{pour } 0 \leq \varepsilon < 1$$

$$= \frac{n^{\frac{\delta}{1+3\delta}}}{\frac{2}{d(1+3\delta)} \log n} e^{\frac{d}{2}(1+2\delta)\varepsilon} .$$

De ceci, on peut conclure que $P\,(D_{n+1}<k) = o(1)$.

En manipulant correctement les expressions obtenues, on peut voir que

$$\forall \varepsilon > 0, P\,(D_n > \frac{2}{d} \log n + \varepsilon) = o(1) \text{ et } P\,(D_n < \frac{2}{d} \log n - \varepsilon) = o(1),$$

ce qui nous permet de conclure que

$$\forall \varepsilon > 0, \lim_{n\to\infty} P\left[\left| \frac{D_n}{\log n} - \frac{2}{d} \right| > \varepsilon \right] = 0,$$

prouvant ainsi la première partie du théorème 3.1.

Maintenant, ce dernier résultat implique naturellement que

$$\liminf_{n\to\infty} \frac{E(D_n)}{\log n} \geq \frac{2}{d} .$$

Aussi, pour une petite constante $\delta > 0$ et pour une autre constante $M > 1$, toutes deux arbitraires,

$$E(D_n) = \int_0^n P\,(D_n > t)dt$$

$$= \int_0^{1+\left\lfloor \frac{2}{d(1-3\delta)} \log n \right\rfloor} P(D_n > t)dt + \int_{1+\left\lfloor \frac{2}{d(1-3\delta)} \log n \right\rfloor}^{M \log n} P(D_n > t)dt + \int_{M \log n}^{n} P(D_n > t)dt$$

$$\leq 1 + \frac{2}{d(1-3\delta)} \log n + M \log n \, P\left[D_n > 1 + \left\lfloor \frac{2}{d(1-3\delta)} \log n \right\rfloor \right] + nP(D_n > M \log n).$$

Pour la dernière expression, il suffit d'utiliser le fait que si $f(x) \geq 0$ est décroissante, alors $\int_a^b f(x)dx \leq bf(a)$. On a vu précédemment que $P(D_{n+1} \geq k) = O(\log^{-R} n)$, ce qui permet de conclure que le troisième terme de la dernière expression est $o(1)$. Pour le dernier terme, utilisons l'expression (3.5) avec q comme spécifié dans ce contexte et $k = \lfloor M \log n \rfloor$. On obtient

$$P(D_n > M \log n)$$

$$\leq n e^{-\frac{d}{2}(1-2\delta)(\lfloor M \log n \rfloor - 1)} + d\left[e^{\frac{1}{2}} - \frac{1}{4} e^{\frac{\delta}{2}(\lfloor M \log n \rfloor - 1)} + (2e)^{\frac{\delta^2}{1-\delta}} e^{-\frac{\delta^2}{2}(\lfloor M \log n \rfloor - 1)} \right]$$

$$\leq n e^{-\frac{d}{2}(1-2\delta)(M \log n - 2)} + d\left[e^{\frac{1}{2}} - \frac{1}{4} e^{\frac{\delta}{2}(M \log n - 2)} + (2e)^{\frac{\delta^2}{1-\delta}} e^{-\frac{\delta^2}{2}(M \log n - 2)} \right]$$

$$= n^{1 - \frac{d}{2}(1-2\delta)M} e^{d(1-2\delta)} + d\left[e^{\frac{1}{2}} e - \frac{1}{4} n^{\frac{\delta}{2}M} e^{\frac{1}{2}e^{\frac{\delta}{2}}} + (2e)^{\frac{\delta^2}{1-\delta}} n^{-\frac{\delta^2}{2}M} e^{\delta^2} \right].$$

Pour que $P(D_n > M \log n)$ soit $o(1/n)$, il faut que

$$-\frac{d}{2}(1-2\delta)M \leq -1 \quad \text{et} \quad -\frac{\delta^2}{2}M \leq -1, \text{ c'est-à-dire que } M \geq \max\left[\frac{2}{\delta^2}, \frac{2}{d(1-2\delta)} \right].$$

Donc,

$$E(D_n) \leq 1 + \frac{2}{d(1-3\delta)} \log n + o(1).$$

De ceci, on peut conclure que

$$\lim_{n\to\infty} \frac{E(D_n)}{\log n} \leq \frac{2}{d(1-3\delta)},$$

d'où,

$$E(D_n) \sim \frac{2}{d} \log n .$$

Ainsi,

$$E(A_n) = \frac{1}{n}\sum_{i=1}^{n} E(D_i) \sim \frac{2}{d}\log n. \quad \blacksquare$$

Chapitre 4

Étude des nœuds d'un arbre hyperquaternaire

4.1 Introduction

Plusieurs résultats intéressants ont été obtenus concernant les arbres binaires de fouille. Mentionnons entre autres le fait que, dans un arbre binaire de fouille ayant n nœuds, le nombre de nœuds qui sont des feuilles, qui ont un enfant et qui en ont deux sont en même proportion, en moyenne. Ce fait intéressant a été étudié par Mahmoud (1986). Une autre étude, effectuée par Poblete et Munro (1985), montre que si, au fur et à mesure que l'on ajoute des nœuds dans un arbre binaire de fouille, on restructure localement l'arbre pour diminuer la hauteur du sous-arbre impliqué, on diminue la profondeur moyenne des nœuds ajoutés. Dans un arbre non rebalancé, cette profondeur est d'environ $1.39 \log_2 N$, tandis qu'avec l'heuristique de rebalancement, on a environ $1.19 \log_2 N$. Ceci est a peu près à mi-chemin entre les arbres parfaitement balancés et les arbres formés aléatoirement. Cette étude est appelée "analyse de la frange". Nous verrons dans ce chapitre comment on peut généraliser certains résultats au cas à plusieurs dimensions, et que, dans certains cas, il devient extrêmement compliqué, voire presqu'impossible, même à deux dimensions, d'obtenir des résultats équivalents.

4.2 Étude des nœuds d'un arbre hyperquaternaire.

Pour l'arbre binaire de fouille, il a été montré que (Mahmoud, 1986), en moyenne, il y a autant de feuilles, de nœuds à un enfant et de nœuds à deux enfants. La technique utilisée consiste premièrement à calculer l'espérance du nombre de feuilles. Posons $E_{i,n}$ comme étant le nombre moyen de nœuds possédant i enfants dans un arbre hyperquaternaire de n nœuds, $i = 0, 1, 2, 3, 4$, $n \geq 1$. On obtient, pour l'arbre binaire, la récurrence suivante:

$$E_{0,n} = \frac{1}{n}\sum_{i=0}^{n-1}(E_{0,i} + E_{0,n-1-i}),$$

puisque la probabilité d'avoir i nœuds dans le sous-arbre gauche dans un arbre de n nœuds est $1/n$ (voir chapitre précédent) et que le nombre de feuilles dans l'arbre est égal à la somme du nombre de feuilles dans chacun des deux sous-arbres. La récurrence se résoud facilement, et on obtient:

$$E_{0,n} = \frac{n+1}{3}.$$

Pour obtenir le nombre moyen de nœuds à un enfant et à deux enfants, on utilise le fait que la somme des nœuds des trois types donne n, et que si l'on pose v_i comme étant le nombre de nœuds de degré i ($i = 1, 2, 3$), on a

$$v_1 + 2v_2 + 3v_3 = 2n - 2.$$

Il suffit donc de solutionner ce système et on obtient

$$v_1 = \frac{n+1}{3} + \frac{2}{n},$$

$$v_2 = \frac{n+4}{3} - \frac{4}{n},$$

$$v_3 = \frac{n-5}{3} + \frac{2}{n}.$$

Ceci a donné lieu au théorème suivant:

Théorème 4.1 (Mahmoud, 1986)

Soit v_i l'espérance du nombre de nœuds de degré i, $i = 1, 2, 3$. Alors

$$v_i = \frac{n}{3} + O(1), i = 1, 2, 3.$$

Voyons comment cette méthode peut se généraliser au cas à deux dimensions. Ici, nous nous intéresserons au nombre moyen de nœuds ayant i enfants, $i = 0, 1, 2, 3, 4$. Ceci constitue une légère différence avec l'approche de Mahmoud qui travaille plutôt avec le degré des nœuds, mais celui-ci doit faire de la racine, un cas particulier. Un des principaux résultats, dont la preuve sera donnée plus loin est le théorème suivant:

Théorème 4.2

Le nombre moyen de feuilles dans un arbre hyperquaternaire à deux dimensions, contenant n nœuds est

$$8H_n^{(2)}(3n+1) + 11 - 39n - \frac{4}{n}, \quad n \geq 2, \text{ où } H_n^{(2)} = 1 + \frac{1}{2^2} + \ldots + \frac{1}{n^2}.$$

Corollaire 4.3

La proportion moyenne de feuilles dans un arbre hyperquaternaire de dimension deux de n points est donnée par la formule asymptotique suivante:

$$(4\pi^2 - 39)\left(1+\frac{1}{3n}\right) - \frac{4}{3n^4} + O\left(\frac{1}{n^5}\right).$$

En particulier, on a $\lim_{n\to\infty} \frac{E_{0,n}}{n} = 0.478417604....$

Preuve du corollaire 4.3.

Il suffit d'appliquer la formule d'Euler-Maclaurin à l'expression obtenue au théorème 4.2 pour obtenir l'expression du corollaire. ■

Théorème 4.4

Le nombre moyen de nœuds à un enfant dans un arbre hyperquaternaire à deux dimensions, contenant n nœuds est

$$E_{1,n} = \frac{4}{n^2} + \frac{16}{n} - \frac{1171}{27} + \frac{2393}{9}n + 4(6-\frac{1}{n})H_n - 52(3n+1)H_n^{(2)}$$

$$- 8(3n+1)H_n^{(3)} + 8(3n+1)\sum_{i=4}^{n} \frac{H_i}{i^2}, \quad n \geq 3.$$

Corollaire 4.5

La proportion moyenne de nœuds à un enfant dans un arbre hyperquaternaire de dimension deux de n points est donnée par la formule asymptotique suivante

$$\lambda \cdot \left(1+\frac{1}{3n}\right) - \frac{4}{3n^3} + O\left(\frac{\log n}{n^4}\right)$$

où $\lambda = 12(2\alpha_1 + 19 - 2\zeta(3) - 13\zeta(2)) = 0.239651196\ldots,$

$\alpha_1 = \sum_{i=1}^{\infty} H_i/i^2 = 2.404113806\ldots$ et $\zeta(r)$ est la fonction zêta de Riemann.

En particulier, on a $\lim_{n\to\infty} \frac{E_{1,n}}{n} = \lambda = 0.239651196\ldots$.

Preuve du corollaire 4.5.

En appliquant la formule sommatoire d'Euler-Maclaurin à l'aide de Maple aux expressions H_n, $H_n^{(2)}$, $H_n^{(3)}$ et $\sum_{i=1}^{n} H_i/i^2$ on arrive au résultat. Cependant, l'expression $\alpha_1 = \sum_{i=1}^{\infty} H_i/i^2$ converge extrêmement lentement. Nous avons donc accéléré sa convergence de la façon suivante:

Soit $\alpha_k(n) = \sum_{i=1}^{n} H_i/i^2(i+1)\ldots(i+k-1)$. Posons $\alpha_k = \alpha_k(\infty)$. Il peut être démontré que

$$\alpha_1 = H_k \pi^2/6 - \theta_k + k!\alpha_{k+1} \quad \text{où } \theta_k = \frac{1}{2}\sum_{i=1}^{k-1} \frac{\binom{k}{i+1}}{i+1}\left(H_i^2 + H_i^{(2)}\right).$$

En prenant une valeur de k appropriée (ici, on a pris $k = 10$) et une valeur de n assez grande (ici, on a utilisé $n = 200$), on arrive à la valeur numérique donnée pour α_1. Les détails complets de cette preuve ainsi que d'autres termes de l'expression asymptotique

peuvent être trouvés dans Labelle, Laforest (1990). ■

Examinons maintenant le problème plus général encore à deux dimensions. Posons $a_{i,n} = P$(la racine a i enfants) pour $i = 0, 1, 2, 3, 4$, et $n \geq 1$. On a entre autres que

$$a_{0,1} = 1,$$

$$a_{0,n} = 0, \text{ pour } n > 1,$$

$$a_{1,1} = 0,$$

$$a_{1,n} = \frac{4}{n^2}, \text{ pour } n > 1.$$

En effet, au chapitre précédent, on avait calculé que

$$P\,(N_1 = i, N_2 = j, N_3 = k, N_4 = n-1-i-j-k) = \frac{\binom{n-1}{i,\, j,\, k,\, n-1-i-j-k}}{n^2 \binom{n-1}{i+j}\binom{n-1}{i+k}}$$

où N_l est la cardinalité du $l^{\text{ième}}$ sous-arbre, et où $0 \leq i+j+k \leq n-1$, $i \geq 0$, $j \geq 0$ et $k \geq 0$. Alors, pour calculer P (la racine a un enfant), il suffit de sommer les quatre possibilités où un des quatre sous-arbres possède $n-1$ enfant(s) et où les autres sont vides, ce qui donne le résultat recherché. Pour calculer les autres $a_{i,n}$, il suffit de faire les sommes appropriées, ce qui est technique mais cela ne donne pas de "belles" expressions pour $i > 1$.

Voyons maintenant les espérances. Nous avons alors que

$$E_{i,n} = a_{i,n} + 4 \sum_{j=0}^{n-1} E_{i,j}\, P\,(N_1 = j).$$

En utilisant le fait que $P\,(N_1 = j) = \dfrac{H_n - H_j}{n}$, calculée au chapitre précédent, on obtient que

$$E_{i,n} = a_{i,n} + \frac{4}{n} \sum_{j=0}^{n-1} E_{i,j}\,(H_n - H_j).$$

Pour calculer $E_{0,n}$, qui est le nombre moyen de feuilles, on utilise $a_{0,n} = 0$ pour $n > 1$. On peut donc démontrer le théorème.

Preuve du théorème 4.2.

Nous allons laisser tomber l'indice $i = 0$ pour plus de commodité. Donc, dans cette preuve, $E_{0,n} = E_n$. Comme conditions initiales on a $E_0 = 0, E_1 = 1$. On a donc, pour $n \geq 2$ que

$$E_n = \frac{4}{n}\sum_{i=0}^{n-1} E_i\,(H_n - H_i), \tag{4.1}$$

et, de ceci, on conclut que, pour $n \geq 3$,

$$\frac{n-1}{4}E_{n-1} = \sum_{i=0}^{n-2} E_i\,(H_{n-1} - H_i) = \sum_{i=0}^{n-2} E_i\,(H_n - H_i) - \frac{1}{n}\sum_{i=0}^{n-2} E_i\,.$$

En remplaçant ceci dans l'équation (4.1), on obtient

$$E_n = \frac{n-1}{n}E_{n-1} + \frac{4}{n^2}\sum_{i=0}^{n-1} E_i \tag{4.2}$$

dont on peut conclure que, pour $n \geq 4$,

$$\sum_{i=0}^{n-2} E_i = \frac{(n-1)^2}{4}(E_{n-1} - \frac{n-2}{n-1}E_{n-2}),$$

et, en remplaçant cette dernière expression dans (4.2), on a

$$E_n = \frac{n-1}{n}E_{n-1} + \frac{4}{n^2}E_{n-1} + \frac{(n-1)^2}{n^2}(E_{n-1} - \frac{n-2}{n-1}E_{n-2})$$

$$= \frac{2n^2 - 3n + 5}{n^2}E_{n-1} - \frac{n^2 - 3n + 2}{n^2}E_{n-2}.$$

Ceci nous permet d'écrire

$$E_n - E_{n-1} = \frac{3}{n^2}E_{n-1} + \frac{(n-1)(n-2)}{n^2}(E_{n-1} - E_{n-2}).$$

Posons maintenant $F_n = E_n - E_{n-1}$, pour $n \geq 1$, et réécrivons la dernière équation. Nous obtenons, pour $n \geq 4$,

$$F_n = \frac{(n-1)(n-2)}{n^2} F_{n-1} + \frac{3}{n^2} \sum_{i=1}^{n-1} F_i. \tag{4.3}$$

De ceci, on conclut que, pour $n \geq 5$,

$$\sum_{i=1}^{n-2} F_i = \frac{(n-1)^2}{3} \left(F_{n-1} - \frac{(n-2)(n-3)}{(n-1)^2} F_{n-2} \right)$$

Si on substitue cette dernière équation dans l'équation (4.3), on trouve

$$F_n = \frac{2n^2 - 5n + 6}{n^2} F_{n-1} - \frac{n^2 - 5n + 6}{n^2} F_{n-2}.$$

On peut maintenant écrire que

$$F_n - F_{n-1} = \frac{(n-2)(n-3)}{n^2} (F_{n-1} - F_{n-2}), \text{ pour } n \geq 5.$$

Comme précédemment, posons $G_n = F_n - F_{n-1}$, et on a alors que, pour $n \geq 5$,

$$G_n = \frac{(n-2)(n-3)}{n^2} G_{n-1} = \frac{(n-2)(n-3)}{n^2} \frac{(n-3)(n-4)}{(n-1)^2} G_{n-2} = \ldots = \frac{288}{n^2(n-1)^2(n-2)} G_4.$$

Puisque $G_4 = F_4 - F_3 = E_4 - 2E_3 + E_2 = \frac{37}{18} - 2\frac{14}{9} + 1 = -\frac{1}{18}$, on obtient

$$G_n = \frac{-16}{n^2(n-1)^2(n-2)}, \text{ pour } n \geq 4.$$

On peut maintenant trouver une expression pour F_n. En effet, on a que

$$
\begin{aligned}
F_n &= F_3 + \sum_{i=4}^{n} G_i = \frac{5}{9} - 16 \sum_{i=4}^{n} \frac{1}{i^2 (i-1)^2 (i-2)} \\
&= -16 \sum_{i=4}^{n} \left(\frac{-1/2}{i^2} - \frac{5/4}{i} - \frac{1}{(i-1)^2} + \frac{1}{i-1} + \frac{1/4}{i-2} \right) \\
&= 8\left(H_n^{(2)} - 1 - \frac{1}{4} - \frac{1}{9} \right) + 20\left(H_n - 1 - \frac{1}{2} - \frac{1}{3} \right) + 16\left(H_n^{(2)} - 1 - \frac{1}{4} \right) \\
&\qquad -16\left(H_{n-1} - 1 - \frac{1}{2} \right) - 4\left(H_{n-1} - 1 \right) \\
&= 24 H_n^{(2)} - 39 + \frac{4}{n^2 (n-1)} (6n^2 - 9n + 4).
\end{aligned}
$$

Cette dernière équation est bonne pour $n \geq 3$. Enfin, pour calculer E_n, on utilise la même technique que pour F_n, c'est-à-dire,

$$
\begin{aligned}
E_n &= E_2 + \sum_{i=3}^{n} F_i = 1 + \sum_{i=3}^{n} \left(24 H_i^{(2)} - 39 + \frac{4}{i^2 (i-1)} (6i^2 - 9i + 4) \right) \\
&= 1 + \sum_{i=3}^{n} \left(24 H_i^{(2)} - 39 + \frac{20}{i} - \frac{16}{i^2} + \frac{4}{i-1} \right) \\
&= 1 + 24\left((n+1) - H_n^{(2)} - H_n - 1 - 1 - \frac{1}{4} \right) - 39(n-2) + 20\left(H_n - 1 - \frac{1}{2} \right) \\
&\qquad -16\left(H_n^{(2)} - 1 - \frac{1}{4} \right) + 4\left(H_n - \frac{1}{n} - 1 \right) \\
&= 8 H_n^{(2)} (3n+1) + 11 - 39n - \frac{4}{n}.
\end{aligned}
$$

Cette expression est bonne pour $n \geq 2$. ■

Nous allons démontrer maintenant le deuxième théorème, dont les premières étapes de la démarche pourraient être utilisées pour étudier les cas où $i = 2$, 3 et 4. Nous allons voir que le cas qui nous intéresse ici est assez ardu à calculer et que l'aide du programme

Maple s'est avérée une aide très précieuse. On trouvera les détails du programme en annexe.

Preuve du théorème 4.4.

Comme pour le nombre moyen de feuilles, on laissera tomber l'indice i. On a donc remplacé $E_{1,n}$ par E_n. La démarche calculatoire est la même que pour le nombre moyen de feuilles. Nous en indiquerons alors les principales étapes. On a donc que $E_0 = 0$, et

$$E_n = a_n + \frac{4}{n}\sum_{j=1}^{n-1} E_j(H_n - H_j), n \geq 2,$$

$$= a_n + \frac{n-1}{n}(E_{n-1} - a_{n-1}) + \frac{4}{n^2}\sum_{j=1}^{n-1} E_j, n \geq 3.$$

$$E_n - a_n = \frac{2n^2 - 3n + 5}{n^2}(E_{n-1} - a_{n-1}) - \frac{(n-1)(n-2)}{n^2}(E_{n-2} - a_{n-2}) + \frac{4}{n^2}a_{n-1}, n \geq 4.$$

On pose maintenant $E_n^* = E_n - a_n$, et on obtient

$$E_n^* - E_{n-1}^* = \frac{3}{n^2}E_{n-1}^* + \frac{4}{n^2}a_{n-1} + \frac{(n-1)(n-2)}{n^2}(E_{n-1}^* - E_{n-2}^*), n \geq 4.$$

Ensuite, on pose $F_n = E_n^* - E_{n-1}^*$, ce qui nous permet d'écrire

$$F_n = \frac{3}{n^2}\sum_{i=2}^{n-1} F_i + \frac{4}{n^2}f_{n-1} + \frac{(n-1)(n-2)}{n^2}F_{n-1}, n \geq 4,$$

puisque, pour tous les cas, sauf les feuilles,

$$\sum_{i=2}^{n-1} F_i = \sum_{i=2}^{n-1}(E_i^* - E_{i-1}^*) = E_{n-1}^* - E_1^* = E_{n-1}^*.$$

On obtient donc,

$$F_n = \frac{4}{n^2}(a_{n-1} - a_{n-2}) + \frac{2n^2 - 5n + 6}{n^2}F_{n-1} - \frac{(n-1)(n-2)}{n^2}F_{n-2}, n \geq 5.$$

On pose $G_n = F_n - F_{n-1}$ et $g_n = a_n - a_{n-1}$, ce qui donne

$$\begin{aligned} G_n &= \frac{4}{n^2} g_{n-1} + \frac{(n-2)(n-3)}{n^2} G_{n-1} \\ &= \frac{4}{n^2(n-1)^2(n-2)} \sum_{i=4}^{n-1} i^2(i-1)g_i + \frac{(16)(9)(2)}{n^2(n-1)^2(n-2)} G_4, \; n \geq 5. \end{aligned}$$

Jusqu'à maintenant, nous ne nous sommes pas souciés de la valeur de a_n; la dernière expression obtenue est bonne pour le cas à 1, 2, 3 ou 4 enfants. Spécifions maintenant la valeur de a_n afin de calculer le nombre moyen de nœuds à un enfant. Comme on l'a vu précédemment, $a_n = \frac{4}{n^2}$, pour $n > 1$. De plus,

$$G_4 = F_4 - F_3 = E_4^* - 2E_3^* + E_2^* = E_4 - 2E_3 + E_2 - a_4 + 2a_3 - a_2 = -\frac{1}{2}$$

et

$$g_n = a_n - a_{n-1} = \frac{4(1-2n)}{n^2(n-1)^2}.$$

On obtient donc

$$\begin{aligned} G_n &= \frac{-8}{n^2(n-1)^2(n-2)} \left\{ 2 \sum_{i=4}^{n-1} \left(1 + \frac{i}{i-1} \right) + 3 \right\} \\ &= -\frac{16(2n + H_{n-2} - 8)}{n^2(n-1)^2(n-2)}, \; n \geq 4. \end{aligned}$$

Maintenant, pour revenir à l'objet qui nous intéresse, il suffit de voir que

$$E_n = E_n^* + a_n = \sum_{i=4}^{n} F_i + E_3^* + a_n = \sum_{i=4}^{n} \left(F_3 + \sum_{j=4}^{i} G_i \right) + E_3^* + a_n.$$

Puisque $E_3^* = E_3 - a_3 = 8/9 - 4/9 = 4/9$, et que $F_3 = E_3^* - E_2^* = 4/9 - E_2 + a_2 = 4/9$, on obtient

$$E_n = \frac{4}{n^2}+\frac{4}{9}+\sum_{i=4}^{n}\left(\frac{4}{9}+\sum_{j=4}^{i}-\frac{16(2j+H_{j-2}-8)}{j^2(j-1)^2(j-2)}\right)$$

$$= \frac{4}{n^2}+\frac{4}{9}(n-2)-16\sum_{j=4}^{n}\frac{(n-j+1)(2j+H_{j-2}-8)}{j^2(j-1)^2(j-2)},\ n\geq 4.$$

Finalement, en utilisant Maple, dont le programme est donné en annexe, on obtient

$$E_n = \frac{4}{n^2}+\frac{16}{n}-\frac{1171}{27}+\frac{2393}{9}n+4\left(6-\frac{1}{n}\right)H_n-52(3n+1)H_n^{(2)}$$

$$-8(3n+1)H_n^{(3)}+8(3n+1)\sum_{i=4}^{n}\frac{H_i}{i^2},\ n\geq 3.$$ ■

4.3 Arbres hyperquaternaires et étude de la frange.

Dans leur article sur la frange d'un arbre binaire, Poblete et Munro (1985) développent et analysent une heuristique visant à améliorer le temps de fouille dans un arbre binaire. La frange qu'ils considèrent est, pour un arbre binaire donné, l'ensemble de tous les sous-arbres obtenus en enlevant de l'arbre considéré tous les nœuds internes, sauf ceux ayant au moins un nœud externe comme enfant. Autrement dit, on enlève les nœuds ayant au plus un enfant. Comme mentionné au début de ce chapitre, leur heuristique consiste à effectuer un rebalancement au bas de l'arbre lorsque le nœud ajouté, toujours dans la frange, produit localement un débalancement, selon certains critères. Donc, au fur et à mesure que l'on construit l'arbre binaire de fouille, la structure de l'arbre est changée par rapport à celle que l'on obtiendrait si l'on ajoutait les nœuds sans rebalancement.

Pour les arbres binaires de fouille, sans rebalancement, l'espérance du nombre de comparaisons pour un arbre à n nœuds internes, pour une fouille sans succès est $2H_{n+1}-2$, avec une variance de $2H_{n+1}-4H_{n+1}^{(2)}+2$ (voir Knuth, 1973.) Ceci est équi-

valent à la profondeur moyenne du nœud ajouté à un arbre de n nœuds, le nœud n'étant pas préalablement dans l'arbre. Poblete et Munro montrent que cette espérance s'améliore lorsque l'on utilise leur heuristique. En effet, cette espérance devient égale à $\frac{12}{7}H_{n+1} - \frac{75}{49}$ et la variance devient $\frac{300}{343}H_{n+1} - \frac{144}{49}H^{(2)}_{n+1} + \frac{5956}{2401} + \frac{2304}{343}\frac{(n-6)!}{(n+1)!}$, pour $n \geq 6$. Puisque pour un arbre binaire complet, où tous les niveaux sont remplis, la profondeur du $n+1^{\text{ième}}$ nœud dans un arbre à n nœuds est de $\lfloor \log_2(n+1) \rfloor$, on a que les arbres produits par l'heuristique se situent à mi-chemin entre les arbres parfaitement balancés, relativement à l'espérance du nombre de comparaisons effectuées lors de l'ajout d'un nœud.

Toute l'étude est basée sur le fait que tous les nœuds externes de l'arbre binaire considéré sont équiprobables quant à l'endroit où un nœud, non élément de l'arbre, serait ajouté, en considérant le modèle uniforme bien sûr. Ceci est effectivement le cas pour les arbres binaires, qui constituent le cas des arbres hyperquaternaires de dimension un. Mais, pour les arbres hyperquaternaires de dimension supérieure à un, ceci n'est plus vrai. Considérons la probabilité suivante: P = P (le troisième nœud ajouté est dans le même hyperquadrant que le deuxième nœud). Si tous les nœuds externes étaient équiprobables en tant que candidat possible pour le troisième nœud, on aurait que

$$\mathrm{P} = \frac{2^d}{2^d + 2^d - 1} = \frac{2^d}{2^{d+1} - 1},$$

où d est la dimension considérée. Cependant, si on calcule P, sachant que les points générés sont uniformes $[0,1]^d$, et où les coordonnées sont indépendantes, on a que

$$\begin{aligned} \mathrm{P} &= \sum_{i=1}^{2^d} P \text{ (les noeuds 2 et 3 tombent dans le } i^{\text{ième}} \text{ hyperquadrant)} \\ &= 2^d P \text{ (les noeuds 2 et 3 tombent dans le troisième hyperquadrant)} \\ &= 2^d E\left((U_1 U_2 \ldots U_d)^2\right) = 2^d E\left(U_1^2 U_2^2 \ldots U_d^2\right) = 2^d (E\left(U^2\right))^d = \left(\frac{2}{3}\right)^d < \frac{2^d}{2^{d+1} - 1}. \end{aligned}$$

Ici, $U_1, U_2, \ldots, U_d, U$, sont des variables aléatoires uniformes [0,1] indépendantes.

On en conclut donc que l'on ne peut considérer les $(2^d-1)(n-1)+1$ nœuds externes de l'arbre hyperquaternaire comportant $n-1$ nœuds internes comme équiprobables pour le $n^{\text{ième}}$ nœud.

4.4 Étude empirique

Puisque l'analyse théorique ne nous a pas permis jusqu'à maintenant d'obtenir de résultats intéressants concernat l'espérance du nombre de nœuds, dans un arbre hyperquaternaire à deux dimensions, ayant deux, trois ou quatre enfants, nous avons calculé numériquement ces espérances. En disposant des formules exactes pour les probabilités impliquées dans ces calculs, nous pouvons calculer la valeur exacte de ces espérances. Le tableau qui suit nous donne ces résultats. Ces calculs ont été effectués à l'aide d'un programme Pascal, dont on peut trouver le code en annexe.

n	$a_{0,n}$	$a_{1,n}$	$a_{2,n}$	$a_{3,n}$	$a_{4,n}$
10	0.00000000	0.04000000	0.32634921	0.39888889	0.23476190
20	0.00000000	0.01000000	0.18059936	0.32834921	0.48105143
30	0.00000000	0.00444444	0.12461010	0.27011141	0.60083404
40	0.00000000	0.00250000	0.09506779	0.23021873	0.67221348
50	0.00000000	0.00160000	0.07683411	0.20146821	0.72009768
60	0.00000000	0.00111111	0.06446396	0.17973010	0.75469482
70	0.00000000	0.00081633	0.05552240	0.16266832	0.78099295
80	0.00000000	0.00062500	0.04875812	0.14888272	0.80173416
90	0.00000000	0.00049383	0.04346247	0.13748561	0.81855810
100	0.00000000	0.00040000	0.03920413	0.12788685	0.83250902
110	0.00000000	0.00033058	0.03570557	0.11967838	0.84428547
120	0.00000000	0.00027778	0.03278015	0.11256864	0.85437343
130	0.00000000	0.00023669	0.03029772	0.10634346	0.86312213
140	0.00000000	0.00020408	0.02816476	0.10084182	0.87078934
150	0.00000000	0.00017778	0.02631232	0.09594017	0.87756973
160	0.00000000	0.00015625	0.02468850	0.09154204	0.88361321
170	0.00000000	0.00013841	0.02325342	0.08757092	0.88903724
180	0.00000000	0.00012346	0.02197601	0.08396534	0.89393519
190	0.00000000	0.00011080	0.02083162	0.08067528	0.89838230
200	0.00000000	0.00010000	0.01980051	0.07765960	0.90243989

Tableau 4.1 – Probabilités que la racine ait entre 0 et 4 enfants dans un arbre quaternaire de n nœuds.

n	$E_{0,n}/n$	$E_{1,n}/n$	$E_{2,n}/n$	$E_{3,n}/n$	$E_{4,n}/n$
10	0.49424	0.24689	0.15308	0.07622	0.02957
20	0.48638	0.24351	0.14906	0.07582	0.04523
30	0.48373	0.24227	0.14807	0.07546	0.05047
40	0.48240	0.24163	0.14762	0.07527	0.05309
50	0.48161	0.24124	0.14735	0.07515	0.05465
60	0.48108	0.24098	0.14718	0.07507	0.05570
70	0.48070	0.24079	0.14706	0.07501	0.05644
80	0.48041	0.24065	0.14697	0.07497	0.05700
90	0.48019	0.24054	0.14690	0.07493	0.05744
100	0.48001	0.24045	0.14685	0.07490	0.05778
110	0.47987	0.24038	0.14680	0.07488	0.05807
120	0.47975	0.24032	0.14677	0.07486	0.05831
130	0.47964	0.24027	0.14674	0.07485	0.05851
140	0.47956	0.24022	0.14671	0.07483	0.05868
150	0.47948	0.24018	0.14669	0.07482	0.05883
160	0.47941	0.24015	0.14666	0.07481	0.05896
170	0.47936	0.24012	0.14665	0.07480	0.05907
180	0.47930	0.24009	0.14663	0.07480	0.05918
190	0.47926	0.24007	0.14662	0.07479	0.05927
200	0.47921	0.24005	0.14660	0.07478	0.05935

Tableau 4.2 – Proportion du nombre de nœuds ayant entre 0 et 4 enfants dans un arbre quaternaire de n nœuds.

On peut conclure, à la lumière de ces résultats qu'un peu moins de la moitié des nœuds seront des feuilles. En fait, ceci corrobore les résultats théoriques énoncés au corollaire 4.3, qui donne comme valeur asymptotique $4\pi^2 - 39 \approx 0.4784$. La proportion de nœuds à enfant unique est de moins de 1/4 asymptotiquement, ce qui va dans le même sens que les résultats énoncés au théorème 4.4. On remarque de plus que le nombre de nœuds à deux enfants, quoiqu'assez faible, est deux fois plus élevé que le nombre de nœuds à trois enfants. Le nombre de nœuds à quatre enfants est d'environ 80% du nombre de nœuds à trois enfants, asymptotiquement.

Chapitre 5

Arbres pseudo-hyperquaternaires

5.1 Introduction

Une alternative intéressante des arbres hyperquaternaires est la structure d'arbre pseudo-hyperquaternaire introduite par Overmars et van Leeuwen (1982). Cette structure est en fait la version "fouille des feuilles" des arbres hyperquaternaires. Les points de l'ensemble, qui sont autant des nœuds internes que des feuilles dans les arbres hyperquaternaires sont représentés uniquement par des feuilles dans les arbres pseudo-hyperquaternaires, les nœuds internes étant des points non présents de l'ensemble de points. Nous pouvons donner une définition de ces arbres:

Définition: Un *arbre pseudo-hyperquaternaire de points* est soit une feuille, dans le cas où l'ensemble de points ne contient qu'un point, soit un arbre dont la racine n'est pas un point de l'ensemble, mais représente le point de rencontre de deux segments perpendiculaires, parallèles aux axes, qui séparent l'ensemble de points en 2^d sous-ensembles, chacun étant représenté par un arbre pseudo-hyperquaternaire de point.

La figure suivante nous donne un exemple d'arbre pseudo-hyperquaternaire de points à deux dimensions.

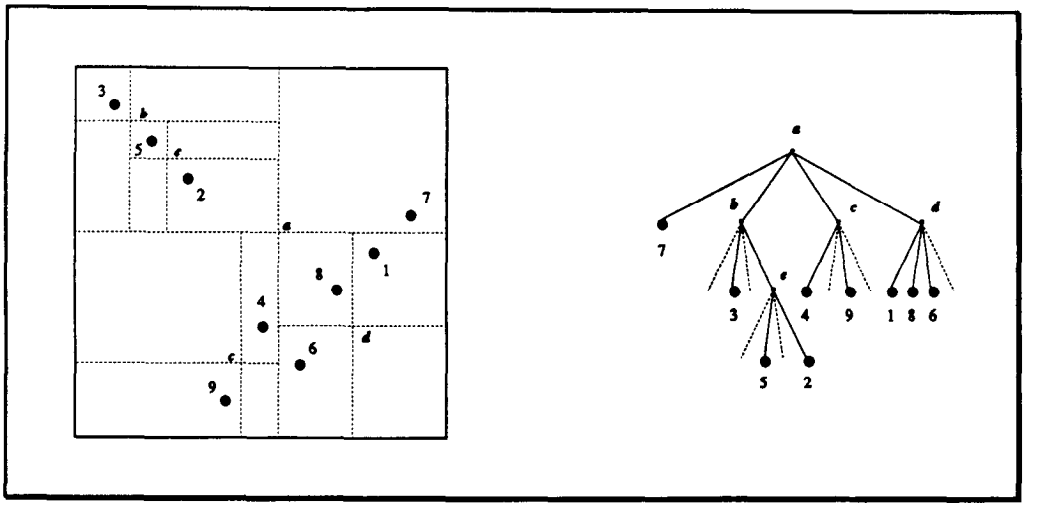


Fig. 5.1 – Arbre pseudo-hyperquaternaire à deux dimensions.

Que ce soit dans un contexte statique où nous disposons de tous les points avant de construire la structure, ou dans un contexte dynamique où les points sont ajoutés un à un, les arbres pseudo-hyperquaternaires permettent la suppression de points beaucoup plus facilement que dans le cas des arbres hyperquaternaires. De plus, les opérations que l'on peut effectuer à l'aide des arbres hyperquaternaires, peuvent tout aussi facilement être effectuées lorsque les points sont représentés avec un arbre pseudo-hyperquaternaire.

Nous verrons comment on construit un arbre pseudo-hyperquaternaire et comment on ajoute et enlève des points dans cette structure. Nous verrons que (Overmars, 1983) si un ensemble de n points nous est donné, en d dimensions, on peut construire un arbre pseudo-hyperquaternaire, de hauteur d'au plus $\lceil \log_{d+1} n \rceil$, en un temps $O(n \log n)$. Le temps requis pour construire un arbre hyperquaternaire, étant donnés n points en d dimensions est aussi de $O(n \log n)$. Ceci s'avère intéressant car nous savons que même l'arbre hyperquaternaire optimal construit pour un ensemble de points peut avoir une hauteur de $\log_2 n$ dans le cas, par exemple, où les points sont sur une diagonale. Dans un

contexte dynamique, nous verrons aussi qu'il existe un algorithme effectuant N insertions et suppressions dans un arbre pseudo-hyperquaternaire de façon à ce que la hauteur soit d'au plus $\log_{d+1-\delta} n + O(1)$, n étant le nombre courant de points dans l'arbre, δ étant une constante fixée, $0 < \delta < d$. Le temps moyen de transaction est borné par $O(\frac{1}{\delta} \log^2 N)$.

5.2 Construction des arbres pseudo-hyperquaternaires

Les résultats importants relatifs aux arbres pseudo-hyperquaternaires, et qui peuvent être retrouvés, ainsi que leurs démonstrations, dans Overmars et van Leeuwen (1982) ainsi que dans Overmars (1983) sont énoncés dans les sections suivantes. Nous y traitons des cas statique et dynamique. On doit assumer qu'aucun point ne peut avoir la même valeur pour une coordonnée donnée qu'un autre point de l'ensemble, et ce, pour toutes les coordonnées.

Contexte statique

Le théorème qui suit, ainsi que le corollaire en découlant, nous assure d'une méthode de construction d'un arbre pseudo-hyperquaternaire pour un ensemble de points fixés.

Théorème 5.1 (Overmars et van Leeuwen, 1982)

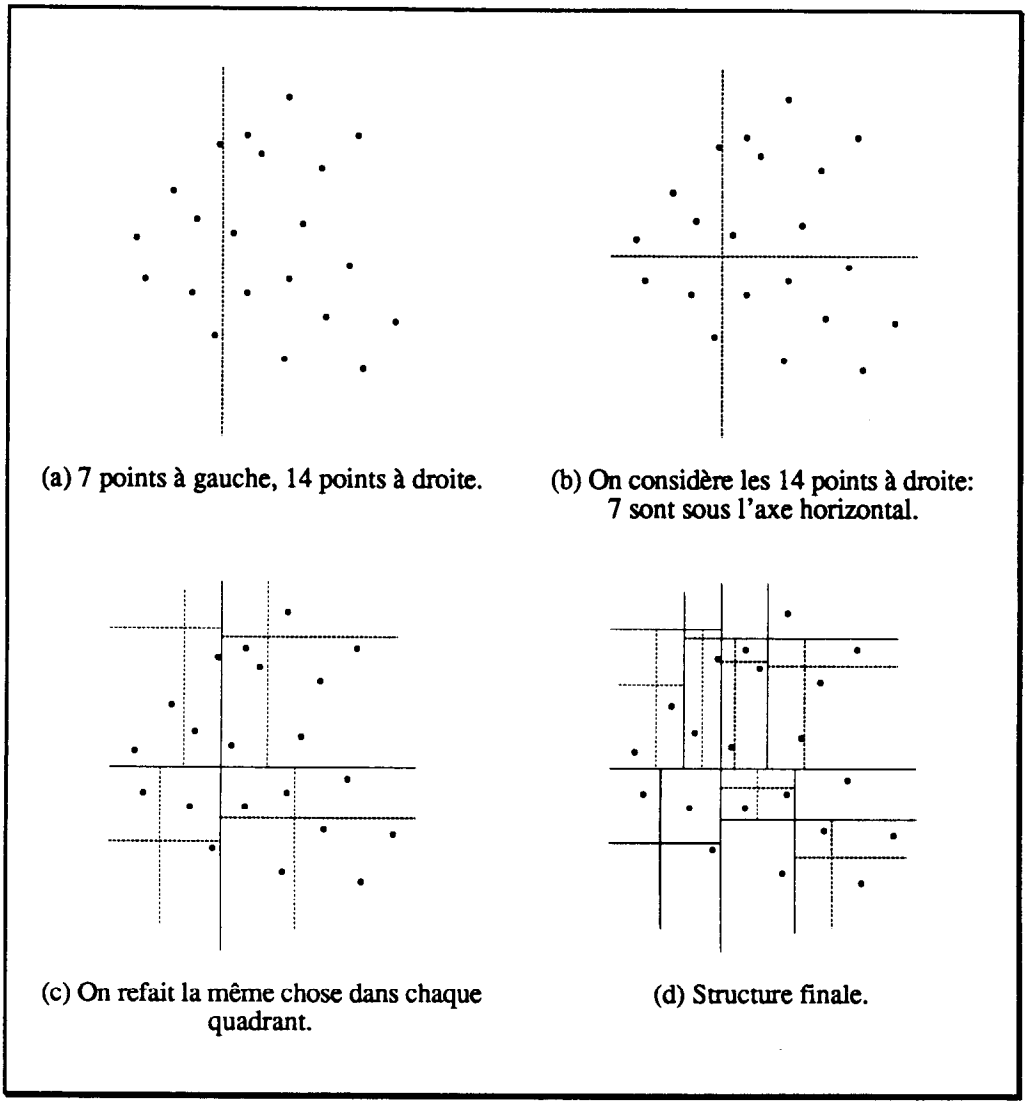
Étant donné un ensemble de n points $X_1, X_2, \ldots, X_n$ dans un espace d-dimensionnel, il existe un point séparateur $h = (h_1, h_2, \ldots, h_d)$ tel que chaque hyperquadrant induit par h contient au plus $\left\lceil \frac{1}{d+1} n \right\rceil$ points.

Ce théorème amène le corollaire suivant.

Corollaire 5.2 (Overmars et van Leeuwen, 1982)

Étant donné un ensemble de n points dans un espace d-dimensionnel, il existe un arbre pseudo-hyperquaternaire représentant cet ensemble, de hauteur d'au plus $\lceil \log_{d+1} n \rceil$.

La façon de construire un arbre pseudo-hyperquaternaire avec n points fixés est la suivante: On choisit h_1 tel que $\left\lceil \frac{1}{d+1} n \right\rceil$ points de l'ensemble ont leur coordonnée x_1 plus petite que h_1, les $\left\lfloor \frac{d}{d+1} n \right\rfloor$ autres points ayant leur coordonnée x_1 plus grande que h_1. Il reste $d-1$ coordonnées à déterminer. On ne considère maintenant que les points ayant leur coordonnée x_1 plus grande que h_1. On choisit h_2 de façon à avoir $\left\lceil \frac{1}{d+1} n \right\rceil$ points de coordonnée x_2 plus petite que h_2, les $\left\lfloor \frac{d-1}{d+1} n \right\rfloor$ points restants ayant leur coordonnée x_2 plus grande que h_2. On continue de la même manière jusqu'à ce qu'on ait déterminé les d coordonnées de h. On peut voir que les 2^d quadrants induits par h contiennent chacun au plus $\left\lceil \frac{1}{d+1} n \right\rceil$ points. On retrouve un exemple de cette construction dans le cas à deux dimensions dans la figure suivante.



(a) 7 points à gauche, 14 points à droite.

(b) On considère les 14 points à droite: 7 sont sous l'axe horizontal.

(c) On refait la même chose dans chaque quadrant.

(d) Structure finale.

Fig. 5.2 – Construction d'un arbre pseudo-hyperquaternaire à 2 dimensions dans le cas statique.

Le résultat du corollaire est optimal comme on peut le constater par le théorème suivant.

Théorème 5.3 (Overmars et van Leeuwen, 1982)

Il existe un ensemble de n points $X_1, X_2, \ldots, X_n$ dans un espace d-dimensionnel tel qu'aucun arbre pseudo-hyperquaternaire représentant cet ensemble de points peut avoir une hauteur plus petite que $\lceil \log_{d+1} n \rceil$.

Par exemple, si tous les n points sont sur une diagonale, on aura un arbre de hauteur $\lceil \log_{d+1} n \rceil$. Le théorème suivant nous assure qu'on peut construire efficacement un tel arbre pseudo-hyperquaternaire.

Théorème 5.4 (Overmars et van Leeuwen, 1982)

Étant donnés n points dans un espace d-dimensionnel, un arbre pseudo-hyperquaternaire de hauteur d'au plus $\lceil \log_{d+1} n \rceil$ représentant l'ensemble de points peut être construit en un temps $O(n \log n)$.

La façon de construire l'arbre pseudo-hyperquaternaire dont il est question dans le théorème précédent est décrite suite au corollaire de la page précédente.

Contexte dynamique

On peut utiliser la même structure, soit celle d'arbre pseudo-hyperquaternaire pour insérer un à un des points. La façon de construire l'arbre est de déterminer dans quel hyperquadrant, par rapport à la racine, se trouve le point à insérer. Si l'hyperquadrant est

vide, on y "place" le point. Si l'hyperquadrant contient un point, alors il faut choisir un point dans cet hyperquadrant, différent des deux points qui servira à séparer cet hyperquadrant en 4 sous-hyperquadrants de façon à ce que les deux points se retrouvent dans deux hyperquadrants différents. Un choix facile pour le point de séparation est le point médian. Si l'hyperquadrant est déjà subdivisé, on réapplique récursivement la procédure en prenant comme racine, la racine de l'hyperquadrant en question. Si on veut plutôt enlever un point, il suffit de descendre jusqu'au nœud parent du point et d'enlever le point. S'il ne reste qu'un seul point ou un seul sous-arbre au même niveau que le nœud enlevé, alors le parent devient le nœud ou le sous-arbre qui reste. On peut retrouver ces deux algorithmes dans ce qui suit:

```
Enlever un point p

parent ← noeud parent de p
p ← Λ
Si le nombre d'enfants de parent = 1
    alors parent ← enfant
```

```
Ajouter un point p

parent ← noeud futur parent de p
Si parent est une feuille
    alors m ← point "entre" p et parent
          enfants ( m ) ← { p, parent }
          parent ← m
    sinon enfants ( parent ) ← { enfants ( parent ), p }
```

La figure suivante donne un exemple d'application de ces algorithmes.

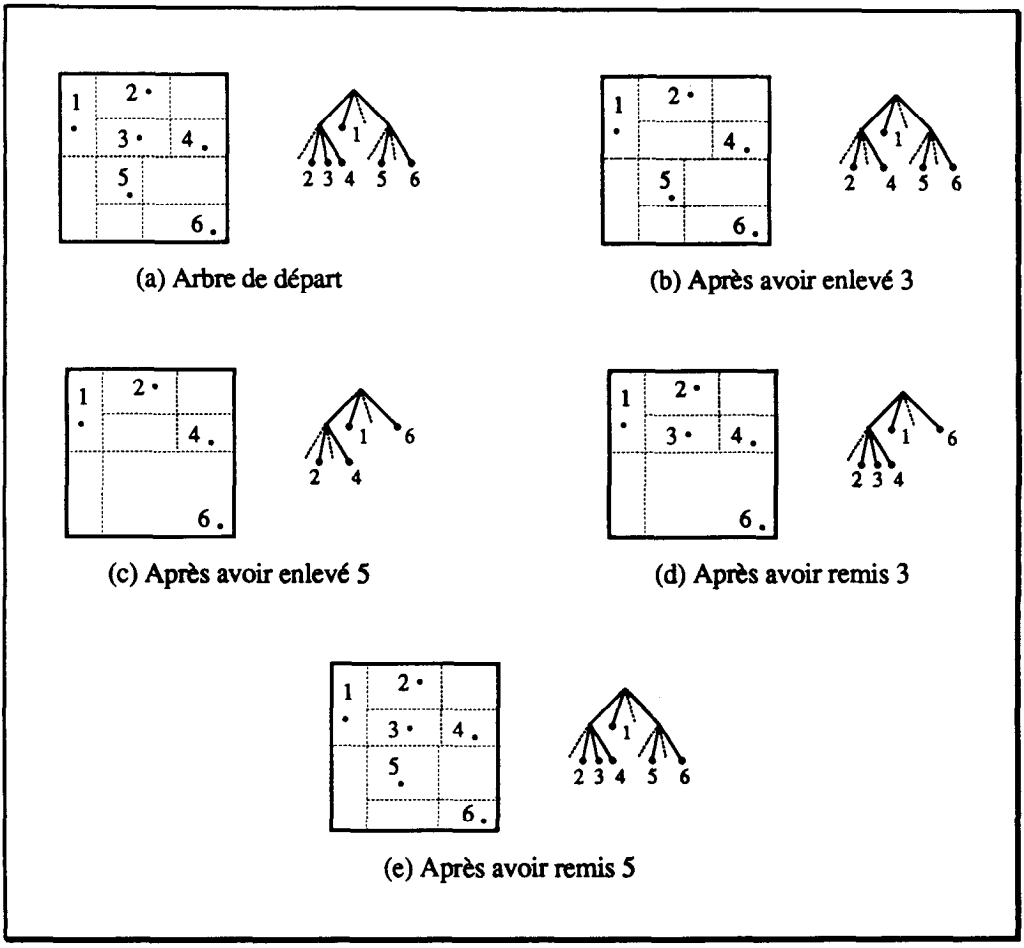


Fig. 5.3 – Suppressions et ajouts de points dans un arbre pseudo-hyperquaternaire à deux dimensions.

Cependant, cette façon de procéder ne nous permet pas de conserver la propriété selon laquelle l'arbre a une hauteur d'au plus $\lceil \log_{d+1} n \rceil$, comme indiqué par un théorème de la section précédente. Nous pouvons néanmoins obtenir un arbre ayant une hauteur qui s'y rapproche en effectuant, à un moment jugé opportun, une restructuration locale, comme on peut le constater par le théorème suivant.

Théorème 5.5 (Overmars et van Leeuwen, 1982)

Pour toute constante fixée δ telle que $0 < \delta < d$, il existe un algorithme permettant d'effectuer, dans un espace d–dimensionnel, N insertions et suppressions dans un arbre pseudo-hyperquaternaire initialement vide tel que sa hauteur est toujours d'au plus $\log_{d+1-\delta} n + O(1)$ et où le temps moyen de transaction est borné par $O(\frac{1}{\delta}\log^2 N)$. Ici, n désigne le nombre courant de points, $0 \leq n \leq N$.

Ceci nous dit que plus on veut avoir un arbre se rapprochant de l'idéal, soit ayant une hauteur d'au plus $\lceil \log_{d+1} n \rceil$, plus on aura à "travailler fort". En fait, l'algorithme mentionné dans le théorème est le suivant: Pour chaque nœud interne h, ayant un total de k points dans ses sous-arbres, on fait en sorte que chacun de ces sous-arbres contienne au plus $\left\lceil \frac{1}{d+1-\delta} k \right\rceil$ points. Si, lorsque l'on ajoute ou l'on enlève un point, la condition précédente n'est plus respectée, on cherche sur le chemin du nœud, ajouté ou enlevé, vers la racine, celui, le plus élevé où le débalancement persiste. Appelons ce nœud h. On reconstruit alors l'arbre dont la racine est h, en utilisant l'algorithme décrit à la section précédente.

5.3 Étude théorique et empirique des arbres pseudo-quaternaires aléatoires dans le cas dynamique.

Dans cette section, nous présentons les résultats nouveaux relatifs aux arbres pseudo-hyperquaternaires à deux dimensions dans le cas dynamique. Nous allons démontrer le théorème suivant qui nous donne le $m^{\text{ième}}$ moment non-centré de la variable aléatoire D_n, la profondeur du dernier nœud ajouté dans un arbre pseudo-hyperquaternaire

de n nœuds.

Théorème 5.6

$$\mu_{n,m} = \sum_{j=0}^{m-1} \binom{m}{j} \mu_{n,j} (-1)^{m-1-j} + \frac{2}{n-2} \sum_{i=1}^{n-2} i T_n(i)(\mu_{i+1,m} + \mu_{i,m}), \text{ pour } n \geq 3,$$

où $\mu_{n,0} = 1$ pour $n \geq 0$, $\mu_{0,m} = 0$ et $\mu_{1,m} = 0$ pour $m > 0$, $\mu_{2,m} = 1$ pour $m \geq 0$,

et où $T_n(i) = \binom{n-2}{i} \sum_{j=0}^{n-2-i} \binom{n-2-i}{j} (-1)^j \left[\frac{1 + 2^{2i+2j+2} - 2^{i+j+2}}{2^{2i+2j-2}(i+j+1)^2(i+j+2)^2} \right]$.

Puisque, comme mentionné précedemment, le choix des quatre quadrants se fait de façon à avoir deux points dans deux quadrants différents, on a choisi d'utiliser le point médian entre les deux points. Nous aurons besoin pour l'étude de la distribution de ce point médian. Ceci nous amène le lemme suivant.

Lemme 5.7

Soient X et Y, deux variables indépendantes uniformes [0,1]. Soit $M = \frac{1}{2}(X+Y)$.

Alors, la fonction de densité de M est donnée par

$$f_M(m) = \begin{cases} 4m & \text{pour } 0 \leq m \leq \frac{1}{2} \\ 4 - 4m & \text{pour } \frac{1}{2} \leq m \leq 1. \end{cases}$$

Preuve du lemme 5.7.

Voir Devroye (1986), page 22. ■

On aura besoin aussi de la distribution du nombre de points, parmi n, qui sont situés dans le troisième quadrant induit par le point médian issu des deux premiers points de l'ensemble de points. Pour ce faire, nous devons connaître au préalable la distribution de l'aire de ce troisième quadrant. Celle-ci est présentée dans le corollaire suivant.

Corollaire 5.8

Soient X_1 et X_2 deux points indépendants uniformes $[0,1]^2$, où $X_1 = (x_{1,1}, x_{1,2})$ et $X_2 = (x_{2,1}, x_{2,2})$. Soient $M_x = \frac{1}{2}(x_{1,1}+x_{2,1})$ et $M_y = \frac{1}{2}(x_{1,2}+x_{2,2})$ et $S = M_x M_y$. Alors, la fonction de densité de S nous est donnée par

$$f_S(s) = \begin{cases} 32s - 64s\log(2) - 16s\log(s) & \text{pour } 0 \le s \le \frac{1}{4} \\ 32 + 32\log(2) - 96s + 64s\log(2) + 16\log(s) + 48s\log(s) & \text{pour } \frac{1}{4} \le s \le \frac{1}{2}. \\ -32 + 32s - 16\log(s) - 16s\log(s) & \text{pour } \frac{1}{2} \le s \le 1 \end{cases}$$

Preuve du corollaire 5.8.

Nous allons calculer la densité cherchée par les techniques usuelles. En tenant compte des différentes bornes pour m et s on obtient

$$f_S(s) = \int_0^1 \frac{1}{m} f_{M_x}(s/m) f_{M_y}(m) dm$$

$$= \begin{cases} \displaystyle\int_s^{2s} \frac{1}{m} 4(1-s/m)4m\,dm + \int_{2s}^{\frac{1}{2}} \frac{1}{m} 4(s/m)4m\,dm + \int_{\frac{1}{2}}^{1} \frac{1}{m} 4(s/m)(4-4m)\,dm, & 0 \le s \le \frac{1}{4} \\ \displaystyle\int_s^{\frac{1}{2}} \frac{1}{m} 4(1-s/m)4m\,dm + \int_{\frac{1}{2}}^{2s} \frac{1}{m} 4(1-s/m)(4-4m)\,dm + \int_{\frac{1}{2}}^{1} \frac{1}{m} 4(s/m)(4-4m)\,dm, & \frac{1}{4} \le s \le \frac{1}{2}. \\ \displaystyle\int_s^{1} \frac{1}{m} 4(1-s/m)(4-4m)\,dm, & \frac{1}{2} \le s \le 1 \end{cases}$$

Il suffit maintenant d'effectuer les intégrales obtenues dans l'expression précédente. Nous les avons résolues à l'aide d'un programme Maple. On pourra trouver le programme Maple en annexe. ■

Passons maintenant à la distribution du nombre de points dans le troisième quadrant.

Corollaire 5.9

Soient $X_1, X_2, \ldots, X_n$, n points indépendants, de coordonnées indépendantes, uniformes $[0,1]^2$ et $M = \frac{1}{2}(X_1 + X_2)$. Soit X le nombre de points parmi $X_1, X_2, \ldots, X_n$ qui sont dans le troisième quadrant induit par M. Alors,

$$P(X = i) = \begin{cases} \frac{1}{2}T_n(0) & \text{pour } i = 0 \\ \frac{1}{2}T_n(n-2) & \text{pour } i = n-1 \\ \frac{1}{2}(T_n(i-1) + T_n(i)) & \text{pour } 0 < i < n-1, \end{cases}$$

$$\text{où } T_n(i) = \binom{n-2}{i} \sum_{j=0}^{n-2-i} \binom{n-2-i}{j} (-1)^j \left[\frac{1 + 2^{2i+2j+2} - 2^{i+j+2}}{2^{2i+2j-2}(i+j+1)^2(i+j+2)^2} \right].$$

Preuve du corollaire 5.9.

Tout d'abord, il faut constater que deux situations différentes peuvent se produire, et qu'elles influent sur le calcul des probabilités qui nous intéressent. Le premier cas est celui où X_1 ou X_2 se situe dans le premier quadrant induit par M, et par conséquent, l'autre se situe dans le troisième quadrant; le deuxième cas est celui où X_1 ou X_2 se situe dans le deuxième quadrant induit par M, et l'autre se situe donc dans le quatrième quadrant. On peut visualiser ces deux cas dans la figure suivante.

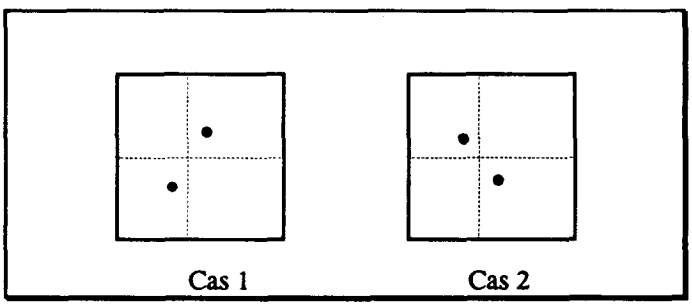


Fig. 5.4 – Deux cas pour deux points.

Il est assez aisé de montrer que ces deux cas sont équiprobables. Maintenant, posons

$$T_n(i) = \int_0^1 P(Y = i | Y \sim B(n-2,s)) f_S(s) ds,$$

où $B(n, p)$ signifie une binomiale de n essais et de probabilité de succès p. Alors, il découle immédiatement les expressions données dans l'énoncé du corollaire pour $P(X = i)$, pour $0 \leq i \leq n-1$. Pour ce qui est de $T_n(i)$, on a que

$$T_n(i) = \int_0^1 \binom{n-2}{i} s^i (1-s)^{n-2-i} f_S(s) ds$$
$$= \binom{n-2}{i} \sum_{j=0}^{n-2-i} \binom{n-2-i}{j} (-1)^j \int_0^1 s^{i+j} f_S(s) ds.$$

Pour trouver l'expression finale du corollaire, il faut utiliser la fonction de densité de S, donnée dans le corollaire précédent, $f_S(s)$. Le travail a été effectué en utilisant un programme Maple qu'on trouvera en annexe. ■

Intéressons nous maintenant à la quantité que nous avons étudiée pour les arbres hyperquaternaires, soit l'espérance de la profondeur du dernier nœud ajouté dans un arbre pseudo-hyperquaternaire. Comme effectué pour le cas des arbres hyperquaternaires, étudions tout d'abord $p_{n,l}$, la probabilité que le $n^{\text{ième}}$ nœud inséré le soit à une profondeur de l. Il est à remarquer que, contrairement à l'étude faite pour les arbres hyperquaternaires, nous ne considérerons que le cas bidimensionnel pour les arbres pseudo-hyperquaternaires. Nous avons le corollaire suivant.

Corollaire 5.10

$$p_{n,l} = \frac{2}{n-2} \sum_{i=1}^{n-2} i T_n(i)(p_{i+1,l-1} + p_{i,l-1}), \quad \text{pour } n \geq 3,$$

où $p_{1,0} = 1$, $p_{2,1} = 1$, $p_{n,0} = 0$ pour $n \neq 1$, $p_{n,l} = 0$ pour $l \geq n$.

Preuve du corollaire 5.10.

Nous savons que les deux premiers points de l'ensemble de points déterminent les quatre quadrants et que deux quadrants opposés contiennent un point (voir figure 5.4.) En

fixant un cas, la probabilité que le dernier nœud inséré soit dans un quadrant, fixé au préalable, les n points étant fixés est de $i/n-2$; la probabilité que le quadrant fixé contienne i points, excluant les deux premiers points, est $T_n(i)$. Maintenant, le quadrant fixé contient au départ, après détermination des quadrants, aucun ou un point selon le cas. S'il en contient un alors la probabilité que le dernier noeud inséré soit à une profondeur de l dans le quadrant fixé est de

$$\sum_{i=0}^{n-2} \frac{i}{n-2} T_n(i) p_{i+1,l-1},$$

tandis que si le quadrant fixé ne contient pas de point, la probabilité est de

$$\sum_{i=0}^{n-2} \frac{i}{n-2} T_n(i) p_{i,l-1},$$

car, si le point est à une profondeur de l dans l'arbre, il le sera à une profondeur de $l-1$ dans le sous-arbre associé au quadrant dans lequel il est tombé. Par symétrie, on obtient le résultat cherché. ■

Nous allons utiliser la technique de la fonction génératrice des moments pour calculer $E(D_n)$, où D_n est la profondeur du dernier nœud inséré dont la fonction de probabilité nous est donnée dans le corollaire précédent. Posons

$$\phi_n^{(m)}(t) = E(e^{tD_n^m}) \quad \text{et} \quad \mu_{n,m} = E(D_n^m).$$

Nous savons que

$$\phi_n(0) = 1 \quad \text{et} \quad \phi_n^{(m)}(t)\big|_{t=0} = \mu_{n,m}.$$

Le théorème suivant, annoncé au début de la section, nous donne une récurrence pour $\mu_{n,m}$. On remarquera une similitude avec l'expression obtenue pour les arbres hyperquaternaires.

Théorème 5.6

$$\mu_{n,m} = \sum_{j=0}^{m-1} \binom{m}{j} \mu_{n,j}(-1)^{m-1-j} + \frac{2}{n-2} \sum_{i=1}^{n-2} iT_n(i)(\mu_{i+1,m} + \mu_{i,m}), \text{ pour } n \geq 3,$$

où $\mu_{n,0} = 1$ pour $n \geq 0$, $\mu_{0,m} = 0$ et $\mu_{1,m} = 0$ pour $m > 0$, $\mu_{2,m} = 1$ pour $m \geq 0$.

Preuve du théorème 5.6.

Tout d'abord, trouvons une expression pour $\phi_n^{(m)}(t)$. Nous avons

$$\begin{aligned}
\phi_n(t) &= \sum_{l=0}^{n-1} e^{tl} p_{n,l} \\
&= \frac{2}{n-2} \sum_{l=1}^{n-1} e^{tl} \sum_{i=l-1}^{n-2} iT_n(i)(p_{i+1,l-1} + p_{i,l-1}) \\
&= \frac{2}{n-2} \sum_{i=0}^{n-2} iT_n(i) \sum_{l=1}^{i+1} e^{tl}(p_{i+1,l-1} + p_{i,l-1}) \\
&= \frac{2}{n-2} e^t \sum_{i=0}^{n-2} iT_n(i) \sum_{l=0}^{i} e^{tl}(p_{i+1,l} + p_{i,l}) \\
&= \frac{2}{n-2} e^t \sum_{i=0}^{n-2} iT_n(i)(\phi_{i+1}(t) + \phi_i(t)).
\end{aligned}$$

En dérivant le nombre de fois nécessaires, on obtient

$$\phi_n^{(m)}(t) = \frac{2}{n-2} e^t \sum_{j=0}^{m} \binom{m}{j} \sum_{i=0}^{n-2} iT_n(i)(\phi_{i+1}^{(j)}(t) + \phi_i^{(j)}(t)),$$

d'où

$$\mu_{n,m} = \frac{2}{n-2}\sum_{j=0}^{m}\binom{m}{j}\sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,j} + \mu_{i,j}).$$

L'expression finale du théorème s'obtient en effectuant une preuve par récurrence sur m. Pour $m = 1$ on obtient la même expression avec les deux formes de $\mu_{n,1}$, soit

$$\mu_{n,1} = 1 + \sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,1} + \mu_{i,1})$$

car

$$\sum_{i=0}^{n-2} iT_n(i) = E(T_n) = \int_0^1 f_S(s)E(Y|Y \sim B(n-2,s))ds = (n-2)\int_0^1 sf_S(s)ds = \frac{n-2}{4}.$$

La dernière égalité a été obtenue en utilisant Maple. Maintenant, supposons que l'expression suivante est vraie

$$\mu_{n,m-1} = \sum_{j=0}^{m-2}\binom{m-1}{j}\mu_{n,j}(-1)^{m-2-j} + \frac{2}{n-2}\sum_{i=1}^{n-2} iT_n(i)(\mu_{i+1,m-1} + \mu_{i,m-1}),$$

alors, on a que

$$\sum_{i=1}^{n-2} iT_n(i)(\mu_{i+1,m-1} + \mu_{i,m-1}) = \frac{n-2}{2}\left\{\mu_{n,m-1} - \sum_{j=0}^{m-2}\binom{m-1}{j}\mu_{n,j}(-1)^{m-2-j}\right\}. \quad (5.1)$$

On sait que

$$\mu_{n,m} = \frac{2}{n-2}\sum_{j=0}^{m}\binom{m}{j}\sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,j} + \mu_{i,j})$$

$$= \frac{2}{n-2}\left\{\sum_{j=0}^{m-1}\binom{m}{j}\sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,j} + \mu_{i,j}) + \sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,m} + \mu_{i,m})\right\}.$$

En utilisant (5.1), on obtient

$$\mu_{n,m} = \frac{2}{n-2}\left\{\sum_{j=0}^{m-1}\binom{m}{j}\frac{n-2}{2}\left[\mu_{n,j} - \sum_{i=0}^{j-1}\binom{j}{i}\mu_{n,i}(-1)^{j-1-i}\right] + \sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,m} + \mu_{i,m})\right\}$$

$$= \sum_{j=0}^{m-1}\binom{m}{j}\sum_{i=0}^{j}\binom{j}{i}\mu_{n,i}(-1)^{j-i} + \frac{2}{n-2}\sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,m}+\mu_{i,m})$$

$$= \sum_{i=0}^{m-1}\mu_{n,i}\sum_{j=i}^{m-1}\binom{j}{i}\binom{m}{j}(-1)^{j-i} + \frac{2}{n-2}\sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,m}+\mu_{i,m})$$

$$= \sum_{i=0}^{m-1}\binom{m}{i}\mu_{n,i}\sum_{j=i}^{m-1}\binom{m-i}{j-i}(-1)^{j-i} + \frac{2}{n-2}\sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,m}+\mu_{i,m})$$

$$= \sum_{i=0}^{m-1}\binom{m}{i}\mu_{n,i}\sum_{j=0}^{m-1-i}\binom{m-i}{j}(-1)^{j} + \frac{2}{n-2}\sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,m}+\mu_{i,m})$$

$$= \sum_{i=0}^{m-1}\binom{m}{i}\mu_{n,i}(-1)^{m-1-i} + \frac{2}{n-2}\sum_{i=0}^{n-2} iT_n(i)(\mu_{i+1,m}+\mu_{i,m}).$$ ■

Si on calcule $\mu_{n,1} = E(D_n)$ à l'aide de l'expression obtenue au théorème précédent, on obtient le corollaire suivant.

Corollaire 5.11

$$\mu_{n,1} = 1 + \frac{2}{n-2}\sum_{i=1}^{n-2} iT_n(i)(\mu_{i+1,1}+\mu_{i,1}) \text{ pour } n \geq 3,\ \mu_{1,1} = 0,\ \mu_{2,1} = 1.$$

$$\mu_{n,2} = 2\mu_{n,1} - 1 + \frac{2}{n-2}\sum_{i=1}^{n-2} iT_n(i)(\mu_{i+1,2}+\mu_{i,2}) \text{ pour } n \geq 3,\ \mu_{1,2} = 0,\ \mu_{2,2} = 1.$$

Preuve du corollaire 5.11.

Il suffit de remplacer $m = 1$ et $m = 2$ dans l'expression du théorème 5.6 et on obtient l'expression désirée. ■

Comme on peut le constater, l'expression obtenue pour $\mu_{n,1}$ est une récurrence difficile à résoudre. Cependant, à défaut de pouvoir la résoudre comme nous y étions parvenus pour le cas des arbres hyperquaternaires, nous la calculerons numériquement pour quelques valeurs de n, et nous verrons qu'empiriquement, la profondeur du dernier nœud inséré dans un arbre pseudo-hyperquaternaire semble être inférieure à celle du dernier nœud inséré dans un arbre hyperquaternaire de dimension deux.

Nous avons utilisé un programme Maple pour calculer l'espérance et la variance du dernier nœud ajouté dans les deux types d'arbres. Rappelons que pour le cas des arbres pseudo-quaternaires, les deux premiers moments non centrés sont

$$\mu_{n,1} = 1 + \frac{2}{n-2}\sum_{i=1}^{n-2} iT_n(i)(\mu_{i+1,1} + \mu_{i,1}) \text{ pour } n \geq 3,\ \mu_{1,1} = 0,\ \mu_{2,1} = 1 \text{ et}$$

$$\mu_{n,2} = 2\mu_{n,1} - 1 + \frac{2}{n-2}\sum_{i=1}^{n-2} iT_n(i)(\mu_{i+1,2} + \mu_{i,2}) \text{ pour } n \geq 3,\ \mu_{1,2} = 0,\ \mu_{2,2} = 1,$$

et que dans le cas des arbres quaternaires, les deux premiers moments non centrés sont

$$\mu_{n,1} = H_n - \tfrac{1}{6} - \tfrac{2}{3n}$$

$$\mu_{n,2} = H_n^2 + H_n^{(2)} + \frac{H_n}{6} - \frac{4H_n}{3n} + \frac{7}{9n} - \frac{77}{36}.$$

Le tableau 5.1 nous donne la comparaison entre les deux types d'arbres pour les espérances et les variances. Les valeurs tabulées sont $\mu_{n,1} / \log n$ et $\mu_{n,2} - \mu_{n,1}^2$ pour les deux modèles d'arbres. Rappelons que Maple calcule avec les valeurs exactes sans simplifier les fractions impliquées. Ce n'est qu'au moment de l'impression qu'il transforme les valeurs en notation point flottant. Il s'agit donc des "vraies" valeurs.

n	$\mu^*_{n,1} / \log_e(n)$	$\mu_{n,1} / \log_e(n)$	$\mu^*_{n,2} - \mu^{*2}_{n,1}$	$\mu_{n,2} - \mu^2_{n,1}$
5	1.216603939	1.232314287	.428320910	.531944444
10	1.119546271	1.170699371	.597938558	.898696303
15	1.081622719	1.147363472	.679705186	1.107949842
20	1.059830747	1.134193361	.735334043	1.255033072
25	1.045215990	1.125431685	.777338827	1.368546937
30	1.034513374	1.119046564	.810973272	1.461001708
35	1.026217943	1.114115149	.838994570	1.539001162
40	1.019529248	1.110150049	.863009327	1.606459927
45	1.013977363	1.106866355	.884025128	1.665890521
50	1.009265709	1.104084852	.902711893	1.719002069
55	1.005196492	1.101686366	.919536730	1.767010062
60	1.001631960	1.099588119	.934838250	1.810809860
65	.998472757	1.097730552	.948869881	1.851080095
70	.995645136	1.096069553	.961826367	1.888347425

Tableau 5.1 – Espérance et variance de D_n pour les arbres hyperquaternaires et pseudo-hyperquaternaires (*) de dimension 2.

On peut voir que la profondeur moyenne du dernier nœud inséré dans un arbre pseudo-quaternaire est inférieure à celle du dernier nœud inséré dans un arbre quaternaire. La variance de la même variable aléatoire semble encore être plus petite que celle des arbres quaternaires. En conclusion, les arbres pseudo-quaternaires semblent être une structure plus intéressante que les arbres quaternaires puisque la suppression de points dans une telle structure est plus facile et ne demande pas une restructuration de l'arbre, et qu'en moyenne, ce genre d'arbre donne une profondeur moyenne des nœuds plus petite.

Annexe A

Programmes

Programme 1 – Calcul de l'espérance du nombre de nœuds à un enfant dans un arbre hyperquaternaire de dimension deux.

```
### --- Calcul de 4/n^2+4/9*(n-2)-16* la somme pour j allant de 4 a n de
### ---        (n-j+1)*(2*j+h-8-1/j-1/(j-1))/(j^2*(j-1)^2*(j-2))
### ---  Correspondance entre les termes et les variables Maple:
### ---    h       ------>   Hn, somme des inverses
### ---    hdeux   ------>   H(2)n, somme de l'inverse des carres
### ---    htrois  ------>   H(3)n, somme de l'inverse des cubes
### ---    h1      ------>   Hn-1
### ---    h2      ------>   Hn-2
### ---    L       ------>   somme pour i de 4 a n de Hi/i^2
### ---    L1      ------>   comme L mais pour i de 3 a n-1
expression:=(n-j+1)*(2*j+h-8-1/j-1/(j-1))/(j^2*(j-1)^2*(j-2));
### ---  Fractions partielles
expression:=convert((n-j+1)*(2*j+h-8-1/j-1/(j-1))
                        /j/j/(j-1)^2/(j-2),parfrac,j);
### --- changements de variables pour avoir un polynome:
### ---    1/j    ---> i, 1/(j-1)    ---> i1, 1/(j-2)    ---> i2
### ---    1/j^2  ---> k, 1/(j-1)^2  ---> k1, 1/(j-2)^2  ---> k2
### ---    1/j^3  ---> m, 1/(j-1)^3  ---> m1, 1/(j-2)^3  ---> m2
expressionik:=subs((1/j=i,  1/(j-1)=i1,  1/(j-2)=i2,  1/j^2=k,
                    1/(j-1)^2=k1,  1/(j-2)^2=k2,  1/j^3=m,
                    1/(j-1)^3=m1,  1/(j-2)^3=m2),   expression);
### --- calcul des coefficients de chacune des 9 variables;
ci:=coeff(expressionik,i,1);        ci1:=coeff(expressionik,i1,1);
ci2:=coeff(expressionik,i2,1);      ck:=coeff(expressionik,k,1);
ck1:=coeff(expressionik,k1,1);      ck2:=coeff(expressionik,k2,1);
cm:=coeff(expressionik,m,1);        cm1:=coeff(expressionik,m1,1);
cm2:=coeff(expressionik,m2,1);
### --- changement au niveau des nombres harmoniques;
ci1:=subs(h=h1+i,ci1);   ci2:=subs(h=h2+i+i1,ci2);
ck1:=subs(h=h1+i,ck1);   ck2:=subs(h=h2+i+i1,ck2);
cm1:=subs(h=h1+i,cm1);   cm2:=subs(h=h2+i+i1,cm2);
### --- reformer l'expression
```

```
nexpression:=expand(i*ci+i1*ci1+i2*ci2+k*ck+k1*ck1+k2*ck2
                    +m*cm+m1*cm1+m2*cm2);
### --- remettre la formule en terme de j
nexpression:=subs({i=1/j,  i1=1/(j-1),  i2=1/(j-2),  k=1/j^2,
                   k1=1/(j-1)^2,  k2=1/(j-2)^2,  m=1/j^3,
                   m1=1/(j-1)^3,  m2=1/(j-2)^3},  nexpression):
nexpression:=normal(nexpression);
### --- fractions partielles
nexpression:=convert(nexpression,parfrac,j);
### --- revenir aux variables i, i1,... m2
nexpressionik:=subs({1/j=i,  1/(j-1)=i1,  1/(j-2)=i2,  1/j^2=k,
                     1/(j-1)^2=k1,  1/(j-2)^2=k2,  1/j^3=m,
                     1/(j-1)^3=m1,  1/(j-2)^3=m2 }, nexpression);
### --- calculer les facteurs
ci:=coeff(nexpressionik,i,1);     ci1:=coeff(nexpressionik,i1,1);
ci2:=coeff(nexpressionik,i2,1);   ck:=coeff(nexpressionik,k,1);
ck1:=coeff(nexpressionik,k1,1);   ck2:=coeff(nexpressionik,k2,1);
cm:=coeff(nexpressionik,m,1);     cm1:=coeff(nexpressionik,m1,1);
cm2:=coeff(nexpressionik,m2,1);
### --- terme constant de l'expression:
constante:=expand(nexpressionik-i*ci-i1*ci1-i2*ci2-k*ck-k1*ck1
                  -k2*ck2-m*cm-m1*cm1-m2*cm2);
### --- faire la somme pour j allant de 4 a n
si:= coeff(ci,h,0)*(H-1-1/2-1/3)
    +coeff(ci,h,1)*(1/2*(H^2+Hdeux)-1/2*((1+1/2+1/3)^2+1+1/4+1/9));
si1:= coeff(ci1,h1,0)*(H-1-1/2-1/n)
     +coeff(ci1,h1,1)*(1/2*((H-1/n)^2+Hdeux-1/n/n)
                  -1/2*((1+1/2)^2+1+1/4));
si2:= coeff(ci2,h2,0)*(H-1-1/n-1/(n-1))
     +coeff(ci2,h2,1)*(1/2*((H-1/n-1/(n-1))^2+Hdeux-1/n/n-1/(n-1)^2)
                  -1/2*(1+1));
sk:=coeff(ck,h,0)*(Hdeux-1-1/4-1/9)   +  coeff(ck,h,1)*L;
sk1:=coeff(ck1,h1,0)*(Hdeux-1/n/n-1-1/4)   +  coeff(ck1,h1,1)*L1;
sm:=cm*(Htrois-1-1/8-1/27);
sm1:=cm1*(Htrois-1/n/n/n-1-1/8);
s:=expand(subs(L1=L-H/n/n+11/54,si+si1+si2+sk+sk1+sm+sm1)):
### expression finale
somme:=expand(normal((4/n/n+4/9*(n-2)-16*s)));
### facteur de H, Hdeux, Htrois, L
cH:= normal(coeff(somme,H,1)); cHdeux:= normal(coeff(somme,Hdeux,1));
cHtrois:=normal(coeff(somme,Htrois,1));cL:= normal(coeff(somme,L,1));
### facteur constant
cste:=expand(normal(somme-H*cH-Hdeux*cHdeux-Htrois*cHtrois-L*cL));
```

Exécution du programme 1

```
    |\^/|
._|\|   |/|_. Mathematics and Computer Science.
 \  MAPLE  /  Version 4.1 --- May 1987
 <____ ____>  For on-line help, type  help();
      |
### --- Calcul de 4/n^2+4/9*(n-2)-16* la somme pour j allant de 4 a n de
### ---         (n-j+1)*(2*j+h-8-1/j-1/(j-1))/(j^2*(j-1)^2*(j-2))
### ---  Correspondance entre les termes et les variables Maple:
### ---    h       ------>   Hn, somme des inverses
### ---    hdeux   ------>   H(2)n, somme de l'inverse des carres
### ---    htrois  ------>   H(3)n, somme de l'inverse des cubes
### ---    h1      ------>   Hn-1
### ---    h2      ------>   Hn-2
### ---    L       ------>   somme pour i de 4 a n de Hi/i^2
### ---    L1      ------>   comme L mais pour i de 3 a n-1
```

```
                                                                    1
                                  (n - j + 1) (2 j + h - 8 - 1/j - -------)
                                                                  j - 1
          expression := ------------------------------------------------
                                                2         2
                                               j  (j - 1)  (j - 2)

### ---  Fractions partielles
                          6 h + 10 n h - 37 - 75 n
expression := - 1/8 --------------------------
                                    j

          - 17 + 2 h + 2 n h - 19 n          1 + n      h + n h - 6 - 8 n
   - 1/4 --------------------------- + 1/2 ------- + --------------------
                         2                    3             j - 1
                        j                    j

      n h + 1 - 6 n          n             2 n h + 11 - 2 h - 11 n
   - -------------- + ---------- + 1/8 -------------------------
                2             3                    j - 2
         (j - 1)       (j - 1)

### --- changements de variables pour avoir un polynome:
### ---    1/j    ---> i, 1/(j-1)    ---> i1, 1/(j-2)    ---> i2
### ---    1/j^2  ---> k, 1/(j-1)^2  ---> k1, 1/(j-2)^2  ---> k2
### ---    1/j^3  ---> m, 1/(j-1)^3  ---> m1, 1/(j-2)^3  ---> m2
expressionik := - 1/8 (6 h + 10 n h - 37 - 75 n) i

   - 1/4 (- 17 + 2 h + 2 n h - 19 n) k + 1/2 (1 + n) m + (h + n h - 6 - 8 n) i1

   - (n h + 1 - 6 n) k1 + n m1 + 1/8 (2 n h + 11 - 2 h - 11 n) i2

### --- calcul des coefficients de chacune des 9 variables;
                    ci := - 3/4 h - 5/4 n h + 37/8 + 75/8 n

                            ci1 := h + n h - 6 - 8 n

                   ci2 := 1/4 n h + 11/8 - 1/4 h - 11/8 n

                   ck := 17/4 - 1/2 h - 1/2 n h + 19/4 n

                           ck1 := - n h - 1 + 6 n

                                  ck2 := 0

                                cm := 1/2 + 1/2 n

                                  cm1 := n

                                  cm2 := 0

### --- changement au niveau des nombres harmoniques;
                     ci1 := h1 + i + n (h1 + i) - 6 - 8 n

      ci2 := 1/4 n (h2 + i + i1) + 11/8 - 1/4 h2 - 1/4 i - 1/4 i1 - 11/8 n

                          ck1 := - n (h1 + i) - 1 + 6 n

                                  ck2 := 0

                                  cm1 := n

                                  cm2 := 0
```

```
### --- reformer l'expression
nexpression := 1/4 i2 i1 n - k1 i n + 37/8 i + 17/4 k - 1/2 k n h + 11/8 i2

   - k1 - 6 i1 + 1/4 i2 i n - k1 n h1 - 5/4 i n h + i1 n h1 + i1 i n

   + 1/4 i2 n h2 + n m1 - 3/4 i h + 75/8 i n + i1 h1 + i1 i - 8 i1 n + 1/2 m

   - 1/4 i2 h2 - 1/4 i2 i - 1/4 i2 i1 - 11/8 i2 n - 1/2 k h + 19/4 k n + 6 k1 n

   + 1/2 m n

### --- remettre la formule en terme de j
nexpression :=

   -1/4

                  3      6               2                   6
      (- 4 - 17 n j  h - j  n h2 - 4 n h j  - 4 n - 20 j - 4 j  n h1

              5                   2         2       3                  3
        + 3 n j  h2 + 4 h j + 71 j  - 8 h j  - 64 j  + 4 n h j - 3 h j

             4      5          3          2                   5           4
        + 19 j  - 2 j  - 44 n j  + 57 n j  - 24 n j + 20 n j  h1 + 35 n j  h

             6           5           4            3           4          3
        + 5 n j  h - 23 n j  h - 32 n j  h1 + 16 n j  h1 - 3 n j  h2 + n j  h2

             5          4      6          5          4         6          5
        - 6 n j  + 25 n j  + 3 j  h - 13 j  h + 17 j  h - 4 j  h1 + 16 j  h1

             4          3       6          5          4        3
        - 20 j  h1 + 8 j  h1 + j  h2 - 3 j  h2 + 3 j  h2 - j  h2)

            3        3
        / (j  (j - 1)  (j - 2))

### --- fractions partielles
                      5 n h - 15 - 29 n + 3 h
nexpression := - 1/4 -------------------------
                                j

          - 17 + 2 h + 2 n h - 19 n          1 + n
   - 1/4 --------------------------- + 1/2 -------
                       2                      3
                      j                      j

          4 n h1 - 25 n - 19 + 4 h1      n h1 - 5 n + 1          n
   + 1/4 --------------------------- - ---------------- + ----------
                    j - 1                          2              3
                                                (j - 1)        (j - 1)

          n h2 - h2 - 4 n + 4
   + 1/4 ---------------------
                 j - 2

### --- revenir aux variables i, i1,... m2
nexpressionik := - 1/4 (5 n h - 15 - 29 n + 3 h) i

   - 1/4 (- 17 + 2 h + 2 n h - 19 n) k + 1/2 (1 + n) m

   + 1/4 (4 n h1 - 25 n - 19 + 4 h1) i1 - (n h1 - 5 n + 1) k1 + n m1

   + 1/4 (n h2 - h2 - 4 n + 4) i2
```

```
### --- calculer les facteurs
                    ci := - 5/4 n h + 15/4 + 29/4 n - 3/4 h

                       ci1 := n h1 - 25/4 n - 19/4 + h1

                       ci2 := 1/4 n h2 - 1/4 h2 - n + 1

                     ck := 17/4 - 1/2 h - 1/2 n h + 19/4 n

                          ck1 := - n h1 + 5 n - 1

                                 ck2 := 0

                              cm := 1/2 + 1/2 n

                                 cm1 := n

                                 cm2 := 0

### --- terme constant de l'expression:
                               constante := 0

### --- faire la somme pour j allant de 4 a n
                                                                  2                  85
 si := (15/4 + 29/4 n) (H - 11/6) + (- 5/4 n - 3/4) (1/2 H  + 1/2 Hdeux - ----)
                                                                                    36

sil := (- 25/4 n - 19/4) (H - 3/2 - 1/n)

                            2                      1
   + (1 + n) (1/2 (H - 1/n)  + 1/2 Hdeux - 1/2 ---- - 7/4)
                                                  2
                                                 n

                                    1
si2 := (- n + 1) (H - 1 - 1/n - -------)
                                  n - 1

   + (1/4 n - 1/4)

                        1    2                    1             1
      (1/2 (H - 1/n - -------)  + 1/2 Hdeux - 1/2 ---- - 1/2 ---------- - 1)
                      n - 1                       2               2
                                                 n          (n - 1)

                                          49
           sk := (17/4 + 19/4 n) (Hdeux - ----) + (- 1/2 - 1/2 n) L
                                          36

                                           1
                sk1 := (5 n - 1) (Hdeux - ---- - 5/4) - n L1
                                           2
                                          n

                                                 251
                    sm := (1/2 + 1/2 n) (Htrois - -----)
                                                 216

                                           1
                       sm1 := n (Htrois - ---- - 9/8)
                                           3
                                          n
```

```
### expression finale
somme := - 156 n Hdeux - 52 Hdeux + 24 H - 4 H/n + 24 n L + 8 L - 24 n Htrois

                                1171                    1
   - 8 Htrois + 2393/9 n - ------ + 16 1/n + 4 ----
                            27                    2
                                                 n

### facteur de H, Hdeux, Htrois, L
                                      6 n - 1
                                cH := 4 ---------
                                           n

                               cHdeux := - 156 n - 52

                               cHtrois := - 24 n - 8

                                   cL := 24 n + 8

### facteur constant
                                        1171                    1
                     cste := 2393/9 n - ------ + 16 1/n + 4 ----
                                        27                    2
                                                             n
```

Programme 2 – Calcul de l'espérance et de la variance de la profondeur du dernier nœud ajouté dans un arbre pseudo-hyperquaternaire et dans un arbre hyperquaternaire de dimension deux.

```
T := proc (n,i) local j;
     binomial(n-2,i) *
        sum(binomial(n-2-i,j)*(-1)^j*(1+2^(2*i+2*j+2)-2^(i+j+2))
                                    /(i+j+1)^2/(i+j+2)^2/2^(2*i+2*j-2)
                    ,j=0..n-2-i)
     end;

procMU := proc ( debut, fin ) local i, n, t, H, H2;
          H := 3/2; H2 := 5/4;
          for i from 3 to debut-1 do H := H + 1/i; H2 := H2 + 1/i/i od;
          for n from debut to fin do
              for i from 1 to n-2 do t[i] := T(n,i) od; i := 'i';
              MU1[n] := 1 + 2/(n-2)*sum(i*t[i]*(MU1[i+1]+MU1[i]),i=0..n-2);
              MU2[n] := 2*MU1[n] - 1
                        + 2/(n-2)*sum(i*t[i]*(MU2[i+1]+MU2[i]),i=0..n-2);
              H := H + 1/n;   H2 := H2 + 1/n/n;
              MUQ1[n] := H - 1/6 - 2/3/n;
              MUQ2[n] := H*H + H2 + H/6 - 4*H/3/n+7/9/n-77/36 od;
          end;

resultats := proc ( debut, fin ) local n;
     for n from debut to fin do
         if irem ( n, 5 ) = 0 then
             print(n, evalf(MU1[n]/log(n)), evalf(MUQ1[n]/log(n)),
                  evalf(MU2[n]-MU1[n]^2), evalf(MUQ2[n]-MUQ1[n]^2) ) fi od
     end;
```

```
init := proc ();
    MU1[1] := 0; MU1[2] := 1;
    MU2[1] := 0; MU2[2] := 1;
    end;

init():
procMU(3,70):
resultats(3,70):
```

Programme 3 – Calcul de la probabilité que la racine ait entre 0 et 4 enfants et de l'espérance du nombre de nœuds ayant entre 0 et 4 enfants dans un arbre hyperquaternaire de dimension deux.

```
PROGRAM calc;

CONST
     maxN = 200;

TYPE
     cardinal = 0 .. maxInt;
     nbEnfant = 0 .. 4;
     indice   = 1..maxN;
     tabReal  = ARRAY [ indice ] OF real;

{+-------------------------------------------------------------------+
 | H [n]    : somme ( 1/i, i = 1 .. n )                              |
 | E [i, n] : Esperance du nombre de noeuds a i enfants pour les     |
 |            arbres quaternaires a n noeuds                         |
 | A [i, n] : probabilite que la racine ait i enfants (dans un       |
 |            arbre quaternaire de n noeuds)                         |
 | lg [i]   : somme ( ln(k), k = 1 .. i ), lg [0] = 0                |
 +-------------------------------------------------------------------+}

VAR
     H : ARRAY [ indice   ] OF Real;
     E : ARRAY [ nbEnfant ] OF tabReal;
     A : ARRAY [ nbEnfant ] OF tabReal;
     lg : ARRAY [ 0 .. maxN ] OF real;
     hn : real;
     f : text;

FUNCTION NbNonZero (a, b, c, d : integer) : integer;
    BEGIN
      NbNonZero := ORD(a<>0) + ORD(b<>0) + ORD(c<>0) + ORD(d<>0)
    END;

PROCEDURE Init;
    VAR i, j : cardinal;
    BEGIN
      rewrite(f);
      H[1] := 1; lg [0] := 0;

      for i := 0 to 4 do
          E[i, 1] := 0;
      E[0, 1] := 1;

      lg[0] := 0;
      for i := 1 to maxN do lg[i] := lg[i-1] + ln(i)
    END;
```

```
PROCEDURE Calculs;
    VAR n, i, j, k, l, m : cardinal;
        s : real;
    BEGIN
     FOR n := 2 TO maxN DO
         BEGIN
           FOR i := 0 TO 4 DO A [i,n ] := 0;
           FOR i := 0 TO n - 1 DO
               FOR j := 0 TO n - 1 - i DO
                   FOR k := 0 TO n - 1 - i - j DO
                       BEGIN
                         l := n - 1 - i - j - k;
                         m := NbNonZero(i, j, k, l);
                         A[m,n] := A[m,n] + exp(lg[i+j]+lg[n-1-i-j]
                                   + lg[i+k] + lg[n-1-i-k]
                                   -lg[i]-lg[j]-lg[k]-lg[l]-lg[n-1]);
                       END;
            H[n] := H[n - 1] + 1 / n;
            hn := H[n];
            FOR m := 0 TO 4 DO
                BEGIN
                  A[m,n] := A[m,n] / n / n;
                  s := 0;
                  FOR i := 1 TO n - 1 DO
                      s := s + E[m, i] * ( hn - H[i] );
                  E[m, n] := A[m,n] + 4/n * s
                END
         END
    END;

PROCEDURE Resultats;
    VAR i, j : integer;
    BEGIN
      writeln ('Esperance du nombre de noeuds ayant entre 0 et 4 enfants' );
      writeln;
      FOR i := 1 TO maxN div 10 DO
          BEGIN
            write ( 10*i:5 );
            FOR j := 0 TO 4 DO
                write ( E[j,10*i]/i/10:8:5 );
            writeln
          END;
      writeln;
      writeln ('Probabilites que la racine ait entre 0 et 4 enfants' );
      writeln;
      FOR i := 1 TO maxN div 10 DO
          BEGIN
            write ( 10*i:5 );
            FOR j := 0 TO 4 DO
                write ( A[j,10*i]:8:5 );
            writeln
          END;
    END;

BEGIN
  Init;
  Calculs;
  Resultats
END.
```

Programme 4 – Calcul de la fonction de densité de S du corollaire 5.8

```
####################  Calcul de la densité de S  ####################
#                                                                    #
# Soient X, Y, Z, T ~ U[0,1], indépendantes                          #
# Soit S = (X+Y)/2 * (Z+T)/2.                                        #
# La densité de S est donnée par I4, I2 et I1 pour les bornes        #
# mentionnées plus bas                                               #
#                                                                    #
######################################################################

fMx:=4*(1-s/m);  fMy:=4*s/m;
haut := 4/m-4;     bas := 4;
# Pour 0 <= s < 1/4;
I4 := expand ( int ( fMx*bas, m=s .. 2*s ) +
               int ( fMy*bas, m=2*s  .. 1/2) +
               int ( fMy*haut, m=1/2 .. 1) );
# Pour 1/4 <= s < 1/2;
I2 := expand ( int ( fMx * bas, m=s .. 1/2 ) +
               int ( fMx* haut, m=1/2 .. 2*s ) +
               int ( fMy*haut, m=2*s .. 1 ) );
# Pour 1/2 <= s < 1;
I1 := expand ( int ( fMx*haut, m=s .. 1));
```

Programme 5 – Calcul intermédiaire servant à la preuve du corollaire 5.9

```
c:='c': p:='p':
# lint1: integrale de p^a
lint1:=proc(a,p) 'p^(a+1)/(a+1)' end:
# lint2: integrale de ln(P)*p^a
lint2:=proc(a,p) 'p^(a+1)/(a+1)*log(p)-p^(a+1)/(a+1)^2' end:
# Calcul de int ( s^(i+j) * f(s) , s ), ou f(s) a ete calcule auparavent
M1:=array([0,32-64*c,0,-16]):
M2:=array([32*c+32,64*c-96,16,48]):
M3:=array([-32,32,-16,-16]):
lesint:=array([lint1(i+j,p),lint1(i+j+1,p),lint2(i+j,p),lint2(i+j+1,p)]):
R1:=linalg[dotprod](M1,lesint):
R2:=linalg[dotprod](M2,lesint):
R3:=linalg[dotprod](M3,lesint):
c:=ln(4)/2:
p:=1/4:
t1:=R1:
t3:=R2:
p:=1/2:
t2:=R2:
t5:=R3:
p:=1:
t4:=R3:
reponse:=t1+t2-t3+t4-t5:

expand(factor(simplify(numer(reponse)*2^(2*i+2*j-2))));
factor(simplify(denom(reponse)*2^(2*i+2*j-2)));
```

Annexe B

Formules utiles

$$H_n^{(i)} = 1 + \frac{1}{2^i} + \frac{1}{3^i} + \ldots + \frac{1}{n^i}, \quad n \geq 1.$$

$$H_n \triangleq H_n^{(1)}, \quad H_0^{(i)} = 0.$$

$$S_{n,k,j} = \sum_{i=1}^{n} i^k H_i^j = H_n S_{n,k,j-1} - \sum_{l=1}^{n} \frac{S_{l-1,k,j-1}}{l}, \quad k \geq 0, \quad j \geq 0, \quad n \geq 1.$$

$$S_{0,k,j} = 0; \quad S_{n,k,0} = \sum_{i=1}^{n} i^k.$$

$$\sum_{i=1}^{n} H_i^{(m)} = (n+1)H_n^{(m)} - H_n^{(m-1)}.$$

$$\sum_{i=1}^{n} iH_i^{(m)} = \frac{n(n+1)}{2} H_n^{(m)} + \frac{1}{2} H_n^{(m-1)} - \frac{1}{2} H_n^{(m-2)}.$$

$$\sum_{i=1}^{n} i^2 H_i^{(m)} = \frac{n(n+1)(2n+1)}{6} H_n^{(m)} - \frac{1}{6} H_n^{(m-1)} + \frac{1}{2} H_n^{(m-2)} - \frac{1}{3} H_n^{(m-3)}.$$

$$\sum_{i=1}^{n} i^3 H_i^{(m)} = \frac{n^2(n+1)^2}{4} H_n^{(m)} - \frac{1}{4} H_n^{(m-2)} + \frac{1}{2} H_n^{(m-3)} - \frac{1}{4} H_n^{(m-4)}.$$

$$\sum_{i=1}^{n} H_i^2 = (n+1)H_n^2 - (2n+1)H_n + 2n.$$

$$\sum_{i=1}^{n} H_i^3 = (n+1)H_n^3 - \frac{3}{2}(2n+1)H_n^2 + \frac{1}{2}H_n^{(2)} + 3(2n+1)H_n - 6n .$$

$$\sum_{i=1}^{n} \frac{H_i}{i} = \frac{1}{2}(H_n^2 + H_n^{(2)}) .$$

$$\sum_{i=1}^{n} (H_i^{(m)})^2 = (n+1)(H_n^{(m)})^2 - H_n^{(m-1)} H_n^{(m)} - \sum_{i=1}^{n} \frac{H_i^{(m)}}{i^{m-1}} + \sum_{i=1}^{n} \frac{H_i^{(m-1)}}{i^m} .$$

$$\sum_{i=1}^{n} i H_i^2 = \frac{n(n+1)}{2} H_n^2 - \frac{n^2-n-1}{2} H_n + \frac{n(n-3)}{4} .$$

$$\sum_{i=1}^{n} i^2 H_i^2 = \frac{n(n+1)(2n+1)}{6} H_n^2 - \frac{4n^3-3n^2-n+3}{18} H_n + \frac{n(8n^2-15n+25)}{108} .$$

$$\sum_{i=1}^{n} i^3 H_i^2 = \frac{n^2(n+1)^2}{4} H_n^2 - \frac{n(n^2-1)(3n-2)}{24} H_n + \frac{n(n-1)(9n^2-5n+10)}{288} .$$

$$\sum_{i=1}^{n} H_i H_i^{(2)} = (n+1)H_n^{(2)}(H_n - 1) - \frac{1}{2}(H_n^2 - H_n^{(2)}) + H_n .$$

$$\sum_{i=1}^{n} \frac{H_i^{(2)}}{i} = -\sum_{i=1}^{n} \frac{H_i}{i^2} + H_n H_n^{(2)} + H_n^{(3)} .$$

$$\sum_{i=1}^{n} \frac{H_i^2}{i} = \sum_{i=1}^{n} \frac{H_i}{i^2} + \frac{1}{3}(H_n^3 - H_n^{(3)}) .$$

$$\sum_{i=1}^{n} \frac{H_i^2 + H_i^{(2)}}{i} = H_n H_n^{(2)} + \frac{1}{3}H_n^3 + \frac{2}{3}H_n^{(3)} .$$

$$\sum_{i=1}^{n} \frac{H_i}{n+1-i} = H_n^2 - H_n^{(2)} + \frac{2H_n}{n+1} .$$

$$\sum_{i=1}^{n} H_i H_{n+1-i} = (n+2)(H_n^2 - H_n^{(2)}) - \frac{2(n^2+n-1)}{n+1} H_n + 2n .$$

$$\sum_{i=1}^{n} iH_iH_{n+1-i} = \frac{(n+1)(n+2)}{2}(H_n^2 - H_n^{(2)}) - (n^2+n-1)H_n + n(n+1) .$$

$$\sum_{i=1}^{n} iH_{n+1-i} = \frac{(n+1)(n+2)}{2}H_n - \frac{n}{4}(3n+5) .$$

$$\sum_{i=1}^{n} i^2H_{n+1-i} = \frac{(n+1)(n+2)(2n+3)}{6}H_n - \frac{n}{36}(22n+69n+53) .$$

$$\sum_{i=1}^{n}(H_i^2 - H_i^{(2)}) = (n+1)(H_n^2 - H_n^{(2)}) + 2n(1 - H_n) .$$

$$\sum_{i=1}^{n} i(H_i^2 - H_i^{(2)}) = \frac{n(n+1)}{2}(H_n^2 - H_n^{(2)}) - \frac{n(n-1)}{2}H_n + \frac{n(n-1)}{4} .$$

$$\sum_{i=1}^{n} i^2(H_i^2 - H_i^{(2)}) = \frac{n(n+1)(2n+1)}{6}(H_n^2 - H_n^{(2)}) - \frac{n(n-1)(4n+1)}{18}H_n + \frac{n(n-1)(8n+11)}{108} .$$

$$\sum_{i=1}^{n} i^3(H_i^2 - H_i^{(2)}) = \frac{n^2(n+1)^2}{4}(H_n^2 - H_n^{(2)}) - \frac{n(n^2-1)(3n-2)}{24}H_n + \frac{n(n-1)(9n^2+19n-2)}{288} .$$

$$\sum_{i=1}^{n} \frac{H_i^{(m)}}{i^k} = H_n^{(k)}H_n^{(m)} + H_n^{(m+k)} - \sum_{i=1}^{n} \frac{H_i^{(k)}}{i^m} .$$

$$\sum_{i=1}^{n} \frac{H_i^m}{i^m} = \frac{1}{2}((H_n^{(m)})^2 + H_n^{(2m)}) .$$

$$\sum_{k=0}^{n} \frac{(-1)^k\binom{n}{k}}{k+x} = \frac{1}{x\binom{n+x}{n}}, \quad x \notin \{0, -1, -2, \ldots, -n\}.$$

$$\sum_{k=0}^{j-1} \frac{(-1)^k\binom{j}{k}}{j-k} = (-1)^{j-1}H_j, \quad j \geq 0.$$

$$\sum_{k=0}^{j-1} \frac{(-1)^k \binom{j}{k}}{(j-k)(i+k+1)} = \frac{(-1)^{j+1} H_j}{(i+j+1)} + \frac{(-1)^{j+1}}{(i+j+1)^2} + \frac{1}{(i+1)(i+j+1)\binom{i+j+1}{j}}, \quad j \geq 0.$$

$$\sum_{k=0}^{j-1} \frac{(-1)^k \binom{j}{k}}{(i+j-k)} = \frac{(-1)^j}{i} \left(\frac{1}{\binom{i+j}{i}} - 1 \right), \quad i \geq 1.$$

$$\sum_{k=0}^{j-1} \frac{(-1)^k \binom{j}{k}}{(k+1)(i+j-k)} = \begin{cases} \dfrac{(-1)^{j+1}}{i(j+1)} + \dfrac{1}{(j+1)(i+j+1)} + \dfrac{(-1)^j}{i(i+j+1)\binom{i+j}{i}}, & i \geq 1 \\ \dfrac{1}{(j+1)^2} + \dfrac{(-1)^{j+1} H_{j+1}}{j+1}, & i = 0. \end{cases}$$

Références

M. Abramowitz, I.A. Stegun, *Handbook of Mathematical Functions*, Dover Publications, New York, 1972.

A.V. Aho, J.E. Hopcroft, J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Massachusetts, 1974.

D.H. Ballard, "Strip trees: a hierarchical representation for curves," *Communications of the ACM*, vol. 24, pp. 310-321, 1981.

J.L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, pp. 509-517, 1975.

J.L. Bentley, "Multidimensional binary search trees in database applications," *IEEE Transactions on Software Engineering*, vol. SE-5, pp. 333-340, July 1979.

J.L. Bentley, M.I. Shamos, "Divide-and-conquer in multidimensional space," *Proceedings of the 8th Annual ACM Symposium on Theory of Computing*, pp. 220-230, 1976.

J.L. Bentley, D.F. Stanat, "Analysis of range searches in quad trees," *Information Processing Letters*, vol. 3, pp. 170-173, 1975.

J.L. Bentley, D.F. Stanat, E.H. Williams, "The complexity of fixed-radius near neighbor searching," *Information Processing Letters*, vol. 6, pp. 209-212, 1977.

J. Berstel, A.N. Abdallah, "Tétrarbres engendrés par des automates finis," CNRS Université Paris VI et VII, *Informatique théorique et programmation*, 1989.

W.H. Beyer, *CRC Standard Mathematical Tables*, 27th edition, CRC Press, Boca Raton, Florida, 1984.

E. Börger, *Trends in Theoretical Computer Science*, Computer Science Press, 1988.

G.G. Brown, B.O. Shubert, "On random binary trees," *Mathematics of Operations Research*, vol. 9, pp. 43-65, 1984.

H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Annals of Mathematical Statistics*, vol. 23, pp. 493-507, 1952.

J. Culberson, "The effect of updates in binary search trees," *Proceedings of the 17th Annual ACM. Symposium on Theory of Computing*, Providence, R.I., 1985.

L. Devroye, "A note on the height of binary search trees," *Journal of the ACM*, vol. 33, pp. 489-498, 1986.

L. Devroye, "Applications of the theory of records in the study of random trees," *Acta Informatica*, vol. 26, pp. 123-130, 1988.

L. Devroye, "Branching processes in the analysis of the heights of trees," *Acta Informatica*, vol. 24, pp. 277-298, 1987.

L. Devroye, *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, 1986.

L. Devroye, L. Laforest, "An analysis of random *d*-dimensional quadtree," *SIAM Journal on Computing*, à paraître, 1990.

D. Dobkin, R.J. Lipton, "Multidimensional searching problems," *SIAM Journal on Computing*, vol. 5, pp. 181-186, 1976.

C.R. Dyer, A. Rosenfeld, H. Samet, "Region representation: boundary codes from quadtrees," *Communications of the ACM*, vol. 23, pp. 171-179, 1980.

J.L. Eppinger, "An empirical study of insertion and deletion in binary trees," *Communications of the ACM*, vol. 26, 1983.

R.A. Finkel, J.L. Bentley, "Quad trees: a data structure for retrieval on composite keys," *Acta Informatica*, vol. 4, pp. 1-9, 1974.

P. Flajolet, C. Puech, "Partial match retrieval of multidimensional data," *Journal of the ACM*, vol. 33, pp. 371-407, 1986.

P. Flajolet, G. Gonnet, C. Puech, M. Robson, "Analytic variations on quad trees," en préparation. 1987.

P. Flajolet, J.S. Vitter, "Average-case analysis of algorithms and data structures," Rapport de recherche No. 718, INRIA, Paris, France, 1987.

M.L. Fredman, "A lower bound on the complexity of orthogonal range queries," *Journal of the ACM*, vol. 28, pp. 696-705, 1981.

J.D. Gibbons, *Nonparametric Statistical Inference*, McGraw-Hill, New York, 1971.

G.H. Gonnet, *A Handbook of Algorithms and Data Structures*, Addison-Wesley, Reading, Mass., 1984.

L.J. Guibas, "A Principle of Independence for Binary Tree Searching," *Acta Informatica*, vol. 4, pp. 293-298, 1975.

T.N. Hibbard, "Some combinatorial properties of certain trees with applications to searching and sorting," *Journal of the ACM*, vol. 9, 1962.

G.M. Hunter, K. Steiglitz, "Operations on images using quad trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, pp. 145-153, 1979.

V. Klee, "Can the measure of $\bigcup[a_i,b_i]$ be computed in less than $O(n\log n)$ steps," Research Problems Sect., *American Mathematical Monthly*, vol. 84, pp. 284-285, 1977.

D.E. Knuth, *The Art of Computer Programming, vol. 1: Fundamental Algorithms*, Addison-Wesley, Reading, Mass., 1973. 2nd Ed.

D.E. Knuth, *The Art of Computer Programming, vol. 3: Sorting and Searching*, Addison-Wesley, Reading, Mass., 1973.

H.T. Kung, F. Luccio, F.P. Preparata, "On finding the maxima of a set of vectors," *Journal of the ACM*, vol. 22, pp. 469-476, 1975.

G. Labelle, L. Laforest, "Étude asymptotique du nombre moyen de nœuds à un enfant dans un arbre quaternaire," Rapport de recherche, département de mathématiques et d'informatique, Université du Québec à Montréal, à paraître, 1990.

D.T. Lee, C.K. Wong, "Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees," *Acta Informatica*, vol. 9, pp. 23-29, 1977.

D.T. Lee, C.K. Wong, "Quintary trees: a file structure for multidimentional database systems," *ACM Transactions on Database Systems I*, vol. 5, pp. 339-353, 1981.

J. van Leeuwen, D. Wood, "The measure problem for rectangular ranges in *d*–space," *Journal of Algorithms*, vol. 2, pp. 282-300, 1981.

G.S. Lueker, "A data structure for orthogonal range queries," *19th Annual Symposium on Foundations of Computer Science*, Ann Arbor, MI, USA 16-18 Oct 78 (New York, USA: IEEE 1978), pp 28-34. 1978.

W.C. Lynch, "More combinatorial problems on certain trees," *The Computer Journal*, vol. 7, pp. 299-302, 1965.

H.M. Mahmoud, "The expected distribution of degrees in random binary trees," *The Computer Journal*, vol. 29, pp. 36-37, 1986.

H.M. Mahmoud, B. Pittel, "On the most probable shape of a search tree grown from a random permutation," *SIAM Journal of Algebraic and Discrete Methods*, vol. 5, pp. 69-81, 1984.

T.H. Merrett, E. Otoo, "Multidimensional paging for associative searching," Technical Report SOCS 81-18, School of Computer Science, McGill University, Montréal, 1981.

J.A. Orenstein, "Multidimensional tries used for associative searching," Technical Report, School of Computer Science, McGill University, Montréal, 1982.

M.H. Overmars, *The Design of Dynamic Data Structures*, Lecture Notes in Computer Science, vol. 156, Springer-Verlag, 1983.

M.H. Overmars, J. van Leeuwen, "Dynamic multi-dimentional data structures based on quad- and *k–d* trees," *Acta Informatica*, vol. 17, pp. 267-285, 1982.

B. Pittel, "On growing random binary trees," *Journal of Mathematical Analysis and Applications*, vol. 103, pp. 461-480, 1984.

P.V. Poblete, J.I. Munro, "The analysis of a fringe heuristic for binary search trees," *Journal of Algorithms*, vol. 6, pp. 336-350, 1985.

C. Puech, H. Yahia, "Quadtrees, octrees, hyperoctrees: a unified analytical approach to tree data structures used in graphics, geometric modeling and image processing," *Proceedings of the Symposium on Computational Geometry*, pp. 272-280, ACM, New York, 1985.

R. Pyke, "Spacings," *Journal of the Royal Statistical Society Series B*, vol. 7, pp. 395-445,1965.

R. Pyke, "Spacings revisited," *Proceedings of the Sixth Berkeley Symposium*, vol. 1, pp. 417-427, 1972.

R.L. Rivest, "Analysis of associative retrieval algorithms," Technical Report STAN-CS-74-415, Computer Science Department, Stanford University, Stanford, CA., 1974.

R.L. Rivest, "Partial match retrieval algorithms," *SIAM Journal on Computing*, vol. 5, pp. 19-50, 1976.

J.T. Robinson, "The K-B-D tree: a search structure for large multidimensional dynamic indexes," *Proceedings of the ACM SIGMOD*, pp. 10-18, 1981.

J.M. Robson, "The height of binary search trees," *The Australian Computer Journal*, vol. 11, pp. 151-153, 1979.

D. Rutovitz, "Data structures for operations on digital images," *Pictorial Pattern Recognition*, G.C. Cheng et al., Eds. Thompson Book Co., Washington D.C., pp 105-133, 1968.

H. Samet, "Deletion in two-dimentional quad trees," *Communications of the ACM*, vol. 23, pp. 703-710, 1980.

H. Samet, "The quadtree and related hierarchical data structures," *Computing Surveys*, vol. 16, pp. 187-260, 1984.

R. Sedgewick, "Mathematical analysis of combinatorial algorithms," *Probability Theory and Computer Science*, ed. G. Louchard and G. Latouche, pp. 123-205, Academic Press, London, 1983.

Robert J. Serfling, *Approximation Theorems of Mathematical Statistics*, John Wiley & Sons, 1980.

T.A. Standish, *Data Structures Techniques*, Addison-Wesley, 1980.

M. Tamminen, "Order preserving extendible hashing and bucket tries," *BIT*, vol. 21, pp. 419-435, 1981.

M. Tamminen, "The EXCELL method for efficient geometric access to data," *Acta Polytechnica Scandinavica*, vol. Mathematics and Computer Science Series 34, Helsinki, 1981.

M. Tamminen, "The extendible cell method for closest point problems," *BIT*, vol. 22, pp. 27-41, 1982.

D.E. Willard, "Polygon retrieval," *SIAM Journal on Computing*, vol. 11, pp. 149-165, 1982.

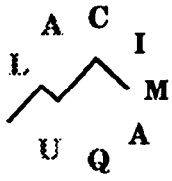
J.R. Woodwark, "The explicit quad tree as a structure for computer graphics," *The Computer Journal*, vol. 25, pp. 235-237, 1982.

Étude des arbres hyperquaternaires

Cet ouvrage traite principalement des arbres hyperquaternaires de points ("point quadtree") qui sont une généralisation des arbres binaires de fouille. En premier lieu, un survol des structures de données hiérarchiques est présenté. Sont décrits entre autres les arbres hyperquaternaires de région, les arbres k-d, les arbres pseudo-hyperquaternaires et pseudo-k-d. Les résultats relatifs aux arbres binaires de fouille ainsi que ceux concernant les arbres pseudo-hyperquaternaires de points sont donnés. Une étude plus poussée des arbres hyperquaternaires nous a permis d'obtenir des résultats concernant la profondeur du dernier nœud inséré, la proportion des divers types de nœuds dans un arbre hyperquaternaire de points. Enfin, nous étudions une alternative de ces arbres, soit les arbres pseudo-hyperquaternaires ce qui nous permet de montrer, après étude du cas à deux dimensions, que ceux-ci semblent plus performants au niveau de la fouille, de l'ajout et de la suppression de points que les arbres hyperquaternaires originaux.

TABLE DES MATIÈRES

Laboratoire de combinatoire et d'informatique mathématique
Département de mathématiques et d'informatique
Université du Québec à Montréal
C.P. 8888, Succ. A
Montréal, Qc.
Canada H3C 3P8



ISBN 2-89276-080-X