

Étude probabiliste d'algorithmes en arbre

THÈSE

présentée et soutenue publiquement le 13 Juillet 2007

pour l'obtention du

Doctorat de l'université Pierre et Marie Curie
(spécialité mathématiques appliquées)

par

Hanène Mohamed

Composition du jury

Rapporteurs : Brigitte Chauvin
Luc Devroye

Examineurs : Christian Lavault
Philippe Bougerol
Philippe Chassaing
Philippe Flajolet
Philippe Robert

Mis en page avec la classe thloria.

حُحِلِقَتَ طَلِيْقًا كَطِيْفِ النَّسِيْمِ وَ حُرًّا كَنُورِ الصُّحَى فِي سَمَاهِ
تُعَرِّدُ كَالطَّيْرِ أَيْنَ إِنْدَفَعْتَ وَ تَشْدُو بِمَا شَاءَ وَحِي الإِلَاهِ
وَ تَمْرُحُ بَيْنَ وُرُودِ الصَّبَاحِ وَ تَنْعَمُ بِالنُّورِ أَنِّي تَرَاهِ
كَذَا صَاغَكَ اللَّهُ يَا ابْنَ الْوُجُودِ وَ أَلْقَتَكَ فِالْكَوْنِ هَذِي الْحَيَاهِ
فَمَا لَكَ تَرْضَى بِذُلِّ الْقُيُودِ وَ تَحْنِي لِئِنْ كَبَّلُوكَ الْجِبَاهِ
وَ تَقْنَعُ بِالْعَيْشِ بَيْنَ الْكُهُوفِ فَأَيْنَ النَّشِيدُ وَ أَيْنَ الإِبَاهِ
أَتَخَشَى نَشِيدَ السَّمَاءِ الْجَمِيلِ أَتَرْهَبُ نُورَ الْقَضَا فِي ضَحَاهِ
أَلَا انْهَضْ وَ سِرْ فِي سَبِيلِ الْحَيَاهِ فَمَنْ نَامَ لَمْ تَنْتَظِرْهُ الْحَيَاهِ
إِلَى النُّورِ فَالنُّورُ عَذْبٌ جَمِيلِ إِلَى النُّورِ فَالنُّورُ ظِلُّ الإِلَاهِ

أَبُو الْقَاسِمِ الشَّائِي

*J'ai une pensée profonde pour ma famille que j'ai laissée loin de moi lors de mon séjour en France ces cinq années,
mon père*

ma mère

ma sœur

et mon frère

qui me manquent énormément. . . Ce travail leur est dédié, il en valait la peine. Aujourd'hui, ma famille s'agrandit, au petit ange qui m'a accompagnée lors de la rédaction de ce mémoire, à

حسين

بية

جيهان

مروان

فهد

Remerciements

Je tiens à remercier Jean Lacroix pour avoir accepté si gentilement de m'encadrer, ainsi que Philippe Bougerol pour avoir accepté de prendre la relève. Merci aussi à Brigitte Chauvin et à Luc Devroye, mes deux rapporteurs, pour leurs commentaires et suggestions. Merci à Philippe Flajolet, Philippe Chassaing, Christian Lavault de m'avoir honorée de leur participation à mon jury de thèse.

Durant ces années, j'ai fait partie du projet RAP à l'INRIA Rocquencourt où j'ai effectué mes travaux de recherche sous la direction de Philippe Robert. Je ne saurais lui exprimer ma gratitude pour ses conseils, sa disponibilité, son écoute, sa patience ainsi que toutes les conditions matérielles et morales dont j'ai bénéficiées tout au long de ma thèse. Il m'a fait découvrir un domaine vivant et captivant et m'a donné goût à la simplicité de l'argument et à l'élégance du résultat. . . Je suis un peu émue en regardant aujourd'hui le chemin parcouru au sein de cette équipe de recherche, et je ne peux qu'avoir une pensée forte pour tous les membres passés et présents, mes amis et collègues.

Je ne saurais oublier les remarques pertinentes et les discussions entre "normaliennes" de Danielle et Christine autour de la différence subtile entre la virgule et le point-virgule, merci pour leurs relectures on ne peut plus sérieuses de ce mémoire. Un clin d'oeil aussi à mon amie et voisine de bureau, Yousra, qui a rendu le bureau moins morose, mais moins silencieux au grand dam de Philippe! Un coucou aussi à notre ange gardien à tous, à Virginie, une pensée nostalgique à toutes les fois où on a failli se perdre sans toi!

Enfin, j'ai une pensée pour mes professeurs de la faculté des sciences de Tunis, Raouf Jaibi, Abdennebi Achour et Houcine Chebli. J'espère qu'ils se souviennent de moi, moi je n'ai pas oublié.

Table des matières

Introduction	ix
1 Arbres aléatoires	ix
2 Contribution de la thèse	xxii

Partie I Cas statique

Chapitre 1

Algorithme en arbre avec arrivées bloquées	1
---	----------

1.1 Introduction	1
1.2 Propriétés Générales	7
1.3 Analyse du Comportement Asymptotique du Coût Moyen	12
1.4 Distribution de l'Algorithme Q -aire Symétrique	17
1.5 Algorithme en Arbre et Processus de Branchement	24

Chapitre 2

Algorithme en arbre avec perte

2.1 Introduction	33
2.2 Coût Moyen de l'Algorithme	35
2.3 Comportement Asymptotique de l'Algorithme	36

Chapitre 3

Algorithme d'élection de leader

3.1	Introduction	41
3.2	Résultats	47
3.3	Coût Moyen de l'Algorithme	49
3.4	Analyse Asymptotique du Coût Moyen	53
3.5	Distribution du Coût de l'Algorithme	58
3.6	Algorithme d'Élection de Leader : Version Markovienne	60

Partie II Cas dynamique

65

Chapitre 4

Algorithme en arbre avec arrivées libres

4.1	Introduction	67
4.2	Formulation du Modèle	68
4.3	Itération d'Opérateurs Fonctionnels	69
4.4	Littérature	71
4.5	Objectifs et Résultats	72
4.6	Existence d'une Région de Stabilité Non Vide	74

Chapitre 5

Arrivées Poissonniennes

5.1	Transformée de Poisson	79
5.2	Condition de Stabilité : le cas conservatif $\varepsilon = 1$	84

Conclusion

91

Annexes

Annexe A

Théorème de renouvellement

A.1 Lois et Processus de *Poisson* 93
A.2 Quelques Résultats autour du Théorème de Renouvellement 94

Annexe B

Élection du leader : preuve du lemme 3

Table des figures 101
Bibliographie 103

Introduction

1 Arbres aléatoires

L'étude des arbres aléatoires en tant qu'objet mathématique ou informatique est motivée par la variété des domaines d'applications ainsi que pour la richesse en phénomènes que présentent de telles structures arborescentes.

Motivation :

Cette thèse s'inscrit dans le cadre d'étude du comportement asymptotique d'algorithmes ayant une structure arborescente, appelés *algorithmes en arbre*. Ces algorithmes de type *diviser pour régner* sont basés sur un *processus récursif de division aléatoire*. Ainsi, l'algorithme est représenté d'une manière naturelle par un arbre. L'analyse du comportement asymptotique d'un tel algorithme revient alors à étudier plusieurs caractéristiques naturelles de l'arbre associé telles que :

- la *profondeur* (ou niveau) P d'un nœud donné u . Cette quantité s'exprime en fonction de la *longueur* $|u|$ du chemin reliant ce nœud à la racine de l'arbre, *i.e.* le nombre de nœuds sur ce chemin. La quantité $|u| - 1$ est égale alors à la *profondeur*, ou encore niveau, du nœud u vu que par convention celle de la racine est 0.
- la *hauteur* H de l'arbre, soit la longueur du plus long chemin reliant une *feuille* de l'arbre à la racine, ou encore la *profondeur maximale* atteinte dans l'arbre, augmentée de 1.
- la *taille* R de l'arbre ; c'est le nombre total des nœuds dans l'arbre.
- le *cheminement externe* E (respectivement *interne* I) de l'arbre, soit la somme des *profondeurs* des positions d'insertions possibles (resp. des nœuds).
- etc...

En effet, différents *coûts* d'algorithmes basés sur ce type de structure peuvent être associés à un des paramètres naturels de l'arbre. Nous passons en revue plusieurs exemples qui illustrent ceci.

1.1 Protocole d'accès

Canal à accès multiple :

Dans les années soixante-dix, il s'agissait d'organiser la communication dans un contexte très décentralisé, dont l'archipel de Hawaï est à la fois un exemple et le point de départ historique dans le domaine des réseaux de communication. La nécessité de faire communiquer un grand nombre de *stations* par l'intermédiaire d'un canal partagé posait le problème des tentatives simultanées de transmission. En effet, l'usage d'un canal unique impose, en plus des contraintes algorithmiques classiques telles que *l'admissibilité*, *la stabilité* ou encore *l'équité*, le problème de *résolution de conflits*, appelés encore *collisions*, vu que ce canal commun de communication ne peut transmettre qu'un seul message par unité de temps. Dans le cas particulier d'un réseau avec détection de *collision*, le canal livre une information ternaire ; chaque *station* peut simultanément envoyer son message vers le réseau et écouter le canal pour détecter

1. une *collision* s'il y a eu au moins deux essais de transmission,
2. un *silence* si aucune station n'a tenté de transmettre son message,
3. ou un *succès* s'il y a eu un seul essai de transmission.

Un tel réseau avec détection de *collision* est appelé *canal à accès multiple*.

Pour faire face à un tel système *distribué* dont la topologie est souvent inconnue vu qu'il est défini simplement à partir d'un ensemble de *nœuds*, et que la notion de file d'attente est incompatible avec l'absence d'un centre de contrôle pour réguler les transmissions, les algorithmes *déterministes*, *i.e.* écrits sur des règles de type déterministe, sont inappropriés.

Ainsi, naquit le premier réseau informatique radio, appelé ALOHA. L'idée de base était de résoudre les conflits éventuels en départageant les opposants par tirage au sort, *l'aléatoire* était ainsi introduit pour désengorger le système afin que les délais moyens d'attente avant transmission ne soient pas trop élevés et que les *collisions* ne se répètent pas indéfiniment. L'évolution des tailles de réseaux ajoutée à un trafic plus intense mettaient en doute la robustesse du protocole ALOHA, dont l'instabilité fut démontrée plus tard (Fayolle *et al.*). Metcalfe [79] proposa alors un protocole plus adaptatif que ALOHA, ETHERNET, et ce en modulant *l'aléatoire*, *i.e.* le tirage au sort, par le nombre d'échecs de transmission. Bien que devenu un vrai succès industriel, ETHERNET s'avère aussi instable (voir Aldous [1]). En 1979, *l'algorithme en arbre binaire* a été proposé comme *protocole d'accès* à un réseau de communication de type *canal de transmission à accès multiple* indépendamment, par Capetanakis [16] et Tsybakov et Mikhaïlov [101]. Voici sa description.

Description :

On rappelle qu'un ensemble d'émetteurs cherchent à transmettre leurs messages à travers un canal unique. Chaque émetteur écoute le canal. Un émetteur noté x et ayant un message à transmettre est muni d'un compteur C_x initialisé à 0. Au début de chaque unité de temps, chaque émetteur doit décider s'il tente, ou pas, une transmission et ce selon la règle suivante :

- Si $C_x = 0$, alors l'émetteur x effectue un essai de transmission.
 1. Si *succès*, c'est terminé pour lui.

-
2. Si *collision*, tirage *pile* ou *face* :
 - si *pile*, $C_x = 0$,
 - sinon, $C_x = 1$.
- Si $C_x > 0$, pas d'essai de transmission. L'émetteur x écoute cependant le canal :
1. Si *collision*, $C_x \rightarrow C_x + 1$.
 2. Sinon, $C_x \rightarrow C_x - 1$.

Contrairement à ALOHA ou ETHERNET, l'*algorithme en arbre* combine l'utilisation, d'une part de l'*aléatoire* pour départager les émetteurs mis en concurrence à un instant donné, et d'autre part d'un *compteur* qui s'adapte à l'information délivrée par le canal, même s'il n'y a pas eu de tentative de transmission. Appliqué à un ensemble initial de n émetteurs dont tous les compteurs sont initialisés à zéro, cet algorithme impose n tirages indépendants à *pile* ou *face* suite à la première collision (on suppose $n \geq 2$). L'ensemble initial des n émetteurs est ainsi divisé en deux sous-ensembles : le premier, noté E_0 , contient ceux ayant conservé un compteur nul suite à un tirage *pile*, le deuxième, noté E_1 , contient les autres émetteurs, soit ceux avec un compteur égal à 1. L'algorithme est alors appliqué au sous-ensemble E_0 selon la règle de priorité des émetteurs à compteurs nuls. E_0 est, soit divisé en deux par le même procédé en émetteurs à compteur nul et d'autres à compteur égal à 1 et ceux du sous-ensemble E_1 voient alors leur compteur augmenté de 1, soit le sous-ensemble E_0 contient au plus un émetteur qui réussit ainsi à transmettre correctement son message à travers le canal, l'algorithme est alors appliqué au sous-ensemble E_1 dont les compteurs associés, diminués de 1, deviennent nuls. Donc, à la fin de cette étape, l'ensemble initial de taille n est divisé en 3 sous-ensembles selon les valeurs prises par les compteurs associés $\{0, 1, 2\}$. L'algorithme est ainsi répété récursivement jusqu'à ce que tous les sous-ensembles obtenus contiennent au plus un émetteur.

Formellement, un *algorithme en arbre* peut être décrit comme suit :

ALGORITHME EN ARBRE $\mathcal{A}(n)$

-
- CONDITION D'ARRÊT.
Si $n < 2$ \longrightarrow ARRÊT.
 - STRUCTURE D'ARBRE.
Si $n \geq 2$, diviser aléatoirement n en $(n_1, n - n_1)$.
 \longrightarrow APPLIQUER $\mathcal{A}(n_1), \mathcal{A}(n - n_1)$.
-

Coût de l'algorithme :

Pour un tel algorithme, une quantité importante est le nombre d'opérations nécessaires pour que l'algorithme arrive à terme, *i.e.* quand tous les sous-groupes contiennent au plus un élément, c'est le *coût de l'algorithme*. Notons R_n cette quantité quand le groupe initial est de taille n . Alors ce *coût* vérifie une relation du type *Diviser pour Régner*;

$$\begin{aligned}
 R_n &\stackrel{\text{dist.}}{=} 1 + R_{S_n} + R_{n-S_n}, \text{ pour } n \geq 2 \\
 R_0 &= R_1 = 1
 \end{aligned}
 \tag{1}$$

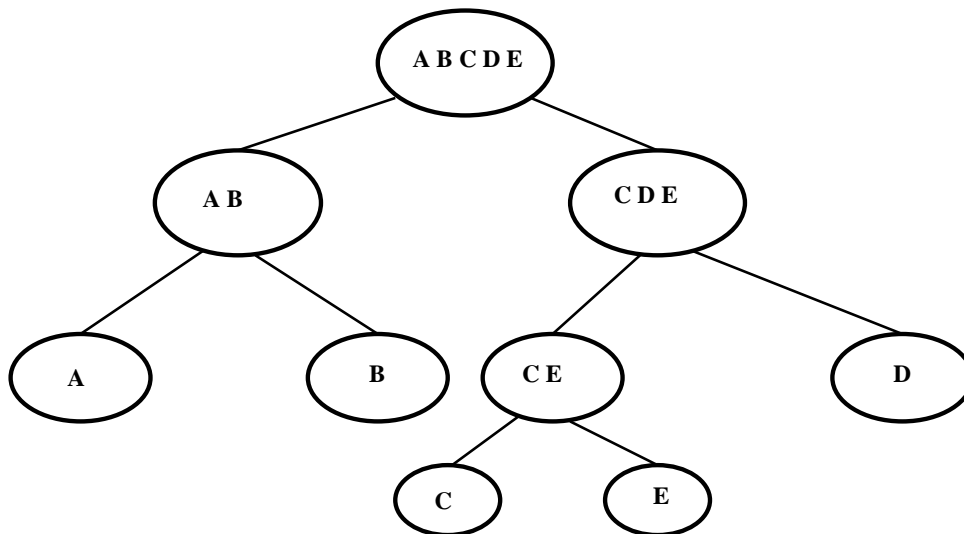


FIG. 1 – Exemple d’arbre binaire.

où S_n désigne la taille du sous-groupe de gauche qui peut être vue comme la somme de n variables aléatoires *i.i.d.* de Bernoulli de paramètre $1/2$. La figure FIG. 1 illustre cette relation de récurrence.

La variable R_n est simplement le nombre de nœuds dans l’arbre associé. Dans le cadre d’un réseau de communication, cette quantité correspond au temps total nécessaire pour transmettre n messages. Si $\mathbb{E}(R_n)$ est sa valeur moyenne, $\mathbb{E}(R_n)/n$ est le temps de transmission moyen d’un message parmi n . D’un point de vue probabiliste, il est naturel d’espérer que la suite (R_n) obéisse à une sorte de loi des grands nombres, *i.e.* que $(\mathbb{E}(R_n)/n)$ converge vers une certaine limite α . La constante α est, en un certain sens, la moyenne asymptotique de temps de transmission d’un message. Curieusement, cette loi des grands nombres n’est pas toujours vérifiée. Dans certains cas, la suite $(\mathbb{E}(R_n)/n)$ ne converge pas du tout, et présente même un comportement oscillatoire.

Quand les éléments sont équitablement divisés entre les deux sous-groupes, ce phénomène est bien connu. Il a été analysé utilisant des techniques d’analyse complexe, des transformations fonctionnelles (et leurs inversions associées) par Knuth [59], Flajolet *et al.* [40], Louchard et Prodinger [69] et plusieurs autres auteurs. Voir Hofri [47], Mahmoud [73] et Flajolet et Sedgewick [99] pour un traitement en détails de cette approche. Voir aussi Devroye [27] pour plus de détails sur le sujet. Robert [93] a proposé comme alternative une méthode élémentaire permettant d’obtenir le comportement asymptotique de certaines séries obéissant à un phénomène oscillatoire et ce sans utiliser les techniques d’analyse complexe.

Quand le degré de division est constant et vaut Q , *i.e.* que le groupe initial est divisé aléatoirement en Q sous-groupes, mais que les éléments ne sont pas équitablement répartis entre ces sous-groupes, les études sont alors plus rares. Utilisant des techniques d’analyse complexe, Fayolle *et al.* [33] obtiennent le comportement asymptotique de la suite associée $(\mathbb{E}(R_n))$ dans le cas binaire $Q = 2$. Mathys et Flajolet [77] donnent une généralisation de

cette étude quand Q est arbitraire.

1.2 Processus de Sélection

Dans des réseaux tels que les réseaux informatiques ou radio, les cas de *défaillance* des liens entre les différents composants du réseau, soit à cause d'éventuelles pertes de messages ou de paquets, soit suite à la défaillance même, temporaire ou définitive, d'un nœud donné, nuisent au bon fonctionnement du système lors de l'exécution d'une tâche collective, telle que la récupération de données brutes ou traitées par des processeurs dans un réseau informatique. Ces défaillances sont d'autant plus probables que la taille du réseau en question augmente. Le besoin de créer dans un réseau distribué un centre de contrôle, même *virtuel*, appelé aussi *leader*, tout en tenant compte des différentes défaillances possibles induisant la perte de cette structure centrée virtuelle est appelé : *problème d'élection de leader*.

On considère un *canal à accès multiple* composé de n stations dont un *leader* par défaut est désigné afin de contrôler et organiser le système entier. En pratique, faute de liaisons fiables entre les stations, il est possible que ce *leader* par défaut soit temporairement hors-service ce qui se traduit par le *silence* du *leader*. Dans ce cas, le système arrête ses opérations en cours et initialise la procédure d'*élection* : le système doit identifier un nouveau *leader* et ce dans un délai raisonnable pour le bon fonctionnement du réseau. On s'intéresse au *coût* de l'algorithme, *i.e.* le temps ou nombre d'unités de temps nécessaires au système pour déterminer un *leader*.

Description :

La taille n du réseau est supposée inconnue. Chaque station est supposée posséder un *numéro identifiant unique*, noté ID. Pour élire un *leader* parmi elles, les stations doivent utiliser le même algorithme. Le cas $n \in \{0, 1\}$ étant trivial, on suppose que $n \geq 2$. L'*algorithme d'élection de leader* dans sa version basique est décrit ainsi :

- Initialisation déterministe : à la première unité de temps, chaque station envoie au réseau un message avec son numéro ID. Comme $n \geq 2$, chacune détecte une *collision*.
- Processus de sélection aléatoire : chaque station S génère, d'une manière indépendante, une variable aléatoire de *Bernoulli* B_S de paramètre p . Seules celles obtenant le résultat $B_S = 1$ sont autorisées à tenter une transmission durant la prochaine unité de temps.

Pour une station donnée S , il y a deux cas :

1. Si $B_S = 1$, la station S est dite *Active*; S tente une transmission et peut ainsi détecter
 - soit un *succès*; seule la station S a effectué un essai de transmission, alors les autres stations reçoivent son message avec son numéro ID et S obtient le statut de *leader*. L'algorithme s'arrête.

— soit une *collision*; la station S n'est pas la seule à être candidate pour devenir *leader*, et donc doit générer encore une variable de *Bernoulli* B_S . Comme la station S vient juste de tenter une transmission, elle ne peut pas détecter de *silence*.

2. Sinon, la station S devient *Inactive*; elle reste candidate à devenir *leader*, continue à écouter le canal sans pour autant participer aux transmissions. Ainsi, elle peut détecter

- un *succès*; une seule station $S' \neq S$ a effectué un essai de transmission. Les autres stations (y compris la station S) reçoivent son numéro ID. La station S' obtient ainsi le statut de *leader*. L'algorithme s'arrête.
- une *collision*; bien que la station S n'a pas participé au processus de sélection, il y a eu au moins deux essais de transmissions de deux stations *Actives*. Dans ce cas, la station S est éliminée.
- un *silence*; toutes les stations sont *Inactives* (y compris S), donc la station S génère une variable de *Bernoulli* B_S pour décider d'envoyer encore, ou pas, un message avec son numéro ID au réseau.

Ainsi, à la fin de l'algorithme, une seule station reste *Active* et devient alors le *leader* du système.

L'exemple suivant illustre la procédure d'élection appliquée à un groupe de 4 stations $\{A, B, C, D\}$. Dans ce cas, le *leader* A est élu au bout de 4 unités de temps.

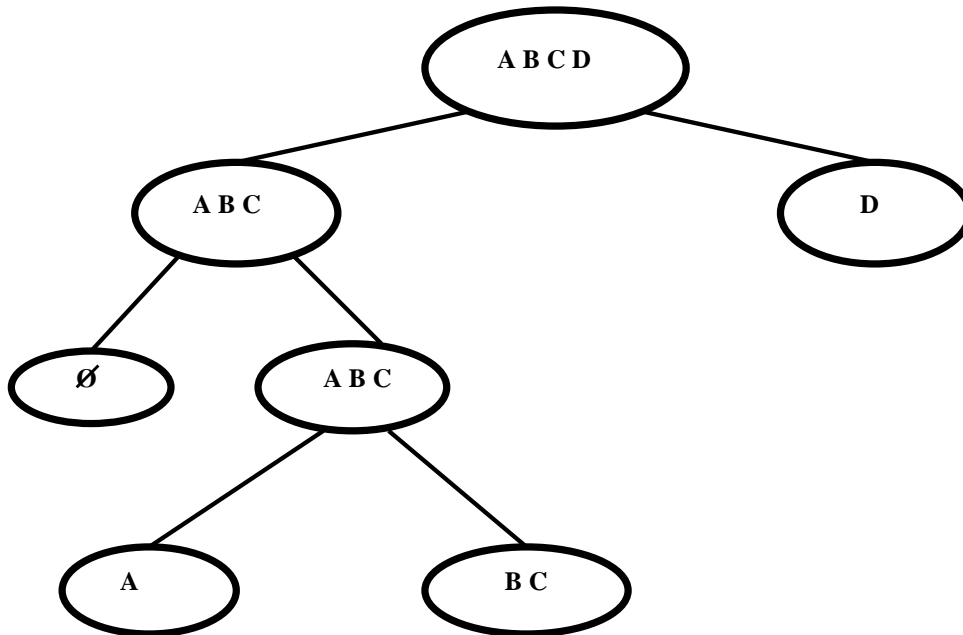


FIG. 2 – Élection du *leader* A ; $H_4 = 4$. Structure d'arbre incomplet.

Ce processus de *division* ou encore d'*éclatement* utilisant des variables aléatoires de *Bernoulli* a été introduit par Capetanakis et Tsybakhov dans le cadre des *protocoles en arbres*.

Pour plus de détails, voir [77].

Formellement, appliqué à un groupe initial de n éléments, l'algorithme divise ce dernier en deux sous-groupes. La probabilité qu'un élément soit mis dans le sous-groupe de gauche est notée p . Ce sous-groupe sera à son tour divisé en deux par le même processus. Les autres éléments seront ignorés, sauf si le premier sous-groupe, celui de gauche, est vide. Dans ce cas, l'algorithme redémarre partant du niveau précédent.

ALGORITHME D'ÉLECTION DE LEADER $\mathcal{E}(n)$

— CONDITION D'ARRÊT.

Si $n = 1$ \longrightarrow ARRÊT.

— PROCESSUS DE SÉLECTION ALÉATOIRE.

Si $n \geq 2$, diviser aléatoirement n en $(n_1, n - n_1)$.

– Si $n_1 \neq 0$ \longrightarrow APPLIQUER $\mathcal{E}(n_1)$.

– Sinon \longrightarrow APPLIQUER $\mathcal{E}(n)$.

Cet algorithme distribué est un processus d'élimination aléatoire admettant une structure d'arbre naturelle (FIG.2). À la racine de l'arbre associé se trouve le groupe initial. Lors de la première division, il génère deux nœuds : celui de gauche sera divisé en deux par le même processus, celui de droite est un nœud dit *terminal*, appelé aussi *feuille*, ne sera traité par l'algorithme que si le nœud de gauche est vide. Dans ce cas, il sera un nœud interne et sera divisé en deux à son tour.

Coût de l'algorithme :

Cette structure d'arbre peut être représentée par un *arbre incomplet* dont seule la branche reliant la racine à la première *feuille* non vide la plus à gauche sera développée, sur la figure FIG. 2, cette branche est celle reliant la racine $ABCD$ à la feuille A . On définit la *hauteur* de l'arbre associé comme étant la longueur du chemin reliant la racine au *leader*, le plus long chemin de l'arbre reliant la racine à une *feuille* (voir [38]). Alors, le *coût* de l'algorithme est égal à la *hauteur* de l'arbre associé. La figure FIG. 2 illustre cette égalité.

Il est connu que le *coût* moyen de l'algorithme d'élection de *leader* admet un *ordre logarithmique* en n et présente asymptotiquement un comportement oscillatoire. Voir [90] pour le cas non biaisé $p = 1/2$, [53] pour celui biaisé $p \neq 1/2$.

Applications :

Les applications d'un tel algorithme d'*élimination aléatoire* sont multiples dans le cadre des systèmes distribués tels que les réseaux de communication sans fil. Dans les réseaux mobiles de type *Ad-hoc*, les échecs de liaisons se produisent dès que les nœuds, étant mobiles, quittent une zone appelée *région de transmission* définie par rapport à la position des nœuds dans l'espace. La topologie instable du réseau rend le problème d'élection de *leader* plus complexe. L'unicité du *leader* n'est pas adaptée à la nature de la topologie dynamique du système ; on parle alors de décomposer le système initial en sous-systèmes temporairement stables dans l'espace et un *leader* est alors associé à chacun de ces sous-systèmes. Pour plus de détails, voir [74]. Élire un *leader* dans un réseau informatique est fondamental pour superviser les communications entre les machines et ainsi synchroniser le système. En effet, l'hypothèse d'un système initialement synchronisé n'assure pas une

synchronisation durable vu la possibilité des *défaillances* temporaires des machines, la perte de paquets envoyés sur le réseau, . . . Le problème d'élection de *leader* devient alors plus complexe dans un système asynchrone, l'exécution même de l'algorithme nécessitant la synchronisation du système. Voir [38]. Le même problème est aussi étudié dans le cadre des réseaux radio. Pour plus de détails, voir [19]. Pour des versions plus élaborées de l'algorithme d'élection de *leader* dans un réseau radio sans détection de *collision*, d'où une perte d'information, ajoutée à une contrainte d'optimalité en terme de *coût* d'énergie due à l'exécution de l'algorithme, voir [64].

1.3 Algorithmes de Recherche

Pour des opérations, telles que la *recherche* ou le *tri*, sur des bases de données, des algorithmes basés sur le principe de dichotomie et donc ayant des structures arborescentes s'avèrent être très performants en termes des différents *coûts* associés à de telles opérations. Dans ce paragraphe, on s'intéressera à deux types d'algorithmes de recherche : l'*arbre binaire de recherche* et le *trie*.

1. Arbre binaire de recherche :

Description :

En informatique, un *arbre binaire de recherche* (**ABR**) est un arbre binaire dans lequel chaque nœud possède une *clé*, telle que chaque nœud du sous-arbre *gauche* ait une clé inférieure ou égale à celle du nœud considéré, et que chaque nœud du sous-arbre *droit* ait une clé supérieure à celle-ci. Selon l'implémentation de l'ABR, on pourra interdire ou non des clés de valeur égale.

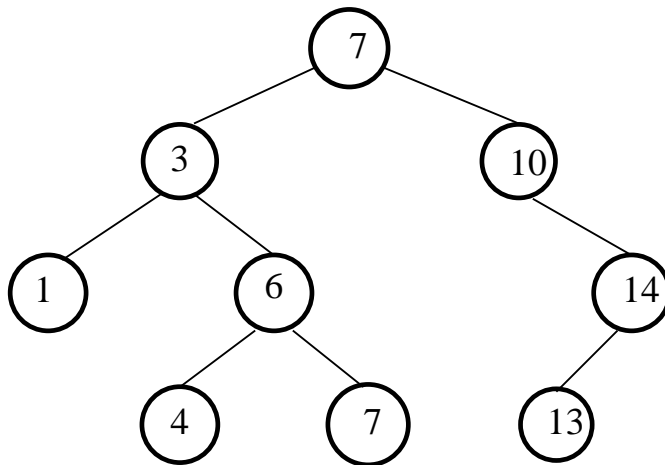


FIG. 3 – Exemple d'arbre binaire de recherche sur la chaîne : {7,3,10,14,1,6,7,4,13}.

Cet algorithme intuitif, introduit pour la première fois par Hibbard (1962), s'avère être très performant pour stocker des données vérifiant une relation d'ordre total en vue d'effectuer des opérations sur cette base de données, telles que la *recherche*, l'*insertion*, la *suppression* ou encore le *tri*.

Recherche, Insertion :

La recherche dans un arbre binaire d'un nœud ayant une clé particulière est un procédé récursif. On commence par examiner la racine ; si sa clé est celle recherchée, l'algorithme s'arrête et renvoie comme résultat la racine. Sinon, deux cas sont possibles ; si elle est strictement inférieure, alors la clé recherchée est dans le sous-arbre de gauche, sur lequel est alors effectuée récursivement la recherche. De même si la clé de la racine est strictement supérieure à la clé recherchée, la recherche continue sur le sous-arbre de droite. Si on atteint un nœud terminal, appelé aussi *feuille*, dont la clé n'est pas celle recherchée, on sait alors que la clé recherchée n'appartient à aucun nœud, *i.e.* la recherche est dite *vaine* ; l'élément recherché n'appartient pas à la liste des données. Dans ce cas, l'*insertion* de l'élément en question se fait en ajoutant un nœud comme *fil*s de cette feuille suivant la relation d'ordre en comparant sa clé à celle de la feuille : si elle est inférieure, le nouveau nœud sera à gauche, sinon il sera à droite. La liste initiale des clés ordonnée ou encore *triée* est obtenue en lisant l'arbre associé en ordre symétrique ou encore l'arbre "aplati". Ainsi, la recherche est l'opération basique sur un arbre binaire, permettant l'évolution d'un ABR au fur et à mesure de l'évolution de la base de données sur laquelle il est construit. En effet, il y a exactement une position pour l'insertion de toute nouvelle clé ; une recherche est alors effectuée pour chaque élément ajouté à liste de données initiale.

Cette opération est comparable à une recherche par dichotomie ; partant de la racine de l'arbre, une branche est exclue à chaque niveau, le domaine de recherche est quasiment divisé par deux à chaque étape, il devient ainsi rapidement de plus en plus petit. D'où le caractère naturel de l'arbre binaire de recherche en tant qu'algorithme récursif pour le traitement d'une base de données totalement ordonnée.

Hauteur d'un ABR :

Considérons un ABR construit sur une liste de n éléments $\{x_1, x_2, \dots, x_n\}$. On aura soit à insérer une nouvelle clé x_{n+1} , ce qui correspond à une *recherche vaine*, soit à rechercher une existante, *recherche réussie*. Notons par H_n la *hauteur* de cet arbre, *i.e.* la longueur du chemin le plus long reliant la racine à une feuille de l'arbre. Alors cette quantité correspond au nombre maximal de comparaisons effectuées lors d'une recherche, *vaine* ou *réussie*, sur l'ABR. Il est clair, qu'en considérant les cas extrêmes, on a

$$\lfloor \log_2 n \rfloor \leq H_n \leq n - 1.$$

Pour une étude de la hauteur d'un ABR, voir Pittel [89], Devroye [25].

Coût de Construction, de Recherche :

Plus généralement, on peut ainsi introduire un *coût de construction* et un *coût de recherche*, mesurés en terme de comparaisons effectuées. Pour les deux opérations, cela consiste à parcourir un par un tous les chemins reliant la racine de l'arbre, soit à la position d'insertion, soit à la clé recherchée. Ainsi, si on désigne par E_n le *coût de construction* (respectivement I_n celui de *recherche*) d'un ABR écrit sur une liste de taille n , alors

$$E_n = \sum_{i \text{ position d'insertion}} P(i), \quad I_n = \sum_{u \text{ noeud}} P(u),$$

où $P(u)$ est la profondeur du nœud u . La quantité E_n (resp. I_n) est dite *cheminement externe* (resp. *interne*) avec la relation suivante

$$E_n = I_n + 2n.$$

Lors des opérations de recherche ou d'insertion, l'arbre sera parcouru plusieurs fois, quasiment proportionnellement à la taille n de la liste initiale sur laquelle il est construit. Ainsi, ces deux *coûts*, de *construction* et de *recherche*, mesurent la performance de l'ABR pour des listes de tailles considérables. En considérant le cas le meilleur, on obtient que le *coût de construction* $E_n = O(n \log n)$, celui le pire $E_n = O(n^2)$.

La question à poser alors est : si on s'intéresse à tous les ABR construits sur des listes de taille fixée n , le pire cas est-il fréquent ?

Pour tenir compte du comportement de l'ABR entre les deux cas extrêmes, on considère le *modèle des permutations uniformes*, *i.e.* qu'on munit l'ensemble \mathcal{S}_n des permutations de $\{1, \dots, n\}$ de la loi uniforme et qu'on associe à toute permutation un ABR écrit sur la liste permutée. Ainsi les listes permutées sont équiprobables et non les ABR, puisque deux permutations différentes peuvent engendrer le même arbre.

Ce modèle, plutôt que celui des *arbres de Catalan*; *i.e.* une loi uniforme sur les arbres binaires eux-mêmes, est le plus pertinent, vu que l'aléatoire est dû à la liste des données et non, pour une liste fixée, à la construction même de l'arbre ; en effet, dès lors que la liste des données initiale est fixée, la construction de l'ABR sera alors basée sur une règle de comparaison *déterministe*, une relation d'ordre. Les résultats pour les deux modèles peuvent être reliés par une transformation de nature combinatoire due à Brink [14], [13].

Sous ce modèle aléatoire, le cas le pire s'avère sans influence puisque l'algorithme admet des *coûts moyens* raisonnables pour des listes de tailles importantes, et que, obéissant à la loi des grands nombres, ces *coûts* tendent en probabilité vers leurs moyennes. Pour plus de détails, voir entre autres Knuth [59], Lynch [71], Brown et Shubert [15], Louchard [68], Mahmoud [72], Régnier [96] ou encore Rösler [98].

2. Tries :

Il existe une autre classe d'algorithmes de recherche qui, contrairement à un arbre binaire de recherche, ne stockent pas les données dans les nœuds de l'arbre ; en effet, c'est la position du nœud dans l'arbre qui détermine la donnée correspondante. Pour tout nœud, ses descendants ont en commun le même *préfixe*, *i.e.* la liste des données correspondant à la branche reliant la racine de l'arbre à ce nœud. La racine est associée à l'ensemble vide. Aucune valeur n'est attribuée aux nœuds, sauf les *feuilles*.

Description :

Un *arbre digital*, appelé aussi *trie*, est une structure de données adaptée aux opérations de recherche, tri, codage ou compression sur un texte assez large tel que des

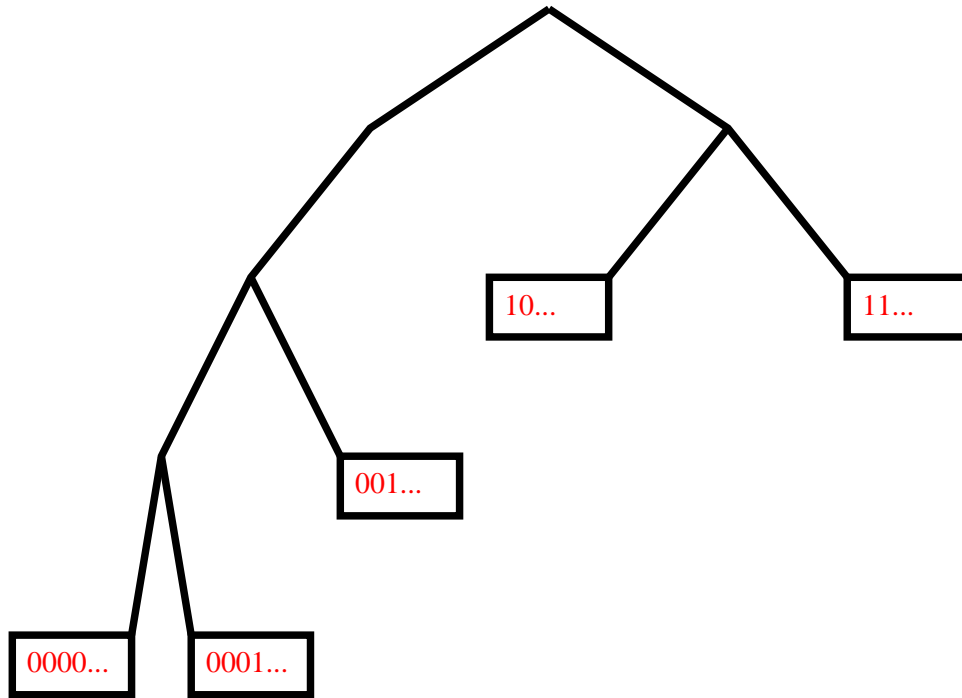


FIG. 4 – Trie binaire pour 5 chaînes de bits 000011, 000111, 001010, 100100, 110011.

chaînes de caractères, ou encore des bases de données de type *Alphabet*. La relation d'ordre est de type *lexicographique* utilisée par exemple dans les dictionnaires ou les répertoires téléphoniques ; les mots sont alors ordonnés selon leurs premières lettres, puis récursivement jusqu'à les séparer les uns des autres. En effet, chaque lettre examinée conduit à restreindre l'ensemble des mots qui partagent le même *préfixe*. Ce mécanisme récursif s'arrête dès lors que chaque mot est isolé. Ces algorithmes à structure arborescente sont très performants pour des applications sur des ensembles de mots très larges.

Structure d'un Trie :

Un *trie* est un arbre de type k -aire, chaque nœud admettant au plus k fils. Il est construit sur un *alphabet* de taille n augmenté d'un *symbole terminal* $\$$ marquant la fin d'un mot. Quelques exemples d'*alphabets* :

- $\{0, 1\}$ pour les fichiers binaires.
- {les 256 caractères ASCII}
- $\{a, b, c, \dots, x, y, z\}$
- $\{A, C, T, G\}$ pour les séquences d'ADN.

Les *tries* ont une structure compacte vu que dès lors qu'un mot peut être isolé, le processus de branchement naturel de l'arbre s'arrête pour ce mot en une *feuille* portant le mot en question. L'exemple de la FIG. 4 illustre ce gain en complexité. Cependant, cette structure peut encore être rendue plus compacte ; en effet, les branches de type *fils unique* sur la structure du *trie* n'apportent rien en termes d'information utile lors d'une opération de recherche, mais représente une perte en

espace. En “écrasant” ces branches et en associant à un nœud interne plus d’un caractère, on obtient une structure appelée *arbre PATRICIA*. Le même exemple est traité avec cette nouvelle structure. Voir la FIG. 5

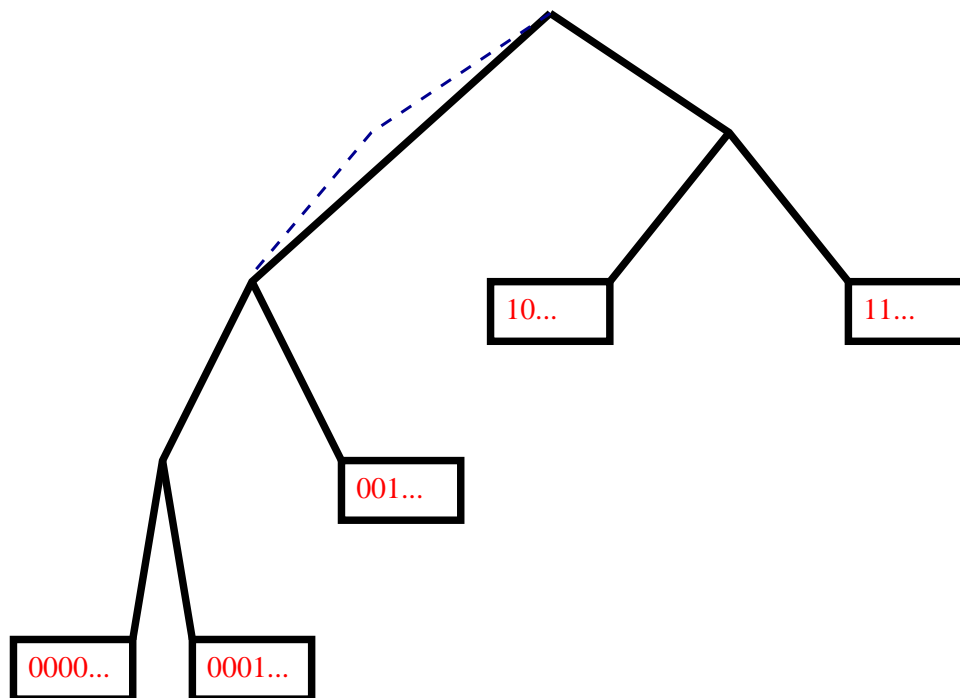


FIG. 5 – Arbre PATRICIA : suppression d’un nœud interne père à fils unique

De nombreux travaux ont été effectués sur des *tries* de type *dynamique*, *i.e.* quand un mécanisme, appelé *source*, génère d’une manière *déterministe* les lettres de l’alphabet à partir d’un *aléa initial* pour ainsi former les mots. Différents types de *sources dynamiques* sont alors considérés dont les classiques sont les *sources sans mémoire*, *markoviennes*. Voir [20].

1.4 Processus de Fragmentation

Une version continue des *algorithmes en arbre* peut être définie comme suit : une masse initiale de taille x est éclatée aléatoirement en plusieurs pièces qui, à leur tour, seront éclatées par le même processus et ainsi de suite. Une classe de tels modèles a été récemment étudiée. La fragmentation de chaque masse a lieu au bout d’intervalles de temps *i.i.d.* de loi exponentielle dont le paramètre peut dépendre de sa masse associée. Voir Bertoin [6], Miermont [80] et autres. Ce changement d’échelle (taille/temps) permet de transformer ce problème de fragmentation en un problème de branchement, et ainsi se mettre d’une manière naturelle dans le cadre des *processus de branchement*.

En effet, un *processus de branchement* général, parmi les *processus de naissance et de mort*, est un arbre généalogique où on associe à l’existence de l’ancêtre le temps $t = 0$. Chaque individu u est né à un instant $\sigma_u > 0$, admet une durée de vie infinie et se reproduit

selon un processus ξ_u sur $[0, \infty[$. On note $\xi_u(t) = \xi_u([0, t])$ le nombre de descendants de l'individu u qui sont nés avant la date t . En particulier, $\xi_u(C) = \xi_u(\infty)$. Ce processus est caractérisé par la quantité suivante, appelée *fonction de reproduction*

$$\mu(t) = \mathbb{E}[\xi(t)].$$

Le caractère récursif de ce processus de branchement se traduit par une équation fonctionnelle, *équation de convolution*, vérifiée par la fonction de reproduction associée μ . Les études de certains paramètres, tels que le nombre moyen d'individus vérifiant un critère donné (une date de naissance avant un certain t_0 par exemple), sont basées sur la *théorie de renouvellement*. Pour plus de détails, voir [36], [45], [52].

Dans le cadre d'un *processus de fragmentation*, les problèmes considérés sont légèrement différents de ceux rencontrés pour les *algorithmes en arbre* : les propriétés de régularité des processus de *Markov* associés, le taux de décroissance des masses individuelles, la perte de masse, les distributions asymptotiques, ... Un *algorithme en arbre* est juste une fragmentation récursive d'un entier en *pièces entières* jusqu'à ce que chacune ait une taille strictement inférieure à un entier fixé D . Dans un contexte continu, l'analogue de l'algorithme en arbre consisterait à arrêter le processus de fragmentation de masse dès que la taille de cette dernière soit inférieure à un certain seuil $\varepsilon > 0$. Voir Bertoin [6] pour une analyse d'un *processus de fragmentation de masse avec perte*.

Un point de vue similaire est de considérer un *algorithme en arbre* comme étant un *processus récursif de décomposition aléatoire* de l'intervalle $[0, 1]$. Par exemple, dans le cas d'un *algorithme en arbre binaire* appliqué à un groupe initial de taille n , diviser ce groupe initial en deux sous-groupes et vérifier sur chacun des deux la condition d'arrêt de l'algorithme, avoir au plus un élément par sous-groupe, puis ainsi de suite, revient à considérer la décomposition dyadique partant de l'intervalle $I_0^0 = [0, 1]$; deux sous-intervalles I_0^1, I_1^1 sont créés et chacun des deux est divisé en deux à son tour ... et à chaque niveau de cette décomposition, la même condition d'arrêt est vérifiée. Ainsi, cette décomposition récursive s'arrête dès que sur un niveau donné ℓ , chacun des sous-intervalles $(I_k^\ell)_{0 \leq k \leq 2^\ell - 1}$ contienne au plus un élément parmi les n initialement répartis aléatoirement sur l'intervalle $[0, 1]$.

Alors, si on considère le modèle poissonnien, *i.e.* la taille du groupe initial est aléatoire suivant une loi de *Poisson*, la condition d'arrêt de l'*algorithme en arbre binaire* s'exprime en fonction des longueurs des sous-intervalles obtenus par la décomposition associée, puisque, sur chaque sous-intervalle obtenu, on a un nombre aléatoire d'éléments suivant une loi de *Poisson* de paramètre la longueur du sous-intervalle considéré. D'où une description similaire à la représentation classique des processus de fragmentation. Le fait de considérer des *poids aléatoires* sur l'*algorithme en arbre binaire*, *i.e.* une probabilité *aléatoire* d'aller à gauche ou à droite dans l'arbre associé, rend cette décomposition *aléatoire* à son tour.

Ce parallèle entre *algorithme en arbre* et *décomposition aléatoire récursive* est très utile pour l'étude, non pas de la moyenne du *coût* de l'algorithme, mais de sa distribution. Des résultats de loi des grands nombres et de théorème de la limite centrale ont ainsi été établis, voir [81].

2 Contribution de la thèse

L'objectif de ce travail est double. D'une part, il est dédié aux *algorithmes en arbre, statiques* ou *dynamiques*, ayant un degré aléatoire (et donc non borné) généralisant ainsi les études précédentes dans ce domaine. D'autre part, il propose une approche probabiliste pour l'analyse de cette classe d'algorithmes. De plus, une nouvelle représentation explicite du comportement asymptotique de l'algorithme est établie.

2.1 La classe des algorithmes en arbre

Dans un cadre général, on définit une classe d'algorithmes appelés *algorithmes en arbre*. Un *algorithme en arbre* est une procédure qui divise récursivement en sous-groupes un groupe initial de n éléments jusqu'à ce que chaque sous-groupe obtenu ait un cardinal strictement inférieur à un nombre fixé D . Formellement, un algorithme en arbre peut être décrit comme suit :

ALGORITHME EN ARBRE $\mathcal{A}(n)$

— CONDITION D'ARRÊT.

Si $n < D$ \longrightarrow ARRÊT.

— STRUCTURE D'ARBRE.

Si $n \geq D$, diviser aléatoirement n en n_1, \dots, n_G , avec $n_1 + \dots + n_G = n$ où G est une variable aléatoire de distribution donnée.

\longrightarrow APPLIQUER $\mathcal{A}(n_1), \mathcal{A}(n_2), \dots, \mathcal{A}(n_G)$.

Description :

Appliqué à un groupe initial de taille n , l'algorithme divise cet ensemble en G sous-groupes, la distribution de G est donnée par $\mathbb{P}(G = \ell) = p_\ell$, où (p_ℓ) est une distribution de probabilité sur $\{2, 3, \dots\}$. Conditionnellement à l'événement $\{G = \ell\}$, pour $1 \leq i \leq \ell$, un élément est mis dans le i -ème sous-groupe avec poids aléatoire $V_{i,\ell}$, où $V_\ell = (V_{i,\ell}; 1 \leq i \leq \ell)$ est un vecteur aléatoire de probabilité. Il peut aussi être considéré comme une marche aléatoire branchante.

Si N_i désigne le cardinal du i -ème sous-groupe, alors, conditionnellement à l'événement $\{G = \ell\}$ et aux variables aléatoires $V_{1,\ell}, V_{2,\ell}, \dots, V_{\ell,\ell}$, la distribution du vecteur (N_1, \dots, N_ℓ) est multinomiale de paramètres n et $(V_{1,\ell}, V_{2,\ell}, \dots, V_{\ell,\ell})$,

$$\mathbb{P}((N_1, \dots, N_\ell) = (m_1, \dots, m_\ell) | V_\ell) = \frac{n!}{m_1! m_2! \dots m_\ell!} \prod_{k=1}^{\ell} (V_{k,\ell})^{m_k},$$

pour $(m_i) \in \mathbb{N}^n$ tels que $m_1 + \dots + m_\ell = n$. Si le i -ème sous-groupe, $1 \leq i \leq n$, est tel que $N_i < D$, l'algorithme s'arrête pour ce sous-groupe. Sinon, il est appliqué pour le i -ème sous-groupe : une variable G_i , de même distribution que G , est engendrée et ce i -ème sous-groupe est divisé en G_i sous-groupes, et ainsi de suite.

Un tel cadre de division aléatoire a été introduit par Devroye [26] où le comportement asymptotique de la *profondeur* de l'arbre associé a été étudié.

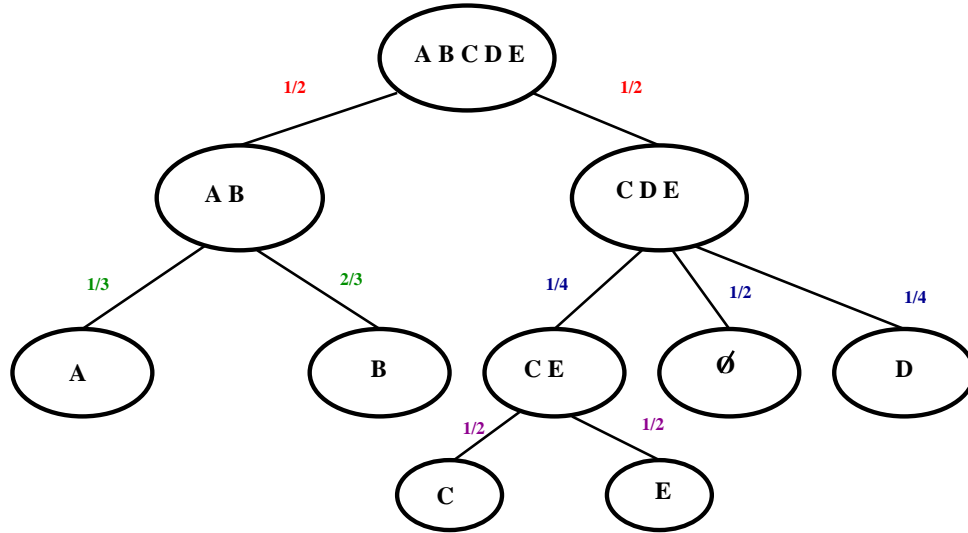


FIG. 6 – Algorithme en arbre avec $D = 2$, deux ensembles de poids aléatoires $(V_{1,2}, V_{2,2})$ et $(V_{1,3}, V_{2,3}, V_{1,3})$, G une variable aléatoire à valeurs dans $\{2, 3\}$, le groupe initial étant $\{A, B, C, D, E\}$.

Exemples.

- Algorithme en Arbre Binaire. Quand $\mathbb{P}(G = 2) = 1$, $D = 2$ et $V_{1,2} \equiv V_{2,2} \equiv 1/2$, on retrouve l'un des premiers algorithmes de ce genre. Il a été étudié par Knuth en 1973.
- Algorithme en Arbre Symétrique. C'est le cas quand $V_{i,n} \equiv 1/n$ pour tout $n \geq 2$ et $1 \leq i \leq n$.
- L'Algorithme Q -aire. Si $\mathbb{P}(G = Q) = 1$ et $D = 2$, on retrouve l'algorithme de résolution de type Q -aire avec arrivées bloquées, algorithme analysé par Mathys et Flajolet [77].

Voir aussi Devroye [26] pour d'autres exemples.

Coût de l'Algorithme en Arbre :

Comme pour l'*algorithme en arbre binaire*, le nombre d'opérations nécessaires pour que l'algorithme arrive à terme, *i.e.* quand tous les sous-groupes obtiennent un cardinal strictement inférieur à D désigne le *coût* de l'algorithme. Notons par R_n cette quantité quand le groupe initial est de taille n .

Algorithme en Arbre Non Conservatif :

On considère un *algorithme en arbre* de degré aléatoire G divisant aléatoirement un groupe initial de taille n en G sous-groupes **avec possibilité de perte**; *i.e.* conditionnellement à l'événement $\{G = \ell\}$, pour $1 \leq i \leq \ell$, un élément du groupe initial est, soit mis dans le i -ième sous-groupe avec probabilité $V_{i,\ell}$, soit perdu avec une probabilité $V_{0,\ell}$, où $V_\ell = (V_{i,\ell}; 0 \leq i \leq \ell)$ est un vecteur aléatoire de probabilité sur $\{0, \dots, \ell\}$.

Dans le cadre des réseaux de communication, le caractère *non conservatif* d'un algorithme traduit la perte de messages lors des tentatives d'accès au canal ; en effet, certains messages quittent le statut de *message en attente* sans pour autant être correctement transmis sur le canal. Ainsi, un *algorithme en arbre non conservatif* de degré G consiste à diviser aléatoirement un groupe initial en $(G + 1)$ sous-groupes, le dernier sous-groupe correspond à tous les messages perdus et donc qui n'interviennent plus dans le processus de division pour l'accès au canal.

Algorithme en Arbre Dynamique :

Une version *dynamique* de l'algorithme en arbre, motivée par les réseaux de communication, a été étudiée par Mohamed et Robert [82]. En effet, les réseaux à accès multiple ne sont pas bloqués durant l'exécution du *protocole d'accès* pour un groupe initial de n messages, soit pour une durée en temps aléatoire égale à R_n . Au cours de l'exécution du *protocole* pour les n messages, de nouveaux messages peuvent rejoindre le groupe initial avec la règle de *priorité pour les nouveaux*, *i.e.* avec un compteur nul. Ceci se traduit sur la structure de l'arbre associé par l'ajout aléatoire à chaque nœud de l'arbre de nouveaux éléments, qui seront à leur tour départagés le long des branches de l'arbre.

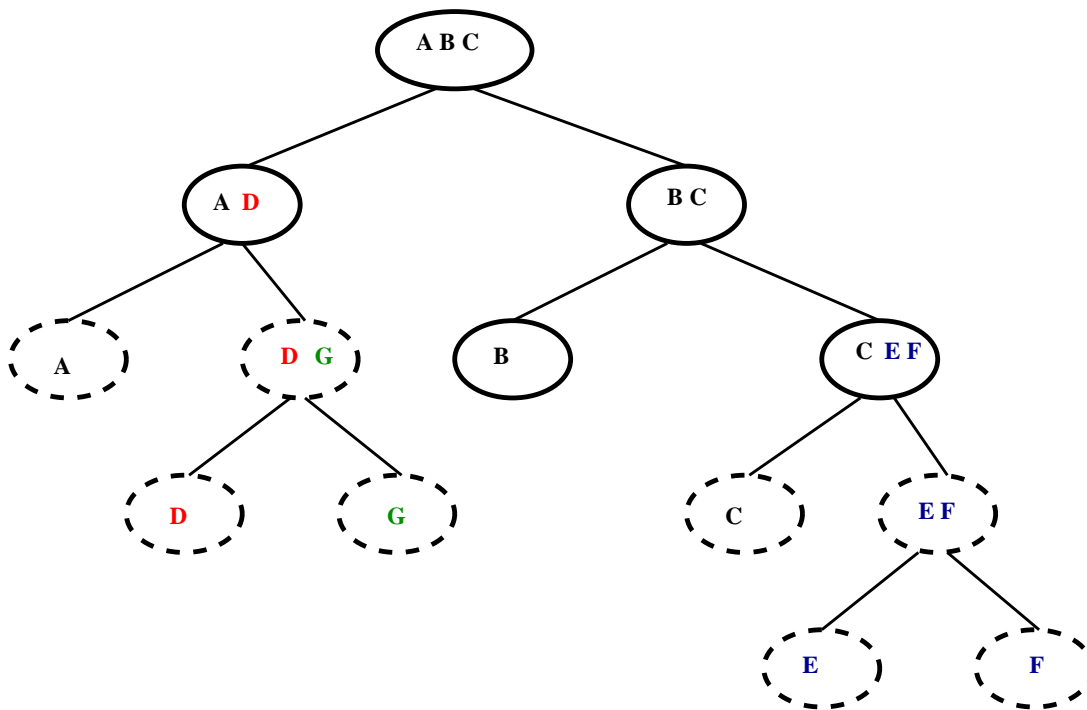


FIG. 7 – Algorithme en arbre binaire avec arrivées libres aléatoires à valeurs dans $\{0, 1, 2\}$, le groupe initial étant $\{A, B, C\}$.

Les études faites sur les *algorithmes en arbre dynamiques* sont assez rares, citons les travaux de Flajolet *et al.* [77] sur les algorithmes Q -aires, *i.e.* de degré fixé Q et correspondant

à un cardinal d'arrêt inférieur strictement à $D = 2$, avec des arrivées libres suivant une *loi de Poisson* de paramètre λ .

Outre l'analyse, exacte ou asymptotique, du *coût* ou encore du *coût moyen*, l'*algorithme en arbre dynamique* met en évidence deux problèmes propres à son caractère *dynamique* ;

- le premier est la *stabilité* de l'algorithme ; existe-t-il un taux d'arrivées critique $\lambda_c > 0$ tel que, quelle que soit la taille du groupe initial et quel que soit le taux d'arrivées libres $\lambda < \lambda_c$, l'algorithme s'arrête en un temps presque sûrement fini ?

Curieusement, l'existence de cette région de stabilité n'a pas été proprement abordée. L'approche classique formule le *coût moyen*, noté r_n^λ , d'un *algorithme en arbre dynamique* comme étant la solution d'une équation de récurrence de type *Diviser pour Régner*, équation dont la suite $(1, \dots, 1, \infty, \dots)$, soit

$$\begin{aligned} r_n^\lambda &= \infty, \text{ pour } n \geq D \\ r_0^\lambda &= \dots = r_{D-1}^\lambda = 1 \end{aligned}$$

est solution. Autrement dit, le temps d'exécution moyen peut être infini pour des groupes de tailles initiales supérieures au cardinal d'arrêt D . Ainsi, en résolvant cette équation de récurrence, l'approche analytique fournit **une solution** possible, celle *finie*, et non pas **la solution** vu qu'il n'y a pas d'unicité et que la solution *infinie* est omise.

- le second est dû à l'écriture même de l'équation fonctionnelle caractérisant un tel algorithme, équation dont la forme basique, le cas binaire, est donnée par

$$r(x) = r(px + \lambda) + r(qx + \lambda) + h_\lambda(x),$$

où $p + q \leq 1$ (le cas conservatif correspond à $p + q = 1$) et h_λ est une fonction donnée qui s'exprime en terme de $D + 1$ constantes $c_i(\lambda)$. Ainsi, le *coût moyen*, exact ou asymptotique, ne pourra que s'exprimer en fonction de ces constantes. La détermination de ces constantes d'une part, et la condition de leur existence, condition écrite sur le taux d'arrivées λ d'autre part, sont deux points essentiels de l'étude d'un tel algorithme. L'approche analytique proposée dans le cas particulier $D = 2$ par Flajolet *et al.* [35] ne paraît pas envisageable à généraliser.

2.2 Résultats et Approche.

Le point de départ de toute étude de cette classe d'algorithmes est une relation de récurrence de type *Diviser pour Régner* vérifiée par la suite des coûts de l'algorithme $(R_n)_{n \geq 0}$. La *transformée de Poisson* de cette suite, étape appelée *poissonisation*, transforme cette relation de récurrence en une équation fonctionnelle dont la forme la plus basique, le cas binaire statique, est du type

$$r(x) = r(px) + r(qx) + h(x),$$

où $p + q \leq 1$ et h est une fonction donnée. Le traitement de cette équation fonctionnelle par l'approche probabiliste proposée ici est significativement différent de celui de l'étude analytique. Après quelques transformations, cette équation est interprétée comme une équation probabiliste qui sera traitée utilisant une suite de variables aléatoire *i.i.d.* appropriées. Suivant la méthode de Robert [93], l'étape suivante sera d'effectuer une dépoissonisation probabiliste et ainsi, en utilisant convenablement le théorème de *Fubini*, obtenir une représentation intégrale de la quantité $\mathbb{E}(R_n)$ en fonction de n variables aléatoires uniformes sur l'intervalle $[0, 1]$. Le point final est une étape cruciale qui s'avère différente de Robert [93]; en effet, elle consiste à utiliser le théorème de *renouvellement* pour obtenir le comportement asymptotique de la suite $(\mathbb{E}(R_n))$.

L'approche est élémentaire, son plus grand avantage par rapport au traitement analytique classique est certainement dû à l'utilisation du théorème de *renouvellement* qui établit *directement* le comportement asymptotique de l'algorithme.

La Section 1.2 donne une représentation intéressante du coût moyen de l'algorithme. Le résultat le plus important pour l'*algorithme en arbre statique* concerne le comportement asymptotique de l'algorithme et se traduit par les Propositions 3 et 4. Ces propositions couvrent et étendent certains résultats dans ce domaine : pour l'algorithme en arbre binaire, Knuth [59] et pour les algorithmes Q -aires avec arrivées bloquées, Mathys et Flajolet [77], voir les Corollaires 2 et 3. Le deuxième chapitre généralise ces résultats pour le cas *non conservatif*, où on tient compte d'une possible perte de messages lors du processus de division aléatoire. La reformulation probabiliste de l'équation fonctionnelle associée à ce type d'algorithme passe alors par une renormalisation des *poids* de ce processus de division, *i.e.* p et q tels que $p + q < 1$ dans le cas binaire.

La *distribution* de la suite (R_n) est étudiée dans la Section 1.4. Pour plus de simplicité, seul le cas où la variable G est constante et où les variables $V_{\cdot,G}$ sont égales à $1/G$ est considéré. Le but de cette section est de montrer que la distribution de la transformée de *Poisson* de la suite des coûts, et plus généralement, celle de la plupart des fonctionnelles de l'arbre associé, peut être exprimée simplement en fonction du processus de *Poisson* et de variables uniformément distribuées sur $[0, 1]$.

Deux représentations de la distribution de la transformée de *Poisson* sous la forme d'une fonctionnelle du processus de *Poisson* sont établies. Comme conséquence, une loi des grands nombres est prouvée quand le groupe d'éléments initial est de taille aléatoire suivant une loi de *Poisson* (transformée de *Poisson*). De plus, le comportement oscillatoire asymptotique de l'algorithme est démontré comme conséquence d'une loi des grands nombres *standard*. Cette étrange loi des grands nombres s'avère ainsi due à une loi des grands nombres classique.

Aussi, le théorème de la limite centrale est prouvé utilisant une méthode similaire pour ce même cas. C'est un résultat classique, voir Jacquet et Régnier [49] ainsi que Mahmoud [73], toujours prouvé utilisant des méthodes d'analyse complexe ainsi que des estimations assez techniques. Dans ce chapitre, il est prouvé comme conséquence du théorème de la limite centrale *standard* pour des variables aléatoires indépendantes. En outre, une nouvelle représentation asymptotique de la variance est obtenue.

La section 1.5 met en valeur un certain parallèle entre *algorithme en arbre* et *processus de branchement*, motivée par un travail de Bertoin et Martinez [6] sur un *processus de fragmentation*. Le comportement asymptotique du modèle *poissonnien* d'un *algorithme en arbre* est ramené à l'étude du *processus de branchement* associé.

Le troisième chapitre a pour objectif d'adapter les techniques utilisées par [81] et basées sur une reformulation probabiliste de l'équation fonctionnelle associée afin d'analyser une quantité additive dans un contexte de structure d'arbre *incomplet*.

Dans la section 3.3, une formule *en série* du *coût moyen* de l'algorithme d'*élection du leader*, noté $E(H_n)$ et similaire au *coût moyen* de l'algorithme en arbre *classique*, est donnée. Dans la section 3.4, le comportement asymptotique de l'algorithme est reformulé en fonction de celui d'un certain temps d'arrêt τ défini par deux suites aléatoires sur l'intervalle $[0, 1]$. Ainsi, il est prouvé que le *coût moyen centré* de l'algorithme, $E(H_n) - \log_p(n)$, est équivalent à une fonction périodique de $\log_p(n)$, fonction dont une formule explicite est établie.

Dans la section 3.5, la distribution du *coût* de l'algorithme est étudiée. Utilisant la décomposition binaire de l'intervalle $[0, 1]$, l'expression exacte de la distribution de H_n est établie. Ce résultat est une variation légèrement différente de la formule asymptotique donnée par [53] pour la distribution du *coût* de l'algorithme H_n dans le cas asymétrique.

La seconde partie de ce mémoire est dédiée aux *algorithmes en arbre dynamiques* avec ou sans perte. Le premier chapitre de cette partie définit formellement le modèle d'un tel algorithme et explique les motivations incitant à prendre en compte les deux caractères ; *dynamique* et *non conservatif* à la fois. Deux problèmes sont alors posés, la stabilité d'un tel algorithme sous un modèle général d'arrivées libres d'une part, et l'expression exacte du *coût moyen* pour des arrivées *poissonniennes* d'autre part.

2.3 Plan

La première partie est dédiée à la classe des algorithmes en arbre statiques ; le premier chapitre est une étude basée sur des outils probabilistes d'un *algorithme en arbre statique conservatif* d'un point de vue comportement asymptotique et distribution. Le deuxième chapitre généralise les résultats du premier au cas des *algorithmes en arbre statiques non conservatifs*. Enfin, le dernier chapitre est une adaptation de l'approche utilisée pour les *algorithmes en arbre* au cas d'un algorithme asymétrique d'un point de vue structure arborescente, appelée *algorithme d'élection de leader*.

La deuxième partie de ce mémoire est dédiée au cas *dynamique*. Dans le premier chapitre, la classe des *algorithmes en arbre dynamiques* est définie. La notion de *région de stabilité* est étudiée et un résultat d'existence est établi pour des arrivées aléatoires non forcément *poissonniennes*. Le deuxième chapitre est une étude d'un *algorithme en arbre dynamique non conservatif* où une formule exacte du *coût moyen* de l'algorithme est établie. La condition de stabilité est établie dans le cas d'arrivées *poissonniennes*.

Première partie

Cas statique

Chapitre 1

Algorithme en arbre avec arrivées bloquées

Sommaire

1.1	Introduction	1
1.1.1	Description	2
1.1.2	Étrange Loi des Grands Nombres	5
1.1.3	Résultats du chapitre	6
1.2	Propriétés Générales	7
1.2.1	Équation Intégrale Fonctionnelle	10
1.3	Analyse du Comportement Asymptotique du Coût Moyen	12
1.3.1	Le Cas Non Arithmétique	12
1.3.2	Le Cas Arithmétique	14
1.4	Distribution de l'Algorithme Q-aire Symétrique	17
1.4.1	Loi des Grands Nombres	18
1.4.2	Représentation de quelques fonctionnelles de l'arbre associé	19
1.4.3	Théorèmes de la Limite Centrale	22
1.4.4	La Distribution de la suite (\mathbf{R}_n)	24
1.5	Algorithme en Arbre et Processus de Branchement	24
1.5.1	Description du modèle	25
1.5.2	Approche	26
1.5.3	Un théorème de renouvellement	28
1.5.4	Comportement asymptotique	29

1.1 Introduction

Un *algorithme en arbre* est une procédure qui divise récursivement en sous-groupes un groupe initial de n éléments jusqu'à ce que chaque sous-groupe obtenu ait un cardinal strictement inférieur à un nombre fixé D . Ces algorithmes ont un large champ d'application.

- Structures de données : Ce sont les algorithmes utilisés pour les recherches sur une base de données, appelés aussi les algorithmes *diviser pour régner*. Voir Cormen *et al.* [21] et Knuth [59] pour une présentation générale et Flajolet et Sedgewick [99], Mahmoud [73] et Szpankowski [100] pour leurs analyses basées sur des méthodes analytiques.
- Réseaux de communication : Ces algorithmes sont utilisés pour donner un accès distribué à un canal commun, qui ne peut transmettre d'un seul message par unité de temps. Voir Capetanakis [16], Tsybakov et Mikhaïlov [101] ainsi que Ephremides et Hajek [30].
- Systèmes distribués : Certains algorithmes utilisent des techniques de division pour sélectionner un sous-groupe d'un ensemble de composantes communicantes identiques, telles que des machines dans un réseau informatique ou des stations dans un réseau radio. Voir Janson et Szpankowski [53] et Raz *et al.* [91].
- Tests statistiques : Un test, effectué sur un groupe d'individus, indique si au moins l'un des ces individus présente certaines caractéristiques (telles qu'une maladie, ou un certain groupe sanguin). Le but est de minimiser le nombre de tests à effectuer afin d'identifier les personnes ayant ces critères. Voir Wolf [109]

Formellement, un algorithme en arbre peut être décrit comme suit :

ALGORITHME EN ARBRE $\mathcal{S}(n)$

— CONDITION D'ARRÊT.

Si $n < D$ \longrightarrow ARRÊT.

— STRUCTURE D'ARBRE.

Si $n \geq D$, diviser aléatoirement n en n_1, \dots, n_G , avec $n_1 + \dots + n_G = n$ où G est une variable aléatoire de distribution donnée.

\longrightarrow APPLIQUER $\mathcal{S}(n_1), \mathcal{S}(n_2), \dots, \mathcal{S}(n_G)$.

1.1.1 Description

Appliqué à un groupe initial de taille n , l'algorithme divise cet ensemble en G sous-groupes, la distribution de G est donnée par $\mathbb{P}(G = \ell) = p_\ell$, où (p_ℓ) est une distribution de probabilité sur $\{2, 3, \dots\}$. Conditionnellement à l'événement $\{G = \ell\}$, pour $1 \leq i \leq \ell$, un élément est mis dans le i -ème sous-groupe avec un poids aléatoire $V_{i,\ell}$, où $V_\ell = (V_{i,\ell}; 1 \leq i \leq \ell)$ est vecteur aléatoire identifiable à une probabilité sur $\{1, \dots, \ell\}$.

Si N_i désigne le cardinal du i -ème sous-groupe, alors, conditionnellement à l'événement $\{G = \ell\}$ et aux variables aléatoires $V_{1,\ell}, V_{2,\ell}, \dots, V_{\ell,\ell}$, la distribution du vecteur (N_1, \dots, N_ℓ) est multinomiale de paramètre n et $(V_{1,\ell}, V_{2,\ell}, \dots, V_{\ell,\ell})$,

$$\mathbb{P}((N_1, \dots, N_\ell) = (m_1, \dots, m_\ell) | V_\ell) = \frac{n!}{m_1! m_2! \dots m_\ell!} \prod_{k=1}^{\ell} (V_{k,\ell})^{m_k},$$

pour $(m_i) \in \mathbb{N}^n$ tel que $m_1 + \dots + m_\ell = n$. Si le i -ème sous-groupe, $1 \leq i \leq n$, est tel que $N_i < D$, l'algorithme s'arrête pour ce sous-groupe. Sinon, il est appliqué pour le i -ème sous-groupe : une variable G_i , de même distribution que G , est engendrée et ce i -ème

sous-groupe est divisé en G_i sous-groupes, et ainsi de suite \dots . Les variables aléatoires G_i sont indépendantes, et de même, les $V_{i,\ell}$ sont *i.i.d.* à ℓ fixé. Voir la FIG 1.1.

Un tel cadre de division aléatoire a été introduit par Devroye [26] qui a étudié le comportement asymptotique de la *profondeur* de l'arbre associé.

Exemples.

- Algorithme binaire : Quand $\mathbb{P}(G = 2) = 1$, $D = 2$ et $V_{1,2} \equiv V_{2,2} \equiv 1/2$, on retrouve l'un des premiers algorithmes de ce genre. Il a été étudié par Knuth en 1973.
- Algorithme en Arbre Symétrique : C'est le cas quand $V_{i,n} \equiv 1/n$ pour tout $n \geq 2$ et $1 \leq i \leq n$.
- L'Algorithme Q -aire : Si $\mathbb{P}(G = Q) = 1$ où $Q \geq 2$ et $D = 2$, on retrouve l'algorithme de résolution Q -aire avec arrivées bloquées, algorithme analysé par Mathys et Flajolet [77].

Voir aussi Devroye [26] pour d'autres exemples. Naturellement, un tel algorithme peut être graphiquement représenté par un arbre comme l'illustre la Figure 1.1.

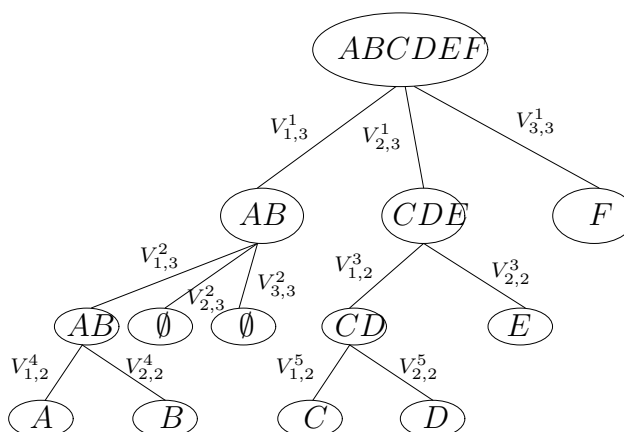


FIG. 1.1 – Algorithme en arbre avec $D = 2$, deux vecteurs de poids aléatoires $(V_{1,2}, V_{2,2})$ et $(V_{1,3}, V_{2,3}, V_{3,3})$, G une variable aléatoire à valeurs dans $\{2, 3\}$, le groupe initial étant $\{A, B, C, D, E, F\}$.

Mesure de division

Comme il sera montré dans ce qui suit, la notion clé pour l'étude de cet algorithme en arbre est la distribution de probabilité \mathcal{W} sur $[0, 1]$ définie ci-dessous et liée à la distribution de branchement (la variable G) et aux poids sur les différents arcs (le vecteur $(V_{1,G}, \dots, V_{G,G})$). Le comportement asymptotique de l'algorithme s'exprime naturellement en fonction de cette distribution \mathcal{W} .

Définition 1. La mesure de division est la distribution de probabilité \mathcal{W} sur $[0, 1]$ définie, pour une fonction borélienne positive ou nulle f , par

$$\int f(x) \mathcal{W}(dx) = \mathbb{E} \left(\sum_{i=1}^G V_{i,G} f(V_{i,G}) \right) = \sum_{\ell=2}^{+\infty} \sum_{i=1}^{\ell} \mathbb{P}(G = \ell) \mathbb{E}(V_{i,\ell} f(V_{i,\ell})). \quad (1.1)$$

Tout au long de ce chapitre, on suppose que $G \geq 2$ presque sûrement et qu'il existe un réel $\delta > 0$ tel que la relation

$$\sup_{\ell \geq 2} \sup_{1 \leq i \leq \ell} V_{i,\ell} \leq \delta < 1 \quad (A)$$

est vraie presque sûrement, en particulier $\mathcal{W}([0, \delta]) = 1$. Ces conditions impliquent en particulier la non-dégénérescence du mécanisme de division aléatoire.

Définition 2. Une mesure \mathcal{W} est exponentiellement arithmétique s'il existe un réel $\rho > 0$ tel que

$$\mathcal{W}(\{e^{-n\rho} : n \geq 1\}) = 1,$$

le plus grand ρ satisfaisant cette relation définit le pas exponentiel de \mathcal{W} .

Si A est une variable aléatoire de distribution \mathcal{W} , alors \mathcal{W} est exponentiellement arithmétique de pas exponentiel ρ si et seulement si la distribution de $-\log(A)$ est arithmétique de pas ρ . Voir Section A.2.

Exemples.

— Algorithme binaire : $\mathbb{P}(G = 2) = 1$, $D = 2$ et $V_{1,2} \equiv V_{2,2} \equiv 1/2$.

Dans ce cas

$$\mathcal{W}(dx) = \delta_{1/2}$$

où δ_x est la distribution de Dirac en x ; \mathcal{W} est exponentiellement arithmétique de pas exponentiel $\log 2$.

— Algorithme en Arbre Symétrique :

$$\mathcal{W}(dx) = \sum_{n \geq 2} P(G = n) \delta_{1/n},$$

le pas exponentiel est $\log K$ où K est le plus grand entier p tel que le support de la variable aléatoire G est contenu dans $\{p^n, n \in \mathbb{N}\}$.

— Algorithme Q -aire : $\mathbb{P}(G = Q) = 1$, $D = 2$, $V_{1,Q} = p_1, \dots, V_{Q,Q} = p_Q$.

$$\mathcal{W}(dx) = p_1 \delta_{p_1} + p_2 \delta_{p_2} + \dots + p_Q \delta_{p_Q},$$

la distribution \mathcal{W} est exponentiellement arithmétique si et seulement si tous les réels $\log p_i / \log p_j$, $1 \leq i < j \leq Q$, sont rationnels.

Coût de l'Algorithme en Arbre

Pour un tel algorithme, une quantité importante est le nombre d'opérations nécessaires pour que l'algorithme arrive à terme, *i.e.* que tous les sous-groupes terminaux soient de cardinal inférieur strictement à D . Notons R_n cette quantité quand le groupe initial est de taille n , alors

- $R_n = 1$ quand $n < D$;
- Pour $n \geq D$,

$$R_n \stackrel{\text{dist.}}{=} 1 + R_{1,N_1^n} + \cdots + R_{G,N_G^n}, \quad (1.2)$$

où conditionnellement à l'événement $\{G = \ell\}$ et aux variables aléatoires $V_{1,\ell}, V_{2,\ell}, \dots, V_{\ell,\ell}$,

1. Le vecteur (N_1^n, \dots, N_ℓ^n) a une distribution multinomiale de paramètres n et $(V_{1,\ell}, V_{2,\ell}, \dots, V_{\ell,\ell})$;
2. Pour $(p_i) \in \mathbb{N}^\ell$, les variables $R_{1,p_1}, \dots, R_{\ell,p_\ell}$ sont indépendantes ;
3. Pour $1 \leq i \leq \ell$, la variable R_{i,p_i} a la même distribution que R_{p_i} .

La variable R_n est simplement le nombre de noeuds dans l'arbre associé, voir la Figure 1.1.

1.1.2 Étrange Loi des Grands Nombres

Notons que, comme le processus de division est aléatoire, la variable R_n est une *variable aléatoire*. Dans le cadre d'un réseau de communication, cette quantité correspond au temps total nécessaire pour transmettre n messages. Si $\mathbb{E}(R_n)$ est sa valeur moyenne, $\mathbb{E}(R_n)/n$ est le temps de transmission moyen d'un message parmi n . D'un point de vue probabiliste, il est naturel d'espérer que la suite (R_n) obéisse à une sorte de loi des grands nombres, *i.e.* que $(\mathbb{E}(R_n)/n)$ converge vers une certaine limite α . La constante α est, en un certain sens, la moyenne asymptotique du temps de transmission d'un message. Curieusement, cette loi des grands nombres n'est pas toujours vérifiée. Dans certains cas même, la suite $(\mathbb{E}(R_n)/n)$ ne converge pas du tout, voire même présente un comportement oscillatoire.

Quand le degré de division est constant égal à 2 et que $V_{1,2} \equiv V_{2,2} \equiv 1/2$ (les éléments sont équitablement divisés entre les deux sous-groupes), ce phénomène est bien connu. Il a été analysé en utilisant des techniques d'analyse complexe, des transformations fonctionnelles (et les inversions associées) par Knuth [59], Flajolet *et al.* [40], Louchard et Prodinger [69] et plusieurs autres. Voir Hofri [47], Mahmoud [73] et Flajolet and Sedgewick [99] pour un traitement complet de cette approche. Voir aussi Devroye [27] pour plus de détails sur le sujet. Robert [93] a proposé comme alternative une méthode élémentaire permettant d'obtenir le comportement asymptotique de certaines séries obéissant à un phénomène oscillatoire, et ce sans utiliser les techniques d'analyse complexe.

Quand le degré de division est constant et vaut Q mais que les éléments ne sont pas équitablement répartis entre les sous-groupes, les études sont alors plus rares. Utilisant des techniques d'analyse complexe, Fayolle *et al.* [33] obtiennent le comportement asymptotique de la suite associée $(\mathbb{E}(R_n))$. Mathys et Flajolet [77] donnent une généralisation de cette étude quand Q est arbitraire.

Quelques Approches Alternatives.

- Certaines lois des grands nombres ont été prouvées par Devroye [27] dans un cadre assez général pour différentes fonctionnelles de l'arbre associé. Les inégalités de concentration de Talagrand sont l'outil principal dans cette étude. En effet, ce là revient à prouver que la distribution de la variable aléatoire $R_n/\mathbb{E}(R_n)$ est concentrée autour de 1 avec une décroissance exponentielle. Les résultats sur les distributions limites du type théorème de la limite centrale ne semblent pas être accessibles par cette méthode.
- Clement *et al.* [20] ont analysé des algorithmes similaires dans le contexte plus général des systèmes dynamiques. Utilisant un cadre hilbertien, ils montrent que le comportement du premier ordre des ces algorithmes s'exprime en fonction du spectre d'un opérateur fonctionnel, appelé *opérateur de transfert*. Obtenir des résultats explicites par cette approche nécessite plus d'information sur certaines valeurs propres de cet opérateur de transfert.

Une version dynamique de cette classe d'algorithmes est étudiée dans Mohamed and Robert [82]. Le processus de division est le même, mais, en utilisant le langage des processus de branchement, une immigration a lieu à chaque feuille de l'arbre associé, *i.e.* de nouveaux éléments arrivent à chaque unité de temps. L'aspect dynamique du processus complique le problème. Dans ce cas, un outil probabiliste additionnel est utilisé : un processus auto-régressif dont la loi à l'équilibre joue un rôle important.

1.1.3 Résultats du chapitre

La Section 1.2 donne une représentation intéressante du coût moyen de l'algorithme. Le résultat le plus important de ce chapitre pour le comportement asymptotique de l'algorithme est le théorème suivant de la Section 1.3. Il regroupe les Propositions 3 et 4.

Théorème 1 (Comportement Asymptotique du Coût Moyen). *Pour un algorithme en arbre, sous la condition*

$$\int_0^1 \frac{|\log(y)|}{y} \mathcal{W}(dy) < +\infty, \quad (1.3)$$

- Si la mesure de décomposition \mathcal{W} n'est pas exponentiellement arithmétique, alors

$$\lim_{n \rightarrow +\infty} \frac{\mathbb{E}(R_n)}{n} = \frac{\mathbb{E}(G)}{(D-1) \int_0^1 |\log(y)| \mathcal{W}(dy)}. \quad (1.4)$$

- Si la mesure de décomposition \mathcal{W} est exponentiellement arithmétique de pas exponentiel $\rho > 0$, alors, quand n tend vers l'infini, l'équivalence

$$\frac{\mathbb{E}(R_n)}{n} \sim F\left(\frac{\log n}{\rho}\right) \quad (1.5)$$

a lieu, où F est une fonction périodique de période 1 définie par, pour $x \geq 0$,

$$F(x) = \frac{\mathbb{E}(G)}{\int_0^1 |\log(y)| \mathcal{W}(dy)} \frac{\rho}{1 - e^{-\rho}} \int_0^{+\infty} \exp\left(-\rho \left\{x - \frac{\log y}{\rho}\right\}\right) \frac{y^{D-2}}{(D-1)!} e^{-y} dy$$

et $\{z\} = z - \lfloor z \rfloor$ désigne la partie fractionnaire de $z \in \mathbb{R}$.

La condition (1.3) n'est pas réellement restrictive vu que la variable G est bornée en pratique. Ce théorème couvre et étend certains résultats dans ce domaine : ceux de Knuth [59] pour l'algorithme binaire, et ceux de Mathys et Flajolet [77] pour les algorithmes Q -aires avec arrivées bloquées, voir les Corollaires 2 et 3.

Quand l'algorithme présente un comportement asymptotique oscillatoire, la fonction périodique F s'exprime généralement en fonction de ses coefficients de *Fourier*. L'expression de F généralise la représentation de Robert [93] obtenue pour l'algorithme binaire.

La *distribution* de la suite (R_n) est étudiée dans la Section 1.4. Pour plus de simplicité, seul est considéré le cas où la variable G est constante et où les variables $V_{i,G}$ sont égales à $1/G$. Le but de cette section est de montrer que la distribution de la transformée de *Poisson* de la suite des coûts, et plus généralement, celle de la plupart des fonctionnelles de l'arbre associé, peut être exprimée simplement en fonction d'un processus de *Poisson* et de variables indépendantes uniformément distribuées sur $[0, 1]$. Ainsi, une loi des grands nombres est prouvée quand le groupe d'éléments initial est de taille aléatoire suivant une loi de *Poisson* (transformée de *Poisson*). De plus, le comportement oscillatoire asymptotique de l'algorithme est démontré comme conséquence d'une loi des grands nombres *standard*.

Par ailleurs, le théorème de la limite centrale est prouvé utilisant une méthode similaire pour ce même cas. C'est un résultat classique, voir Jaquet et Régnier [49] ainsi que Mahmoud [73], toujours prouvé par des méthodes d'analyse complexe ainsi que des estimations assez techniques. Dans ce chapitre, il est prouvé en utilisant le théorème de la limite centrale *standard* pour des variables aléatoires indépendantes. En outre, une nouvelle représentation de la variance asymptotique est obtenue.

1.2 Propriétés Générales

Pour un sommaire sur le processus ainsi que la transformée de *Poisson*, voir A.1. Rappelons que pour $x > 0$, $(\mathcal{N}([0, x]))$ désigne un processus de *Poisson* d'intensité 1 sur l'intervalle $[0, x]$ dont les points sont notés $(t_n)_{n \geq 1}$, $\mathcal{N}([0, x])$ étant indépendant de la suite des variables aléatoires $(R_n)_{n \in \mathbb{N}}$.

L'équation (1.2) ainsi que les conditions au bord pour la suite (R_n) donnent la relation suivante, pour $n \geq 0$,

$$R_n \stackrel{\text{dist.}}{=} 1 + R_{1, N_1^n} + \cdots + R_{G, N_G^n} - G \mathbf{1}_{\{n < D\}}.$$

ou encore,

$$R_n - 1 \stackrel{\text{dist.}}{=} \sum_{i=1}^G (R_{i, N_i^n} - 1) + G \mathbf{1}_{\{n \geq D\}} \quad (1.6)$$

Définition 3. La transformée de *Poisson* de la suite des coûts moyens $(r_n := \mathbb{E}(R_n))_{n \in \mathbb{N}}$ est définie, pour tout $x \geq 0$, par

$$\sum_{n \geq 0} r_n \frac{x^n}{n!} e^{-x} = \mathbb{E}(r_{\mathcal{N}([0, x])}) = \mathbb{E}(R_{\mathcal{N}([0, x])}). \quad (1.7)$$

Par ailleurs, la transformée de Poisson de la suite des coûts $(R_n)_{n \in \mathbb{N}}$ est la variable aléatoire définie par $R_{\mathcal{N}([0,x])}$.

Remarque 1. Notons que, pour n fixé, R_n s'exprime linéairement en fonction de $(R_0, R_1, R_2, \dots, R_{n-1})$. Ainsi, le fait d'avoir des conditions aux bords finies $\mathbb{E}(R_0) = \mathbb{E}(R_1) = 1$ suffit pour assurer l'intégrabilité de la suite des coûts $(R_n)_{n \in \mathbb{N}}$. Un raisonnement par récurrence montre que le coût moyen $\mathbb{E}(R_n)$ est au plus polynomial en n , d'où l'existence de la transformée de Poisson de la suite des coûts moyens sur \mathbb{R}_+ .

La proposition suivante donne une représentation intéressante de la Transformée de Poisson de la suite $(\mathbb{E}(R_n))$.

Proposition 1 (Transformée de Poisson de la suite (R_n)). Pour $x > 0$,

$$\mathbb{E}(R_{\mathcal{N}([0,x])}) = 1 + \mathbb{E}(G) \mathbb{E} \left(\sum_{i=0}^{+\infty} \frac{1}{\prod_{k=1}^i W_k} \mathbf{1}_{\{t_D \leq x \prod_{k=1}^i W_k\}} \right). \quad (1.8)$$

où (W_i) est une suite i.i.d. de variables aléatoires de distribution \mathcal{W} .

Démonstration. Si n suit une loi de Poisson de paramètre x , la propriété de décomposition des variables de Poisson (voir Kingman [55] par exemple) montre que, conditionnellement à l'événement $\{G = \ell\}$ et aux variables $V_{1,\ell}, \dots, V_{\ell,\ell}$, les variables N_i^n , $1 \leq i \leq \ell$ sont indépendantes et N_i^n a une distribution de Poisson de paramètre $xV_{i,\ell}$. Par conséquent, pour $x > 0$, si

$$\Phi(x) \stackrel{\text{def}}{=} \frac{\mathbb{E}(R_{\mathcal{N}([0,x])}) - 1}{x\mathbb{E}(G)}, \quad (1.9)$$

il est facile à vérifier que $\mathbb{E}(G)\Phi(x) \rightarrow R_1 - R_0 = 0$ quand $x \searrow 0$.

Comme $\{\mathcal{N}([0,x]) \geq D\} = \{t_D \leq x\}$, l'équation (1.6) donne la relation

$$\Phi(x) = \sum_{\ell=2}^{+\infty} \mathbb{P}(G = \ell) \mathbb{E} \left(\sum_{i=1}^{\ell} V_{i,\ell} \Phi(xV_{i,\ell}) \right) + \frac{1}{x} \mathbb{P}(t_D \leq x). \quad (1.10)$$

L'équation (1.10) peut alors être réécrite sous la forme suivante, où la variable aléatoire W_1 a pour loi la mesure de division \mathcal{W} définie par (1.1),

$$\Phi(x) = \mathbb{E}(\Phi(xW_1)) + \mathbb{E} \left(\frac{1}{x} \mathbf{1}_{\{t_D \leq x\}} \right). \quad (1.11)$$

L'itération de l'équation (1.11) montre que, pour $n \geq 1$,

$$\Phi(x) = \mathbb{E} \left(\Phi \left(x \prod_{k=1}^n W_k \right) \right) + \mathbb{E} \left(\sum_{i=0}^{n-1} \frac{1}{x \prod_{k=1}^i W_k} \mathbf{1}_{\{t_D \leq x \prod_{k=1}^i W_k\}} \right).$$

Les hypothèses faites sur la variable G ainsi que sur la suite de vecteurs (V_n) impliquent que, presque sûrement, la suite $(\prod_{k=1}^n W_k)$ converge vers 0. La fonction Φ est alors donnée par

$$\Phi(x) = \mathbb{E} \left(\sum_{i=0}^{+\infty} \frac{1}{x \prod_{k=1}^i W_k} \mathbf{1}_{\{t_D \leq x \prod_{k=1}^i W_k\}} \right).$$

La proposition est ainsi démontrée. \square

Tout au long du chapitre, (W_i) désigne une suite *i.i.d.* de variables aléatoires sur $[0, 1]$ de distribution \mathcal{W} .

Proposition 2 (dé-Poissonnisation Probabiliste). *Pour $n \geq D$, on a*

$$\mathbb{E}(R_n) = 1 + \mathbb{E}(G) \mathbb{E} \left(\sum_{i=0}^{T(U_{D,n})-1} \frac{1}{\prod_{k=1}^i W_k} \right), \quad (1.12)$$

où, pour $0 < y < 1$,

$$T(y) = \inf \left\{ i \geq 1 : \prod_{k=1}^i W_k < y \right\}$$

et $U_{D,n}$ est la D -ième plus petite variable parmi n indépendantes, uniformément distribuées sur $[0, 1]$ et indépendantes de la suite (W_i) .

Démonstration. Pour $x > 0$, par décomposition par rapport au nombre de points du processus de Poisson $(\mathcal{N}([0, x]))$ dans l'intervalle $[0, x]$, on obtient, pour $0 < \alpha \leq 1$,

$$\begin{aligned} \mathbb{P}(t_D \leq x\alpha) &= \sum_{n=D}^{+\infty} \mathbb{P}(t_D \leq x\alpha, \mathcal{N}([0, x]) = n) \\ &= \sum_{n=D}^{+\infty} \mathbb{P}(t_D \leq x\alpha \mid \mathcal{N}([0, x]) = n) \mathbb{P}(\mathcal{N}([0, x]) = n). \end{aligned}$$

Pour $n \geq D$, conditionnellement à l'événement $\{\mathcal{N}([0, x]) = n\}$, la variable t_D a la même distribution que la D -ième plus petite variable aléatoire parmi n uniformément distribuées sur $[0, x]$. Quand $x = 1$, notons $U_{D,n}$ une variable aléatoire suivant cette distribution conditionnelle. Il est clair que, par homogénéité, la variable $(t_D \mid \mathcal{N}([0, x]) = n)$ a la même distribution que $xU_{D,n}$. Finalement, on obtient l'identité

$$\mathbb{P}(t_D \leq x\alpha) = \sum_{n=D}^{+\infty} \mathbb{P}(U_{D,n} \leq \alpha) \frac{x^n}{n!} e^{-x} = \mathbb{E} \left(\sum_{n=D}^{+\infty} \mathbf{1}_{\{U_{D,n} \leq \alpha\}} \frac{x^n}{n!} e^{-x} \right).$$

En utilisant l'indépendance des suites (W_i) et t_D dans l'équation (1.8), l'identité ci-dessus donne la relation suivante

$$\mathbb{E}(R_{\mathcal{N}([0, x])}) = 1 + \mathbb{E}(G) \mathbb{E} \left(\sum_{i=0}^{+\infty} \frac{1}{\prod_{k=1}^i W_k} \sum_{n=D}^{+\infty} \mathbf{1}_{\{U_{D,n} \leq \prod_{k=1}^i W_k\}} \frac{x^n}{n!} e^{-x} \right).$$

En utilisant le théorème de *Fubini* et en écrivant que $1 = \exp(x) \exp(-x)$, cette expression devient

$$\sum_{n=0}^{D-1} \frac{x^n}{n!} e^{-x} + \sum_{n=D}^{+\infty} \left(1 + \mathbb{E}(G) \mathbb{E} \left(\sum_{i=0}^{+\infty} \frac{1}{\prod_{k=1}^i W_k} \mathbf{1}_{\{U_{D,n} \leq \prod_{k=1}^i W_k\}} \right) \right) \frac{x^n}{n!} e^{-x}.$$

L'identification avec la représentation (A.1) de $\mathbb{E}(R_{\mathcal{N}([0, x])})$ conduit à la formule (1.12) \square

Corollaire 1 (Algorithme Symétrique de type Q -aire). *Quand $\mathbb{P}(G = Q) = 1$ et que $V_{i,Q} \equiv 1/Q$, pour $i = 1, \dots, Q$, alors pour $n \geq D$,*

$$\mathbb{E}(R_n) = 1 + \frac{Q}{Q-1} (\mathbb{E}(Q^{\lceil -\log_Q U_{D,n} \rceil}) - 1) \quad (1.13)$$

où, pour $0 \leq x \leq 1$,

$$\mathbb{P}(U_{D,n} > x) = \sum_{k=0}^{D-1} \binom{n}{k} x^k (1-x)^{n-k}.$$

Partant de l'équation (1.13), en utilisant le fait que $nU_{D,n}$ converge en distribution quand n tend vers l'infini, il n'est pas difficile d'obtenir le comportement asymptotique de $\mathbb{E}(R_n)$. Le cas général, équation (1.12), est légèrement plus compliqué. On doit étudier l'asymptotique des séries à l'intérieur de l'espérance.

1.2.1 Équation Intégrale Fonctionnelle

Si $R(x) = \mathbb{E}(R_{\mathcal{N}([0,x])})$ désigne la valeur moyenne de la transformée de *Poisson* de la suite (R_n) , alors l'équation (1.6) donne la relation suivante

$$R(x) = 1 + \sum_{\ell=2}^{+\infty} \mathbb{P}(G = \ell) \mathbb{E} \left(\sum_{i=1}^{\ell} R(xV_{i,\ell}) \right) - \mathbb{E}(G) \mathbb{P}(t_D \geq x),$$

en notant

$$h(x) = 1 - \mathbb{E}(G) \int_x^{+\infty} \frac{u^{D-1}}{(D-1)!} du,$$

il est clair que l'identité précédente peut être réécrite sous la forme d'une équation intégrale comme suit

$$R(x) = \int_0^{+\infty} R(xu) \frac{\mathcal{W}(du)}{u} + h(x). \quad (1.14)$$

Rappelons que \mathcal{W} est une distribution de probabilité sur l'intervalle $[0, 1]$. Pour le protocole de type Q -aire considéré par Mathys et Flajolet [77], cette équation est

$$R(x) = \sum_{i=1}^Q R(xp_i) + h(x).$$

Elle est analysée en utilisant la *transformée de Mellin* $R^*(s)$ de $R(x)$ sur une certaine bande verticale \mathcal{S} de \mathbb{C} ,

$$R^*(s) = \int_0^{+\infty} R(u) u^{s-1} du, \quad s \in \mathcal{S}$$

qui est, dans ce cas, donnée par

$$R^*(s) = h^*(s) \left/ \left(1 - \sum_{i=1}^Q \frac{1}{p_i^s} \right) \right.$$

L'approche analytique consiste à analyser les pôles de $R^*(s)$ à droite de \mathcal{S} , ou encore les solutions avec une partie réelle positive de l'équation

$$p_1^{-s} + p_2^{-s} + \cdots + p_Q^{-s} = 1.$$

Alors, par inversion de la *transformée de Mellin* et utilisant des techniques d'analyse complexe, le comportement asymptotique de $(R(x))$ au voisinage de l'infini est décrit en fonction des ces pôles. L'étape finale, une inversion analytique de la *transformée de Poisson* ajoutée à des estimations techniques, établit une relation entre le comportement asymptotique de la fonction $x \rightarrow R(x)$ et celui de la suite (R_n) .

Dans le cas général considéré ici, l'équation (1.14) donne l'expression suivante pour la *transformée de Mellin* de $(R(x))$

$$R^*(s) = h^*(s) \left/ \left(1 - \int_0^{+\infty} \frac{1}{u^{s+1}} \mathcal{W}(du) \right) \right.$$

Une approche analogue à celle analytique consisterait en l'étude des racines $s \in \mathbb{C}$, $\Re(s) \geq 0$, de l'équation

$$\int_0^{+\infty} \frac{1}{u^{s+1}} \mathcal{W}(du) = 1, \tag{1.15}$$

et, si possible, en des inversions successives des transformées de *Mellin* et de *Poisson*.

Remarque 2. *Il n'est pas clair qu'une telle approche puisse être utilisée dans notre cas afin de résoudre complètement le problème du comportement asymptotique de la suite (R_n) , i.e. obtenir des expressions explicites du coût moyen de l'algorithme.*

Comme il sera démontré dans la suite, notre approche directe réduit au minimum les outils techniques nécessaires pour une telle analyse. La transformée de *Poisson* de (R_n) est aussi utilisée dans cette nouvelle approche, la différence est qu'elle est convenablement représentée, voir l'équation (1.8), de telle sorte que la transformation inverse devient immédiate donnant ainsi une expression explicite (1.12) de $\mathbb{E}(R_n)$, expression dont l'analyse du comportement asymptotique de $(\mathbb{E}(R_n))$ est immédiate.

Il est intéressant de noter que, si (L_n) désigne la suite décroissante, au sens large, des longueurs des sous-intervalles de $[0, 1]$ associés à la procédure de décomposition (voir la Section 1.4), la *fonction Zeta* de la suite (L_n) est définie par la fonction méromorphe

$$\zeta(s) = \sum_{n \geq 1} L_n^s, \quad s \in \mathbb{C},$$

Voir Hambly et Lapidus [46] ainsi que Lapidus et van Frankenhuysen [62]. Il n'est pas difficile de voir que la relation

$$\mathbb{E}(\zeta(s)) = \int_0^{+\infty} u^s \mathcal{W}(du) \left/ \left(1 - \int_0^{+\infty} u^s \mathcal{W}(du) \right) \right.$$

est vraie. En particulier, les pôles de la fonction Zeta associée à la suite aléatoire récursive peuvent être exprimés en fonction des solutions de l'équation (1.15).

1.3 Analyse du Comportement Asymptotique du Coût Moyen

Une Marche Aléatoire Associée.

Si (W_i) est une suite de variables aléatoires *i.i.d.* de distribution \mathcal{W} définie par l'équation (1.1), la suite $(B_i) = (-\log(W_i))$ est une suite *i.i.d.* de variables aléatoires positives ou nulles. La marche aléatoire (S_n) associée à (B_i) est donnée par

$$S_n = B_1 + B_2 + \cdots + B_n, \quad n \geq 0.$$

Comme il sera mis en évidence dans la suite, le comportement asymptotique de l'algorithme de décomposition dépend essentiellement de la distribution de (B_i) . Pour $x > 0$, le *temps de dépassement* ν_x du niveau x par (S_n) est défini par

$$\nu_x = \inf\{n : S_n > x\}.$$

Pour $0 < y < 1$, la variable $T(y)$ de la Proposition 2 est simplement $\nu_{-\log(y)}$. La Section A.2 rappelle les résultats importants concernant la Théorie du Renouvellement utilisés ici.

Si Ψ est définie par

$$\Psi(x) = \mathbb{E} \left(\sum_{i=0}^{\nu_x-1} \exp \left(\sum_{k=1}^i B_k \right) \right), \quad x > 0$$

alors, d'après l'équation (1.12),

$$\mathbb{E}(R_n) = 1 + \mathbb{E}(G)\mathbb{E}[\Psi(-\log(U_{D,n}))]. \quad (1.16)$$

Il est clair que $-\log(U_{D,n})$ converge en distribution vers $+\infty$ quand n tend vers l'infini. On commence alors par analyser le comportement asymptotique de Ψ au voisinage de l'infini ; cette fonction est réécrite sous la forme suivante

$$\Psi(x)e^{-x} = \mathbb{E} \left(\sum_{i=0}^{\nu_x-1} e^{S_i-x} \right) = \mathbb{E} \left(\sum_{i=1}^{\nu_x} e^{S_{\nu_x-i}-x} \right). \quad (1.17)$$

1.3.1 Le Cas Non Arithmétique

Dans cette partie, on suppose que la distribution de W_1 n'est pas exponentiellement arithmétique. Voir la Définition 2.

Lemme 1. *Sous la condition*

$$\mathbb{E} \left(\frac{|\log(W_1)|}{W_1} \right) = \int_0^1 \frac{|\log(x)|}{x} \mathcal{W}(dx) < +\infty,$$

la relation

$$\sup_{x \geq 0} \mathbb{E} (e^{S_{\nu_x-x}}) < +\infty$$

est vraie.

Démonstration. Les inégalités de Lorden, voir Lorden [67] et Chang [18], montrent que, pour tout $p \geq 0$,

$$\sup_{x \geq 0} \mathbb{E}((S_{\nu_x} - x)^p) \leq \frac{p+2}{(p+1)\mathbb{E}(B_1)} \mathbb{E}(B_1^{p+1}).$$

Par suite, on obtient la relation

$$\begin{aligned} \sup_{x \geq 0} \mathbb{E}(e^{S_{\nu_x} - x}) &\leq \frac{1}{\mathbb{E}(B_1)} \int_0^{+\infty} (u+2)e^u \mathbb{P}(B_1 \geq u) du \\ &= \mathbb{E}((B_1 + 1)e^{B_1}) - 1 = \mathbb{E}\left(\frac{-\log(W_1) + 1}{W_1}\right) - 1 < +\infty. \end{aligned}$$

□

Pour $i > 1$, le Théorème 8 en annexe A montre que, quand x tend vers l'infini, la variable $S_{\nu_x - i} - x$ converge en distribution vers $-(\tau^* + \tau_1 + \tau_2 + \dots + \tau_{i-1})$, où les variables (τ_n) sont *i.i.d.* de même distribution que B_1 et indépendantes de τ^* dont la distribution est donnée par

$$\mathbb{E}(f(\tau^*)) = \frac{1}{\mathbb{E}(B_1)} \int_0^{+\infty} f(u) \mathbb{P}(B_1 \geq u) du,$$

pour toute fonction borélienne positive sur \mathbb{R} . Suite à l'hypothèse (A) en 1.1.1, les incréments de la marche aléatoire (S_n) sont majorés par $-\log(\delta)$, et donc on obtient la relation suivante, pour tout $1 < K \leq \nu_x$,

$$\sum_{i=K}^{\nu_x} e^{S_{\nu_x - i} - x} = e^{S_{\nu_x} - x} \sum_{i=K}^{\nu_x} e^{S_{\nu_x - i} - S_{\nu_x}} \leq e^{S_{\nu_x} - x} \frac{\delta^K}{1 - \delta}.$$

Du Lemme 1 ainsi que de l'équation (1.17), on déduit alors que

$$\begin{aligned} \lim_{x \rightarrow +\infty} \Psi(x)e^{-x} &= \mathbb{E}\left(\sum_{i=1}^{+\infty} \exp(-\tau^* - \tau_1 - \tau_2 - \dots - \tau_{i-1})\right) \\ &= \frac{1 - \mathbb{E}(\exp(-\tau_1))}{\mathbb{E}(\tau_1)} \times \frac{1}{1 - \mathbb{E}(\exp(-\tau_1))} = \frac{1}{-\mathbb{E}(\log(W_1))}, \end{aligned} \quad (1.18)$$

où, d'après l'équation (A.2), la distribution de τ^* sur \mathbb{R}_+ est donnée par

$$\mathbb{P}(\tau^* > x) = \mathbb{P}(\tau_1 \geq x) / \mathbb{E}(\tau_1), \quad x \geq 0.$$

Proposition 3 (Convergence des Moyennes). *Si la distribution de W_1 n'est pas exponentiellement arithmétique et vérifie*

$$\mathbb{E}\left(\frac{|\log(W_1)|}{W_1}\right) < +\infty,$$

alors, on a

$$\lim_{n \rightarrow +\infty} \frac{\mathbb{E}(R_n)}{n} = \frac{\mathbb{E}(G)}{(D-1)\mathbb{E}(-\log W_1)}.$$

Démonstration. L'équation (1.16) implique que, pour $n \geq 1$,

$$\frac{\mathbb{E}(R_n)}{n} = \frac{1}{n} + \mathbb{E}(G)\mathbb{E}\left(\Psi[-\log(U_{D,n})] \exp(\log(U_{D,n})) \frac{1}{nU_{D,n}}\right).$$

Quand n tend vers l'infini, la variable $nU_{D,n}$ converge en distribution vers une variable aléatoire t_D , somme de D variables exponentielles *i.i.d.* de paramètre 1. De plus,

$$\lim_{n \rightarrow +\infty} \mathbb{E}\left(\frac{1}{nU_{D,n}}\right) = \mathbb{E}(1/t_D) = \frac{1}{D-1}.$$

Pour $\varepsilon > 0$, il existe K tel que, pour $x > K$, $|\Psi(x) \exp(-x) + 1/\mathbb{E}(\log W_1)| < \varepsilon$. Alors, si C désigne le *sup* de $x \rightarrow \Psi(x) \exp(-x)$ sur \mathbb{R}_+ , on a

$$\begin{aligned} & \left| \mathbb{E}\left(\Psi[-\log(U_{D,n})] \exp(\log(U_{D,n})) \frac{1}{nU_{D,n}}\right) - \frac{1}{(D-1)\mathbb{E}(-\log W_1)} \right| \\ & \leq \varepsilon \mathbb{E}\left(\frac{1}{nU_{D,n}}\right) + \left(C + \frac{1}{\mathbb{E}(-\log W_1)}\right) \mathbb{E}\left(\mathbf{1}_{\{U_{D,n} > \exp(-K)\}} \frac{1}{nU_{D,n}}\right) \\ & \quad + \frac{1}{\mathbb{E}(-\log W_1)} \left| \mathbb{E}\left(\frac{1}{nU_{D,n}}\right) - \frac{1}{D-1} \right|. \end{aligned} \quad (1.19)$$

Pour $K_2 > 0$,

$$\begin{aligned} & \limsup_{n \rightarrow +\infty} \mathbb{E}\left(\mathbf{1}_{\{U_{D,n} > \exp(-K)\}} \frac{1}{nU_{D,n}}\right) \\ & \leq \limsup_{n \rightarrow +\infty} \mathbb{E}\left(\mathbf{1}_{\{nU_{D,n} > K_2 \exp(-K)\}} \frac{1}{nU_{D,n}}\right) = \mathbb{E}\left(\mathbf{1}_{\{t_D > K_2 \exp(-K)\}} \frac{1}{t_D}\right), \end{aligned}$$

et ce terme tend vers 0 quand K_2 tend vers l'infini. On conclut que le terme de droite dans la Relation (1.19) est arbitrairement petit quand n tend vers l'infini. La proposition est prouvée. \square

Corollaire 2 (Algorithme Q -aire avec arrivées bloquées.). *Quand $D = 2$, $G \equiv Q$ et $V_{i,Q} = p_i$ pour $1 \leq i \leq Q$ alors, si au moins l'un des nombres réels $\log p_i / \log p_1$, $2 \leq i \leq Q$ est irrationnel, on a la convergence suivante*

$$\lim_{n \rightarrow +\infty} \frac{\mathbb{E}(R_n)}{n} = \frac{Q}{\sum_{i=1}^Q -p_i \log p_i}.$$

1.3.2 Le Cas Arithmétique

On suppose que la distribution de W_1 est exponentiellement arithmétique de pas exponentiel $\rho > 0$. La loi de $-\log(W_1)/\rho$ est une distribution de probabilité sur \mathbb{N} . Pour $i \geq 1$, on définit $C_i = B_i/\rho = -\log(W_i)/\rho$. Dans le cas arithmétique, la marche aléatoire à valeurs entières associée à (C_i) joue un rôle clef, similaire à celui de (S_n) dans le cas non arithmétique. En notant

$$\tau_n = \inf \left\{ k \geq 1 : \sum_{i=1}^k C_i \geq n \right\},$$

l'équation (1.17) devient, pour $x \geq 0$,

$$\Psi(x)e^{-\rho[x/\rho]} = \mathbb{E} \left[\sum_{i=1}^{\tau_{[x/\rho]}} \exp \left(\rho \left(\sum_{k=1}^{\tau_{[x/\rho]}-i} C_k - [x/\rho] \right) \right) \right],$$

où $[y] = \inf\{n \in \mathbb{N} : n > y\}$ pour $y \geq 0$. En utilisant le Théorème 9 de l'annexe A ainsi que ses notations, pour $i \geq 1$, quand n tend vers l'infini, la variable $C_1 + \dots + C_{\tau_n-i} - n$ converge en distribution vers $-(C_1^* + C_2 + \dots + C_i)$. Avec la même méthode que pour le cas non arithmétique, si la variable $|\log(W_1)|/W_1$ est intégrable, alors

$$\lim_{x \rightarrow +\infty} \Psi(x)e^{-\rho[x/\rho]} = \frac{1}{\mathbb{E}(|\log(W_1)|)} \frac{\rho e^{-\rho}}{1 - e^{-\rho}}.$$

Proposition 4 (Oscillations Périodiques Asymptotiques). *Si la distribution de W_1 est exponentiellement arithmétique avec un pas exponentiel $\rho > 0$, et telle que*

$$\mathbb{E} \left(\frac{|\log(W_1)|}{W_1} \right) < +\infty,$$

alors, quand n tend vers l'infini, l'équivalence

$$\frac{\mathbb{E}(R_n)}{n} \sim F \left(\frac{\log n}{\rho} \right)$$

a lieu, où F est une fonction périodique de période 1 définie par, pour $x \geq 0$,

$$F(x) = \frac{\mathbb{E}(G)}{\mathbb{E}(|\log(W_1)|)} \frac{\rho}{1 - e^{-\rho}} \int_0^{+\infty} \exp \left(-\rho \left\{ x - \frac{\log y}{\rho} \right\} \right) \frac{y^{D-2}}{(D-1)!} e^{-y} dy$$

et $\{x\} = x - [x]$.

Démonstration. Pour $n \geq 1$, si $[x] = [x] + 1$,

$$\begin{aligned} & \frac{1}{n} \mathbb{E} [\Psi(-\log(U_{D,n}))] \\ &= \mathbb{E} \left[\Psi(-\log U_{D,n}) e^{-\rho[-\log(U_{D,n}/\rho)]} e^{\rho} \exp \left(-\rho \left\{ \frac{-\log(U_{D,n})}{\rho} \right\} \right) \frac{1}{nU_{D,n}} \right], \end{aligned}$$

comme $nU_{D,n}$ converge en distribution vers t_D quand n tend vers l'infini, avec la même méthode que pour la démonstration de la Proposition 3, on obtient les équivalences suivantes

$$\begin{aligned} & \frac{1}{n} \mathbb{E} [\Psi(-\log(U_{D,n}))] \times \mathbb{E}(|\log(W_1)|) \frac{1 - e^{-\rho}}{\rho} \\ & \sim E \left[\exp \left(-\rho \left\{ \frac{\log(n)}{\rho} - \frac{\log(nU_{D,n})}{\rho} \right\} \right) \frac{1}{nU_{D,n}} \right] \\ & = E \left[\exp \left(-\rho \left\{ \frac{\log(n)}{\rho} - \frac{\log t_D}{\rho} \right\} \right) \frac{1}{t_D} \right]. \end{aligned}$$

On conclut utilisant l'équation (1.16). □

Corollaire 3 (Algorithme Q -aire avec arrivées bloquées). *Quand $D = 2$, $G \equiv Q$ et $V_{i,Q} = p_i$ pour $1 \leq i \leq Q$ alors, si tous les réels $\log p_i / \log p_1$, $2 \leq i \leq Q$, sont rationnels, l'équivalence*

$$\frac{\mathbb{E}(R_n)}{n} \sim F\left(\frac{\log n}{\rho}\right)$$

a lieu, où F est une fonction périodique de période 1 définie par, pour $x \geq 0$,

$$F(x) = \frac{Q}{-\sum_{i=1}^Q p_i \log p_i} \frac{\rho}{1 - e^{-\rho}} \int_0^{+\infty} \exp\left(-\rho \left\{x - \frac{\log y}{\rho}\right\}\right) e^{-y} dy,$$

où $\{x\} = x - \lfloor x \rfloor$ et $\rho = \sup\{y > 0 : \forall i \in \{1, \dots, Q\}, \log p_i \in y\mathbb{Z}\}$.

Remarque 3. *Le cas arithmétique le plus connu et étudié est naturellement celui de l'arbre binaire $Q \equiv 2$, $D = 2$ et $\mathcal{W} = \delta_{1/2}$. Dans ce cas, $\mathbb{E}(R_n)/n \sim F(\log_2 n)$ où la fonction F est donnée par*

$$F(x) = 4 \int_0^{+\infty} \left(\frac{1}{2}\right)^{\{x - \log_2 y\}} e^{-y} dy.$$

La Figure 1.2 représente ces oscillations.

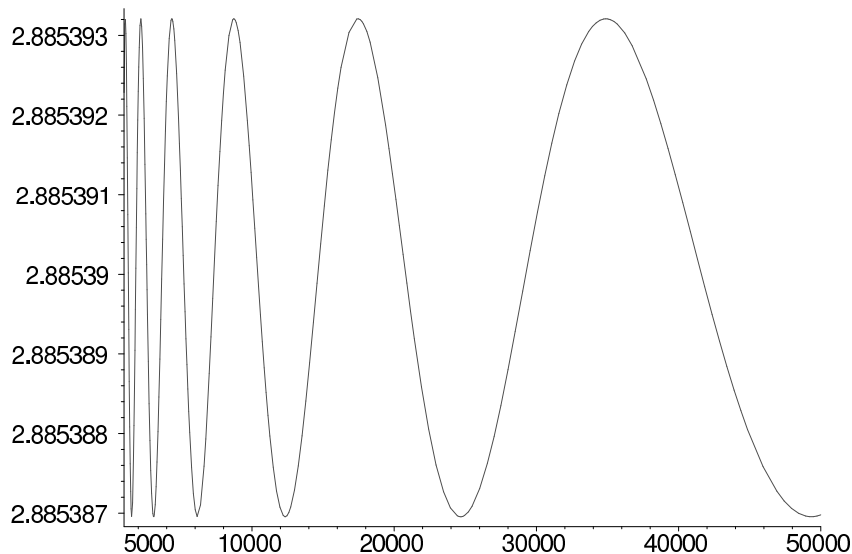


FIG. 1.2 – Algorithme binaire : oscillations de la suite $n \mapsto \mathbb{E}(R_n)/n$.

1.4 Distribution de l'Algorithme Q -aire Symétrique

Dans cette partie, on suppose que le degré de branchement de l'algorithme de décomposition est constant, *i.e.* $\mathbb{P}(G = Q) = 1$, et uniforme, $V_{i,Q} \equiv 1/Q$ pour $1 \leq i \leq Q$. Un groupe de $n \geq D$ éléments est aléatoirement, équitablement divisé en Q sous-groupes. D'après la Proposition 4, on a

$$\mathbb{E}(R_n)/n \sim F_1(\log_Q n)$$

quand n tend vers l'infini, avec

$$F_1(x) = \frac{Q^2}{Q-1} \int_0^{+\infty} Q^{-\{x - \log_Q y\}} \frac{y^{D-2}}{(D-1)!} e^{-y} dy. \quad (1.20)$$

Ceci est typiquement un cas où une loi des grands nombres classique n'est pas valable.

Le but de cette section est d'aller encore plus loin dans l'étude de cette convergence. On étudie la distribution de la transformée de *Poisson* de la suite (R_n) , *i.e.* la variable aléatoire $R_{\mathcal{N}([0,x])}$, et non seulement sa moyenne comme dans les sections précédentes. En particulier, il est démontré que, pour la transformée de *Poisson*, une loi des grands nombres *standard* peut être utilisée pour prouver le comportement oscillatoire de l'algorithme. En d'autres termes, cette loi des grands nombres non usuelle est en fin de compte établie à partir d'une loi des grands nombres classique.

Notations

Dans tout ce qui suit, on suppose que

1. \mathcal{N} est un processus de *Poisson* d'intensité 1 sur \mathbb{R}_+ . Un autre processus de *Poisson* sera utilisé, défini sur le plan $[0, 1] \times \mathbb{R}_+$.
2. La variable \mathcal{M} désigne un processus de *Poisson* sur $[0, 1] \times \mathbb{R}_+$ d'intensité 1, c'est une distribution de points aléatoires sur $[0, 1] \times \mathbb{R}_+$ avec la propriété suivante : si $\mathcal{M}(H)$ désigne le nombre de points qui "tombent" dans l'ensemble $H \subset [0, 1] \times \mathbb{R}_+$,
 - Pour $x \in [0, 1] \times \mathbb{R}_+$, $\mathcal{M}(\{x\}) \in \{0, 1\}$
 - Si G et H sont deux ensembles disjoints de $[0, 1] \times \mathbb{R}_+$, les variables $\mathcal{M}(G)$ et $\mathcal{M}(H)$ sont indépendantes.
 - La distribution de la variable $\mathcal{M}([a, b] \times [y, z])$ est une loi de *Poisson* de paramètre $(b-a)(z-y)$ pour $0 \leq a \leq b \leq 1$ et $0 \leq y \leq z$.
 Notons que les variables aléatoires $\mathcal{N}([0, x])$ et $\mathcal{M}([0, 1] \times [0, x])$ ont une distribution de *Poisson* de paramètre x .
3. La transformée de *Poisson* de la suite (R_n) est une variable aléatoire notée $\mathcal{R}(x)$, $x \geq 0$, telle que

$$\mathcal{R}(x) \stackrel{\text{dist.}}{=} R_{\mathcal{N}([0,x])} \stackrel{\text{dist.}}{=} R_{\mathcal{M}([0,1] \times [0,x])}.$$

Sa moyenne est donnée par l'équation (1.8). Cette section est dédiée à l'étude du comportement asymptotique de la *distribution* de $\mathcal{R}(x)$.

1.4.1 Loi des Grands Nombres

Dans cette section, il est prouvé que la transformée de *Poisson* de la suite (R_n) satisfait une loi forte des grands nombres. Une bonne représentation de cette transformée comme fonctionnelle du processus de *Poisson* est tout d'abord prouvée dans la proposition suivante.

Proposition 5. *La distribution de la transformée de Poisson $\mathcal{R}(x)$ de la suite (R_n) satisfait les relations suivantes,*

$$\mathcal{R}(x) \stackrel{\text{dist.}}{=} \mathcal{R}_1(x) \stackrel{\text{def.}}{=} 1 + Q \sum_{p \geq 0} \sum_{k=0}^{Q^p-1} \phi_{\mathcal{N}} \left(xk/Q^p, x(k+1)/Q^p \right), \quad (1.21)$$

où, pour $0 \leq a \leq b$, $\phi_{\mathcal{N}}(a, b) = 1$ si $\mathcal{N}([a, b]) \geq D$ et 0 sinon,

$$\mathcal{R}(x) \stackrel{\text{dist.}}{=} \mathcal{R}_2(x) \stackrel{\text{def.}}{=} 1 + Q \sum_{p \geq 0} \sum_{k=0}^{Q^p-1} \mathbf{1}_{\{\mathcal{M}([k/Q^p, (k+1)/Q^p] \times [0, x]) \geq D\}}. \quad (1.22)$$

Notons que la fonction $x \rightarrow \mathcal{R}_2(x)$ est clairement croissante. En particulier, si f est une fonction décroissante donnée sur \mathbb{R}_+ , la même propriété est vraie pour

$$x \rightarrow \mathbb{E}[f(\mathcal{R}(x))].$$

La représentation (1.21) sera pratique pour obtenir une loi forte des grands nombres sur les sous-suites, tandis que la représentation (1.22) sera utilisée pour obtenir la convergence en distribution de $\mathcal{R}(x)/x$ quand x tend vers l'infini.

Démonstration. En utilisant la propriété de décomposition des variables de *Poisson*, la relation de récurrence (1.2) écrite sur la suite (R_n) peut être exprimée par

$$R_{\mathcal{N}([0, x])} \stackrel{\text{dist.}}{=} 1 + \sum_{i=1}^Q R_{i, \mathcal{N}([x(i-1)/Q, xi/Q])} - Q \mathbf{1}_{\{\mathcal{N}([0, x]) < D\}},$$

pour $x \geq 0$. Si, pour $0 \leq a < b$,

$$\Phi(a, b) = \frac{1}{Q} (R_{\mathcal{N}([a, b])} - 1), \quad (1.23)$$

la dernière équation peut être réécrite sous la forme suivante

$$\Phi(0, x) \stackrel{\text{dist.}}{=} \sum_{i=1}^Q \Phi_i \left(\frac{i-1}{Q} x, \frac{i}{Q} x \right) + \phi_{\mathcal{N}}(0, x), \quad (1.24)$$

En itérant cette relation, on obtient que

$$\Phi(0, x) \stackrel{\text{dist.}}{=} \sum_{p \geq 0} \sum_{k=0}^{Q^p-1} \phi_{\mathcal{N}} \left(\frac{k}{Q^p} x, \frac{k+1}{Q^p} x \right), \quad (1.25)$$

La fonction $\Phi(0, x)$ est juste la somme des fonctions $\phi_{\mathcal{N}}$ sur les intervalles Q -adiques de $[0, x]$. L'équation (1.22) est prouvée de la même manière. \square

1.4.2 Représentation de quelques fonctionnelles de l'arbre associé

Quand $\mathcal{N}([0, x])$ éléments sont à la racine de l'arbre associé, le nombre total des noeuds de l'arbre $R_{\mathcal{N}([0, x])}$ n'est pas la seule quantité pouvant être représentée par l'équation (1.21) en termes du processus de *Poisson* \mathcal{N} .

La *hauteur*, ou encore *profondeur maximale* $M(x)$ de l'arbre associé quand il y a $\mathcal{N}([0, x])$ éléments à la racine peut être exprimée comme une fonctionnelle du processus de *Poisson*

$$M(x) = \max\{p \geq 1 : \exists k, 0 \leq k < Q^{p-1} - 1, \mathcal{N}(\lfloor k/Q^{p-1}, (k+1)/Q^{p-1} \rfloor) \geq D\}$$

La quantité

$$F(x) = \max\{p \geq 1 : \forall k, 0 \leq k < Q^{p-1} - 1, \mathcal{N}(\lfloor k/Q^{p-1}, (k+1)/Q^{p-1} \rfloor) \geq D\}$$

est le *nombre des niveaux complets* de l'arbre, appelée encore *niveau de saturation*. Voir Knessl et Szpankowski [58]. Remarquons que ces quantités sont directement reliées aux problèmes d'occupation classiques. Le *nombre des noeuds à un niveau* $p \geq 1$ est donné par

$$Q \sum_{k=0}^{Q^{p-1}-1} \mathbf{1}_{\{\mathcal{N}(\lfloor k/Q^{p-1}, (k+1)/Q^{p-1} \rfloor) \geq D\}}.$$

Ceci n'est pas une liste exhaustive des représentations possibles s'exprimant en termes du processus de *Poisson*.

Il est intéressant de faire le parallèle entre les deux représentations possibles des *algorithmes en arbre*; l'une en terme d'*arbre*, l'autre en terme des sous-intervalles Q -adiques de $[0, 1]$. Dans un cas plus général, *i.e.* quand le processus de division n'est pas symétrique, une représentation similaire à celle (1.21) peut être obtenue en utilisant la *décomposition aléatoire* de l'intervalle $[0, 1]$ au lieu de celle Q -adique. Voir Falconer [32].

Loi Forte des Grands Nombres.

L'équation (1.24) montre que, si $N > 0$, la quantité $\Phi(0, yQ^N)$ est la somme de Φ sur les intervalles $[yp, yp + y]$, $0 \leq p < Q^N$, et de $\phi_{\mathcal{N}}$ sur les intervalles $[ykQ^n, y(k+1)Q^n]$ contenus dans $[0, yQ^N]$,

$$\Phi(0, yQ^N) = \sum_{p=0}^{Q^N-1} \Phi(yp, yp + y) + \sum_{n=1}^N \sum_{k=0}^{Q^{N-n}-1} \phi_{\mathcal{N}}(ykQ^n, y(k+1)Q^n). \quad (1.26)$$

Grâce à la propriété d'indépendance du processus de *Poisson* sur des ensembles disjoints, la loi forte des grands nombres classique implique que, presque sûrement,

$$\begin{aligned} \lim_{N \rightarrow +\infty} \frac{1}{Q^N} \sum_{p=0}^{Q^N-1} \Phi(yp, yp + y) &= \mathbb{E}(\Phi(0, y)) = \sum_{p \geq 0} Q^p \mathbb{E}(\phi_{\mathcal{N}}(0, y/Q^p)) \\ &= \sum_{p \geq 0} Q^p \mathbb{P}(\mathcal{N}(\lfloor 0, y/Q^p \rfloor) \geq D). \end{aligned}$$

En utilisant l'équation (1.25), et pour $n > 0$,

$$\lim_{N \rightarrow +\infty} \frac{1}{Q^N} \sum_{k=0}^{Q^{N-n}-1} \phi_{\mathcal{N}}(ykQ^n, y(k+1)Q^n) = \frac{1}{Q^n} \mathbb{E}(\phi_{\mathcal{N}}(0, yQ^n)).$$

Notons que, pour $0 < K < N$,

$$\sum_{n=K}^N \frac{1}{Q^N} \sum_{k=0}^{Q^{N-n}-1} \phi_{\mathcal{N}}(ykQ^n, y(k+1)Q^n) \leq \sum_{n=K}^N \frac{1}{Q^N} Q^{N-n} \leq \frac{1}{Q^{K-1}}.$$

Les trois dernières identités ainsi que la décomposition (1.26) impliquent que, presque sûrement,

$$\lim_{N \rightarrow +\infty} \frac{1}{yQ^N} \Phi(0, yQ^N) = \sum_{n \in \mathbb{Z}} \frac{1}{yQ^n} \mathbb{P}(\mathcal{N}([0, yQ^n]) \geq D).$$

Proposition 6 (Loi Forte des Grands Nombres). *Avec les mêmes notations que la Proposition 5, pour $0 < y \leq Q$, presque sûrement,*

$$\begin{aligned} \lim_{N \rightarrow +\infty} \frac{\mathcal{R}_1(yQ^N)}{yQ^N} &= Q \sum_{n \in \mathbb{Z}} \frac{1}{yQ^n} \mathbb{P}(\mathcal{N}([0, yQ^n]) \geq D) \\ &= Q \sum_{n \in \mathbb{Z}} \frac{1}{yQ^n} \int_0^{yQ^n} \frac{u^{D-1}}{(D-1)!} e^{-u} du = F_1(\log_Q y), \end{aligned} \quad (1.27)$$

où F_1 est la fonction périodique définie par l'équation (1.20).

D'autre part, la proposition établit le résultat intuitif (et classique) que la suite $(\mathbb{E}(R_n)/n)$ et la fonction $x \rightarrow \mathbb{E}(R_{\mathcal{N}([0,x])})/x$ ont le même comportement asymptotique au voisinage de l'infini. Notons que si $G(y)$ est défini comme étant le second terme de l'équation (1.27), alors la fonction $x \rightarrow G(Q^x)$ est clairement périodique de période 1.

Démonstration. Seule la relation $F_1(\log_Q y) = G(y)$ est à prouver. Pour $n \in \mathbb{Z}$, si t_D est le D -ième point du processus de *Poisson* \mathcal{N} , alors

$$\mathbb{P}(\mathcal{N}([0, yQ^n]) \geq D) = \mathbb{P}(t_D \leq yQ^n) = \int_0^{+\infty} \mathbf{1}_{\{u \leq yQ^n\}} \frac{u^{D-1}}{(D-1)!} e^{-u} du.$$

En additionnant ces termes, et en utilisant le théorème de *Fubini*, on obtient que

$$\begin{aligned} G(y) &= Q \int_0^{+\infty} \sum_{n \in \mathbb{Z}} \frac{1}{yQ^n} \mathbf{1}_{\{u \leq yQ^n\}} \frac{u^{D-1}}{(D-1)!} e^{-u} du \\ &= \frac{Q^2}{Q-1} \int_0^{+\infty} \frac{1}{yQ^{\lceil \log_Q(u/y) \rceil}} \frac{u^{D-1}}{(D-1)!} e^{-u} du \\ &= \frac{Q^2}{Q-1} \int_0^{+\infty} \frac{1}{Q^{-\{\log_Q(u/y)\}}} \frac{u^{D-2}}{(D-1)!} e^{-u} du = F_1(\log_Q y). \end{aligned}$$

La proposition est prouvée. □

La proposition suivante établit une loi faible des grands nombres pour la *transformée de Poisson* de la suite (R_n) . Devroye [27] a obtenu des résultats similaires dans un cadre plus général en utilisant les inégalités de concentration de Talagrand.

Théorème 2 (Loi des Grands Nombres). *La convergence en distribution suivante est vraie, pour tout $\varepsilon > 0$,*

$$\lim_{x \rightarrow +\infty} \mathbb{P} \left(\left| \frac{\mathcal{R}(x)}{xF_1(\log_Q x)} - 1 \right| \geq \varepsilon \right) = 0,$$

où F_1 est la fonction définie par l'équation (1.20).

Démonstration. Pour $x > 0$, on définit $N_x = \lfloor \log_Q x \rfloor$, $u_x = x/Q^{N_x}$ et, pour $p \geq 1$, $z_x = \lfloor u_x p \rfloor Q^{N_x}/p$. Notons que $\sup_{x \geq 1} |x/z_x - 1|$ converge vers 0 quand p tend vers l'infini; alors, par continuité de F_1 ,

$$\lim_{p \rightarrow +\infty} \sup_{x \geq 1} \left| \frac{x F_1(\log_Q x)}{z_x F_1(\log_Q z_x)} - 1 \right| = 0.$$

La proposition 6 montre que pour $p \geq 1$ et k tel que $0 \leq k \leq p$, on a presque sûrement

$$\lim_{N \rightarrow +\infty} \frac{\mathcal{R}_1(y_k Q^N)}{y_k Q^N F_1(\log_Q y_k)} = 1,$$

où $y_k = k/p$. Par suite, si $p \geq 1$ est fixé, presque sûrement,

$$\lim_{x \rightarrow +\infty} \frac{\mathcal{R}_1(z_x)}{z_x F_1(\log_Q z_x)} = 1,$$

La monotonie de la fonction $x \rightarrow \mathcal{R}_2(x)$ donne la relation

$$\mathcal{R}_2 \left(\frac{\lfloor u_x p \rfloor}{p} Q^{N_x} \right) \leq \mathcal{R}_2(x),$$

finalement, on obtient que

$$\begin{aligned} \mathbb{P} \left(\frac{\mathcal{R}(x)}{xF_1(\log_Q x)} < 1 - \varepsilon \right) &= \mathbb{P} \left(\frac{\mathcal{R}_2(x)}{xF_1(\log_Q x)} < 1 - \varepsilon \right) \\ &\leq \mathbb{P} \left(\frac{\mathcal{R}_2(z_x)}{xF_1(\log_Q x)} < 1 - \varepsilon \right) = \mathbb{P} \left(\frac{\mathcal{R}_1(z_x)}{z_x F_1(\log_Q z_x)} \frac{z_x F_1(\log_Q z_x)}{xF_1(\log_Q x)} < 1 - \varepsilon \right), \end{aligned}$$

et donc

$$\lim_{x \rightarrow +\infty} \mathbb{P} \left(\frac{\mathcal{R}(x)}{xF_1(\log_Q x)} < 1 - \varepsilon \right) = 0.$$

L'inégalité analogue est obtenue de la même façon. Le théorème est ainsi prouvé. \square

1.4.3 Théorèmes de la Limite Centrale

Pour $N \geq 1$ et $0 < x \leq Q$, avec Φ définie par l'équation (1.23), la variance de la variable $\Phi(0, x)$ est d'abord analysée. L'expression (1.25) donne

$$[\Phi(0, x) - \mathbb{E}(\Phi(0, x))]^2 = \sum_{p \geq 0} \sum_{k=0}^{Q^p-1} \sum_{p' \geq 0} \sum_{k'=0}^{Q^{p'}-1} \Delta_{k,p}(x) \Delta_{k',p'}(x),$$

où

$$\Delta_{k,p}(x) = \phi_{\mathcal{N}} \left(\frac{k}{Q^p} x, \frac{k+1}{Q^p} x \right) - \mathbb{E} \left(\phi_{\mathcal{N}} \left(0, \frac{1}{Q^p} x \right) \right).$$

La moyenne de la variable $\Delta_{k,p}(x) \Delta_{k',p'}(x)$ est non nulle seulement si $p \leq p'$ et $kQ^{p'-p} \leq k' \leq (k+1)Q^{p'-p} - 1$ ou sous la condition symétrique obtenue en échangeant (p, k) et (p', k') .

$$\begin{aligned} \mathbb{E} \left[\left(\Phi(0, x) - \mathbb{E}(\Phi(0, x)) \right)^2 \right] &= \sum_{p \geq 0} \sum_{k=0}^{Q^p-1} \mathbb{E} [\Delta_{k,p}(x)^2] \\ &\quad + 2 \sum_{p \geq 0} \sum_{k=0}^{Q^p-1} \sum_{p' > p} \sum_{k'=kQ^{p'-p}}^{(k+1)Q^{p'-p}-1} \mathbb{E} [\Delta_{k,p}(x) \Delta_{k',p'}(x)]. \end{aligned}$$

En utilisant des identités élémentaires

$$\begin{aligned} \mathbb{E} [\Delta_{k,p}(x)^2] &= \mathbb{E} [\phi_{\mathcal{N}}(0, x/Q^p)] (1 - \mathbb{E} [\phi_{\mathcal{N}}(0, x/Q^p)]) \\ &= \mathbb{P}(t_D \leq x/Q^p) \mathbb{P}(s_D \geq x/Q^p), \\ \mathbb{E} [\Delta_{k,p}(x) \Delta_{k',p'}(x)] &= \mathbb{E} [\phi_{\mathcal{N}}(0, x/Q^{p'})] (1 - \mathbb{E} [\phi_{\mathcal{N}}(0, x/Q^p)]) \\ &= \mathbb{P}(t_D \leq x/Q^{p'}) \mathbb{P}(s_D \geq x/Q^p), \end{aligned}$$

où t_D et s_D sont deux variables aléatoires indépendantes de même distribution que le D -ième point du processus de *Poisson* \mathcal{N} , on obtient alors la relation

$$\begin{aligned} \mathbb{E} \left[\left(\Phi(0, x) - \mathbb{E}(\Phi(0, x)) \right)^2 \right] &= \sum_{p \geq 0} Q^p \mathbb{P}(t_D \leq x/Q^p \leq s_D) \\ &\quad + 2 \sum_{p' > p \geq 0} Q^{p'} \mathbb{P}(t_D \leq x/Q^{p'}, x/Q^p \leq s_D). \end{aligned}$$

En permutant somme et intégrale, on obtient finalement que

$$\begin{aligned} (Q-1) \mathbb{E} \left[\left(\Phi(0, x) - \mathbb{E}(\Phi(0, x)) \right)^2 \right] &= Q \mathbb{E} \left(\left(Q^{\lfloor \log_Q(x/t_D) \rfloor} - Q^{\lfloor \log_Q(x/s_D) \rfloor} \right) \mathbf{1}_{\{\lfloor \log_Q(x/t_D) \rfloor > \lfloor \log_Q(x/s_D) \rfloor\}} \right) \\ &\quad + 2Q \mathbb{E} \left(\left(\lfloor \log_Q(x/t_D) \rfloor - \lfloor \log_Q(x/s_D) \rfloor - 1 \right)^+ Q^{\lfloor \log_Q(x/t_D) \rfloor} \right) \\ &\quad - 2 \frac{Q}{Q-1} \mathbb{E} \left(\left(Q^{\lfloor \log_Q(x/t_D) \rfloor} - Q^{\lfloor \log_Q(x/s_D) \rfloor + 1} \right)^+ \right), \quad (1.28) \end{aligned}$$

où $a^+ = \max(a, 0)$ pour $a \in \mathbb{R}$. Cette identité conduit à la proposition suivante. Une proposition similaire a été prouvée par Jacquet et Régnier [49] ainsi que Régnier et Jacquet [97] dans le cas où $Q = D = 2$ sans condition de symétrie comme c'est le cas ici. Voir aussi Mahmoud [73] Chapitre 5.

Proposition 7 (Variance Asymptotique). *La variance de la transformée de Poisson de la suite (R_n) satisfait l'équivalence suivante, quand x vers l'infini,*

$$\frac{1}{x} \text{Var}(\mathcal{R}(x)) \sim F_2(\log_Q(x)),$$

où F_2 est la fonction périodique continue de période 1 définie par, pour $y \geq 0$,

$$F_2(y) = \int_{\mathbb{R}_+^2} f_2(\{y - \log_Q(u)\}, \{y - \log_Q(v)\}, u, v) \times \frac{u^{D-1}}{(D-1)!} \frac{v^{D-1}}{(D-1)!} e^{-(u+v)} dudv, \quad (1.29)$$

où $\{z\} = z - [z]$ pour $z \in \mathbb{R}$ et pour $u > 0$, $v > 0$ et $y \in \mathbb{R}$,

$$f_2(a, b, u, v) = \frac{Q}{Q-1} \left(\frac{Q^a}{u} - \frac{Q^b}{v} \right) \mathbf{1}_{\{\log_Q(v/u) + b > a\}} + \frac{2Q}{Q-1} (\log_Q(v/u) - a + b - 1)^+ - \frac{2Q}{(Q-1)^2} \left(\frac{Q^a}{u} - \frac{Q^{b+1}}{v} \right)^+$$

où $z^+ = \max(z, 0)$.

Notons qu'une expression encore plus détaillée de la variance peut être obtenue par la formule (1.28).

Proposition 8 (Théorème de la Limite Centrale pour la Transformée de Poisson). *Pour $0 < y < Q$, quand N tend vers l'infini, la variable*

$$\frac{1}{\sqrt{Q^N}} (\mathcal{R}(yQ^N) - \mathbb{E}(\mathcal{R}(yQ^N)))$$

converge en distribution vers une variable Gaussienne centrée de variance $yF_2(\log_Q y)$, où F_2 est définie par l'équation (1.29).

Démonstration. Il suffit de prouver la proposition pour la variable $\Phi(0, x)$ définie par

$$\Phi(0, x) = \frac{1}{Q} (R_{\mathcal{N}([0, x])} - 1),$$

L'équation (1.26) implique que, pour $K \geq 1$,

$$\begin{aligned} \Phi(0, yQ^N) - \mathbb{E}(\Phi(0, yQ^N)) &= \sum_{p=0}^{Q^N-1} [\Phi(yQ^N, yQ^N + p) - \mathbb{E}(\Phi(0, yQ^N))] \\ &+ \sum_{n=1}^K \sum_{k=0}^{Q^N-n-1} [\phi_{\mathcal{N}}(ykQ^n, y(k+1)Q^n) - \mathbb{E}(\phi_{\mathcal{N}}(0, yQ^n))] + \Delta_K. \end{aligned}$$

où Δ_K est le terme résiduel de la série. En utilisant la même méthode que pour le calcul de la variance, il n'est pas difficile d'établir que, pour tout $\varepsilon > 0$, il existe un certain $K > 0$ tel que la moyenne de $(\Delta_K/Q^N)^2$ soit inférieure à ε , pour N assez grand.

En regroupant les termes de l'équation précédente suivant les intervalles Q -adiques $[yk/Q^K, y(k+1)/Q^K]$ pour $0 \leq k < Q^{N-K}$ et en utilisant la propriété d'indépendance du processus de Poisson \mathcal{N} , la quantité $\Phi(0, yQ^N) - \mathbb{E}(\Phi(0, yQ^N)) - \Delta_K$ s'écrit sous la forme d'une somme de Q^{N-K} variables aléatoires *i.i.d.* Alors, le théorème central limite classique s'applique. La proposition est ainsi prouvée. \square

1.4.4 La Distribution de la suite (R_n)

La proposition suivante décrit la distribution de la variable R_n en fonction de n variables aléatoires uniformes *i.i.d.* sur l'intervalle $[0, 1]$. Cette caractérisation est en général implicitement utilisée afin d'établir différents équivalents asymptotiques de la profondeur de l'arbre associé, voir Mahmoud [73] et Pittel [89].

Proposition 9. *Pour $n \geq 0$, la variable aléatoire R_n a la même distribution que*

$$R_n \stackrel{\text{dist.}}{=} 1 + Q \sum_{p \geq 0} \sum_{k=0}^{Q^p-1} \phi_{\mathcal{U}_n}([k/Q^p, (k+1)/Q^p]), \quad (1.30)$$

où, pour $0 \leq a \leq b \leq 1$, $\phi_{\mathcal{U}_n}([a, b]) = 1$ si $\mathcal{U}_n([a, b]) \geq D$ et 0 sinon. La variable \mathcal{U}_n est la mesure de masse sur $[0, 1]$ définie par

$$\mathcal{U}_n = \delta_{U_1} + \delta_{U_2} + \dots + \delta_{U_n},$$

(U_1, \dots, U_n) sont des variables uniformes *i.i.d.* sur $[0, 1]$, en particulier $\mathcal{U}_n([a, b])$ est le nombre des U_i dans l'intervalle $]a, b]$.

Démonstration. Supposons que \mathcal{N} est un processus de Poisson de paramètre 1. Par définition

$$(R_{\mathcal{N}([0, x])} \mid \mathcal{N}([0, x]) = n) \stackrel{\text{dist.}}{=} R_n.$$

D'après la Proposition 5, la distribution de la transformée de Poisson $R_{\mathcal{N}([0, x])}$ s'exprime en fonction des points du processus de Poisson sur l'intervalle $[0, x]$. Mais, comme dans la démonstration de la Proposition 2, conditionnellement à l'événement $\{\mathcal{N}([0, x]) = n\}$, ces points ont la même distribution que les xU_i , $1 \leq i \leq n$, où (U_i) sont des uniformes *i.i.d.* sur l'intervalle $[0, 1]$. L'équation (1.30) est alors une conséquence directe de la Relation (1.21). \square

1.5 Algorithme en Arbre et Processus de Branchement

Cette section a été motivée par le travail de Bertoin et Martinez [7] sur les processus de fragmentation, avec ou sans perte, d'une masse donnée. Le parallèle entre ce type de processus et le mécanisme de décomposition dans un algorithme en arbre suggérait un

nouveau point de vue de ces algorithmes en tant que *processus de branchement*. Le travail de Nerman [83] pour le cas non-arithmétique ainsi que celui de Gatzouras [43] pour le cas arithmétique, tous les deux basés sur l'étude faite par Jagers [52] de tels *processus de branchement*, fournissent à la fois le cadre théorique et les principaux outils utilisés dans ce chapitre pour obtenir le comportement asymptotique d'un *algorithme en arbre* en fonction de celui d'un *processus de branchement*.

1.5.1 Description du modèle

Un *processus de branchement* de type *Crump-Mode-Jagers* est un arbre généalogique dans lequel on associe à l'ancêtre, noté \emptyset , le temps $t = 0$. Chaque individu u naît à un instant $\sigma_u > 0$, admet une ligne de vie infinie et se reproduit selon un processus ξ_u sur $[0, \infty[$, les $(\xi_u)_u$ étant *i.i.d.* de même loi qu'un processus noté ξ . On note $\xi_u(t) = \xi_u([0, t])$ le nombre de descendants de l'individu u qui sont nés avant la date t . On introduit aussi la fonction de reproduction

$$\mu(t) = \mathbb{E}[\xi(t)].$$

D'autre part, on considère un *algorithme en arbre* de degré fixé Q dont la condition d'arrêt est d'avoir un cardinal inférieur strictement à un entier fixé $D \geq 2$. Ainsi, l'algorithme est appliqué initialement à un groupe de n éléments qui sera divisé en Q sous-groupes. La probabilité qu'un objet soit mis dans le i -ème sous-groupe est notée p_i où $0 \leq i \leq Q-1$. On définit la distribution de probabilité \mathcal{W} sur $[0, 1]$ comme étant la *mesure de branchement* de l'algorithme

$$\mathcal{W} = \sum_{i=0}^{Q-1} p_i \delta_{p_i}.$$

On rappelle que le *coût* de l'algorithme est le nombre d'étapes nécessaires pour arriver à vérifier la condition d'arrêt sur chaque sous-groupe terminal généré par l'algorithme. On note R_n cette quantité quand le groupe initial est de taille n , alors

$$\begin{aligned} R_i &= 1, \text{ pour } 0 \leq i < D \\ R_n &\stackrel{\text{dist.}}{=} 1 + \sum_{i=0}^{Q-1} R_{i, S_i(n)}, \text{ pour } n \geq D, \end{aligned} \quad (1.31)$$

où,

- $(A_k)_{1 \leq k \leq n}$ étant une suite *i.i.d.* de variables aléatoires de loi \mathcal{W} , pour $0 \leq i \leq Q-1$

$$S_i(n) = \sum_{k=1}^n \mathbf{1}_{\{A_k = p_i\}},$$

- Pour $(m_i)_{0 \leq i \leq Q-1} \in \mathbb{N}^Q$, les variables $R_{0, m_0}, \dots, R_{Q-1, m_{Q-1}}$ sont indépendantes;
- Pour $0 \leq i \leq Q-1$, la variable R_{i, m_i} a la même distribution que R_{m_i} .

Alors, pour $n \geq 0$, l'équation de récurrence sur R_n peut être réécrite

$$R_n \stackrel{\text{dist.}}{=} 1 + \sum_{i=0}^{Q-1} R_{i, S_i(n)} - Q \mathbf{1}_{(n < D)}. \quad (1.32)$$

Rappelons que $(\mathcal{N}([0, x]))$ est un processus de *Poisson* d'intensité 1 sur l'intervalle $[0, x]$, (t_n) désigne la suite de ses points.

1.5.2 Approche

Afin d'étudier la variable aléatoire R_n , on utilise la décomposition en base Q pour définir récursivement une suite de sous-intervalles de $[0, 1]$. L'étude du modèle *poissonnien* revient alors à celle de la répartition de $(\mathcal{N}([0, x]))$ points sur une suite de sous-intervalles de $[0, 1]$.

Décomposition en base Q :

On définit récursivement la suite d'intervalles (I_k^n) , où $n \geq 0$ et $0 \leq k \leq Q^n - 1$, par

$$\begin{cases} I_0^0 &= [0, 1] \\ I_k^{n+1} &= (I_{k-1}^{n+1})_+ + p_{k-Q \lfloor k/Q \rfloor} I_{\lfloor k/Q \rfloor}^n, \end{cases}$$

où $(I)_+$ désigne l'extrémité droite de l'intervalle I . La suite ainsi obtenue est donnée par

$$([0, 1], [0, p_0], [p_0, p_1], \dots, [1 - p_{Q-1}, 1], [0, p_0^2] \dots).$$

Il est facile de voir que $(I_k^n)_+ = \sum_{i=0}^k |I_i^n|$, où $|I|$ est la longueur de l'intervalle I .

Soient $n \in \mathbb{N}$ et $0 \leq k < Q^{n+1} - 1$. On considère la décomposition en base Q de k à l'ordre n

$$k = a_0 + a_1 Q + \dots + a_n Q^n.$$

Pour $0 \leq i \leq n$, $a_i \in \{0, \dots, Q-1\}$. Alors, la longueur de l'intervalle I_k^{n+1} est donnée par

$$|I_k^{n+1}| = \prod_{i=0}^n p_{a_i}.$$

Si on considère un groupe initial de taille *poissonnienne* $\mathcal{N}([0, x])$, alors, par itération, l'équation (1.32) devient

$$R_{\mathcal{N}([0, x])} \stackrel{\text{dist.}}{=} 1 + Q \sum_{n \geq 0} \sum_{k=0}^{Q^n - 1} \mathbf{1}_{(\mathcal{N}_k([0, |I_k^n| x]) \geq D)},$$

où, pour un n fixé, $(\mathcal{N}_k)_{0 \leq k \leq Q^n - 1}$ désigne une suite *i.i.d.* de processus de *Poisson* indépendants. Notons que la série ci-dessus est presque sûrement finie, vu qu'à partir d'un certain niveau n , tous les sous-intervalles obtenus contiennent au plus $D - 1$ éléments parmi les $\mathcal{N}([0, x])$ initialement lancés sur $[0, x]$.

Processus de branchement associé :

On peut associer à l'algorithme ainsi décrit un *processus de branchement* dont le processus de naissance ξ (de même loi que ξ_\emptyset , processus de naissance de l'ancêtre) est défini par

$$\xi(t) \stackrel{\text{dist.}}{=} \xi_\emptyset(t) = \sum_{i=0}^{Q-1} \mathbf{1}_{(\sigma_i^1 < t)},$$

où $\sigma_k^n = -\log(|I_k^n|)$ pour tout couple (k, n) tel que $n \geq 0$ et $0 \leq k \leq Q^n - 1$. Ainsi, l'instant de naissance du k -ième individu de la n -ième génération s'exprime en fonction de la taille du k -ième sous-intervalle du n -ième niveau. D'où le parallèle entre *algorithme en arbre* et *arbre généalogique*, parallèle où la décomposition Q -adique joue la transition.

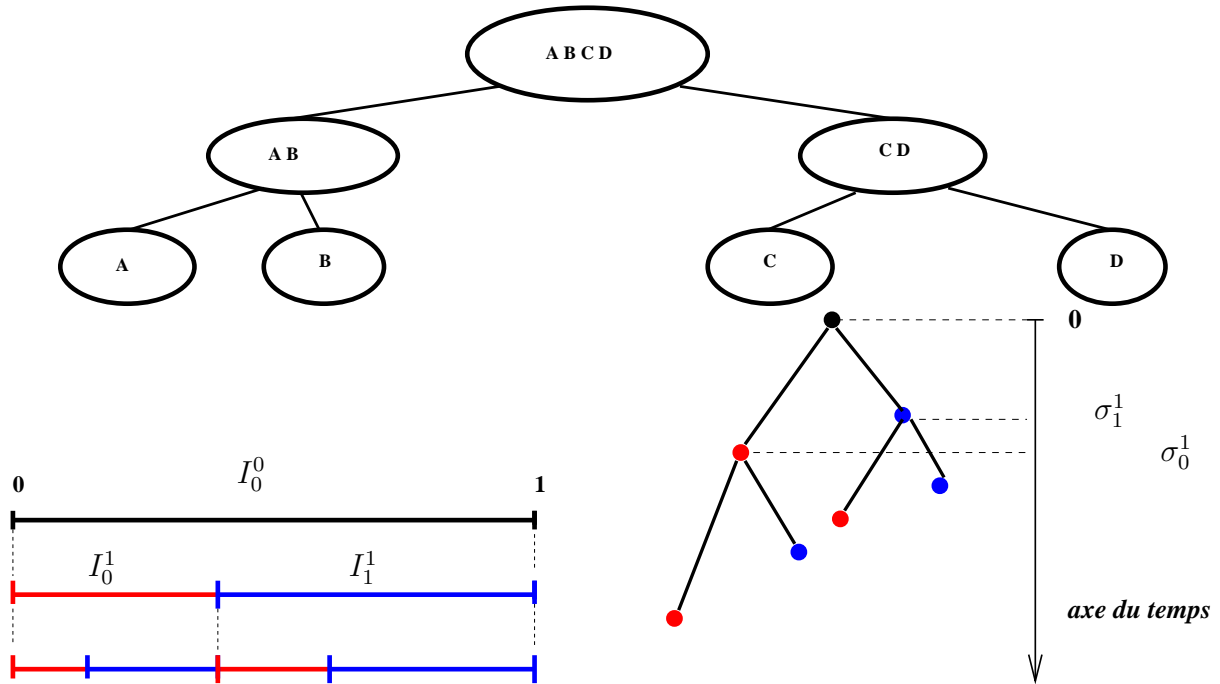


FIG. 1.3 – Décomposition de $[0, 1]$: transition entre algorithme en arbre et arbre généalogique.

La fonction de reproduction étant définie par $\mu(t) = \mathbb{E}(\xi(t))$, alors

$$\mu(t) = \sum_{i=0}^{Q-1} \mathbf{1}_{(p_i \geq e^{-t})}.$$

Comme

$$\mu(dt) = \sum_{i=0}^{Q-1} \delta_{-\log(p_i)}(t),$$

on obtient que

$$\int_0^\infty e^{-t} \mu(dt) = 1.$$

Donc ce processus de branchement est de type Bellman-Harris dont le paramètre Malthusien est égal à 1. Autrement dit, le *processus de décomposition est conservatif*. Ceci est dû au fait que l'*algorithme en arbre* considéré est conservatif, i. e. que $\sum p_i = 1$.

On définit le processus ϕ sur tout \mathbb{R} par

$$\phi(t) = \mathbf{1}_{(\mathcal{N}([0, e^t]) \geq D)},$$

et

$$Z_t^\phi = \sum_{n \geq 0} \sum_{k=0}^{Q^n - 1} \phi_k(t - \sigma_k^n),$$

le nombre d'individus, nés *avant ou après* t indifféremment, vérifiant $(\mathcal{N}([0, e^t]) \geq D)$. Ainsi défini, le processus ϕ traduit la condition de non-arrêt de l'algorithme en arbre, vu que, à un niveau donné n , si l'un des sous-intervalles I_k^n où $0 \leq k \leq Q^n - 1$ contient au moins D points parmi les $\mathcal{N}([0, x])$ initialement répartis sur l'intervalle $[0, 1]$, alors ce sous-intervalle sera divisé en Q sous-sous-intervalles et l'algorithme atteindra ainsi le niveau $(n + 1)$. Notons que, pour $n \geq 0$ fixé, la suite (ϕ_k) est *i.i.d.* de même loi que ϕ .

Remarque 4. Si on note α l'individu dont l'ordre dans l'arbre associé est (n, k) , i.e. le k -ième du n -ième niveau, et $\phi_\alpha := \phi(\cdot - \sigma_k^n)$, alors le processus ϕ_α ne dépendant pas uniquement de l'individu α mais de toute sa vie : parents et enfants, i.e. ϕ_α dépend de toute la branche contenant α . Pour plus de détails, voir Nerman [83].

Par le changement d'échelle $t = \log(x)$, pour tout réel $x > 0$, on obtient la relation suivante entre le coût de l'algorithme en arbre d'une part, et le nombre d'individus de l'arbre généalogique caractérisés par le processus ϕ d'autre part ;

$$R_{\mathcal{N}([0, x])} = 1 + QZ_t^\phi,$$

et donc pour étudier le comportement asymptotique au voisinage de l'infini du ratio $R_{\mathcal{N}([0, x])}/x$, il est équivalent d'étudier celui de

$$e^{-t} Z_t^\phi,$$

quand t tend vers l'infini.

1.5.3 Un théorème de renouvellement

L'idée de base pour l'étude d'un processus de branchement est l'équation (1.33) exprimant la récursivité de ce processus.

Proposition 10. *Le processus de branchement Z_t vérifie l'équation de renouvellement suivante*

$$Z_t^\phi \stackrel{dist.}{=} \phi_\emptyset(t) + \sum_{k=1}^{Q-1} Z_{k, t - \sigma_k^1}^\phi, \quad (1.33)$$

où $Z_{k, t}^\phi$ est un processus de branchement d'ancêtre le k -ième fils de l'ancêtre par défaut \emptyset et de caractéristique ϕ . Les processus $(Z_{k, t}^\phi)$ sont des copies indépendantes de Z_t^ϕ , indépendantes de ξ_\emptyset .

L'équation de récurrence (1.33) traduit en langage des *processus de branchement* celle de type *Diviser pour Régner* caractérisant le coût d'un *algorithme en arbre* (1.31). Elle exprime simplement la propriété suivante :

à un instant donné t , le nombre d'individus dont l'ancêtre est \emptyset et vérifiant la caractéristique ϕ , nombre noté Z_t^ϕ , est égal à celui des descendants des $Q - 1$ fils de \emptyset vérifiant la même caractéristique ϕ , augmenté de 1 si l'ancêtre par défaut \emptyset vérifie cette même caractéristique, d'où le terme $\phi_\emptyset(t)$.

Remarque 5. L'équation (1.33) diffère de celle étudiée par Nerman dans [83] au sens où on considère ici les descendants de tous les fils de l'ancêtre \emptyset , et non seulement ceux nés avant l'instant fixé t comme dans [83]. Ceci est dû à la définition du processus ϕ sur tout \mathbb{R} et non seulement sur \mathbb{R}_+ . L'instant fixé t impose une seule restriction, d'ailleurs de type aléatoire, sur les dates de naissances des individus de l'arbre généalogique, soit

$$\mathcal{N}([0, e^{t-\sigma_u}]) \geq D, \quad \text{où } \sigma_u \text{ désigne l'instant de naissance d'un individu } u.$$

Il est ainsi clair qu'il n'est pas nécessaire que l'individu u soit né avant l'instant t .

Si on pose

$$m_t^\phi = \mathbb{E}[e^{-t} Z_t^\phi],$$

alors, m_t^ϕ vérifie l'équation de *renouvellement*

$$m_t^\phi = e^{-t} \mathbb{E}[\phi(t)] + \int_0^\infty m_{t-s}^\phi e^{-s} \mu(ds) \quad (1.34)$$

Soit $\tilde{\mu}$ la mesure définie sur $[0, +\infty[$ par

$$\tilde{\mu}(t) = \int_0^t e^{-s} \mu(ds),$$

l'équation (1.34) s'écrit alors

$$m^\phi = f + m^\phi * \tilde{\mu}. \quad (1.35)$$

où f est définie sur tout \mathbb{R} par $f(t) = e^{-t} \mathbb{E}[\phi(t)]$. L'équation (1.35) est une *équation de renouvellement classique* étudiée notamment par Feller [36], Grimmett et Stirzaker [45], Jagers [52] et autres auteurs. Étant donné que f est bornée, l'équation (1.35) admet alors une unique solution bornée sur tout intervalle fini, solution donnée par $m = f * \nu$, où $\nu = \sum_{n \geq 0} (\tilde{\mu})^{*n}$, la série des produits de convolution de la mesure $\tilde{\mu}$.

1.5.4 Comportement asymptotique

L'objectif de cette section est de se mettre dans le cadre théorique établi par Jagers [52], et complété par Nerman [83] (cas non-arithmétique) et Gatzouras [43] (cas arithmétique), pour ainsi arriver à étudier le comportement asymptotique de $e^{-t} Z_t^\phi$ quand t tend vers l'infini.

En effet, dans ce cadre, bien que seul le cas d'un processus ϕ nul sur \mathbb{R}_- ait été traité, *i.e.* Z_t^ϕ correspond au nombre d'individus nés *avant* un instant donné $t > 0$ et caractérisés

par ϕ , la généralisation de ce traitement au cas d'un processus bien défini sur tout \mathbb{R} est élémentaire. D'ailleurs, dans son traitement de l'équation (1.35), Jagers [52] évoque cette généralisation assez intuitive.

Cas non-arithmétique :

On suppose que la distribution de $\tilde{\mu}$ n'est pas arithmétique, ce qui revient à faire l'hypothèse que \mathcal{W} n'est pas *exponentiellement* arithmétique. Partant de l'équation de renouvellement (1.35), on obtient le résultat suivant concernant le comportement asymptotique de $m := f * \tilde{\mu}$ au voisinage de $+\infty$.

Proposition 11 (Convergence des Moyennes). *Quand t tend vers l'infini, m_t^ϕ tend vers*

$$m_\infty^\phi : t \longmapsto m_\infty^\phi(t) := \frac{\int_{-\infty}^{\infty} e^{-t} \mathbb{E}[\phi(t)] dt}{\int_0^{\infty} t e^{-t} \mu(dt)}.$$

Schéma de preuve due à Jagers [52]. Soit (r_n) une suite réelle arbitraire tendant vers l'infini. Alors (r_n) admet une sous-suite telle que, quitte à remplacer r_n par cette sous-suite, la suite des mesures associées translatées ν_{r_n} définies par

$$\nu_{r_n} = \nu(r_n + \cdot)$$

tend vaguement vers ν quand n tend vers l'infini. Alors, on a

$$\begin{aligned} m(t + r_n) &= \int_0^{t+r_n} f(r_n + t - u) \nu(du) \\ &= \int_{-r_n}^t f(t - u) \nu_n(du) \rightarrow \int_{-\infty}^t f(t - u) \nu(du) \end{aligned}$$

pourvu que la fonction f soit assez régulière, par exemple, il suffit que, pour $t \in \mathbb{R}$, et $n \in \mathbb{N}$, la fonction $f(t - \cdot)$ soit uniformément intégrable par rapport à la mesure ν_{r_n} , *i.e.* pour tout $\varepsilon > 0$, il existe un réel $K > 0$ tel que, pour tout $n \in \mathbb{N}$, on a

$$\int_{|u|>K} f(t - u) \nu_{r_n}(du) < \varepsilon.$$

Or

$$\begin{aligned} \int_{|u|>K} f(t - u) \nu_{r_n}(du) &= \int_K^{+\infty} f(t - u) \nu_{r_n}(du) + \int_{-\infty}^{-K} f(t - u) \nu_{r_n}(du) \\ &= \int_{K-t}^{+\infty} f(-u) \nu_{t+r_n}(du) + \int_{-\infty}^{-K-t} f(-u) \nu_{t+r_n}(du) \\ &\leq \sum_{i \geq \lfloor K-t \rfloor} \int_i^{i+1} f(-u) \nu_{t+r_n}(du) + \sum_{i \leq \lceil -K-t \rceil} \int_i^{i+1} f(-u) \nu_{t+r_n}(du) \\ &\leq \sum_{i \geq \lfloor K-t \rfloor} \sup_{i \leq u \leq i+1} f(-u) (\nu(r_n + t + i + 1) - \nu(r_n + t + i)) \\ &+ \sum_{i \leq \lceil -K-t \rceil} \sup_{i \leq u \leq i+1} f(-u) (\nu(t + r_n + i + 1) - \nu(t + r_n + i)) \end{aligned}$$

En utilisant le lemme (5.2.3) de [52], on sait qu'il existe une constante A telle que ;

$$\nu(t + r_n + i + 1) - \nu(t + r_n + i) \leq A.$$

En notant que pour K assez grand, ceci implique que

$$\int_{|u|>K} f(t-u)\nu_{r_n}(du) \leq A \sum_{|i|\geq K} f(i),$$

il suffit alors de remarquer que la série $\sum_{i\in\mathbb{Z}} f(i)$ est convergente, où on rappelle que, dans notre cas particulier,

$$f(t) := e^{-t} \mathbb{E}(\phi(t)) = \mathbb{P}(\mathcal{N}([0, e^{-t}]) \geq D).$$

Reste alors à déterminer explicitement la limite m_∞^ϕ définie sur \mathbb{R} par

$$m_\infty^\phi(t) = \int_{-\infty}^t f(t-u)\nu(du).$$

Ainsi définie, la fonction m_∞^ϕ vérifie la propriété suivante

$$m_\infty^\phi(t) = \int m_\infty^\phi(t-u)\tilde{\mu}(du).$$

D'après [45], ou encore [52], Corollaire (5.2.4) et Théorème (5.2.6), la solution de cette dernière équation est donnée par

$$\frac{\int_{-\infty}^{\infty} e^{-t}\mathbb{E}[\phi(t)]dt}{\int_0^{\infty} t\tilde{\mu}(dt)}.$$

La proposition est ainsi prouvée. □

Ce résultat établit la convergence, en moyenne, du ratio $R_{\mathcal{N}([0,x])}/x$ quand x tend vers l'infini dans le cas non arithmétique. Des résultats plus forts concernant la convergence presque sûre ont été établis, par Nerman [83] dans le cas non arithmétique, et par Gatzouras [43] dans le cas arithmétique, pour un processus de branchement Z_t caractérisé par une fonction ϕ nulle sur \mathbb{R}_-^* .

Remarque 6. *En reprenant le travail de Nerman [83], moyennant des changements mineurs dû à la définition du processus ϕ sur tout \mathbb{R} , on peut obtenir un résultat plus fort : la convergence presque sûre vers la même limite que dans la Proposition 11.*

Dans notre cas, celui de l'algorithme en arbre de degré Q , on a

$$m_\infty^\phi = \frac{1}{(D-1) \int_0^1 |\log(y)|\mathcal{W}(du)}.$$

Donc la limite, quand x tend vers l'infini, de $\mathbb{E}[R_{\mathcal{N}([0,x])}]/x$ ainsi que celle de $\mathbb{E}(R_n)/n$ est égale à

$$\frac{Q}{(D-1) \int_0^1 |\log(y)| \mathcal{W}(du)}.$$

Cas arithmétique :

On suppose que $\tilde{\mu}$ est arithmétique, ou encore que la distribution de \mathcal{W} est exponentiellement arithmétique, de pas $\rho > 0$. Dans ce cas, en étudiant l'équation de renouvellement (1.35), on obtient le résultat suivant

Proposition 12 (Convergence des Moyennes). *Soit $s \in [0, \rho[$. Quand n tend vers l'infini, $m_{n\rho+s}^\phi$ tend vers*

$$\rho \frac{\sum_{j \in \mathbb{Z}} e^{-(j\rho+s)} \mathbb{E}[\phi(j\rho+s)]}{\int_0^\infty te^{-t} \mu(dt)}.$$

Schéma de preuve dû à Jagers [52]. C'est une variante d'un résultat dû à Jagers [52], facile à établir vu que la condition

$$\sum_{j \in \mathbb{Z}} f(j\rho+s) < \infty,$$

est vérifiée. On rappelle que

$$f(t) = e^{-t} \mathbb{P}(N([0, e^{-t}]) \geq D).$$

□

Remarque 7. *L'analogue du travail de Nerman [83] est celui de Gatzouras [43] dans le cas arithmétique dont résulte, moyennant des changements analogues au cas non arithmétique, la convergence presque sûre, au sens où :*

Pour tout $s \in [0, \rho[$, il existe un ensemble \mathcal{A}_s de probabilité 1 tel que, pour tout $\omega \in \mathcal{A}_s$, on a

$$\lim_{n \rightarrow \infty} e^{-n\rho} Z_{n\rho+s}^\phi(\omega) = \rho \frac{\sum_{j \in \mathbb{Z}} e^{-(j\rho+s)} \mathbb{E}[\phi(j\rho+s)]}{\int_0^\infty te^{-t} \mu(dt)}.$$

Ainsi, pour obtenir le comportement asymptotique de $e^{-t} Z_t^\phi$ quand t tend vers l'infini, on décompose la variable t sous la forme suivante

$$t = \left\lfloor \frac{t}{\rho} \right\rfloor \rho + \left\{ \frac{t}{\rho} \right\} \rho.$$

Par conséquent,

$$e^{-t} Z_t^\phi = e^{-\{t/\rho\}\rho} \left(e^{-\lfloor t/\rho \rfloor \rho} Z_{\lfloor t/\rho \rfloor \rho + \{t/\rho\}\rho}^\phi \right).$$

Le phénomène de fluctuations du comportement asymptotique du coût d'un algorithme en arbre dans le cas arithmétique est représenté dans cette étude par les oscillations du terme $e^{-\{t/\rho\}\rho}$ quand t tend vers l'infini.

Chapitre 2

Algorithme en arbre avec perte

Sommaire

2.1	Introduction	33
2.1.1	Motivation	33
2.1.2	Description du modèle	34
2.2	Coût Moyen de l'Algorithme	35
2.3	Comportement Asymptotique de l'Algorithme	36
2.3.1	Cas non arithmétique	37
2.3.2	Cas arithmétique	39
2.3.3	Un cas particulier : le cas binaire avec perte	39

2.1 Introduction

2.1.1 Motivation

Dans le cadre des réseaux de communication, le caractère *non conservatif* d'un algorithme traduit la perte de messages lors des tentatives d'accès au canal ; en effet, certains messages quittent le statut de *message en attente* sans pour autant être correctement transmis sur le canal. Ainsi, un *algorithme en arbre non conservatif* de degré G consiste à diviser aléatoirement un groupe initial en $(G + 1)$ sous-groupes, le dernier sous-groupe correspond à tous les messages perdus et donc qui n'interviennent plus dans le processus de division pour l'accès au canal.

L'efficacité d'un protocole de communication avec perte n'est plus mesurée par rapport au temps nécessaire pour transmettre un nombre donné n de messages. En effet, ce temps ne correspond qu'à la transmission d'une fraction de n ; les messages, parmi n initialement en conflit pour l'accès au canal, non perdus durant le processus de division récursif.

Un problème similaire a été étudié par Bertoin et Martinez [7] dans un cadre différent, celui des processus de fragmentation. En effet, une masse initiale est décomposée en fragments successivement avec possibilité de perte de masse, traduisant ainsi une *réelle perte physique*

de masse lors de ce processus de décomposition. Cette fragmentation s'arrête alors dès que la taille de tous les fragments de masse obtenus soit inférieure à un certain seuil η .

Le lien entre l'analyse faite au premier chapitre du cas conservatif, *i.e.* sans perte de messages, et cette étude est la normalisation des poids du processus de décomposition, *i.e.* les probabilités pour un élément du groupe initial d'être mis à droite p ou à gauche q , dans le cas d'un algorithme en arbre binaire par exemple. Dans ce cas, on n'a pas forcément l'égalité classique $p + q = 1$, mais plutôt l'inégalité $p + q \leq 1$. La normalisation de ces poids, selon une méthode proposée par Erdős *et al.* [31], et d'établir l'existence et l'unicité d'un réel $\varepsilon \in]0, 1]$ tel que $p^\varepsilon + q^\varepsilon = 1$. Ainsi, on définit, par analogie au cas conservatif, une mesure de branchement *normalisée* dont l'expression dans le cas binaire est donnée par $\mathcal{W} = p^\varepsilon \delta_p + q^\varepsilon \delta_q$. Pour une définition dans le cas général, voir Définition 4.

Dans ce chapitre, on démontre qu'un algorithme en arbre non conservatif admet un comportement asymptotique similaire à celui sans pertes, au sens où, le coût moyen d'un tel algorithme obéit à un phénomène oscillatoire si la mesure de branchement normalisée \mathcal{W} est exponentiellement arithmétique. Ainsi, le temps de transmission de n messages est proportionnel, moyennant un terme périodique de faible amplitude, à n^ε où ε désigne le taux de conservation de l'algorithme, voir Remarque 9. Tandis que, dans le cas non arithmétique, on obtient une vraie Loi des Grands Nombres.

2.1.2 Description du modèle

On considère un algorithme en arbre de degré aléatoire G qui divise aléatoirement un groupe initial de taille n en G sous-groupes avec possibilité de perte; *i.e.* conditionnellement à l'événement $\{G = \ell\}$, pour $1 \leq i \leq \ell$, un élément du groupe initial est, soit mis dans le i -ème sous-groupe avec probabilité $V_{i,\ell}$, soit perdu avec une probabilité $V_{0,\ell}$ dont la moyenne est donnée par

$$\mathbb{E}(V_{0,\ell}) := 1 - \sum_i \mathbb{E}(V_{i,\ell}),$$

où $V_\ell = (V_{i,\ell}; 0 \leq i \leq \ell)$ est un vecteur aléatoire de probabilités sur $\{0, \dots, \ell\}$.

Si N_i est le nombre d'éléments parmi n mis dans le i -ème sous-groupe (N_0 celui de ceux perdus), alors, conditionnellement à l'événement $\{G = \ell\}$ est aux variables aléatoires $V_{0,\ell}, V_{1,\ell}, \dots, V_{\ell,\ell}$, la distribution du vecteur (N_0, \dots, N_ℓ) est multinomiale de paramètres n et $(V_{0,\ell}, V_{1,\ell}, \dots, V_{\ell,\ell})$,

$$\mathbb{P}((N_0, \dots, N_\ell) = (m_0, \dots, m_\ell) | V_\ell) = \frac{n!}{m_0! m_1! \dots m_\ell!} \prod_{k=0}^{\ell} (V_{k,\ell})^{m_k},$$

pour $(m_i) \in \mathbb{N}^{\ell+1}$ tels que $m_0 + m_1 + \dots + m_\ell = n$.

Remarque 8. Comme pour le cas conservatif, on fait l'hypothèse suivante, pour assurer la non dégénérescence de l'algorithme; il existe un réel $\delta > 0$ tel que

$$\sup_{\ell \geq 2} \sup_{1 \leq i \leq \ell} V_{i,\ell} \leq \delta < 1. \tag{A}$$

2.2 Coût Moyen de l'Algorithme

Le coût moyen de l'algorithme vérifie l'équation de récurrence suivante

- $R_n = 1$ si $n < D$;
- pour $n \geq D$,

$$R_n \stackrel{\text{dist.}}{=} 1 + R_{1, N_1^n} + \cdots + R_{G, N_G^n}, \quad (2.1)$$

où conditionnellement à l'événement $\{G = \ell\}$ et aux variables aléatoires $V_{0,\ell}, V_{1,\ell}, \dots, V_{\ell,\ell}$,

1. le vecteur (N_0^n, \dots, N_ℓ^n) a une distribution multinomiale de paramètre n et $(V_{0,\ell}, V_{1,\ell}, \dots, V_{\ell,\ell})$;
2. pour $(p_i) \in \mathbb{N}^\ell$, les variables $R_{1,p_1}, \dots, R_{\ell,p_\ell}$ sont indépendantes;
3. Pour $1 \leq i \leq \ell$, la variable R_{i,p_i} a la même distribution que R_{p_i} .

La transformée de *Poisson* du coût moyen de l'algorithme est notée r , elle vérifie l'équation fonctionnelle suivante :

$$r(x) = 1 + \mathbb{E} \left(\sum_{i=1}^G r(V_{i,G} x) \right) - \mathbb{E}(G) \mathbb{P}(\mathcal{N}([0, x]) \leq D - 1).$$

Et donc, en dérivant, on obtient que

$$r'(x) = \mathbb{E} \left(\sum_{i=1}^G V_{i,G} r'(V_{i,G} x) \right) + \mathbb{E}(G) \frac{x^{D-1}}{(D-1)!} e^{-x}. \quad (2.2)$$

Suivant la même méthode de normalisation des poids utilisée par Erdős *et al.* [31], on définit la fonction

$$g(u) = \mathbb{E} \left(\sum_{i=1}^G (V_{i,G})^u \right).$$

La fonction g est strictement décroissante sur \mathbb{R} , donc il existe un unique réel ε , dont un encadrement est $]0, 1]$ vu que $g(0) = \mathbb{E}(G) > 1$, et que $g(1) \leq 1$, tel que $g(u) = 1$, *i.e.* que

$$\mathbb{E} \left(\sum_{i=1}^G (V_{i,G})^\varepsilon \right) = 1.$$

On introduit le changement de fonctions suivant $r'(x) = x^{\varepsilon-1} f(x)$. L'équation fonctionnelle basique devient alors

$$f(x) = \mathbb{E} \left(\sum_{i=1}^G (V_{i,G})^\varepsilon f(V_{i,G} x) \right) + \mathbb{E}(G) \frac{x^{D-\varepsilon}}{(D-1)!} e^{-x}. \quad (2.3)$$

Définition 4. On définit la mesure de décomposition normalisée \mathcal{W} sur $[0, 1]$ par

$$\int f(x) \mathcal{W}(dx) = \mathbb{E} \left(\sum_{i=1}^G (V_{i,G})^\varepsilon f(V_{i,G}) \right),$$

où f est une fonction borélienne positive.

Remarque 9. Le paramètre $\epsilon \in]0, 1]$ mesure, en un certain sens, le taux de conservation du processus de décomposition de l'algorithme. Le cas conservatif correspond à $\epsilon = 1$, on retrouve ainsi l'expression du coût moyen établie dans le premier chapitre.

Soit A une variable aléatoire de loi \mathcal{W} . L'équation fonctionnelle (2.3) s'écrit alors sous la forme suivante

$$f(x) = \mathbb{E}(f(Ax)) + h(x),$$

où

$$h(x) = \mathbb{E}(G) \frac{x^{D-\epsilon}}{(D-1)!} e^{-x}.$$

Par itération, et comme $f(0) = 0$ (car $r'(0) = 0$ et $\epsilon < 1$), on obtient que

$$f(x) = \sum_{k \geq 0} \mathbb{E}(h(X_k)), \text{ avec } X_k = A_k \cdot X_{k-1}, (A_k) \text{ i.i.d. de loi } \mathcal{W} \text{ et } X_0 = x > 0$$

Si on note par $\pi_k = \prod_{i=0}^{k-1} A_i$, alors $X_k = x\pi_k$ et donc

$$f(x) = \mathbb{E}(G) \sum_{k \geq 0} \mathbb{E} \left(\frac{(\pi_k x)^{D-\epsilon}}{(D-1)!} e^{-\pi_k x} \right).$$

Tenant compte de la condition au bord $r(0) = 1$, on obtient alors l'expression de la transformée de *Poisson* du coût moyen

$$r(x) = 1 + \mathbb{E}(G) \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k^\epsilon} \mathbf{1}_{(t_D < \pi_k x)} \right).$$

Par le même procédé de dépoissonisation probabiliste utilisé précédemment, on obtient alors le résultat suivant

Proposition 13. *Le coût moyen d'un algorithme en arbre dont le taux de conservation est noté $\epsilon \in]0, 1]$ est donné par*

$$\mathbb{E}(R_n) = 1 + \mathbb{E}(G) \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k^\epsilon} \mathbf{1}_{(U_{D,n} < \pi_k)} \right),$$

où $U_{D,n}$ désigne la D -ième plus petite variable aléatoire parmi n uniformes indépendantes sur $[0, 1]$.

2.3 Comportement Asymptotique de l'Algorithme

L'objectif de cette section est d'établir un équivalent asymptotique du coût moyen de l'algorithme en tenant compte d'une possible perte de données avec un taux $\epsilon \in]0, 1]$. Utilisant la même approche que le cas conservatif (sans perte), on définit la marche aléatoire S_k de pas $-\log A$ et partant de zéro. Le temps de dépassement d'une barrière

donnée $x > 0$ par cette marche aléatoire est noté $\nu(x)$. Ainsi, le coût moyen de l'algorithme s'écrit sous la forme suivante

$$\mathbb{E}(R_n) = 1 + \mathbb{E}(G) \mathbb{E} \left(\sum_{k \geq 0}^{\nu(-\log(U_{D,n})-1)} \frac{1}{\pi_k^\varepsilon} \right).$$

On définit la fonction ψ par

$$\Psi(x) = \mathbb{E} \left(\sum_{k \geq 0}^{\nu(x)-1} e^{\varepsilon S_k} \right).$$

L'étude asymptotique du coût moyen revient à étudier celui de

$$\Psi(x) e^{-\varepsilon x} = \mathbb{E} \left(\sum_{k \geq 1}^{\nu(x)} e^{\varepsilon(S_{\nu(x)-k}-x)} \right).$$

Utilisant les inégalités de Lorden d'une manière analogue à celle du Lemme 8 dans [81], le résultat suivant est immédiat ;

Lemme 2. *Sous la condition*

$$\mathbb{E} \left(\frac{-\log A}{A^\varepsilon} \right) < \infty,$$

on a

$$\sup_{x \geq 0} \mathbb{E} (e^{\varepsilon(S_{\nu(x)}-x)}) < \infty$$

Soit $1 \leq K \leq \nu(x)$, on a

$$\sum_{k \geq K}^{\nu(x)} e^{\varepsilon(S_{\nu(x)-k}-x)} \leq e^{\varepsilon(S_{\nu(x)}-x)} \sum_{k \geq K}^{\nu(x)} e^{\varepsilon(S_{\nu(x)-k}-S_{\nu(x)})} \leq e^{\varepsilon(S_{\nu(x)}-x)} \frac{(\delta^\varepsilon)^K}{1 - \delta^\varepsilon}.$$

2.3.1 Cas non arithmétique

Dans ce cas, le *théorème de renouvellement (version continue)* appliqué à la marche aléatoire arrêtée (S_k) montre que la variable $S_{\nu(x)-k} - x$ converge en loi, quand x tend vers l'infini, vers $-(\tau^* + \tau_1 + \dots + \tau_{k-1})$, où les variables (τ_i) sont *i.i.d.* de loi \mathcal{W} indépendantes de τ^* dont la densité est donnée par $\mathbb{P}(-\log A \geq x) / \mathbb{E}(-\log A)$.

Conclusion, $\Psi(x) e^{-\varepsilon x}$ tend, quand x tend vers l'infini, vers

$$\mathbb{E} \left(\sum_{k \geq 1}^{\infty} e^{-\varepsilon(\tau^* + \tau_1 + \dots + \tau_{k-1})} \right) = \frac{1}{-\varepsilon \mathbb{E}(\log A)}.$$

Proposition 14 (Cas non arithmétique). *Si la mesure \mathcal{W} n'est pas exponentiellement arithmétique et sous la condition*

$$\mathbb{E} \left(\frac{-\log A}{A^\varepsilon} \right) < \infty,$$

on a

$$\lim_{n \rightarrow \infty} \frac{\mathbb{E}(R_n)}{n^\varepsilon} = \frac{\mathbb{E}(G)\Gamma(D - \varepsilon)}{(D - 1)!\mathbb{E}(-\log A)},$$

où Γ désigne la fonction Gamma d'Euler.

Démonstration. Pour $n \geq 1$, on a

$$\frac{\mathbb{E}(R_n)}{n^\varepsilon} = \frac{1}{n^\varepsilon} + \mathbb{E}(G)\mathbb{E} \left(\Psi[-\log(U_{D,n})] \exp(\varepsilon \log(U_{D,n})) \frac{1}{(nU_{D,n})^\varepsilon} \right).$$

Quand n tend vers l'infini la variable $nU_{D,n}$ converge en distribution vers une variable aléatoire t_D , somme de D variables exponentielles *i.i.d.* de paramètre 1, de plus,

$$\lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{(nU_{D,n})^\varepsilon} \right) = \mathbb{E}(1/t_D^\varepsilon) = \frac{\Gamma(D - \varepsilon)}{(D - 1)!},$$

où Γ désigne la fonction Gamma D'Euler.

Pour $\varepsilon_0 > 0$, il existe K tel que, pour $x > K$, $|\Psi(x) \exp(-\varepsilon x) + 1/\varepsilon \mathbb{E}(\log A)| < \varepsilon_0$, si C désigne le *sup* de $x \rightarrow \Psi(x) \exp(-\varepsilon x)$ sur \mathbb{R}_+ , alors

$$\begin{aligned} & \left| \mathbb{E} \left(\Psi[-\log(U_{D,n})] \exp(\varepsilon \log(U_{D,n})) \frac{1}{(nU_{D,n})^\varepsilon} \right) - \frac{\Gamma(D - \varepsilon)}{(D - 1)!\mathbb{E}(-\log A)} \right| \\ & \leq \varepsilon_0 \mathbb{E} \left(\frac{1}{(nU_{D,n})^\varepsilon} \right) + \left(C + \frac{1}{\varepsilon \mathbb{E}(-\log A)} \right) \mathbb{E} \left(\mathbf{1}_{\{U_{D,n} > \exp(-K)\}} \frac{1}{(nU_{D,n})^\varepsilon} \right) \\ & \quad + \frac{1}{\varepsilon \mathbb{E}(-\log A)} \left| \mathbb{E} \left(\frac{1}{(nU_{D,n})^\varepsilon} \right) - \frac{\Gamma(D - \varepsilon)}{(D - 1)!} \right|. \end{aligned} \quad (2.4)$$

Pour $K_2 > 0$,

$$\begin{aligned} & \limsup_{n \rightarrow +\infty} \mathbb{E} \left(\mathbf{1}_{\{U_{D,n} > \exp(-K)\}} \frac{1}{(nU_{D,n})^\varepsilon} \right) \\ & \leq \limsup_{n \rightarrow +\infty} \mathbb{E} \left(\mathbf{1}_{\{nU_{D,n} > K_2 \exp(-K)\}} \frac{1}{(nU_{D,n})^\varepsilon} \right) = \mathbb{E} \left(\mathbf{1}_{\{t_D > K_2 \exp(-K)\}} \frac{1}{t_D^\varepsilon} \right), \end{aligned}$$

et ce terme tend vers 0 quand K_2 tend vers l'infini. On conclut que le terme de droite dans la relation (2.4) est arbitrairement petit quand n tend vers l'infini. La proposition est prouvée. \square

2.3.2 Cas arithmétique

Si la mesure \mathcal{W} est exponentiellement arithmétique de pas $\rho > 0$, alors le *théorème de renouvellement (version discrète)* est appliqué à la marche aléatoire normalisée $(S_k^*) := (S_k/\rho)$ dont le temps d'arrêt associé est défini par

$$\nu^* \left(\frac{x}{\rho} \right) = \nu^* \left(\left\lfloor \frac{x}{\rho} \right\rfloor \right) := \nu(x).$$

Le même raisonnement utilisé pour la cas non arithmétique implique que

$$\lim_{x \rightarrow +\infty} \Psi(x) e^{-\varepsilon \rho \lceil x/\rho \rceil} = \frac{1}{-\varepsilon \mathbb{E}(\log A)} \frac{\rho e^{-\varepsilon \rho}}{1 - e^{-\varepsilon \rho}}.$$

Proposition 15 (Cas arithmétique). *Si la mesure \mathcal{W} est exponentiellement arithmétique de pas $\rho > 0$ et sous la condition*

$$\mathbb{E} \left(\frac{-\log A}{A^\varepsilon} \right) < \infty,$$

alors, quand n tend vers l'infini,

$$\frac{\mathbb{E}(R_n)}{n^\varepsilon} \sim F \left(\frac{\log n}{\rho} \right),$$

où F est une fonction périodique de période 1 définie pour tout $x \geq 0$ par

$$F(x) = \frac{\mathbb{E}(G)}{-\varepsilon \mathbb{E}(\log A)} \frac{\rho}{1 - e^{-\varepsilon \rho}} \int_0^\infty \exp \left(-\varepsilon \rho \left\{ x - \frac{\log y}{\rho} \right\} \right) \frac{y^{D-1-\varepsilon}}{(D-1)!} e^{-y} dy,$$

$\{z\} := z - \lfloor z \rfloor$ désignant la partie fractionnaire de z .

Démonstration. Il suffit de remarquer que, utilisant le même raisonnement que le cas non-arithmétique, on a l'équivalence suivante

$$\frac{1}{n^\varepsilon} \mathbb{E}(\Psi(-\log(U_{D,n}))) \cdot (\varepsilon \mathbb{E}(-\log A)) \frac{1 - e^{-\varepsilon \rho}}{\rho} \sim \mathbb{E} \left(\exp \left(-\varepsilon \rho \left\{ \frac{\log n}{\rho} - \frac{\log t_D}{\rho} \right\} \right) \frac{1}{t_D^\varepsilon} \right).$$

□

2.3.3 Un cas particulier : le cas binaire avec perte

On considère le cas particulier d'un *algorithme en arbre binaire avec ou sans perte*, soit $G \equiv 2$, $D = 2$, $V_{1,2} \equiv p$ et $V_{2,2} \equiv q$ où $p + q \leq 1$. Alors, il existe un unique réel $\varepsilon \in]0, 1]$ tel que $p^\varepsilon + q^\varepsilon = 1$. La mesure \mathcal{W} est donnée par $\mathcal{W} = p^\varepsilon \delta_p + q^\varepsilon \delta_q$. L'arithméticité de la mesure $-\log \mathcal{W}$ est alors équivalente au fait que le quotient $\log p / \log q$ soit rationnel.

Cas non arithmétique :

Si $\log p / \log q \notin \mathbb{Q}$, alors le coût moyen de l'algorithme $\mathbb{E}(R_n)$ est équivalent, quand n tend vers l'infini, à

$$-2 n^\varepsilon \frac{\Gamma(2 - \varepsilon)}{p^\varepsilon \log p + q^\varepsilon \log q}.$$

Cas arithmétique :

S'il existe un couple d'entiers premiers entre eux $(a, b) \in (\mathbb{N}^*)^2$ tel que $\log p / \log q = a/b$, alors la mesure $-\log \mathcal{W}$ est arithmétique de pas $\rho = -\log p/a$. Le coût moyen de l'algorithme $\mathbb{E}(R_n)$ est équivalent, quand n tend vers l'infini, à

$$C n^\varepsilon F\left(\frac{\log n}{\rho}\right),$$

où C est une constante donnée par

$$\frac{-2}{\varepsilon(p^\varepsilon \log p + q^\varepsilon \log q)} \frac{\rho}{1 - e^{-\varepsilon\rho}},$$

et F est une fonction périodique définie, pour tout $x \geq 0$, par

$$F(x) = \int_0^\infty \exp\left(-\varepsilon\rho \left\{x - \frac{\log y}{\rho}\right\}\right) y^{1-\varepsilon} e^{-y} dy.$$

Chapitre 3

Algorithme d'élection de leader

Sommaire

3.1	Introduction	41
3.1.1	Problème d'élection de leader	42
3.1.2	Processus de division et structure d'arbre	44
3.1.3	Travaux précédents	45
3.1.4	Variantes de l'algorithme d'élection de leader	47
3.2	Résultats	47
3.3	Coût Moyen de l'Algorithme	49
3.3.1	Coût moyen	49
3.3.2	Poissonisation	50
3.3.3	Dépoissonisation	51
3.4	Analyse Asymptotique du Coût Moyen	53
3.4.1	Deux suites aléatoires, un temps d'arrêt	53
3.4.2	Phénomène de fluctuations asymptotiques	54
3.4.3	Estimation du reste	56
3.5	Distribution du Coût de l'Algorithme	58
3.6	Algorithme d'Élection de Leader : Version Markovienne	60
3.6.1	Définition du coût de l'algorithme	60
3.6.2	Étude du coût moyen	61
3.6.3	Cas particulier : le cas indépendant	63

3.1 Introduction

Un réseau de communication de type *single-hop* est un système distribué de n noeuds, appelés aussi *stations*, partageant un canal commun de communication qui ne peut transmettre qu'un seul message par unité de temps. Dans le cas particulier d'un réseau avec détection de *collision*, le canal livre une information ternaire ; chaque *station* peut simultanément envoyer son message vers le réseau et écouter le canal pour détecter

1. une *collision* s'il y a eu au moins deux essais de transmission,
2. un *silence* si aucune station n'a tenté de transmettre son message,
3. ou un *succès* s'il y a eu un seul essai de transmission.

Un tel réseau avec détection de *collision* est appelé *canal à accès multiple*.

On considère un *canal à accès multiple* composé de n stations dont un *leader par défaut* est désigné et ce afin de contrôler et organiser le système entier. En pratique, faute de liaisons fiables entre les stations, il est possible que ce *leader par défaut* soit temporairement hors service ce qui se traduit par le *silence* du *leader*. Dans ce cas, le système arrête ses opérations en cours et initialise la procédure d'*élection* : le système doit identifier un nouveau *leader* et le faire dans un délai raisonnable pour le bon fonctionnement du réseau. On s'intéresse au *coût* de l'algorithme, *i.e.* le temps, ou nombre d'unités de temps, nécessaire au système pour déterminer un *leader*.

3.1.1 Problème d'élection de leader

La taille n du réseau est supposée inconnue. Chaque station est supposée posséder un *numéro identifiant unique*, noté ID. Pour élire un *leader* parmi elles, les stations doivent utiliser le même algorithme. Le cas $n \in \{0, 1\}$ étant trivial, on suppose que $n \geq 2$. L'algorithme d'*élection de leader* dans sa version basique est le suivant :

- Initialisation déterministe : à la première unité de temps, chaque station envoie au réseau un message avec son numéro ID. Comme $n \geq 2$, chacune détecte une *collision*.
- Processus de sélection aléatoire : chaque station S génère, d'une manière indépendante, une variable aléatoire de *Bernoulli* B_S de paramètre p . Seules celles obtenant le résultat $B_S = 1$ sont autorisées à tenter une transmission durant la prochaine unité de temps.

Pour une station donnée S , il y a deux cas :

1. Si $B_S = 1$, la station S est dite *Active*; S tente une transmission et peut ainsi détecter
 - soit un *succès*; seule la station S a effectué un essai de transmission, alors les autres stations reçoivent son message avec son numéro ID et S obtient le statut de *leader*. L'algorithme s'arrête.
 - soit une *collision*; la station S n'est pas la seule à être candidate pour devenir *leader*, et donc doit générer encore une variable de *Bernoulli* B_S .

Comme la station S vient juste de tenter une transmission, elle ne peut pas détecter de *silence*.

2. Sinon, la station S devient *Inactive*; elle reste candidate à devenir *leader*, continue à écouter le canal sans pour autant participer aux transmissions. Ainsi, elle peut détecter

- un *succès*; une seule station $S' \neq S$ a effectué un essai de transmission. Les autres stations (y compris la station S) reçoivent son numéro ID. La station S' obtient ainsi le statut de *leader*. L'algorithme s'arrête.
- une *collision*; bien que la station S n'ait pas participé au processus de sélection, il y a eu au moins deux essais de transmissions de deux stations *Actives*. Donc, la station S est éliminée.
- un *silence*; toutes les stations sont *Inactives* (y compris S), donc la station S génère une variable de *Bernoulli* B_S pour décider d'envoyer encore, ou pas, un message avec son numéro ID au réseau.

Ainsi, à la fin de l'algorithme, une seule station reste *Active* et devient alors le *leader* du système.

L'exemple suivant illustre la procédure d'élection appliquée à un groupe de 4 stations $\{A, B, C, D\}$. Dans ce cas, le *leader* A est élu au bout de 4 unités de temps.

unités de temps	1	2	3	4
Stations Actives	A B C D	A B C		A
Stations Inactives		D	A B C	B C
Stations Éliminées			D	D
État du Canal	<i>Collision</i>	<i>Collision</i>	<i>Silence</i>	<i>Succès</i>

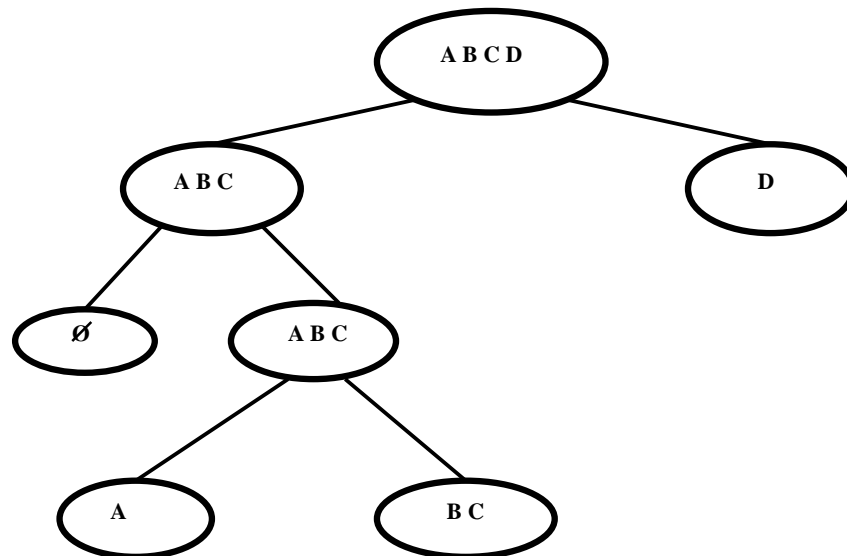


FIG. 3.1 – Élection du *leader* A ; $H_4 = 1 + P_4 = 4$.

Définition 5 (Coût de l'algorithme). *Le nombre d'étapes nécessaires pour déterminer le leader est le coût de l'algorithme, noté H_n quand la taille du réseau est égale à n .*

Les applications d'un tel algorithme d'élimination aléatoire sont multiples dans le cadre des systèmes distribués tels que les réseaux de communication sans fil. Dans les réseaux mobiles de type *Ad-hoc*, les échecs de liaisons se produisent dès que les noeuds, étant mobiles, quittent une zone appelée *région de transmission* définie par rapport à la positions des noeuds dans l'espace. La topologie instable du réseau rend le problème d'élection de *leader* plus complexe. Pour plus de détails, voir [74]. Élire un *leader* dans un réseau informatique est fondamental pour superviser les communications entre les machines et ainsi synchroniser le système. Voir [38]. Le même problème est aussi étudié dans le cadre des réseaux radio. Pour plus de détails, voir [19]. Des versions plus élaborées de l'algorithme d'élection de *leader* dans un réseau radio sans détection de *collision*, sous une contrainte d'optimalité de coût d'énergie, voir [64].

3.1.2 Processus de division et structure d'arbre

Formellement, appliqué à un groupe initial de n éléments, l'algorithme divise ce dernier en deux sous-groupes. La probabilité qu'un élément soit mis dans le sous-groupe de gauche est notée p . Ce sous-groupe sera à son tour divisé en deux par le même processus. Les autres éléments seront ignorés, sauf si le premier sous-groupe, celui de gauche, est vide. Dans ce cas, l'algorithme redémarre partant du niveau précédent.

ALGORITHME D'ÉLECTION DE LEADER $\mathcal{E}(n)$

- CONDITION D'ARRÊT.
Si $n = 1$ \longrightarrow ARRÊT.
 - PROCESSUS DE SÉLECTION ALÉATOIRE.
Si $n \geq 2$, diviser aléatoirement n en $(n_1, n - n_1)$.
 - Si $n_1 \neq 0 \longrightarrow$ APPLIQUER $\mathcal{E}(n_1)$.
 - Sinon \longrightarrow APPLIQUER $\mathcal{E}(n)$.
-

Cet algorithme distribué est un processus d'élimination aléatoire admettant une structure naturelle d'arbre (Fig.3.1). À la racine de l'arbre associé se trouve le groupe initial. Lors de la première division, il génère deux noeuds : celui de gauche sera divisé en deux par le même processus, celui de droite est un noeud dit *terminal*, appelé aussi *feuille*, ne sera traité par l'algorithme que si le noeud de gauche est vide, dans ce cas, il sera un noeud interne et sera divisé en deux à son tour.

Cette structure d'arbre peut être représentée par un arbre incomplet dont seule la branche de gauche sera développée. On définit la *hauteur* de l'arbre associé comme étant la longueur du chemin reliant la racine au *leader*, le plus long chemin de l'arbre reliant la racine à une *feuille* (voir [38]). Alors, le *coût* de l'algorithme est égal à la *hauteur* de l'arbre associé. C'est aussi la *profondeur maximale* atteinte dans l'arbre, ajoutée de 1, vu que, par convention, la profondeur de la racine est 0. Fig.3.1 illustre cette égalité. Si on désigne par P_n la *profondeur maximale* atteinte dans un arbre incomplet engendré par n éléments, on a alors l'égalité suivante $H_n = 1 + P_n$.

3.1.3 Travaux précédents

Il est connu que le *coût* moyen de l'algorithme d'élection de *leader* admet un ordre logarithmique en n et présente asymptotiquement un comportement oscillatoire. Voir [90] pour le cas non biaisé $p = 1/2$, [53] pour celui biaisé $p \neq 1/2$.

Considérons le modèle *poissonien* de l'algorithme d'élection de *leader* où le processus d'élection est appliqué à un groupe de taille aléatoire notée $\mathcal{N}([0, x])$ suivant une loi de *Poisson* (voir A.1). Soit h la *transformée de Poisson* de la suite des coûts moyens de l'algorithme $(\mathbb{E}(H_n))_{n \geq 0}$.

Définition 6 (*Transformée de Poisson*). Pour $x > 0$, la transformée de Poisson de la suite $(\mathbb{E}(H_n))$ est la fonction h définie par

$$h(x) = \mathbb{E}(H_{\mathcal{N}([0, x])}) = \sum_{n=2}^{+\infty} \mathbb{E}(H_n) \frac{x^n}{n!} e^{-x}.$$

Alors, la fonction h est solution d'une équation fonctionnelle appelée *équation fonctionnelle basique* associée à l'algorithme

$$h(x) = h(px) + h(qx) e^{-px} + f(x), \quad (3.1)$$

où $p + q = 1$ et f est une fonction donnée. L'équation (3.1) est le point de départ de toute étude faite sur cet algorithme.

Le cas non biaisé

Quand le processus de division suit une variable aléatoire de *Bernoulli* de paramètre $1/2$, l'algorithme d'élection de *leader* est dit *symétrique*; la probabilité qu'un élément soit mis dans le sous-groupe de gauche, ou de droite est égale à $1/2$. Remarquons que, dans ce cas, l'équation fonctionnelle (3.1) est résolue par des itérations directes. En effet, la *transformée de Poisson* h vérifie

$$h(x) = h(x/2) (1 + e^{-x/2}) + f(x),$$

ce qui s'écrit

$$g(x) = g(x/2) + \frac{f(x)}{1 - e^{-x}},$$

où $g(x) = h(x)/(1 - e^{-x})$.

La première analyse de l'algorithme d'élection de *leader* a été proposée par Prodinger [90]. Il a étudié différents paramètres tels que la *hauteur*, appelée *profondeur* dans son article, la *taille* de l'arbre associé, *i.e.* le nombre total de noeuds... Utilisant des techniques combinatoires, il a établi des expressions exactes ainsi que des équivalents asymptotiques pour ces quantités dans le cas symétrique. Ainsi, il a démontré que, pour un groupe initial de taille n , l'algorithme s'arrête en moyenne au bout de $\log_2 n$ étapes.

Utilisant les techniques d'analyse complexe telles que la transformée et la transformée inverse de Mellin, Fill *et al.* [38] ont étudié le comportement asymptotique des deux

premiers moments du *coût* de l'algorithme. De plus, ils ont obtenu l'expression exacte ainsi que le comportement asymptotique de la distribution de la variable aléatoire H_n et montré que la distribution limite du *coût centré* de l'algorithme $H_n - \lfloor \log_2 n \rfloor$ n'existe pas. Pour une étude analytique du cas non biaisé, voir [38]. Pour plus de détails sur la transformée de Mellin, voir [40].

Le cas biaisé

Si le processus de *division* est biaisé, *i.e.* la probabilité qu'un élément soit mis dans le sous-groupe de gauche est $p \neq 1/2$, l'algorithme est dit *asymétrique*. Les études du cas biaisé sont plus rares. Cet algorithme a été étudié par Janson et Szpankowski [53] utilisant des techniques d'analyse complexe. Le comportement asymptotique des deux premiers moments du *coût* de l'algorithme H_n est établi en fonction de la suite des valeurs numériques de ce dernier $(\mathbb{E}(H_j))_{j \in \mathbb{N}}$, suite dont les termes sont calculés numériquement à l'aide de deux équations récurrentes.

Les auteurs obtiennent que, modulo $O(1/n)$, la moyenne du *coût* moyen centré de l'algorithme $H_n - \lfloor -\log_p(n) \rfloor$ est asymptotiquement équivalent à une certaine fonction périodique $\delta_{\mathcal{T}}$ de $-\log_p(n)$

$$\mathbb{E}(H_n) - \lfloor -\log_p(n) \rfloor = \delta_{\mathcal{T}}(-\log_p(n)) + O\left(\frac{1}{n}\right),$$

où δ s'exprime en fonction d'une suite de constantes $\mathcal{T} = (T_k)_{k \in \mathbb{Z}}$

$$T_k = \sum_{j=2}^{+\infty} \frac{\mathbb{E}(H_j) q^j}{j!} \Gamma(j - 2\pi i k / \log p),$$

et Γ est la fonction Gamma d'Euler.

Pour un entier fixé $k \in \mathbb{Z}$, et pour évaluer la constante T_k , il est nécessaire de calculer tous les termes de la suite $(\mathbb{E}(H_j))_{j \geq 2}$. Dans [53], les auteurs utilisent la fonction génératrice de H_n pour obtenir une relation de récurrence en $\mathbb{E}(H_n)$ avec laquelle la valeur numérique du *coût* moyen est calculée.

$$\mathbb{E}(H_n) = 1 + q^n \mathbb{E}(H_n) + \sum_{k=0}^n \binom{n}{k} p^k q^{n-k} \mathbb{E}(H_k).$$

Ainsi, le comportement asymptotique du *coût* moyen de l'algorithme $\mathbb{E}(H_n)$ est exprimé en fonction de tous les termes de ses valeurs exactes $(\mathbb{E}(H_j))_{j \geq 2}$. La même méthode est utilisée pour l'étude de la variance avec un résultat similaire.

Cette dépendance implicite est due à l'asymétrie de l'équation fonctionnelle (3.1) obtenue par *Poissonisation*

$$h(x) = h(px) + h(qx) e^{-px} + f(x).$$

Le coefficient e^{-px} rend inapproprié l'établissement d'un schéma itératif, tel que celui utilisé par Fayolle *et al.* [34] dans le cadre des protocoles pour un canal de transmission à accès multiple. Appliquer la transformée de Mellin à l'équation (3.1) avant de la résoudre engendre cette dépendance dans la formule obtenue par Janson et Szpankowski dans [53].

3.1.4 Variantes de l'algorithme d'élection de leader

Élection de leader dans un réseau de taille connue

Considérons un simple algorithme d'élection de *leader* dans le cadre des réseaux de communication ; à chaque niveau, la probabilité p pour une station de participer, ou non, aux essais de transmission dépend du nombre n des stations non éliminée par le processus de sélection aléatoire ; $p = 1/n$. Le temps d'exécution moyen est de l'ordre de $O(1)$ mais il est nécessaire de connaître le nombre des stations *actives*, ce qui contredit le caractère distribué du réseau. Toutefois, il est possible d'estimer ce nombre, voir [108] pour une estimation de l'ordre de $\log \log n + O(1/n)$.

Cette variante de l'algorithme d'élection de *leader* ne présente pas de comportement oscillatoire. En effet, le *coût* moyen de l'algorithme est asymptotiquement équivalent à une certaine constante L . Pour plus de détails, voir [63].

Schéma de compression de données LZ'77

Considérons une variante de l'algorithme d'élection de *leader* obtenue en introduisant un modérateur qui détermine le processus d'élimination ; chacun des participants, ainsi que le modérateur, tire d'une manière indépendante à pile ou face et seuls ceux qui obtiennent le même résultat que le modérateur continuent le processus. Voir [106] pour le cas biaisé, [90] pour celui non biaisé.

Soit M_n le nombre de participants restants dans le dernier tour non trivial d'un groupe initial de n éléments. Il est asymptotiquement équivalent à la multiplicité des phrases dans le schéma de compression de données LZ'77. Cet algorithme présente un comportement oscillatoire similaire à celui des algorithmes en arbre étudiés dans [81].

3.2 Résultats

Dans le premier chapitre, une approche directe des algorithmes en arbre basée sur une reformulation probabiliste de l'équation fonctionnelle associée est proposée. Le but de ce chapitre est d'adapter les techniques utilisées par Mohamed et Robert dans [81] et ce afin d'analyser une quantité additive dans un contexte de structure d'arbre incomplet.

Dans la section 3.3, une formule *en série* similaire du *coût* moyen de l'algorithme $E(H_n)$ est donnée.

Théorème 3 (Représentation Probabiliste du *coût* moyen). *Pour $n \geq 2$,*

$$\mathbb{E}(H_n) = \mathbb{E} \left(\sum_{i=0}^{\tau(U_{1,n}, U_{2,n})-1} \frac{1}{\pi_i} \right),$$

où,

- pour $0 < x < y < 1$,

$$\begin{aligned}\tau(x, y) &= \min(\nu(x); \mu(y)), \text{ avec} \\ \nu(x) &= \inf \{i \geq 1 : \alpha_i > x\}, \\ \mu(y) &= \inf \{i \geq 1 : \alpha_i + \pi_i < y\},\end{aligned}$$

- $(\alpha_k)_{k \geq 0}$ et $(\alpha_k + \pi_k)_{k \geq 0}$ sont deux suites de variables aléatoires dans $[0, 1]$;

$$\begin{aligned}\pi_0 &= 1 \text{ et, pour } i \geq 1, \pi_i = \prod_{j=0}^{i-1} A_j \\ \alpha_0 &= 0 \text{ et, pour } i \geq 1, \alpha_i = \sum_{j=0}^{i-1} \pi_j B_j\end{aligned}$$

- $(A_j, B_j)_{j \in \mathbb{N}}$ sont des réalisations i.i.d. du couple de variables aléatoires (A, B) de distribution

$$\mathbb{P}(A = p, B = 0) = p, \quad \mathbb{P}(A = q, B = p) = q,$$

- et $U_{i,n}$ est la i ème plus petite variable aléatoire parmi n indépendantes, uniformément distribuées dans $[0, 1]$ indépendantes de la suite (A_j, B_j) .

Dans la section 3.4, le comportement asymptotique de l'algorithme est reformulé en fonction de celui du temps d'arrêt τ .

Théorème 4 (Comportement asymptotique du coût moyen de l'algorithme). *Le coût moyen de l'algorithme $\mathbb{E}(H_n)$ admet l'équivalent asymptotique suivant*

$$\mathbb{E}(H_n) = -\log_p(n) + \mathbb{E}(\lceil \log_p(t_2) \rceil) + F(\log_p(n)) + \mathcal{R}(n),$$

où t_2 est la somme de 2 variables aléatoires exponentielles i.i.d. de paramètre 1, F est une fonction périodique définie pour tout $z > 0$ par

$$F(z) = \int_0^\infty y(1 - p^{1 - \{\log_p y - z\}}) \left(\lceil \log_p \left(\frac{1 - p^{1 - \{\log_p y - z\}}}{1 - p} \right) + \log_p y - z \rceil - \lfloor \log_p y - z \rfloor \right) e^{-y} dy,$$

$\{z\} = z - \lfloor z \rfloor$ est la partie fractionnaire de z et $\mathcal{R}(n)$ est un reste étudié dans la section 3.4.3.

Dans la section 3.5, la distribution du coût de l'algorithme est étudiée. Utilisant la décomposition binaire de l'intervalle $[0, 1]$, l'expression exacte de la distribution de H_n est établie.

Proposition 16.

$$\mathbb{P}(H_n \leq k) = n \int_0^1 (1 - t)^{n-1} d\mu_k(t),$$

où la mesure de probabilité μ_k est décrite dans la section (3.5). De plus, quand n tend vers l'infini, on a l'équivalence suivante

$$\mathbb{P}(H_n > k) \sim \int_0^1 \mathbb{P}(U_{2,n} < t) d\mu_k(t).$$

Ce résultat donne une expression légèrement différente de la formule asymptotique obtenue dans [53] pour la distribution du coût de l'algorithme H_n dans le cas asymétrique.

3.3 Coût Moyen de l'Algorithme

3.3.1 Coût moyen

Le *coût* de l'algorithme est le nombre d'étapes nécessaires pour déterminer le *leader*, ou d'une manière équivalente la *hauteur* de l'arbre associé, ou encore la *profondeur maximale* atteinte dans l'arbre. Notons par H_n (resp. P_n) ces quantités quand le groupe initial est de taille n , alors ces deux variables aléatoires vérifient la *même* relation de récurrence et on a, pour $n \geq 0$, $H_n = 1 + P_n$, et pour $n \geq 2$,

$$P_n \stackrel{dist.}{=} 1 + P_{1,S_n} \mathbf{1}_{\{S_n \neq 0\}} + P_{2,n} \mathbf{1}_{\{S_n = 0\}},$$

avec $P_0 = P_1 = 0$, et

$$S_n = \sum_{i=1}^n B_i(p).$$

où $(B_i(p))_{1 \leq i \leq n}$ désignent n variables aléatoires indépendantes de *Bernoulli* de paramètre p . De plus,

- pour $(m, n) \in \mathbb{N}^2$, $P_{1,m}$ et $P_{2,n}$ sont indépendantes.
- pour $i = 1, 2$, la variable $P_{i,m}$ a la même distribution que P_m .

Alors, pour $n \geq 0$, la relation de récurrence vérifiée par la suite (P_n) peut être réécrite sous la forme suivante

$$P_n \stackrel{dist.}{=} 1 + P_{S_n} + P_n \mathbf{1}_{\{S_n = 0\}} - \mathbf{1}_{\{n \leq 1\}}. \quad (3.2)$$

Remarque 10. *Le choix des conditions aux bords $H_0 = H_1 = 1$ est cohérent avec la définition du coût de l'algorithme d'élection de leader en tant que hauteur de l'arbre associé, où encore la longueur du chemin le plus long dans l'arbre. Cette convention est différente de celle utilisée par [53], mais le comportement asymptotique de l'algorithme n'en dépend pas.*

Remarque 11. *Remarquons que $\forall n \in \mathbb{N}$, on a $P_n \leq R_n$, taille totale de l'arbre, d'où l'intégrabilité de la variable aléatoire P_n et la bonne définition de sa transformée de Poisson sur R_+ .*

3.3.2 Poissonisation

Considérons la modèle *poissonien*, *i.e.* la taille du groupe initial est aléatoire notée $\mathcal{N}([0, x])$ suivant une loi de *Poisson* d'intensité $x \geq 0$. La proposition suivante donne une représentation probabiliste de la *transformée de Poisson* de la suite des profondeurs moyennes (P_n) .

Proposition 17. *Pour $x > 0$,*

$$\mathbb{E}(P_{\mathcal{N}([0, x])}) = \mathbb{E} \left(\sum_{i=0}^{+\infty} \frac{1}{\pi_i} \mathbf{1}_{\{t_1 > x\pi_i; t_2 \leq x(\alpha_i + \pi_i)\}} \right),$$

où, si (A_j, B_j) est une suite de réalisations *i.i.d.* du couple des variables aléatoires (A, B) dont la distribution est donnée par

$$\mathbb{P}(A = p, B = 0) = p, \quad \mathbb{P}(A = q, B = p) = q.$$

alors les suites $(\pi_i)_{i \geq 0}$ et $(\alpha_i)_{i \geq 0}$ sont définies pour $i \geq 1$ par

$$\pi_i = \prod_{j=0}^{i-1} A_j, \quad \alpha_i = \sum_{j=0}^{i-1} \pi_j B_j, \quad (3.3)$$

avec $\pi_0 = 1$ et $\alpha_0 = 0$.

Démonstration. Soit h la *transformée de Poisson* des profondeurs moyennes

$$h(x) = \mathbb{E}(P_{\mathcal{N}([0, x])}).$$

Alors, la relation de récurrence (3.2) pour la suite $(P_n)_{n \geq 0}$ devient

$$h(x) = h(px) + h(qx) e^{-px} + 1 - (1+x)e^{-x}.$$

Suivant l'approche de [81], un schéma itératif direct devient possible utilisant une formulation probabiliste de la dernière équation comme suit

$$h(x) = h(px) + h(qx) e^{-px} + f(x) = \mathbb{E} \left(\frac{h(Ax)}{A} e^{-Bx} \right) + f(x), \quad (3.4)$$

où (A, B) est un couple de variables aléatoires de distribution

$$\mathbb{P}(A = p, B = 0) = p, \quad \mathbb{P}(A = q, B = p) = q,$$

et

$$f(x) = (1 - (1+x)e^{-x}).$$

On définit la suite *i.i.d* des réalisations $(A_i, B_i)_{i \in \mathbb{N}}$ du couple (A, B) . On note aussi, pour $n \in \mathbb{N}$ et $x \geq 0$,

$$\begin{aligned} X_{n+1} &= A_n X_n, \\ Y_{n+1} &= B_n X_n, \end{aligned}$$

avec les conditions initiales $X_0 = x$, $Y_0 = 0$.

Par itérations de l'équation (3.4), on obtient au $(n+1)^{ieme}$ niveau

$$h(x) = \mathbb{E} \left(\frac{h(X_{n+1})}{\prod_{i=0}^n A_i} e^{-\sum_{i=0}^{n+1} Y_i} \right) + \mathbb{E} \left(\sum_{i=0}^n e^{-\sum_{j=0}^i Y_j} \frac{f(X_i)}{\prod_{j=0}^{i-1} A_j} \right).$$

Comme, presque sûrement, quand $n \rightarrow +\infty$

$$X_{n+1} = x \left(\prod_{i=0}^n A_i \right) \rightarrow 0,$$

et $h'(0) = 0$, alors

$$h(x) = \mathbb{E} \left(\sum_{i=0}^{+\infty} \frac{1}{\pi_i} (1 - (1 + \pi_i x) e^{-\pi_i x}) e^{-\alpha_i x} \right),$$

où les suites $(\pi_i)_{i \geq 0}$ et $(\alpha_i)_{i \geq 0}$ sont définies par (3.3).

Comme les suites $(\alpha_i)_{i \geq 0}$ et $(\alpha_i + \pi_i)_{i \geq 0}$ sont, presque sûrement, dans l'intervalle $[0, 1]$, la fonction h peut être représentée comme suit

$$h(x) = \mathbb{E} \left(\sum_{i=0}^{+\infty} \frac{1}{\pi_i} \mathbf{1}_{\{t_1 > \alpha_i x; t_2 < (\alpha_i + \pi_i) x\}} \right). \quad (3.5)$$

La proposition est ainsi prouvée. \square

Tout le long de ce chapitre, on conservera les notations introduites dans cette preuve.

3.3.3 Dépoissonisation

L'étape suivante est la dépoissonisation probabiliste de (3.5) suivant la méthode de [93] et ce afin d'obtenir l'expression exacte du coût moyen de l'algorithme $\mathbb{E}(H_n)$.

Proposition 18. *Pour $n \geq 2$,*

$$\mathbb{E}(H_n) = 1 + \mathbb{E} \left(\sum_{i=0}^{\tau(U_{1,n}, U_{2,n})-1} \frac{1}{\pi_i} \right),$$

où, pour $0 < x < y < 1$,

$$\begin{aligned} \tau(x, y) &= \min(\nu(x); \mu(y)), \\ \nu(x) &= \inf \{i \geq 1 : \alpha_i > x\}, \\ \mu(y) &= \inf \{i \geq 1 : \alpha_i + \pi_i < y\}, \end{aligned}$$

et $U_{i,n}$ est la i -ème plus petite variable parmi n variables aléatoires indépendantes, uniformément distribuées sur $[0, 1]$, indépendantes de la suite $(A_j, B_j)_{j \geq 0}$.

Démonstration. Pour $x > 0$, en décomposant par rapport au nombre de points ($\mathcal{N}([0, x])$) dans l'intervalle $[0, x]$ d'un processus de *Poisson* d'intensité 1, on obtient que, pour $0 < a < b < 1$,

$$\begin{aligned} \mathbb{P}(t_1 > ax, t_2 < bx) &= \sum_{n=2}^{+\infty} \mathbb{P}(t_1 > ax, t_2 < bx, \mathcal{N}([0, x]) = n) \\ &= \sum_{n=2}^{+\infty} \mathbb{P}(t_1 > ax, t_2 < bx | \mathcal{N}([0, x]) = n) \mathbb{P}(\mathcal{N}([0, x]) = n) \end{aligned}$$

Pour $n \geq 2$, sachant l'évènement $\{\mathcal{N}([0, x]) = n\}$, le couple de variables (t_1, t_2) a la même distribution que le couple $(xU_{1,n}, xU_{2,n})$ des deux plus petites variables aléatoires parmi n indépendantes uniformément distribuées sur $[0, x]$. Alors, on obtient l'identité suivante

$$\begin{aligned} \mathbb{P}(t_1 > ax, t_2 < bx) &= \sum_{n=2}^{+\infty} \mathbb{P}(U_{1,n} > a, U_{2,n} < b) \frac{x^n}{n!} e^{-x} \\ &= \mathbb{E} \left(\sum_{n=2}^{+\infty} \mathbf{1}_{\{U_{1,n} > a, U_{2,n} < b\}} \frac{x^n}{n!} e^{-x} \right). \end{aligned}$$

L'indépendance de la suite (A_i, B_i) et du couple (t_1, t_2) implique que

$$\mathbb{E}(P_{\mathcal{N}([0, x])}) = \mathbb{E} \left(\sum_{i=0}^{+\infty} \frac{1}{\pi_i} \sum_{n=2}^{+\infty} \mathbf{1}_{\{U_{1,n} > \pi_i, U_{2,n} < (\alpha_i + \pi_i)\}} \frac{x^n}{n!} e^{-x} \right).$$

Utilisant le théorème de Fubini, cette expression peut être réécrite ainsi

$$\sum_{n=2}^{+\infty} \left(\mathbb{E} \left(\sum_{i=0}^{+\infty} \frac{1}{\pi_i} \mathbf{1}_{\{U_{1,n} > \pi_i, U_{2,n} < (\alpha_i + \pi_i)\}} \right) \right) \frac{x^n}{n!} e^{-x}.$$

L'identification de la représentation de la *transformée de Poisson* (voir la Définition 6) $\mathbb{E}(P_{\mathcal{N}([0, x])})$ avec l'identité précédente implique que, pour $n \geq 2$

$$E(P_n) = \mathbb{E} \left(\sum_{i=0}^{+\infty} \frac{1}{\pi_i} \mathbf{1}_{\{U_{1,n} > \pi_i, U_{2,n} < (\alpha_i + \pi_i)\}} \right).$$

Comme, presque sûrement, la suite $(\alpha_i)_{i \geq 0}$ est croissante vers une variable aléatoire notée $\alpha \in [0, 1]$ et que, d'autre part, la suite $(\alpha_i + \pi_i)_{i \geq 0}$ est décroissante vers la même limite α , l'égalité suivante est établie

$$\{i \geq 0 : U_{1,n} > \pi_i, U_{2,n} < \alpha_i + \pi_i\} = [0, \tau(U_{1,n}, U_{2,n}) - 1],$$

où, pour $0 < x < y < 1$,

$$\begin{aligned} \tau(x, y) &= \min(\nu(x); \mu(y)), \\ \nu(x) &= \inf \{i \geq 1 : \alpha_i > x\}, \\ \mu(y) &= \inf \{i \geq 1 : \alpha_i + \pi_i < y\}. \end{aligned}$$

□

Remarque 12. Dans le cas non biaisé, le couple (A, B) est donné par $1/2(1, \delta_{1/2})$, où δ désigne la mesure de Dirac. Ainsi,

$$\pi_n = \frac{1}{2^n}, \quad \alpha_n = \frac{1}{2} \sum_{i=0}^{n-1} \frac{1}{2^i} \delta_{1/2},$$

et donc la limite α vérifie $\mathbb{E}(\alpha) = 1/2$.

3.4 Analyse Asymptotique du Coût Moyen

On a l'identité presque-sûre suivante

$$\mathbb{E} \left(\sum_{i=0}^{\tau(U_{1,n}, U_{2,n})-1} \frac{1}{\pi_i} | U_{1,n}, U_{2,n} \right) = \Phi(U_{1,n}, U_{2,n}),$$

où

$$\Phi(x, y) = \mathbb{E} \left(\sum_{i=0}^{\tau(x,y)-1} \frac{1}{\pi_i} \right).$$

3.4.1 Deux suites aléatoires, un temps d'arrêt

Il est clair que la clé de l'analyse du comportement asymptotique de l'algorithme est le temps d'arrêt τ défini par rapport aux deux suites aléatoires $(\alpha_i)_{i \geq 0}$ et $(\alpha_i + \pi_i)_{i \geq 0}$.

Soit (γ_i) la suite des variables aléatoires définie par

$$(\gamma_i)_{i \in \mathbb{N}} = \{j \geq 0 : B_j = p\}.$$

Ce sont les instants de sauts de la suite aléatoire $(\alpha_i)_{i \geq 0}$ et, inversement, ceux d'arrêt de la suite $(\pi_i + \alpha_i)_{i \geq 0}$. Il est simple à vérifier que ces instants peuvent être définis récursivement comme suit : $\gamma_0 = G_0$ et $\gamma_{n+1} = 1 + \gamma_n + G_{n+1}$, où $(G_n)_{n \geq 0}$ est une suite *i.i.d* de variables aléatoires de loi géométrique de paramètre q notée $Geo(q)$ telle que, pour $k \in \mathbb{N}$, on a

$$Geo(q)(k) = q p^k.$$

Pour tout entier $i \geq 0$, on a

$$\alpha_{1+\gamma_i} = p \sum_{j=0}^i \pi_{\gamma_j}.$$

Alors, il est facile de voir que

$$\begin{aligned} \nu(x) &\in \{1 + \gamma_i : i \in \mathbb{N}\}, \\ \mu(y) &\notin \{1 + \gamma_i : i \in \mathbb{N}\}. \end{aligned}$$

Utilisant une discussion sur la position du temps d'arrêt τ par rapport à la suite γ , on établit le lemme suivant dont la démonstration est donnée en B.

Lemme 3. La fonction $\phi : (x, y) \mapsto \mathbb{E} \left(\sum_{i=0}^{\tau(x,y)-1} 1/\pi_i \right)$ vérifie l'égalité suivante

$$\begin{aligned} \Phi(x, y) &= \lceil \log_p(y) \rceil + (\lceil \log_p(\rho(\log_p(y))y) \rceil - \lceil \log_p(y) \rceil) \mathbf{1}_{\Omega(x,y)} \\ &+ \mathbb{E} \left(\sum_{i=1+\lceil \log_p(\rho(\log_p(y))y) \rceil}^{\tau(x,y)-1} \frac{1}{\pi_i} \mathbf{1}_{\{\gamma_0=\lceil \log_p(y) \rceil; \gamma_1=\lceil \log_p(\rho(\log_p(y))y) \rceil\}} \right) \mathbf{1}_{\Omega(x,y)}. \end{aligned}$$

où

$$\begin{aligned} \Omega(x, y) &= \{(x, y) \in]0, 1]^2 : \lceil \log_p(y) \rceil = \lceil \log_p(x) \rceil\} \\ \rho(z) &= \frac{1 - p^{1-\{z\}}}{1 - p} \text{ est une fonction périodique d'amplitude 1.} \end{aligned}$$

Remarque 13. L'ensemble Ω peut être défini, d'une manière équivalente, par

$$\Omega(x, y) = \{(x, y) \in [0, 1]^2; p^{\lceil \log_p(y) \rceil} < x < y\}.$$

On en déduit que sa mesure vérifie

$$\begin{aligned} \mathbb{P}(\Omega(U_{1,n}, U_{2,n})) &= n(n-1) \int_0^1 \left(\int_p^{\lceil \log_p y \rceil} dx \right) (1-y)^{n-2} dy \\ &= n(n-1) \int_0^1 y(1-p^{1-\{\log_p(y)\}})(1-y)^{n-2} dy \\ &\sim g(\log_p(n)) \end{aligned}$$

quand n tend vers l'infini, où g est une fonction périodique définie par

$$g(z) = \int_0^\infty y(1-p^{1-\{\log_p(y)-z\}})e^{-y} dy.$$

3.4.2 Phénomène de fluctuations asymptotiques

Proposition 19. Le coût moyen $\mathbb{E}(H_n)$ admet la formule asymptotique suivante

$$\mathbb{E}(H_n) = -\log_p(n) + \mathbb{E}(\lceil \log_p(t_2) \rceil) + F(\log_p(n)) + \mathcal{R}(n),$$

où F est une fonction périodique définie pour tout $z > 0$ par

$$F(z) = \int_0^\infty y(1-p^{1-\{\log_p y - z\}}) \left(\lceil \log_p \left(\frac{1-p^{1-\{\log_p y - z\}}}{1-p} \right) + \log_p y - z \rceil - \lceil \log_p y - z \rceil \right) e^{-y} dy$$

et $\mathcal{R}(n)$ est un reste définie par

$$\mathcal{R}(n) = \mathbb{E} \left(\sum_{i=1+\lceil \log_p(\rho(\log_p(U_{2,n}))U_{2,n}) \rceil}^{\tau(U_{1,n}, U_{2,n})-1} \frac{1}{\pi_i} \mathbf{1}_{\{\gamma_0=\lceil \log_p(U_{2,n}) \rceil; \gamma_1=\lceil \log_p(\rho(U_{2,n})U_{2,n}) \rceil\}} \mathbf{1}_{\Omega(U_{1,n}, U_{2,n})} \right).$$

Démonstration. Utilisant le lemme 3, on a

$$\begin{aligned} \mathbb{E}(H_n) &= \mathbb{E}(\lceil \log_p(U_{2,n}) \rceil) + \mathbb{E}(\lceil \log_p(\rho(\log_p(U_{2,n}))U_{2,n}) \rceil - \lfloor \log_p(U_{2,n}) \rfloor) \mathbf{1}_{\Omega_n}) \\ &+ \mathbb{E} \left(\left(\sum_{i=1+\lceil \log_p(\rho(\log_p(y))y) \rceil}^{\tau(U_{1,n}, U_{2,n})-1} \frac{1}{\pi_i} \right) \mathbf{1}_{\{\gamma_0 = \lfloor \log_p(U_{2,n}) \rfloor; \gamma_1 = \lceil \log_p(\rho(U_{2,n})U_{2,n}) \rceil\}} \mathbf{1}_{\Omega_n} \right), \end{aligned}$$

où

$$\Omega_n = \Omega(U_{1,n}, U_{2,n}).$$

Les seuls termes non négligeables sont

$$\begin{aligned} \mathcal{T}_1(n) &= \mathbb{E}(\lceil \log_p(U_{2,n}) \rceil) \\ \mathcal{T}_2(n) &= \mathbb{E}(\lceil \log_p(\rho(\log_p(U_{2,n}))U_{2,n}) \rceil - \lfloor \log_p(U_{2,n}) \rfloor) \mathbf{1}_{\Omega_n}) \end{aligned}$$

Quand n tend vers l'infini, $nU_{2,n}$ converge en distribution vers une variable aléatoire t_2 , somme de deux variables *i.i.d.* exponentielles de paramètre 1. Alors, le premier terme satisfait l'égalité suivante

$$\mathcal{T}_1(n) = \mathbb{E}(\lceil \log_p(t_2) - \log_p(n) \rceil) + O\left(\frac{1}{n}\right).$$

Soit \mathcal{D} , fonction de $-\log_p(n)$, la différence

$$\mathcal{D}(-\log_p(n)) = \mathbb{E}(\lceil \log_p(t_2) - \log_p(n) \rceil) - (\mathbb{E}(\lceil \log_p(t_2) \rceil) - \log_p(n))$$

Il est simple à vérifier que

$$\mathcal{D}(z) = \mathcal{D}(\{z\}) - \lfloor z \rfloor,$$

donc

$$\lim_{n \rightarrow +\infty} n \mathcal{D}(-\log_p(n)) = \lim_{z \rightarrow +\infty} p^{-z} \mathcal{D}(z) = 0,$$

ainsi on obtient que

$$\mathcal{T}_1(n) = -\log_p(n) + \mathbb{E}(\lceil \log_p(t_2) \rceil) + O\left(\frac{1}{n}\right).$$

Le deuxième terme, qui s'écrit

$$\mathcal{T}_2(n) = n(n-1) \int_0^1 (\lceil \log_p(\rho(\log_p(y))y) \rceil - \lfloor \log_p y \rfloor) y(1-p^{1-\{\log_p(y)\}}) (1-y)^{n-2} dy,$$

est asymptotiquement équivalent, modulo $O(1/n)$, à $F(\log_p(n))$ où F est une fonction périodique définie pour $z > 0$ par

$$\begin{aligned} F(z) &= \\ &\int_0^\infty y(1-p^{1-\{\log_p(y)-z\}}) \left(\lceil \log_p\left(\frac{1-p^{1-\{\log_p(y)-z\}}}{1-p}\right) + \log_p(y) - z \rceil - \lfloor \log_p(y) - z \rfloor \right) e^{-y} dy. \end{aligned}$$

En effet,

$$\begin{aligned} |F(\log_p(n)) - \mathcal{T}_2(n)| &\leq \int_0^n \left| \left(1 - \frac{y}{n}\right)^{n-2} - e^{-y} \right| dy + \int_n^\infty \log_p(\rho(\log_p(y/n))) y e^{-y} dy \\ &\quad + \frac{1}{n} F(\log_p(n)) + 2e^{-n} \end{aligned}$$

Observons que

$$\int_n^\infty \log_p(\rho(\log_p(y/n))) y e^{-y} dy = n^2 \int_1^\infty \log_p(\rho(\log_p y)) y e^{-ny} dy.$$

Par décomposition sur la suite d'intervalles $([p^{k+1}, p^k])$, on obtient que la dernière intégrale est dominée par une somme géométrique et donc l'inégalité suivante, pour $n > 2$,

$$\int_1^\infty \log_p(\rho(\log_p(y))) y e^{-ny} dy \leq \frac{p^{n-2}}{1-p}.$$

Alors,

$$F(\log_p(n)) - \mathbb{E} \left(([\log_p(\rho(\log_p(U_{2,n}))U_{2,n})] - [\log_p(U_{2,n})]) \mathbf{1}_{\Omega_n} \right) = O\left(\frac{1}{n}\right).$$

□

3.4.3 Estimation du reste

L'étape finale est d'estimer le reste

$$\mathcal{R}(n) = \mathbb{E} \left(\left(\sum_{i=1+\lceil \log_p(\rho(\log_p(U_{2,n}))U_{2,n}) \rceil}^{\tau(U_{1,n}, U_{2,n})-1} \frac{1}{\pi_i} \mathbf{1}_{\{\gamma_0 = \lfloor \log_p(U_{2,n}) \rfloor; \gamma_1 = \lceil \log_p(\rho(U_{2,n})U_{2,n}) \rceil\}} \right) \mathbf{1}_{\Omega_n} \right).$$

De simples calculs impliquent que, pour $x, y \in [0, 1]$, et $K > k > 0$, on a

$$\mathbb{E} \left(\sum_{i=K}^{\tau(x,y)-1} \frac{1}{\pi_i} \mathbf{1}_{\{\gamma_0=k; \gamma_1=K\}} \right) \leq (1-\delta)^K \sqrt{\mathbb{E} \left(\left(\frac{1}{\delta^2} \right)^{\tau(x,y)} \right)},$$

où $\delta = \min(p, q)$.

Le résultat suivant sera admis.

Conjecture 1. *Le temps d'arrêt τ satisfait*

$$\sup_{x \in [0,1]} \mathbb{E} \left(\left(\frac{1}{\delta^2} \right)^{\tau(x,x)} \right) < \infty.$$

Remarque 14. La conjecture 1 est une restriction intuitive sur le moment exponentiel du temps d'arrêt τ . Elle est motivée par des simulations numériques (Fig.3.2,3.3) de $x \mapsto \mathbb{E} \left(\left(\frac{1}{\delta^2} \right)^{\tau(x,x)} \right)$ utilisant la méthode de Monte-Carlo.

Observons que, pour le cas biaisé (Fig.3.2), le maximum correspond aux valeurs numériques de x autour de 0.5, moyenne de la limite presque-sûre α des deux suites (α_i) et $(\alpha_i + \pi_i)$ lorsque i tend vers l'infini. Ce maximum est de l'ordre de 10^{14} , ce qui est raisonnable vu que

$$\mathbb{E}(\tau) \leq 14 \log_4(10) \approx 23.25$$

Pour le cas biaisé (Fig.3.3), $p = 0.2$, $p = 0.8$, comme $\delta = 0.2$, le fait que le maximum est de l'ordre de 10^{80} est aussi acceptable ; $\mathbb{E}(\tau) \leq 57.22$.

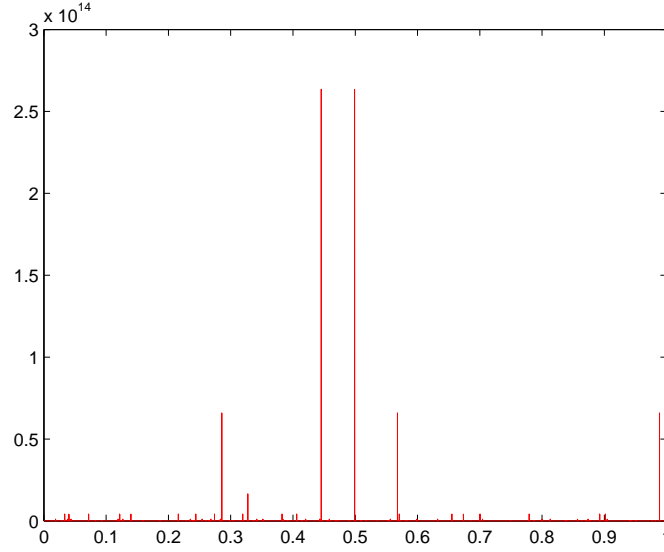


FIG. 3.2 – Cas non biaisé : simulations de $x \mapsto \mathbb{E} \left(4^{\tau(x,x)} \right)$.

Il est simple de voir que, pour $0 \leq x < y \leq 1$,

$$\tau(x, y) \leq \max(\tau(x, x), \tau(y, y)).$$

Alors, utilisant la conjecture 1, on obtient que

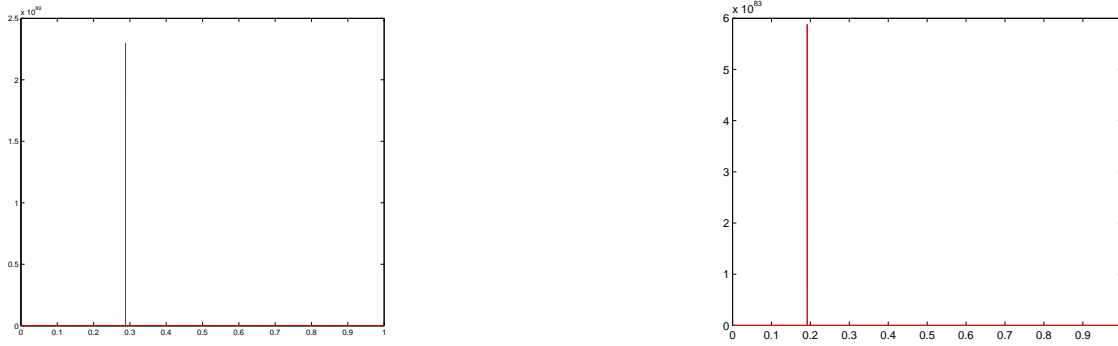
$$\mathcal{R}(n) \leq C \mathbb{E} \left((1 - \delta)^{\lceil \log_p(\rho(U_{2,n})U_{2,n}) \rceil} \right),$$

où

$$C = \sup_{x \in [0,1]} \sqrt{\mathbb{E} \left(\left(\frac{1}{\delta^2} \right)^{\tau(x,x)} \right)}.$$

Utilisant la même méthode que pour la fonction F , où n tend vers l'infini, on a

$$\mathbb{E} \left((1 - \delta)^{\lceil \log_p(\rho(U_{2,n})U_{2,n}) \rceil} \right) \leq \mathbb{E} \left((1 - \delta)^{\lceil \log_p(U_{2,n}) \rceil} \right) \sim \left(\frac{1}{n} \right)^{\log_p(1-\delta)}.$$


 FIG. 3.3 – Cas biaisé : $p = 0.2$, $p = 0.8$. Simulations de $x \mapsto \mathbb{E} \left(\left(\frac{1}{\delta^2} \right)^{\tau(x,x)} \right)$.

Ceci implique que

$$\mathcal{R}(n) = O\left(\frac{1}{n^{\log_p(1-\delta)}}\right).$$

On conclut que

$$\mathbb{E}(H_n) = -\log_p(n) + \mathbb{E}(\lceil \log_p(t_2) \rceil) + F(\log_p(n)) + O\left(\frac{1}{n^{\log_p(1-\delta)}}\right).$$

3.5 Distribution du Coût de l'Algorithme

Dans cette section, il sera plus approprié d'utiliser les notations suivantes ; $p_0 = p$, $p_1 = q$, pour définir récursivement la suite d'intervalles (I_k^n) associés à la décomposition binaire de l'intervalle $[0, 1]$ dans la base (p_0, p_1)

$$\begin{cases} I_0^0 &= [0, 1] \\ I_k^{n+1} &= (I_{k-1}^{n+1})_+ + p_{k-2\lfloor k/2 \rfloor} I_{\lfloor k/2 \rfloor}^n, \end{cases}$$

où $(I)_+$ désigne l'extrémité droite de l'intervalle I . Il est clair que

$$(I_k^n)_+ = \sum_{i=0}^k |I_i^n|,$$

où $|I|$ est la longueur de l'intervalle I .

Soit $n \in \mathbb{N}$ et $0 \leq k \leq 2^{n+1} - 1$. Considérons la décomposition binaire de k à l'ordre n

$$k = a_0 + a_1 2 + \dots + a_n 2^n.$$

Pour $0 \leq i \leq n$, $a_i \in \{0, 1\}$. Alors, la longueur de l'intervalle I_k^{n+1} est

$$|I_k^{n+1}| = \prod_{i=0}^n p_{a_i}.$$

Rappelons que H_n est la longueur du chemin reliant la racine de l'arbre au *leader*. Considérons le modèle *poissonien*; pour $k \in \mathbb{N}$ et $x > 0$, on obtient l'identité suivante

$$\{H_{\mathcal{N}([0,x])} > k\} = \{\exists i \in [0, 2^k - 1] : \mathcal{N}(xI_0^k) = \dots = \mathcal{N}(xI_{i-1}^k) = 0, \mathcal{N}(xI_i^k) \geq 2\}.$$

Alors,

$$\mathbb{P}(H_{\mathcal{N}([0,x])} \leq k) = e^{-x} + x \sum_{i=0}^{2^k-1} |I_i^k| e^{-(I_i^k)_+}. \quad (3.6)$$

On définit la suite des mesures de probabilité (μ_k) par

$$\mu_k(t) = \sum_{i=0}^{2^k-1} |I_i^k| \delta_{(I_i^k)_+}. \quad (3.7)$$

Alors, l'équation (3.6) peut être réécrite sous la forme suivante

$$\mathbb{P}(H_{\mathcal{N}([0,x])} \leq k) = e^{-x} + x \int_0^1 e^{-xt} d\mu_k(t). \quad (3.8)$$

Pour obtenir la distribution de H_n , on utilise une dépoissonisation probabiliste de cette dernière équation.

Proposition 20. *Pour $n \geq 2$,*

$$\mathbb{P}(H_n \leq k) = n \int_0^1 (1-t)^{n-1} d\mu_k(t),$$

où la mesure de probabilité μ_k est donnée par (3.7).

Démonstration. En décomposant l'équation (3.8) par rapport au nombre de points $(\mathcal{N}([0,x]))$ dans l'intervalle $[0,x]$ d'un processus de *Poisson* d'intensité 1, on obtient que

$$\begin{aligned} \mathbb{P}(H_{\mathcal{N}([0,x])} \leq k) &= e^{-x} \left(1 + x \int_0^1 e^{x(1-t)} d\mu_k(t) \right) \\ &= \sum_{n=0}^{+\infty} \mathbb{P}(H_n \leq k) \frac{x^n}{n!} e^{-x}, \end{aligned}$$

d'où

$$\mathbb{P}(H_{\mathcal{N}([0,x])} \leq k) = e^{-x} \left(1 + \sum_{n=0}^{+\infty} (n+1) \left(\int_0^1 (1-t)^n d\mu_k(t) \right) \frac{x^{n+1}}{(n+1)!} \right).$$

Par identification avec la définition de la *transformée de Poisson*, on obtient le résultat voulu. \square

En utilisant l'identité suivante

$$1 - n(1-t)^{n-1} = 1 - nt(1-t)^{n-1} - n(1-t)^n,$$

le corollaire suivant est immédiat.

Corollaire 4. Pour $k \in \mathbb{N}$, on a l'équivalence suivante quand n tend vers l'infini,

$$\mathbb{P}(H_n > k) \sim \int_0^1 \mathbb{P}(U_{2,n} < t) d\mu_k(t),$$

où $U_{2,n}$ est la deuxième plus petite variable aléatoire parmi n indépendantes uniformément distribuées sur $[0, 1]$.

3.6 Algorithme d'Élection de Leader : Version Markovienne

L'algorithme divise un groupe initial de n éléments en deux sous-groupes selon les poids p et q où $q = 1 - p$; autrement dit la probabilité qu'un élément soit mis dans le premier sous-groupe est p . Ce dernier est divisé à son tour en deux selon les poids $(p(0, 0), p(0, 1))$. Le second sous-groupe est ignoré, sauf si le premier est vide. Dans ce cas, le second sous-groupe est divisé par l'algorithme selon les poids $(p(1, 0), p(1, 1))$. Le processus de sélection aléatoire suit alors une chaîne de Markov $(X_k)_{k \in \mathbb{N}}$ dont la matrice de transition est donnée par

$$\begin{pmatrix} p(0, 0) & p(0, 1) \\ p(1, 0) & p(1, 1) \end{pmatrix}.$$

3.6.1 Définition du coût de l'algorithme

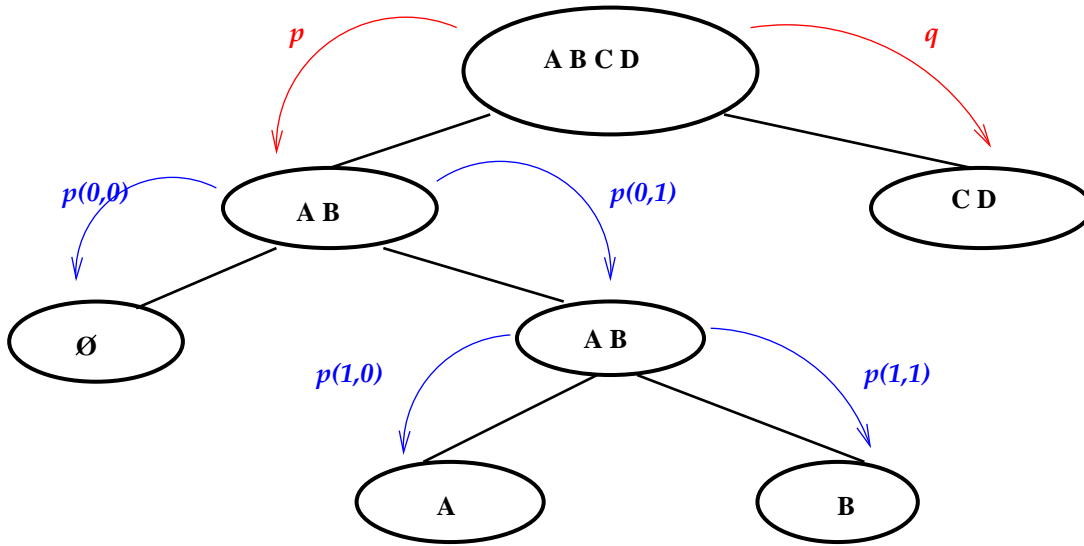


FIG. 3.4 – Élection du leader A ; $H_4^0 = 4$, $H_2^1 = 3$, $H_2^2 = 2$, $H_1^3 = 1$.

Supposons que, après k itérations, le nombre d'éléments non éliminés par l'algorithme est n . Notons alors H_n^k le nombre d'étapes nécessaires pour déterminer le leader partant de cette position. D'une part, il est clair que, pour tout entier $k \in \mathbb{N}$, on a $H_0^k = H_1^k = 1$.

D'autre part, à chaque niveau $k \geq 0$ de l'arbre, la relation de récurrence pour $n \geq 2$ s'écrit, pour la *hauteur* comme pour la *profondeur maximale*, sous la forme suivante :

$$P_n^k \stackrel{\text{dist.}}{=} 1 + P_{1, S_n(X_k)}^{k+1} \mathbf{1}_{(S_n(X_k) \neq 0)} + P_{2, n}^{k+1} \mathbf{1}_{(S_n(X_k) = 0)},$$

où

- si B désigne une variable aléatoire dont $(B_i)_{1 \leq i \leq n}$ sont n réalisations indépendantes, alors la variable aléatoire $S_n(B)$ est définie par

$$S_n(B) = \sum_{i=1}^n \mathbf{1}_{\{B_i=0\}},$$

- pour $(m, n, k) \in \mathbb{N}^3$, $P_{1, m}^k$ et $P_{2, n}^k$ sont indépendantes,
- pour $i \in \{1, 2\}$, la variable $P_{i, m}^k$ a la même distribution que P_m^k .

Alors, au $k^{\text{ième}}$ niveau, l'équation de récurrence pour la suite doublement indexée $(P_n^k)_{n, k \in \mathbb{N}}$ s'écrit

$$P_n^k \stackrel{\text{dist.}}{=} 1 + P_{S_n(X_k)}^{k+1} + P_n^{k+1} \mathbf{1}_{(S_n(X_k) = 0)} - \mathbf{1}_{(n \leq 1)}, \quad (3.9)$$

avec comme conditions initiales : $P_0^k = P_1^k = 0$.

3.6.2 Étude du coût moyen

Considérons un groupe de n éléments situé au $k^{\text{ième}}$ niveau de l'arbre associé à l'algorithme d'élection, $k \geq 1$. Rappelons l'équation (3.9)

$$P_n^k \stackrel{\text{dist.}}{=} 1 + P_{S_n(X_k)}^{k+1} + P_n^{k+1} \mathbf{1}_{(S_n(X_k) = 0)} - \mathbf{1}_{(n \leq 1)}.$$

Notons par $e_{n, k-1}$ le *passé commun* de ces n éléments situés au $k^{\text{ième}}$ niveau défini par leurs positions sur l'arbre durant les niveaux précédents, branche de gauche ou de droite.

On a

$$e_{n, k-1} = \prod_{i=1}^n X_{k-1}^i = \begin{cases} 1 & \text{si, pour tout indice } 1 \leq j \leq n, X_{k-1}^j = 1, \\ 0 & \text{sinon.} \end{cases}$$

Comme les n éléments ont le même passé, $e_{n, k-1} = X_{k-1}^j$ pour un indice arbitraire $j \in \{1, \dots, n\}$. On définit la suite de fonctions $(h_k)_{k \geq 0}$ par

$$h_k(x) = \mathbb{E}(P_{\mathcal{N}([0, x])}^k),$$

où $\mathcal{N}([0, x])$ désigne le nombre de points $(t_n)_{n \geq 1}$ dans l'intervalle $[0, x]$ d'un processus de *Poisson* d'intensité 1. Alors,

Proposition 21. *Pour $x > 0$,*

$$h_0(x) = \mathbb{E} \left(\sum_{j=0}^{+\infty} \frac{1}{\pi_j} \mathbf{1}_{(t_1 > x\pi_j; t_2 \leq x(\pi_j + \alpha_j))} \right), \quad (3.10)$$

où $(\pi_j)_{j \geq 0}$ et $(\alpha_j)_{j \geq 0}$ sont deux suites de variables aléatoires définies par (3.12).

Démonstration. Soit $k \geq 1$. Par Poissonisation de l'équation (3.9), on obtient que

$$h_k(x) = \mathbb{E} \left(P_{S_{\mathcal{N}([0,x])}(X_k)}^{k+1} + P_{\mathcal{N}([0,x])}^{k+1} \mathbf{1}_{(S_{\mathcal{N}([0,x])}(X_k)=0)} | X_{k-1}^1 = 0 \right) \mathbb{P}(X_{k-1} = 0) \\ + \mathbb{E} \left(P_{S_{\mathcal{N}([0,x])}(X_k)}^{k+1} + P_{\mathcal{N}([0,x])}^{k+1} \mathbf{1}_{(S_{\mathcal{N}([0,x])}(X_k)=0)} | X_{k-1}^1 = 1 \right) \mathbb{P}(X_{k-1} = 1) + \mathbb{P}(t_2 \geq x),$$

ou d'une manière équivalente une équation fonctionnelle

$$h_k(x) = (h_{k+1}(p(0,0) x) + h_{k+1}(p(0,1) x) e^{-p(0,0) x}) \mathbb{P}(X_{k-1} = 0) \\ + (h_{k+1}(p(1,0) x) + h_{k+1}(p(1,1) x) e^{-p(1,0) x}) \mathbb{P}(X_{k-1} = 1) + \mathbb{P}(t_2 \geq x),$$

qu'on transforme ainsi

$$h_k(x) = \mathbb{E} (h_{k+1}(p(X_{k-1}, 0) x) + h_{k+1}(p(X_{k-1}, 1) x) e^{-p(X_{k-1}, 0) x}) + \mathbb{P}(t_2 \geq x). \quad (3.11)$$

L'équation (3.11) est équivalente à

$$h_k(x) = \mathbb{E} \left(\frac{h_{k+1}(U^{X_{k-1}} x)}{A^{X_{k-1}}} e^{-B^{X_{k-1}} x} \right) + \mathbb{P}(t_2 \geq x)$$

où, sachant X_k , (A^{X_k}, B^{X_k}) est un couple de variables aléatoires de distribution

$$\mathbb{P}(A^{X_k} = p(X_k, 0), B^{X_k} = 0 | X_k) = p(X_k, 0), \mathbb{P}(A^{X_k} = p(X_k, 1), B^{X_k} = p(X_k, 0) | X_k) = p(X_k, 1).$$

Pour $k = 0$, *i.e.* à la racine de l'arbre associé à l'algorithme d'élection, on a

$$p = \mathbb{P}(X_0 = 0), \quad q = \mathbb{P}(X_0 = 1).$$

Donc

$$h_0(x) = \mathbb{E} \left(\frac{h_1(U x)}{A} e^{-B x} \right) + \mathbb{P}(t_2 \geq x)$$

où le couple (A, B) est défini par

$$P(A = p, B = 0) = p, \quad P(A = q, B = p) = q.$$

Utilisant ce schéma itératif, on a, pour $n \geq 0$,

$$h_0(x) = \mathbb{E} \left(\frac{h_n(\pi_n x)}{\pi_n} e^{-\alpha_n x} \right) + \mathbb{E} \left(\sum_{j=0}^{n-1} \frac{e^{-\alpha_j x}}{\pi_j} \mathbf{1}_{(t_2 \geq \pi_j x)} \right),$$

où, pour $i, j \in \{0, 1\}$ fixés, la suite $(A_k^i, B_k^i)_{k \geq 1}$ est i.i.d. de même distribution que le couple (A^i, B^j) tel que

$$\mathbb{P}(A^i = p(i, 0), B^i = 0) = p(i, 0), \quad \mathbb{P}(A^i = p(i, 1), B^i = p(i, 0)) = p(i, 1).$$

Les suites $(\pi_j)_{j \geq 0}$ et $(\alpha_j)_{j \geq 0}$ sont deux suites de variables aléatoires définies, pour $j \geq 1$, par

$$\pi_j = A \prod_{l=1}^{j-1} A_l^{X_l}, \quad \alpha_j = B + \sum_{l=1}^{j-1} B_l^{X_l} \pi_l, \quad (3.12)$$

avec $\pi_0 = 1$ et $\alpha_0 = 0$.

Comme la suite $(\pi_j)_{j \geq 0}$ converge presque sûrement vers 0, et en utilisant les conditions initiales $H_0^k = H_1^k = 0$ pour $k \in \mathbb{N}$ fixé, on obtient que

$$\lim_{n \rightarrow +\infty} \left(\frac{h_n(\pi_n x)}{\pi_n} \right) = 0$$

Ceci montre que la fonction h_0 peut être représentée sous la forme suivante

$$h_0(x) = \mathbb{E} \left(\sum_{j=0}^{+\infty} \frac{e^{-\alpha_j x}}{\pi_j} \mathbf{1}_{(t_2 \geq \pi_j x)} \right),$$

Les suites (π_j) et $(\pi_j + \alpha_j)$ sont presque sûrement dans l'intervalle $[0, 1]$, donc la fonction h_0 est donnée par

$$h_0(x) = \mathbb{E} \left(\sum_{j=0}^{+\infty} \frac{1}{\pi_j} \mathbf{1}_{(t_1 > \pi_j x; t_2 < (\pi_j + \alpha_j)x)} \right). \quad (3.13)$$

La proposition est ainsi prouvée. \square

La dépoissonisation de l'équation (3.13) est similaire à celle effectuée en (3.3.3). Ainsi, le résultat suivant est obtenu

Proposition 22. *Pour $n \geq 2$, alors*

$$\mathbb{E}(H_n^0) = 1 + \mathbb{E} \left(\sum_{i=0}^{\tau(U_{1,n}, U_{2,n})-1} \frac{1}{\pi_i} \right),$$

où, pour $0 < x < y < 1$,

$$\begin{aligned} \tau(x, y) &= \min(\tau(x); \tau(y)), \\ \tau(x) &= \inf \{i \geq 1 : \alpha_i > x\}, \\ \tau(y) &= \inf \{i \geq 0 : \pi_i + \alpha_i < y\}, \end{aligned}$$

et $U_{i,n}$ est la i ème plus petite variable aléatoire parmi n indépendantes, uniformément distribuées sur $[0, 1]$, indépendantes de la suite $(A_j^{X_j}, B_j^{X_j})_{j \geq 0}$.

3.6.3 Cas particulier : le cas indépendant

Dans le cas indépendant, la matrice de transition devient

$$\begin{pmatrix} p & q \\ p & q \end{pmatrix}$$

et la suite des variables aléatoires $(X_k)_{k \geq 0}$ a la même distribution que

$$p \delta_0 + q \delta_1.$$

Comme la suite des réalisations $(A^{X_k}, B^{X_k})_{k \geq 0}$ est *i.i.d.* et a la même distribution que le couple de variables aléatoires (A, B) de distribution

$$P(A = p, B = 0) = p, \mathbb{P}(A = q, B = p) = q,$$

et que $H_n^k \stackrel{dist.}{=} H_n^0 := H_n$, alors la Proposition (18) page 53 est immédiate.

Deuxième partie

Cas dynamique

Chapitre 4

Algorithme en arbre avec arrivées libres

Sommaire

4.1	Introduction	67
4.2	Formulation du Modèle	68
4.3	Itération d'Opérateurs Fonctionnels	69
4.3.1	Réurrences d'Arbre	70
4.3.2	Représentation Markovienne	70
4.4	Littérature	71
4.5	Objectifs et Résultats	72
4.5.1	Stabilité de l'algorithme en arbre	72
4.5.2	Analyse de récurrences d'arbre générales.	73
4.6	Existence d'une Région de Stabilité Non Vide	74

4.1 Introduction

Cette partie est dédiée à l'analyse d'une version *dynamique* d'algorithmes probabilistes qui décomposent, *avec ou sans perte*, récursivement un ensemble d'éléments en un nombre aléatoire de sous-ensembles jusqu'à ce que tous ces sous-ensembles aient un cardinal strictement inférieur à un nombre fixé D . Ces algorithmes sont largement utilisés en informatique ainsi que dans différents protocoles de communication sur des systèmes distribués. Voir Mohamed et Robert [81] pour plus de détails.

Pour un ensemble donné de cardinal $n \geq D$, un arbre est naturellement associé à un tel algorithme : la racine contient le groupe initial, les noeuds contiennent les autres sous-groupes et ainsi de suite À la fin de ce processus, l'arbre obtenu est caractérisé par des *feuilles* contenant moins de D éléments tandis que tous ses *nœuds internes* en contiennent au moins D . Cet algorithme qualifié de *statique* peut être vu comme étant un processus de branchement qui meurt dès qu'une certaine condition d'arrêt est satisfaite.

La version *dynamique* consiste à introduire une *immigration* dans le langage des processus de branchement : quand $n \geq D$, à chacun des sous-ensembles engendrés par la décomposition aléatoire s'ajoute un nombre aléatoire d'éléments *extérieurs* (nouvelles arrivées)

nécessitant le même traitement. L'algorithme en arbre est alors appliqué à chacun de ces sous-ensembles augmentés. Ce modèle est assez naturel pour les protocoles de communication vu que la condition $n \geq D$ correspond au fait qu'aucune des n tentatives de transmission n'a réussi durant l'unité de temps en cours, pendant laquelle de nouveaux messages sont arrivés à ce nœud de communication. Pour plus de détails sur l'utilisation d'un tel algorithme dans le contexte des réseaux de communication, voir Berger [5] et Massey [76].

ALGORITHME EN ARBRE DYNAMIQUE $\mathcal{D}(n)$.

— CONDITION D'ARRÊT.

Si $n < D$ \longrightarrow ARRÊT.

— STRUCTURE D'ARBRE.

Si $n \geq D$, diviser aléatoirement n en n_1, \dots, n_G , avec $n_1 + \dots + n_G \leq n$ (avec ou sans perte)

\longrightarrow APPLIQUER $\mathcal{D}(n_1 + A_1), \mathcal{D}(n_2 + A_2), \dots, \mathcal{D}(n_G + A_G)$ où (A_p)

sont des variables aléatoires *i.i.d.*

Ce modèle général de processus de décomposition a été introduit dans sa version conservatrice par Devroye [26]. Il est facile de voir que, pourvu que le mécanisme de décomposition ne soit pas dégénéré, l'algorithme *statique* s'arrête presque sûrement, et que le nombre total des nœuds de l'arbre est intégrable. Voir [81]. Pour la version *dynamique*, intuitivement, si le nombre des nouvelles arrivées ajoutées à chaque sous-ensemble est suffisamment large (supérieur ou égal à D avec probabilité 1 par exemple pour le cas conservatif), la probabilité que l'algorithme ne s'arrête jamais est alors égale à 1, ce qui signifie que l'arbre associé est presque sûrement infini. C'est une des questions étudiées dans cette partie. De plus, si le processus s'arrête presque sûrement, un autre problème se pose : décrire le comportement asymptotique de la taille moyenne de l'arbre engendré par n éléments quand n tend vers l'infini.

4.2 Formulation du Modèle

L'algorithme est appliqué à un ensemble de n éléments. Si $n < D$ alors il s'arrête. Sinon, l'ensemble est divisé aléatoirement en G sous-ensembles, où G est une variable aléatoire donnée. À cette étape, conditionnellement à l'événement $\{G = \ell\}$, pour $1 \leq i \leq \ell$, chacun des n éléments est envoyé, soit au i -ème sous-ensemble avec probabilité $V_{i,\ell}$, soit perdu avec probabilité $V_{0,\ell}$ où $V_\ell = (V_{i,\ell}; 0 \leq i \leq \ell)$ est un vecteur aléatoire de probabilité sur $\{0, \dots, \ell\}$.

Soit un vecteur (A_1, \dots, A_ℓ) de variables aléatoires indépendantes de même distribution qu'une variable aléatoire donnée A . Un nombre A_i d'arrivées nouvelles est ajouté au i -ème sous-ensemble. Si n_i est le cardinal du i -ème sous-ensemble, alors, conditionnellement à l'événement $\{G = \ell\}$ et aux variables aléatoires $V_{0,\ell}, V_{1,\ell}, \dots, V_{\ell,\ell}$, la distribution du vecteur (n_0, \dots, n_ℓ) est multinomiale de paramètres n et $(V_{0,\ell}, V_{1,\ell}, \dots, V_{\ell,\ell})$. Si le i -ème sous-ensemble, $1 \leq i \leq n$, est tel que $n_i + A_i < D$, l'algorithme s'arrête pour cet ensemble.

Sinon, il est appliqué au i -ème sous-ensemble : une variable G_i , de même distribution que G , est générée et ce i -ème ensemble est divisé en G_i sous-ensembles, et ainsi de suite.

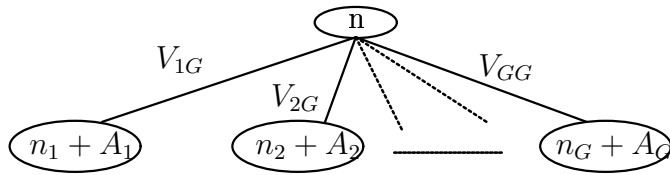


FIG. 4.1 – Premier niveau d'un algorithme en arbre dynamique décomposant $n \geq D$, $n_1 + \dots + n_G \leq n$ et avec arrivées (A_i)

L'algorithme Q -aire considéré par Mathys et Flajolet [77] correspond au cas conservatif où G est constante égale à Q et au vecteur de poids $(V_{i,Q}, 1 \leq i \leq Q)$ déterministe, $(V_{0,Q})$ étant presque sûrement nul.

4.3 Itération d'Opérateurs Fonctionnels

D'un point de vue mathématique, ces algorithmes dits *dynamiques* sont un vrai défi. Utilisant un schéma itératif, une famille d'opérateurs fonctionnels simples $(P_v^\lambda, v \in]0, 1[)$ joue un rôle central dans leur analyse. Ces opérateurs sont définis par

$$P_v^\lambda(f)(x) = \frac{1}{v^\varepsilon} f(xv + \lambda),$$

où $f: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ désigne une fonction continue. L'algorithme *statique* correspond alors au cas $\lambda = 0$, celui *sans perte* correspond à un taux de conservation $\varepsilon = 1$. Une quantité d'intérêt dans ce contexte est donnée par la série

$$\sum_{n=1}^{+\infty} \int_{[0,1]^n} P_{v_1}^\lambda \circ P_{v_2}^\lambda \circ \dots \circ P_{v_n}^\lambda(f)(x) \bigotimes_{i=1}^n \mathcal{W}(dv_i), \quad x \geq 0, \quad (4.1)$$

où \mathcal{W} est une distribution de probabilité donnée sur $[0, 1]$.

Il est facile de voir que les opérateurs $(P_v^\lambda, v \in [0, 1])$ commutent seulement quand $\lambda = 0$. Ceci complique significativement l'analyse du comportement asymptotique de l'expression (4.1) quand x tend vers l'infini. Dans le cas *statique*, on a

$$P_{v_1}^0 \circ P_{v_2}^0 \circ \dots \circ P_{v_n}^0 = P_{v_1 v_2 \dots v_n}^0$$

ce qui induit une représentation multiplicative de l'expression (4.1) pouvant être analysée par la transformée de Mellin dans un contexte analytique (voir Flajolet *et al.* [40]) ou en utilisant une méthode de marche aléatoire dans une approche probabiliste (voir Mohamed et Robert [81]). Quand $\lambda > 0$, une telle formulation multiplicative n'est plus valide, c'est pourquoi ces méthodes doivent être adaptées. En effet, une simple généralisation n'est pas envisageable.

4.3.1 Récurrences d'Arbre

Dans ce contexte, la quantité d'intérêt la plus importante est le nombre total de nœuds, noté R_n^A , dans l'arbre associé quand l'algorithme est appliqué à n éléments. Dans ce cas, quand $n \geq D$, avec les mêmes notations que précédemment, on obtient une relation de récurrence

$$R_n^A \stackrel{\text{dist.}}{=} 1 + (R_{1, n_1+A_1}^A + R_{2, n_2+A_2}^A + \dots + R_{G, n_G+A_G}^A) \mathbf{1}_{(n \geq D)}, \quad (4.2)$$

où, pour $i \geq 1$, $(R_{i,n}^A, n \geq 0)$ sont des variables aléatoires indépendantes de même distribution que $(R_n^A, n \geq 0)$.

Pourvu que les variables (R_n^A) soient intégrables, la transformée de *Poisson* $\phi(x)$ de la suite $(\mathbb{E}(R_n^A))$ est définie par

$$\phi(x) \stackrel{\text{def.}}{=} \sum_{n \geq 0} \mathbb{E}(R_n^A) \frac{x^n}{n!} e^{-x}. \quad (4.3)$$

En prenant l'espérance de l'équation (4.2) et en itérant l'équation fonctionnelle ainsi obtenue, on obtient une représentation de la transformée de *Poisson* (4.1) pour une certaine fonction f s'exprimant en termes de certaines constantes *inconnues*.

4.3.2 Représentation Markovienne

Cet algorithme peut être représenté par une chaîne de *Markov* (\underline{L}_n) défini sur l'ensemble $\mathcal{S} = \cup_{n \geq 0} \mathbb{N}^n$ des suites finies sur \mathbb{N} , ses transitions sont décrites comme suit : si $(\underline{L}_0) = (l_i, i \geq 0)$

$$(\underline{L}_1) = \begin{cases} (l_1 + A_1, l_2, \dots, l_n, \dots) & \text{si } l_0 < D \\ (n_1 + A_1, n_2, \dots, n_G, l_2, l_3, \dots, l_n, \dots) & \text{si } l_0 \geq D, \end{cases} \quad (4.4)$$

si l'entier $n = l_0$ est décomposé en (n_1, n_2, \dots, n_G) par le processus de décomposition aléatoire décrit ci-dessus.

Pour $n \geq 1$, si l'état initial est $\underline{L}_0 = (n, 0, \dots, 0, \dots)$, alors il est simple de vérifier que

$$R_n^A \stackrel{\text{dist.}}{=} \inf\{t \geq 1 : (\underline{L}_t = (0, \dots, 0, \dots))\},$$

la variable R_n^A est ainsi le temps d'atteinte de l'état $(0, \dots, 0, \dots)$ par la chaîne de *Markov* (\underline{L}_n) . L'ergodicité de cette chaîne de *Markov* est alors équivalente au fait que les variables $R_{A_1}^A$ soient intégrables. Dans le cas où les arrivées sont *poissonniennes* de paramètre λ , ceci revient à prouver que la transformée de *Poisson* de la suite $(\mathbb{E}(R_n^A))$ est bien définie en $x = \lambda$.

Exemple : Arbre Binaire. Le mécanisme de division est binaire, le degré de branchement G est déterministe égal à 2, $G \equiv 2$, avec des poids déterministes, $V_{11} \equiv p$ et $V_{12} \equiv q = 1 - p$ tels que $p \in (0, 1)$. Les variables (A_i) sont supposées *poissonniennes* de paramètre λ . L'exposant A dans (R_n^A) est remplacé par λ dans ce cas pour alléger les notations. Pourvu que les variables R_n^λ soient intégrables, et si $\alpha_n^\lambda = \mathbb{E}(R_n^\lambda)$, l'intégration de l'équation (4.2) implique les identités $\alpha_0^\lambda = \alpha_1^\lambda = 1$ et, pour $n \geq 2$,

$$\alpha_n^\lambda = 1 + \sum_{i=0}^n \binom{n}{i} p^i q^{n-i} \sum_{k, \ell \geq 0} \frac{\lambda^k}{k!} e^{-\lambda} \frac{\lambda^\ell}{\ell!} e^{-\lambda} (\alpha_{i+k}^\lambda + \alpha_{n-i+\ell}^\lambda). \quad (4.5)$$

Dans ce cas, il n'est pas difficile de vérifier que la transformée de *Poisson* ϕ satisfait l'équation

$$\phi(x) = \phi(px + \lambda) + \phi(qx + \lambda) + h(x), \quad (4.6)$$

où $x \rightarrow h(x)$ est une fonction donnée s'exprimant en terme de certains coefficients inconnus.

4.4 Littérature

Ces problèmes ont été analysés dans différents contextes dans le passé. Motivés par les protocoles de communication, Tsybakov et Mikhaïlov [101] ainsi que Capetanakis [16] ont effectués les premières études dans ce domaine (et, au même moment, proposé l'utilisation de ces algorithmes dans le contexte des systèmes distribués), voir aussi Tsybakov et Vvedenskaya [102].

Dans les années 80, Flajolet et ses co-auteurs dans une série d'articles [33, 35, 77] ont obtenu des équivalents asymptotiques rigoureux pour des solutions de l'équation de type (4.5) dans différents cas. Dans un premier article, Fayolle *et al.* [33] ont étudié la récurrence en arbre (4.5) pour l'arbre binaire : pour λ plus petit qu'un certain seuil critique, ils ont prouvé qu'il existe une unique suite de réels (α_n^λ) , solution de (4.5). L'analyse asymptotique de (α_n^λ) est faite en trois étapes :

1. En itérant cette équation fonctionnelle, exprimer la transformée de *Poisson* ϕ de (α_n^λ) sous la forme (4.1),

$$\phi(x) = \sum_{n \geq 0} \sum_{i \in \{1, 2\}^n} h(\sigma_{i_1} \circ \sigma_{i_2} \circ \dots \circ \sigma_{i_n}(x)), \quad (4.7)$$

où $\sigma_1(x) = px + \lambda$ et $\sigma_2(x) = qx + \lambda$.

2. Obtenir l'asymptotique de la transformée de *Poisson* $\phi(x)$ quand x tend vers l'infini à partir de celui de sa transformée de *Mellin* ;
3. Prouver que $\phi(x)$ et R_n^λ ont le même comportement asymptotique quand x (resp n) tend vers l'infini.

Une grande partie de [33] (pages 450-466), est consacrée à l'étape (2) où, par différentes estimations de séries et d'intégrales sur des contours ajoutées à des techniques d'analyse complexe, les auteurs arrivent à identifier dans la série (4.7) les termes qui contribuent

effectivement dans le comportement asymptotique de $\phi(x)$ quand x tend vers l'infini. Étant donné la complexité de la méthode appliquée à l'exemple de l'arbre binaire, une adaptation de la même approche que [33] pour des cas de mécanisme de branchement plus généraux semble délicate. Un sommaire de cette approche est présentée par Mathys et Flajolet dans la Section IV de [77] dans le cas où \mathcal{W} est uniforme. Notons que si \mathcal{W} admet une composante *Lebesgue*-mesurable, une représentation de ϕ sous la forme de série similaire à (4.7) n'est plus valable et on revient alors à la représentation générale (4.1).

Il y a encore un intérêt constant pour de telles questions, voir Boxma *et al.* [12], van Houdt et Blondia [104] ainsi que Velthoven *et al.* [105] par exemple. Pour plus de détails sur les protocoles de communication à accès aléatoire, voir Berger [5], Massey [76] ainsi que Ephremides et Hajek [30].

4.5 Objectifs et Résultats

L'objectif de cette partie est de présenter une approche probabiliste pour ces problèmes pouvant traiter, avec un minimum de techniques, des modèles d'arbres dynamiques assez généraux.

Pour le modèle décrit dans ce chapitre, l'équation (4.6) pour la transformée de *Poisson* est donnée par

$$\phi(x) = \int_0^1 \frac{\phi(wx + \lambda)}{w^\varepsilon} \mathcal{W}(dw) + h(x), \quad (4.8)$$

pour une fonction donnée h et une certaine distribution de probabilité \mathcal{W} sur $(0, 1)$ décrivant le processus de branchement généré par l'algorithme, le paramètre ε désigne le taux de conservation de l'algorithme. L'exemple binaire correspond au cas où \mathcal{W} est portée par deux points, p et $q = 1 - p$. Quand la mesure \mathcal{W} est portée par des masses de *Dirac* à k points de $[0, 1]$, l'expression équivalente pour la série (4.7) fait intervenir les différents produits possibles de k fonctions $(\sigma_m, 1 \leq m \leq k)$.

4.5.1 Stabilité de l'algorithme en arbre

La propriété de stabilité est cruciale pour les algorithmes en arbre utilisés dans le cas des systèmes distribués tels que les réseaux de communication à canal partagé. Admettant que les arrivées soient *poissonniennes* de paramètre λ , ceci revient à l'existence d'un certain seuil critique $\lambda_0 > 0$ tel que, si le taux des arrivées λ est strictement inférieur à λ_0 , alors le processus de *Markov* associé décrivant l'algorithme en arbre est ergodique. Durant les années 70, l'élaboration de protocoles stables ainsi que les preuves mathématiques de leur stabilité ont été des domaines de recherche très actifs. Voir Ephremides et Hajek [30]. Récemment, suite à l'émergence des réseaux mobiles et sans fils, il y a eu un regain d'intérêt pour ces modèles. Les premiers protocoles, Aloha et Ethernet, ont été prouvés instables, *i. e.* pour ces deux algorithmes, le nombre de demandes en attente de transmission tend vers l'infini en distribution quel que soit le taux d'arrivée > 0 . Voir Aldous [1]. L'algorithme correspondant à l'exemple binaire avec $p = q = 1/2$ est le premier protocole dans un système distribué qui admette une région de stabilité non vide, voir Massey [76] ainsi que

Bertsekas et Gallager [8]. Plus tard, l'algorithme en arbre a été amélioré pour atteindre des seuils de stabilité plus grands. La stabilité de l'algorithme en arbre correspond à l'ergodicité de la chaîne de *Markov* définie par les transitions (4.4).

Une lacune dans les preuves de stabilité

Les résultats de stabilité obtenus jusqu'à présent sont relatifs à des d'arrivées *poissonniennes*. Toutes les preuves, à notre connaissance, sont basées sur l'analyse d'une équation de type (4.5) pour $(\mathbb{E}(R_n^\lambda))$: il est alors prouvé qu'il existe un certain $\lambda_0 > 0$ tel que pour $\lambda < \lambda_0$, (4.5) admette une unique solution *finie* (α_n) et donc que la chaîne de *Markov* correspondante est ergodique. Le problème ici est que cette analyse admet implicitement que les variables (R_n^λ) sont *intégrables* (pour écrire l'équation (4.5)) et donc que *la chaîne de Markov est déjà ergodique*. Ainsi, les résultats précédents montrent seulement que le système est instable pour $\lambda \geq \lambda_0$. Même si l'équation (4.5) admet une unique solution finie, on ne peut exclure la possibilité que $\mathbb{E}(R_n^\lambda) = +\infty$ pour tout $n \geq 2$ et tout $\lambda > 0$, vu que, dans ce cas, (α_n) et $(1, 1, +\infty, \dots, +\infty, \dots)$ sont deux solutions de l'équation (4.5).

Cette lacune dans les preuves des résultats de stabilité est discutée dans ce chapitre. De plus, il est prouvé que, sous des hypothèses assez générales sur la distribution des arrivées (A_i) et sur le mécanisme de branchement, l'algorithme en arbre est stable pour un taux d'arrivée assez petit. À notre connaissance, c'est le premier résultat de stabilité sur ces algorithmes pour des arrivées non forcément *poissonniennes*.

4.5.2 Analyse de récurrences d'arbre générales.

Le deuxième chapitre de la partie *dynamique* de ce mémoire est dédié au *coût moyen de l'algorithme*, noté $\mathbb{E}(R_n^A)$, où (R_n^A) est solution de la récurrence (4.2) et où A est une variable aléatoire de *Poisson* de paramètre λ . Pour étudier ces récurrences, l'approche choisie consiste à exprimer la série (4.7) dans le cas d'un arbre binaire, ou (4.1) dans le cas général, comme l'espérance d'une variable aléatoire donnée dépendant d'un certain processus auto-régressif (X_n) défini par $X_0 = x$ et

$$X_{n+1} = W_n X_n + \lambda, \quad n \geq 0, \quad (4.9)$$

où (W_n) est une suite *i.i.d.* de distribution commune \mathcal{W} . L'étape suivante est d'identifier la fonction h admettant D coefficients inconnus. C'est là que le processus auto-régressif joue un rôle important vu qu'une équation linéaire cruciale est obtenue comme une fonctionnelle de sa distribution stationnaire. Dans le cas binaire, cette dernière étape est effectuée autrement dans [35], mais l'argument utilisé ne semble pas généralisable.

Plusieurs ingrédients utilisés pour l'analyse du cas *statique* (voir Mohamed et Robert [81]) sont repris ici : dépoissonisation probabiliste pour obtenir une expression explicite de $\mathbb{E}(R_n^\lambda)$, ainsi que le théorème de *Fubini*. L'équation correspondante est donnée par (4.8) où $\lambda = 0$. Notons que, dans ce cas, le comportement asymptotique de la solution de cette équation fonctionnelle peut être ainsi établi pour *toute* fonction h de classe C^1 (voir Section 4 de Robert [93] pour un exemple déterministe). La distribution invariante du processus auto-régressif (4.9) est la mesure de *Dirac* en 0.

La situation est complètement différente quand $\lambda > 0$, principalement à cause de l'introduction du processus auto-régressif (X_n) . Contrairement au cas $\lambda = 0$, l'expression explicite de h est importante pour obtenir une formule explicite pour $\mathbb{E}(R_n^\lambda)$ exprimant ainsi certaines quantités comme fonctionnelles du processus de *Poisson*. Ce point est crucial dans cette approche. Le fait que la distribution invariante de (X_n) ne soit pas connue complique la détermination de certaines constantes.

Hypothèses et Notations

Comme au premier chapitre pour le cas *statique*, la caractéristique clé de cet algorithme *dynamique* est la distribution de probabilité \mathcal{W} sur $[0, 1]$ définie en fonction de la distribution de branchement (la variable G) et les poids sur chaque chemin, le vecteur $(V_{1,G}, \dots, V_{G,G})$.

Définition 7. *La mesure de division normalisée est la distribution de probabilité \mathcal{W} sur $[0, 1]$ définie, pour toute fonction borélienne positive f , par*

$$\int f(x) \mathcal{W}(dx) = \mathbb{E} \left(\sum_{i=1}^G V_{i,G}^\epsilon f(V_{i,G}) \right) = \sum_{\ell=2}^{+\infty} \sum_{i=1}^{\ell} \mathbb{P}(G = \ell) \mathbb{E}(V_{i,\ell}^\epsilon f(V_{i,\ell})), \quad (4.10)$$

où $\epsilon \in]0, 1]$ désigne le taux de conservation de l'algorithme (voir le chapitre 3 de la première partie).

Dans le contexte des processus de fragmentation, cette mesure est appelée *mesure de dislocation*, voir Bertoin [6]. La mesure \mathcal{W} caractérise complètement le processus de division de l'algorithme en arbre.

Les conditions suivantes seront admises tout au long de la partie *dynamique* de ce mémoire.

Hypothèses

1. *Presque sûrement, $G \geq 2$, et il existe un réel $\delta > 0$ tel que la relation*

$$\sup_{\ell \geq 2} \sup_{1 \leq i \leq \ell} V_{i,\ell} \leq \delta < 1, \quad (\mathbf{H}_1)$$

en particulier $\mathcal{W}([0, \delta]) = 1$.

2. *\mathcal{W} vérifie*

$$\int_0^1 \frac{|\log x|}{x} \mathcal{W}(dx) < +\infty. \quad (\mathbf{H}_2)$$

La condition \mathbf{H}_1 implique en particulier la non dégénérescence du processus de division.

4.6 Existence d'une Région de Stabilité Non Vide

Cette section est une analyse basée, en un certain sens, sur une "perturbation" de l'algorithme en arbre *statique*. Donnons d'abord quelques définitions.

Définition 8. *Pour $n \geq 0$ et $p \geq 1$, on note par $R_n^{A,p}$, le nombre de nœuds situés entre la racine et le p -ième niveau de l'arbre dynamique engendré par un groupe initial de n éléments. A désigne une variable aléatoire de même distribution que la suite i.i.d. des*

arrivées (A_i) . Par convention, $R_n^{A,\infty} = R_n^A$ tandis que $(R_n^{0,p})$ correspond au cas statique, i.e. quand $A \equiv 0$. La variable G_A est définie par

$$G_A = \sum_{i=0}^{+\infty} \sum_{k=1}^{A_i} B_k^i,$$

où, pour $i \geq 0$, $(B_k^i, k \geq 0)$ est une suite i.i.d. de Bernoulli indépendante de A_i telle que $\mathbb{P}(B_k^i = 1) = \delta^i$, les suites $(B^i, i \geq 0)$ étant deux à deux indépendantes.

Notons que si $E(A) < +\infty$, la variable G_A est bien définie et intégrable. Le théorème suivant a été prouvé, dans le cas conservatif $\epsilon = 1$, dans Mohamed et Robert [81], (Théorème 3). Pour le cas avec ou sans perte, voir le chapitre 2 de la première partie de ce mémoire, (Propositions 14 et 15).

Théorème 5. *Sous les hypothèses \mathbf{H}_1 et \mathbf{H}_2 , il existe une constante $C_{\mathcal{W}}^0$ telle que*

$$\mathbb{E}(R_n^0) \leq n^\epsilon C_{\mathcal{W}}^0, \quad \forall n \geq 0.$$

Le lemme suivant établit une relation intéressante pour la taille d'un arbre, le symbole \leq_{st} désigne l'ordre stochastique classique.

Lemme 4 (Inégalité Stochastique). *Pour $n \in \mathbb{N}$,*

$$R_n^{A,p} \leq_{\text{st}} R_n^{0,p} + \sum_{k=1}^{R_n^{0,p}} \tilde{R}_{D-1+G_{A,i}}^{A,p} \mathbf{1}_{(G_{A,i}>0)}, \quad (4.11)$$

où $(\tilde{R}_n^{A,p})$ (resp $(G_{A,i})$) est une suite de variables aléatoires de même distribution que $(R_n^{A,p})$ (resp G_A). Les variables $(\tilde{R}_n^{A,p})$, $(R_n^{0,p})$ et $(G_{A,i})$ sont indépendantes.

Démonstration. Tout d'abord, notons que, pour $n \leq m$, on a clairement que $R_n^{A,p} \leq_{\text{st}} R_m^{A,p}$. Un couplage est utilisé pour prouver cet ordre stochastique. L'algorithme statique est d'abord appliqué au groupe initial des n éléments. Le nombre total de nœuds jusqu'à un niveau p dans l'arbre associé \mathcal{T}^0 est noté $R_n^{0,p}$. Ses feuilles ont au plus $D - 1$ éléments tandis que les nœuds internes en contiennent au moins D .

À cette étape, de nouvelles arrivées sont ajoutées aux nœuds internes et l'algorithme statique est de nouveau appliqué suivant la structure \mathcal{T}^0 jusqu'à ce que chacune de ces nouvelles arrivées arrive dans une feuille de l'arbre \mathcal{T}^0 . Là, pour chaque feuille ayant plus de D éléments, l'algorithme dynamique est appliqué partant de ce nœud. Ainsi, on a besoin d'estimer le nombre d'éléments externes à chaque feuille.

Notons que, suite à l'hypothèse \mathbf{H}_1 , un élément dans un nœud contenant déjà au moins D éléments est envoyé à l'un de ses nœuds-fils avec une probabilité au plus δ . Ainsi, une feuille donnée $L = I_i$ de \mathcal{T}^0 située à un niveau $i \leq p$ contient au plus $D-1$ éléments du groupe initial et une fraction des éléments nouveaux arrivés successivement par les nœuds internes I_2, I_3, \dots, I_{i-1} reliant cette feuille à la racine. Chacun des $A_{L,k-1}$ éléments

externes arrivés au nœud I_{i-1} est mis dans celui I_i avec probabilité au plus δ et donc, en distribution, il y a au plus

$$B_1^1 + B_2^1 + \cdots + B_{A_{L,i-1}}^1$$

éléments en I_i . D'une manière similaire, pour $2 \leq k \leq i-1$, les éléments externes arrivés au nœud I_k atteignent la *feuille* $L = I_i$ avec probabilité au plus δ^{i-k} . Par conséquent, le nombre total N_L d'éléments présents dans la *feuille* L est, en distribution, au plus

$$D - 1 + \sum_{i=1}^{k-2} \sum_{k=1}^{A_i} B_k^i.$$

Si N_L est supérieur ou égal à D , l'algorithme *dynamique* continue partant de ce nœud appliqué à un groupe de taille $N_L + A_0 \leq_{\text{st.}} D - 1 + G_A$. Notons que, pour que cet événement ait lieu, on devrait avoir des nouveaux éléments à cette *feuille* et ainsi que $G_A > 0$. Sinon, il n'y a que des éléments issus du groupe initial dans la *feuille* L et donc l'algorithme s'arrête.

En remarquant que le nombre de *feuilles* de \mathcal{T}^0 dont la *profondeur* est inférieure à p est majoré par $R_n^{0,p}$, alors on obtient la relation désirée

$$R_n^{A,p} \leq_{\text{st.}} R_n^{0,p} + \sum_{k=1}^{R_n^{0,p}} \tilde{R}_{D-1+G_{A,i}}^{A,p} \mathbf{1}_{(G_{A,i}>0)}.$$

□

Théorème 6 (Existence d'un système stable). *Sous les conditions \mathbf{H}_1 et \mathbf{H}_2 , et si A^η est une famille de variables aléatoires intégrables et à valeurs entières telle que*

$$\lim_{\eta \rightarrow 0} A^\eta = 0 \text{ en distribution et } \limsup_{\eta \rightarrow 0} \mathbb{E}(A^\eta \mid A^\eta > 0) < +\infty,$$

alors il existe un certain $\eta_0 > 0$ et une constante finie $C_{\mathcal{W}}^1$ tels que pour tout $\eta \leq \eta_0$, $R_n^{A^\eta}$ est intégrable et

$$\mathbb{E}(R_n^{A^\eta}) \leq n^\varepsilon C_{\mathcal{W}}^1, \quad \forall n \geq 0. \quad (4.12)$$

Notons que les conditions sur la famille (A^η) sont assez faibles vu qu'elles assurent la stabilité de l'algorithme pour η suffisamment petit.

Démonstration. Si F_{A^η} est une variable aléatoire de même distribution que $(G_{A^\eta} \mid G_{A^\eta} > 0)$,

$$\begin{aligned} \mathbb{E}(F_{A^\eta}) &= \frac{1}{\mathbb{P}(G_{A^\eta} > 0)} \mathbb{E}(G_{A^\eta} \mathbf{1}_{G_{A^\eta} > 0}) \\ &\leq \frac{\mathbb{E}(A^\eta)}{\mathbb{P}(A^\eta > 0)(1 - \delta)} = \mathbb{E}(A^\eta \mid A^\eta > 0) \frac{1}{1 - \delta}, \end{aligned}$$

l'hypothèse du théorème implique que $\mathbb{E}(F_{A^\eta})$ est borné par une certaine constante K quand η tend vers 0. Théorème 5 donne la relation $\mathbb{E}(R_{D-1+F_{A^\eta}}^0) \leq (D + K)^\varepsilon C_{\mathcal{W}}^0$.

La variable $R_n^{A^\eta, p}$ est clairement intégrable. La relation (4.11) implique l'inégalité

$$\mathbb{E} \left(R_n^{A^\eta, p} \right) \leq \mathbb{E} \left(R_n^{0, p} \right) + \mathbb{E} \left(R_n^{0, p} \right) \mathbb{P}(G_{A^\eta} > 0) \mathbb{E} \left(R_{D-1+F_{A^\eta}}^{A^\eta, p} \right), \quad n \geq 0, \quad (4.13)$$

et donc la relation suivante

$$\begin{aligned} \mathbb{E} \left(R_{D-1+F_{A^\eta}}^{A^\eta, p} \right) &\leq \mathbb{E} \left(R_{D-1+F_{A^\eta}}^{0, p} \right) \left(1 + \mathbb{P}(G_{A^\eta} > 0) \mathbb{E} \left(R_{D-1+F_{A^\eta}}^{A^\eta, p} \right) \right) \\ &\leq (D + K)^\varepsilon C_{\mathcal{W}}^0 \left(1 + \mathbb{P}(G_{A^\eta} > 0) \mathbb{E} \left(R_{D-1+F_{A^\eta}}^{A^\eta, p} \right) \right) \end{aligned}$$

a lieu. Il est facile de vérifier que les variables (G_{A^η}) convergent vers 0 en distribution quand η tend vers 0. Par conséquent, il existe un certain $\eta_0 > 0$ tel que, si $\eta \leq \eta_0$, alors $\mathbb{P}(G_{A^\eta} > 0)(D + K)^\varepsilon C_{\mathcal{W}}^0 < 1/2$, et donc que

$$\mathbb{E} \left(R_{D-1+F_{A^\eta}}^{A^\eta, p} \right) \leq 2(D + K)^\varepsilon C_{\mathcal{W}}^0,$$

en faisant tendre p vers l'infini, on obtient que $\mathbb{E} \left(R_{D-1+F_{A^\eta}}^{A^\eta} \right) \leq 2(D + K)C_{\mathcal{W}}^0$. En utilisant encore l'équation (4.13) et le théorème ci-dessus, cette dernière inégalité donne la relation suivante

$$\mathbb{E} \left(R_n^{A^\eta, p} \right) \leq n^\varepsilon C_{\mathcal{W}}^0 \left(1 + 2(D + K)^\varepsilon C_{\mathcal{W}}^0 \right),$$

ce qui conclut la démonstration du théorème. \square

Le théorème ainsi prouvé montre que, pour η suffisamment petit, les variables aléatoires $(R_n^{A^\eta})$ sont intégrables, en particulier que la chaîne de *Markov* définie par les transitions (4.4) est ergodique. Un cas particulier important est celui d'un algorithme en arbre sans perte avec des arrivées libres suivant une loi de *Poisson* de paramètre $\lambda > 0$. Le résultat suivant établit l'existence d'un taux d'arrivées critique $\lambda_{\mathcal{W}} > 0$ au delà duquel la probabilité que l'algorithme ne s'arrête jamais et non nulle.

Corollaire 5 (Région de Stabilité pour les Algorithmes en Arbre Conservatifs avec des Arrivées Poissonniennes). *Pour tout algorithme en arbre conservatif ($\varepsilon = 1$) avec des arrivées poissonniennes de paramètre λ dont le processus de branchement est défini par \mathcal{W} satisfaisant les hypothèses \mathbf{H}_1 et \mathbf{H}_2 , il existe $\lambda_{\mathcal{W}} > 0$ tel que, si $\lambda < \lambda_{\mathcal{W}}$, la transformée de Poisson de (R_n^λ) est définie sur \mathbb{R} , en particulier toutes les variables (R_n^λ) sont intégrables. Quand $\lambda \geq \lambda_{\mathcal{W}}$ alors $\mathbb{E}(R_n^\lambda) = +\infty$ pour tout $n \geq D$.*

Démonstration. Si A_1^λ est une variable aléatoire suivant une distribution de Poisson de paramètre λ , la famille (A_1^λ) satisfait clairement les hypothèses du théorème général de stabilité. Alors, il existe un certain $\lambda_0 > 0$ tel que $\mathbb{E}(R_n^\lambda) < +\infty$ pour tout $n \geq 0$.

D'une part, la somme des composantes de la chaîne de Markov définie par les transitions (4.4) décroît d'au plus $D - 1$ au cours d'une unité de temps. D'autre part, les nouvelles arrivées ont pour moyenne λ , donc $\mathbb{E}(R_n^\lambda) = +\infty$ pour tout $n \geq D$ si $\lambda > D - 1$. La quantité

$$\lambda_{\mathcal{W}} = \sup\{\lambda \geq 0 : \mathbb{E}(R_n^\lambda) < +\infty\}$$

est alors positive et finie. De plus, elle ne dépend pas de $n \geq D$. En effet, si $\mathbb{E}(R_n^\lambda) < +\infty$ et pour $m \geq D$, si $m \leq n$, clairement $\mathbb{E}(R_m^\lambda) < \mathbb{E}(R_n^\lambda)$. Si $m \geq n$, partant de l'équation (4.2)

on obtient que $R_n^\lambda \geq R_m^\lambda \mathbf{1}_{(n_1+A_1=m)}$, et donc que R_m^λ est intégrable. Pour $n \in \mathbb{N}$, un argument simple de couplage montre que la fonction : $\lambda \mapsto \mathbb{E}(R_n^\lambda)$ n'est pas décroissante. Ceci achève la démonstration du corollaire. \square

Chapitre 5

Arrivées Poissonniennes

Sommaire

5.1 Transformée de Poisson	79
5.1.1 Équation fonctionnelle basique	80
5.1.2 Processus auto-régressif	81
5.1.3 Coût moyen de l'algorithme	83
5.2 Condition de Stabilité : le cas conservatif $\varepsilon = 1$	84

Dans ce chapitre, le cas particulier d'arrivées A suivant une distribution de *Poisson* de paramètre λ est traité. Comme précédemment, on écrira R_n^λ au lieu de R_n^A .

5.1 Transformée de Poisson

La suite $\mathcal{N} = (t_n), 0 \leq t_1 \leq \dots \leq t_n \leq \dots$ désigne un processus de *Poisson* d'intensité 1 et, pour $x \geq 0$, $\mathcal{N}([0, x])$ est le nombre des t_n dans l'intervalle $[0, x]$:

$$\mathcal{N}([0, x]) = \sup\{n : t_n \leq x\}.$$

La *transformée de Poisson* ϕ_r de la suite $r = (r_n)$ est donnée par

$$\phi_r(x) = \sum_{n \geq 0} r_n \frac{x^n}{n!} e^{-x} = \mathbb{E} (r_{\mathcal{N}([0, x])}) .$$

Pourvu que cette fonction soit bien définie sur \mathbb{R}_+ ,

$$\phi_r'(x) = \sum_{n \geq 0} (r_{n+1} - r_n) \frac{x^n}{n!} e^{-x} = \phi_{\Delta r}(x),$$

où $\Delta r = (r_{n+1} - r_n, n \geq 0)$. Autrement dit, la *transformée de Poisson* commute avec la dérivation : la dérivée de la *transformée de Poisson* de (r_n) est la *transformée de Poisson* de la dérivée (discrète) de (r_n) . La relation suivante est facile à vérifier, pour $n \geq 0$ et $x \geq 0$,

$$\mathbb{E} (r_{n+\mathcal{N}([0, x])}) = \sum_{k=0}^n \binom{n}{k} \phi_r^{(k)}(x). \tag{5.1}$$

5.1.1 Équation fonctionnelle basique

La proposition suivante établit une équation fonctionnelle primordiale pour la *transformée de Poisson*. Pour alléger les notations, la *transformée de Poisson* de (R_n^λ) est notée ϕ_λ au lieu de ϕ_{R^λ} .

Proposition 23 (Transformée de Poisson). *Pourvu que λ soit assez petit, la transformée de Poisson $\phi_\lambda(x)$ de la suite (R_n^λ) satisfait la relation*

$$\phi_\lambda(x) = \mathbb{E} \left(\frac{1}{W_1^\varepsilon} \phi_\lambda(\lambda + W_1 x) \right) + 1 - \phi_C(x), \quad (5.2)$$

où W_1 est une variable aléatoire de distribution \mathcal{W} , $\varepsilon \in]0, 1]$ désigne le taux de conservation de l'algorithme et $C = (C_m)$ est une suite définie par $C_m = 0$ pour $m \geq D$ et, pour $0 \leq m < D$,

$$C_m = \sum_{k=0}^m \binom{m}{k} \mathbb{E} (W_1^{k-\varepsilon}) \phi_\lambda^{(k)}(\lambda).$$

Démonstration. D'après le Théorème 6 ainsi que les relations (4.12), il existe un certain λ_0 tel que ϕ_λ soit définie sur \mathbb{R}_+ quand $\lambda \leq \lambda_0$. En utilisant la propriété de décomposition d'un processus de *Poisson* et en incluant les conditions aux limites de l'équation (4.2), on obtient la relation

$$R_{\mathcal{N}([0,x])}^\lambda \stackrel{\text{dist.}}{=} 1 + \sum_{i=1}^G R_{i,\mathcal{N}([xS_{i-1,G}, xS_{i,G}]) + Z_i}^\lambda - \mathbf{1}_{(\mathcal{N}([0,x]) < D)} \sum_{i=1}^G R_{i,\mathcal{N}([xS_{i-1,G}, xS_{i,G}]) + Z_i}^\lambda, \quad (5.3)$$

où

- Pour $1 \leq i \leq G$, $S_{i,G}$ est la i -ième somme partielle des poids

$$S_{i,G} = V_{1,G} + V_{2,G} + \cdots + V_{i,G},$$

en particulier $S_{0,G} = 0$ et $S_{G,G} = 1 - V_{0,G}$ (*possibilité de perte*);

- les variables $R_{i,j}$, $i \geq 1$, $j \geq 0$ sont indépendantes et $R_{i,n}$ a la même distribution que R_n pour tout $n \geq 0$;
- (Z_i) est une suite i.i.d de variables de *Poisson* de paramètre λ .

Pour $k \geq 0$, la propriété d'homogénéité d'un processus de *Poisson* implique que

$$\begin{aligned} \mathbb{E} \left(\mathbf{1}_{(\mathcal{N}([0,x])=k)} \sum_{i=1}^G R_{i,\mathcal{N}([xS_{i-1,G}, xS_{i,G}]) + Z_i}^\lambda \right) &= \mathbb{E} \left(\mathbf{1}_{(\mathcal{N}([0,x])=k)} \sum_{i=1}^G R_{\mathcal{N}([0,xV_{i,G}]) + Z_i}^\lambda \right) \\ &= \mathbb{E} \left(\mathbf{1}_{(\mathcal{N}([0,x])=k)} \frac{R_{\mathcal{N}([0,xW_1]) + Z_1}^\lambda}{W_1^\varepsilon} \right) \\ &= \mathbb{E} \left(\frac{R_{\mathcal{N}([0,W_1]) + Z_1}^\lambda}{W_1^\varepsilon} \middle| \mathcal{N}([0,1]) = k \right) \frac{x^k}{k!} e^{-x}, \end{aligned}$$

où W_1 est une variable aléatoire de distribution \mathcal{W} . En appliquant l'espérance à l'équation (5.3), on obtient la relation suivante

$$\begin{aligned}\phi_\lambda(x) &= \mathbb{E} \left(\sum_{i=1}^G \phi_\lambda(\lambda + xV_{i,G}) \right) - \sum_{k=0}^{D-1} \mathbb{E} \left(\frac{R_{\mathcal{N}([0,W_1])+Z_1}^\lambda}{W_1^\varepsilon} \middle| \mathcal{N}([0,1]) = k \right) \frac{x^k}{k!} e^{-x}, \\ &= \mathbb{E} \left(\frac{1}{W_1^\varepsilon} \phi_\lambda(\lambda + xW_1) \right) - \sum_{k=0}^{D-1} \mathbb{E} \left(\frac{R_{\mathcal{N}([0,W_1])+Z_1}^\lambda}{W_1^\varepsilon} \middle| \mathcal{N}([0,1]) = k \right) \frac{x^k}{k!} e^{-x}.\end{aligned}$$

Pour $k \geq 0$, d'après l'équation (5.1)

$$\begin{aligned}\mathbb{E} \left(\frac{R_{\mathcal{N}([0,W_1])+Z_1}^\lambda}{W_1^\varepsilon} \middle| \mathcal{N}([0,1]) = k, W_1 \right) &= W_1^{-\varepsilon} \sum_{\ell=0}^k \binom{k}{\ell} W_1^\ell (1 - W_1)^{k-\ell} \mathbb{E} (R_{\ell+Z_1}^\lambda) \\ &= W_1^{-\varepsilon} \sum_{\ell=0}^k \sum_{m=0}^{\ell} \binom{k}{\ell} \binom{\ell}{m} W_1^\ell (1 - W_1)^{k-\ell} \phi_\lambda^{(m)}(\lambda) \\ &= \sum_{m=0}^k \binom{k}{m} W_1^{m-\varepsilon} \phi_\lambda^{(m)}(\lambda)\end{aligned}$$

La proposition est prouvée. \square

Remarque 15. Le cas conservatif, correspondant à $\varepsilon = 1$, obéit à l'équation fonctionnelle suivante

$$\phi_\lambda(x) = \mathbb{E} \left(\frac{1}{W_1} \phi_\lambda(\lambda + W_1 x) \right) + 1 - \phi_C(x).$$

où la suite C des constantes est donnée par la Proposition 23.

5.1.2 Processus auto-régressif

L'équation (5.2) suggère l'introduction d'une suite de variables aléatoires (X_n^x) définies par $X_0^x = x$ et, pour $n \geq 1$,

$$X_n^x = W_n X_{n-1}^x + \lambda, \quad (5.4)$$

où (W_n) est une suite *i.i.d.* de variables aléatoires de même distribution que W_1 . Les termes de la suite (X_n^x) peuvent être exprimés comme suit, pour $n \geq 0$,

$$X_n^x = x \prod_{i=1}^n W_i + \sum_{p=1}^n \lambda \prod_{i=p+1}^n W_i \stackrel{\text{dist.}}{=} \pi_n x + \lambda \sum_{p=0}^{n-1} \pi_p = \pi_n x + X_n^0, \quad (5.5)$$

où, pour $1 \leq k$, $\pi_k = \prod_{i=1}^k W_i$. Pour $x = 0$, l'indice en exposant est omis. La suite (X_n^x) converge en distribution vers une variable aléatoire X_∞ telle que

$$X_\infty \stackrel{\text{dist.}}{=} W_0 X_\infty + \lambda \text{ soit } X_\infty \stackrel{\text{dist.}}{=} \lambda \sum_{p=0}^{+\infty} \pi_p.$$

On note que la distribution de X_∞ n'est pas explicitement connue en général.

La dérivation de l'équation (5.2) par rapport à x implique que

$$\phi'_\lambda(x) = \mathbb{E} (W_1^{1-\varepsilon} \phi'_\lambda(X_1^x)) - \phi_{\Delta C}(x). \quad (5.6)$$

L'équation (5.6) exprime ϕ'_λ comme la solution d'une équation de *Poisson* associée à la chaîne de *Markov* (X_n) et à la fonction $\phi_{\Delta C}$. Notons que la fonction $\phi_{\Delta C}$ dépend de ϕ_λ à travers ses dérivées successives en λ .

Remarque 16. L'équation (5.6) implique que, pour $\varepsilon \in]0, 1]$, par invariance, on a

$$\mathbb{E} ((W^{1-\varepsilon} - 1) \phi'_\lambda(X_\infty)) = \mathbb{E}(\phi_{\Delta C}(X_\infty)), \quad (5.7)$$

en particulier, dans le cas conservatif $\varepsilon = 1$, que

$$\mathbb{E}(\phi_{\Delta C}(X_\infty)) = 0.$$

L'itération de l'équation (5.6) donne, pour $n \geq 1$,

$$\phi'_\lambda(x) = \mathbb{E} (\pi_n^{1-\varepsilon} \phi'_\lambda(X_n^x)) - \sum_{k=0}^{n-1} \mathbb{E} (\pi_k^{1-\varepsilon} \phi_{\Delta C}(\pi_k x + X_k)),$$

ainsi, par passage à la limite, on obtient la formule suivante

— **Cas conservatif :** $\varepsilon = 1$

$$\phi'_\lambda(x) = \mathbb{E} (\phi'_\lambda(X_\infty)) - \sum_{k=0}^{+\infty} \mathbb{E} (\phi_{\Delta C}(\pi_k x + X_k)). \quad (5.8)$$

— **Cas non conservatif :** $\varepsilon \in]0, 1[$

$$\phi'_\lambda(x) = - \sum_{k=0}^{+\infty} \mathbb{E} (\pi_k^{1-\varepsilon} \phi_{\Delta C}(\pi_k x + X_k)). \quad (5.9)$$

Notons que, pour $\varepsilon = 1$ et $k \geq 0$,

$$\mathbb{E} \left[\frac{1}{\pi_k} (\phi_C(\pi_k x + X_k) - \phi_C(X_k)) \right] = \mathbb{E} \left[\frac{1}{\pi_k} (\phi_C(\pi_k x + X_k) - \phi_C(X_k) - \pi_k x \phi_{\Delta C}(X_\infty)) \right]$$

et que

$$\begin{aligned} & |\phi_C(\pi_k x + X_k) - \phi_C(X_k) - \pi_k x \phi_{\Delta C}(X_\infty)| \\ & \leq \frac{1}{2} (\pi_k x)^2 \|\phi_{\Delta^2 C}\|_\infty + \pi_k x |\phi_{\Delta C}(X_\infty) - \phi_{\Delta C}(X_k)| \\ & \leq \pi_k^2 \left(\frac{x^2}{2} + \frac{x}{1-\delta} \right) \|\phi_{\Delta^2 C}\|_\infty, \end{aligned}$$

vu que $|X_\infty - X_k| \leq \pi_k/(1 - \delta)$ d'après l'hypothèse \mathbf{H}_1 .

D'autre part, pour $\varepsilon \in]0, 1[$ et $k \geq 0$, on a

$$\pi_k^{-\varepsilon} |\phi_C(\pi_k x + X_k) - \phi_C(X_k)| \leq \pi_k^{1-\varepsilon} x \|\phi_{\Delta C}\|_\infty.$$

L'intégration des équations (5.8) et (5.9) terme par terme est alors valide. Ceci établit finalement la représentation suivante.

Proposition 24 (Représentation de la Transformée de Poisson). *Pourvu que λ soit assez petit, la transformée de Poisson $\phi_\lambda(x)$ de la suite (R_n^λ)*

$$\phi_\lambda(x) = 1 + x \mathbb{E}(\phi'_\lambda(X_\infty)) \mathbf{1}_{(\varepsilon=1)} + \mathbb{E} \left(\sum_{k=0}^{+\infty} \frac{1}{\pi_k^\varepsilon} \left[\phi_C(X_k) - \phi_C(\pi_k x + X_k) \right] \right), \quad (5.10)$$

où $\varepsilon \in]0, 1]$ est le taux de conservation de l'algorithme, C est la suite définie dans la Proposition 23, (W_n) est une suite de variables *i.i.d.* de distribution \mathcal{W} et, pour $n \in \mathbb{N}$, $\pi_n = \prod_{i=1}^n W_i$ et (X_n) est le processus auto-régressif défini par $X_0 = 0$ et $X_{n+1} = W_n X_n + \lambda$ pour $n \geq 0$ et X_∞ est sa limite en distribution.

5.1.3 Coût moyen de l'algorithme

Si \mathcal{F}_k est la tribu engendrée par les variables W_1, \dots, W_k et \mathcal{N}_1 est un deuxième processus *Poisson* d'intensité 1, indépendant de \mathcal{N} et de (W_n) , alors, pour $k \geq 1$,

$$\begin{aligned} \phi_C(\pi_k x + X_k) &= \mathbb{E} (C_{\mathcal{N}([0, x\pi_k]) + \mathcal{N}_1([0, X_k])} \mid \mathcal{F}_k) \\ &= \sum_{m \geq 0} \mathbb{E} (C_{\mathcal{N}([0, x\pi_k]) + \mathcal{N}_1([0, X_k])} \mid \mathcal{F}_k, \mathcal{N}([0, x]) = m) \frac{x^m}{m!} e^{-x} \\ &= \sum_{m \geq 0} \mathbb{E} (C_{\mathcal{N}([0, \pi_k]) + \mathcal{N}_1([0, X_k])} \mid \mathcal{F}_k, \mathcal{N}([0, 1]) = m) \frac{x^m}{m!} e^{-x}. \end{aligned}$$

Utilisant l'équation (5.10), on obtient la relation

$$\begin{aligned} \phi_\lambda(x) &= 1 + x \mathbb{E}(\phi'_\lambda(X_\infty)) \mathbf{1}_{(\varepsilon=1)} \\ &\quad + \sum_{m \geq 0} \frac{x^m}{m!} e^{-x} \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k^\varepsilon} \left[C_{\mathcal{N}([0, X_k])} - C_{\mathcal{U}_m([0, \pi_k]) + \mathcal{N}([0, X_k])} \right] \right), \end{aligned}$$

où, si U_1, \dots, U_n sont n variables *i.i.d.* uniformément distribuées sur $[0, 1]$, pour $0 \leq x \leq 1$, $\mathcal{U}_n([0, x])$ désigne le nombre des U_i dans l'intervalle $[0, x]$. En identifiant les coefficients dans l'expression ci-dessus, on obtient que, pour $n \geq 0$,

$$\begin{aligned} \mathbb{E} (R_n^\lambda) &= 1 + n \mathbb{E}(\phi'_\lambda(X_\infty)) \mathbf{1}_{(\varepsilon=1)} \\ &\quad + \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k^\varepsilon} \left[C_{\mathcal{N}([0, X_k])} - C_{\mathcal{U}_n([0, \pi_k]) + \mathcal{N}([0, X_k])} \right] \right). \quad (5.11) \end{aligned}$$

L'équation (5.11) donne la représentation suivante, pour $n \geq D$,

$$\mathbb{E}(R_n^\lambda) = 1 + n\mathbb{E}(\phi'_\lambda(X_\infty)) \mathbf{1}_{(\varepsilon=1)} - \sum_{i=0}^{D-1} \sum_{k \geq 0} \mathbb{E}\left(\frac{1}{\pi_k^\varepsilon} \Delta C_{i+\mathcal{N}([0, X_k])} \mathbf{1}_{(U_{i+1, n} \leq \pi_k)}\right), \quad (5.12)$$

où (ΔC_i) est la suite $(C_{i+1} - C_i)$ et, pour $1 \leq i \leq n$, $U_{i, n}$ est le i -ième plus petit terme parmi n variables aléatoires indépendantes uniformes sur $[0, 1]$.

Cas particuliers

— **Cas statique** : $\lambda = 0$.

Dans ce cas, la suite des inconnues $C = (C_i)_{0 \leq i \leq D-1}$ est constante et est égale au degré moyen de l'arbre

$$C \equiv \mathbb{E}(W_1^{-\varepsilon}) = \mathbb{E}(G).$$

Donc, pour tout $n \geq 0$,

$$\mathbb{E}(R_n^0) = 1 + \mathbb{E}(G) \sum_{k \geq 0} \mathbb{E}\left(\frac{1}{\pi_k^\varepsilon} \mathbf{1}_{(U_{D, n} \leq \pi_k)}\right).$$

On retrouve ainsi l'expression établie dans le chapitre 3 pour la le coût moyen d'un algorithme en arbre non forcément conservatif.

— **Cas binaire dynamique** : $G = D = 2$.

La suite des inconnues C est réduite à deux coefficients et est donnée par

$$C_0 = 2\phi_\lambda(\lambda), \quad C_1 = 2\phi_\lambda(\lambda) + (p + q)\phi'_\lambda(\lambda).$$

Le coût moyen de l'algorithme est alors donné par

$$\begin{aligned} \mathbb{E}(R_n^\lambda) &= 1 + n\mathbb{E}(\phi'_\lambda(X_\infty)) \mathbf{1}_{(\varepsilon=1)} - (p + q) \phi'_\lambda(\lambda) \sum_{k \geq 0} \mathbb{E}\left(\frac{e^{-X_k}}{\pi_k^\varepsilon} \mathbf{1}_{(U_{1, n} \leq \pi_k)}\right) \\ &\quad + (2\phi_\lambda(\lambda) + (p + q) \phi'_\lambda(\lambda)) \sum_{k \geq 0} \mathbb{E}\left(\frac{(1 + X_k)e^{-X_k}}{\pi_k^\varepsilon} \mathbf{1}_{(U_{2, n} \leq \pi_k)}\right). \end{aligned}$$

Le cas binaire *statique* correspond à $\phi_0(0) = 1$, $\phi'_0(0) = 0$, le coût moyen est alors donné par

$$\mathbb{E}(R_n^0) = 1 + 2 \sum_{k \geq 0} \mathbb{E}\left(\frac{1}{\pi_k^\varepsilon} \mathbf{1}_{(U_{2, n} \leq \pi_k)}\right).$$

5.2 Condition de Stabilité : le cas conservatif $\varepsilon = 1$

Pour obtenir la représentation explicite de la suite (R_m^λ) , les D coefficients inconnus C_0, \dots, C_{D-1} (rappelons que les autres C_k sont nuls) ainsi que la constante $C_\infty = \mathbb{E}(\phi'_\lambda(X_\infty))$ doivent être déterminés.

Les conditions aux bords $\mathbb{E}(R_m^\lambda) = 1$ pour $1 \leq m \leq D - 1$ sont traduites en $D - 1$ équations linéaires reliant ces $D + 1$ inconnues,

$$M_{m,D}^\lambda C_\infty + \sum_{\ell=0}^D \sum_{\ell=0}^{D-1} M_{m,\ell}^\lambda C_\ell = 0, \quad 1 \leq m \leq D - 1, \quad (5.13)$$

où, pour $1 \leq m \leq D - 1$, $0 \leq \ell \leq D - 1$,

$$M_{m,\ell}^\lambda = \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k} \left[\mathbf{1}_{(\mathcal{N}([0, X_k]) = \ell)} - \mathbf{1}_{(\mathcal{U}_m([0, \pi_k]) + \mathcal{N}([0, X_k]) = \ell)} \right] \right)$$

et $M_{m,D}^\lambda = m$.

Remarque 17. Pour $m = 0$, l'équation (5.11) est triviale, ceci est une conséquence du fait que la relation $R_0^\lambda = 1$ a déjà été utilisée pour obtenir l'équation (5.10).

L'équation (5.7) donne une relation additionnelle

$$M_{D,0}^\lambda C_0 + M_{D,1}^\lambda C_1 + \cdots + M_{D,D-1}^\lambda C_{D-1} = 0, \quad (5.14)$$

où

$$M_{D,\ell}^\lambda = \mathbb{E} \left(\frac{(X_\infty)^\ell}{\ell!} (\ell X_\infty^{-1} - 1) e^{-X_\infty} \right), \quad 0 \leq \ell \leq D - 1.$$

L'équation finale obtenue en posant $x = \lambda$ dans l'équation (5.10) implique que, comme $C_0 = \mathbb{E}(G) \phi(\lambda)$,

$$\begin{aligned} -1 &= \lambda C_D - \frac{1}{\mathbb{E}(G)} C_0 \\ &+ \sum_{\ell=0}^{D-1} C_\ell \mathbb{E} \left(\sum_{k=0}^{+\infty} \frac{1}{\pi_k} \left[\frac{X_k^\ell}{\ell!} e^{-X_k} - \frac{(\pi_k \lambda + X_k)^\ell}{\ell!} e^{-(\pi_k \lambda + X_k)} \right] \right) \end{aligned} \quad (5.15)$$

Le théorème suivant est le résultat majeur concernant l'ergodicité de l'*algorithme en arbre*.

Théorème 7 (Stabilité de l'Algorithme en Arbre avec Arrivées Poissonniennes). *La matrice $\mathcal{M}_\lambda = (M_{m,\ell}^\lambda, 1 \leq m \leq D+1, 0 \leq \ell \leq D)$ associée à l'algorithme en arbre est définie par*

$$M_{m,\ell}^\lambda = \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k} \left[\mathbf{1}_{(\mathcal{N}([0, X_k]) = \ell)} - \mathbf{1}_{(\mathcal{U}_m([0, \pi_k]) + \mathcal{N}([0, X_k]) = \ell)} \right] \right),$$

pour $1 \leq m \leq D-1$, $0 \leq \ell \leq D-1$, et

$$\begin{aligned}
 M_{D,\ell}^\lambda &= \mathbb{E} \left(\frac{(X_\infty)^{\ell-1}}{(\ell-1)!} \left(1 - \frac{X_\infty}{\ell} \right) e^{-X_\infty} \right), & 1 \leq \ell \leq D-1, \\
 M_{D,0}^\lambda &= -\mathbb{E} \left(e^{-X_\infty} \right), \\
 M_{m,D}^\lambda &= m, & 1 \leq m \leq D-1, \\
 M_{D+1,\ell}^\lambda &= \mathbb{E} \left(\sum_{k=0}^{+\infty} \frac{1}{\pi_k} \left[\frac{X_k^\ell}{\ell!} e^{-X_k} - \frac{(\pi_k \lambda + X_k)^\ell}{\ell!} e^{-(\pi_k \lambda + X_k)} \right] \right), & 1 \leq \ell \leq D-1, \\
 M_{D+1,0}^\lambda &= \mathbb{E} \left(\sum_{k=0}^{+\infty} \frac{1}{\pi_k} \left[e^{-X_k} - e^{-(\pi_k \lambda + X_k)} \right] \right) - \frac{1}{\mathbb{E}(G)},
 \end{aligned}$$

où $M_{D+1,D+1}^\lambda = \lambda$ et le coefficient restant est nul. Le processus auto-régressif $(X_n, 0 \leq n \leq +\infty)$ est introduit dans la Proposition 24. Si

$$\lambda_c = \inf\{\lambda > 0 : \det \mathcal{M}_\lambda = 0\}, \quad (5.16)$$

alors, pour tout $\lambda < \lambda_c$, les variables (R_n^λ) sont intégrables et par conséquent la chaîne de Markov associée à l'algorithme en arbre est ergodique.

Remarque 18. La matrice \mathcal{M}_λ est constituée de 3 composantes ;

$$\mathcal{M}_\lambda = \begin{pmatrix} \begin{pmatrix} M_{1,0}^\lambda & M_{1,1}^\lambda & \cdots & M_{1,D-1}^\lambda & 1 \\ M_{2,0}^\lambda & M_{2,1}^\lambda & \cdots & M_{2,D-1}^\lambda & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M_{D-1,0}^\lambda & M_{D-1,1}^\lambda & \cdots & M_{D-1,D-1}^\lambda & D-1 \end{pmatrix} \\ \begin{pmatrix} M_{D,0}^\lambda & M_{D,1}^\lambda & \cdots & M_{D,D-1}^\lambda & 0 \end{pmatrix} \\ \begin{pmatrix} M_{D+1,0}^\lambda & M_{D+1,1}^\lambda & \cdots & M_{D+1,D-1}^\lambda & M_{D+1,D}^\lambda \end{pmatrix} \end{pmatrix}$$

où les $(D-1)$ premières lignes (en bleu) sont dues aux conditions de bords $R_\ell^\lambda = 1$ pour $1 \leq \ell \leq D-1$, la D -ième ligne (en rouge) traduit l'équilibre du processus auto-régressif $(X_n, 0 \leq n \leq +\infty)$, à savoir $\mathbb{E}(\phi_{\Delta_c}(X_\infty)) = 0$, et enfin la dernière ligne traduit la bonne définition de la transformée de Poisson de la suite (r_n) .

Démonstration. Pour $\lambda < \lambda_c$, désignons par $C^\lambda = (C_0^\lambda, C_1^\lambda, \dots, C_{D-1}^\lambda, C_\infty^\lambda)$ l'unique solution du système linéaire $\mathcal{M}_\lambda \cdot C^\lambda = (0, 0, \dots, 0, -1)$.

Supposons aussi que $\lambda < \lambda_W$, d'après le Corollaire 5, la transformée de Poisson de la suite (R_n^λ) est finie sur \mathbb{R} . D'après les équations (5.13) (5.14) et (5.15), on a nécessairement que pour $i = 0, \dots, D-1$, $C_i^\lambda = C_i$ et $C_\infty^\lambda = \mathbb{E}(\phi'_\lambda(X_\infty))$. Or

$$\lim_{\lambda \rightarrow \lambda_W} \phi_\lambda(\lambda) = +\infty$$

encore d'après le Corollaire 5, on obtient que $\lambda_W \geq \lambda_c$. Le théorème est prouvé. \square

Cas particuliers

Le cas statique $\lambda = 0$:

Le cas *statique* correspond à la matrice \mathcal{M}_0 donnée par

$$\mathcal{M}_0 = \begin{pmatrix} M_{1,0}^0 & M_{1,1}^0 & \cdots & M_{1,D-1}^0 & 1 \\ M_{2,0}^0 & M_{2,1}^0 & \cdots & M_{2,D-1}^0 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M_{D-1,0}^0 & M_{D-1,1}^0 & \cdots & M_{D-1,D-1}^0 & D-1 \\ -1 & 1 & 0 & \cdots & 0 \\ -1/\mathbb{E}(G) & 0 & \cdots & 0 & 0 \end{pmatrix}$$

Un développement du déterminant de \mathcal{M}_0 par rapport à la dernière ligne, suivi d'un autre par rapport à l'avant dernière, implique que

$$\det(\mathcal{M}_0) = \frac{-1}{\mathbb{E}(G)} \begin{vmatrix} M_{1,2}^0 & M_{1,3}^0 & \cdots & M_{1,D-1}^0 & 1 \\ M_{2,2}^0 & M_{2,3}^0 & \cdots & M_{2,D-1}^0 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M_{D-1,2}^0 & M_{D-1,3}^0 & \cdots & M_{D-1,D-1}^0 & D-1 \end{vmatrix}.$$

Or, pour $1 \leq m, \ell \leq D-1$,

$$M_{m,\ell}^0 = - \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k} \mathbf{1}_{(\mathcal{U}_m([0,\pi_k])=\ell)} \right),$$

alors, $M_{m,\ell}^0 = 0$ dès que $\ell \geq m$. Ainsi, par un développement par rapport à la première ligne, on obtient que

$$\det(\mathcal{M}_0) = \frac{-1}{\mathbb{E}(G)} M_{2,2}^0 \cdots M_{D-1,D-1}^0.$$

Le cas binaire :

Pour un *algorithme en arbre binaire*, i.e. $G \equiv 2$ et $D = 2$, la *mesure de branchement* est donnée par $\mathcal{W} = p\delta_p + q\delta_q$, où p est la probabilité d'aller à gauche dans l'arbre associé, $q = 1 - p$.

La seule condition aux bords non-triviale étant $R_1^\lambda = 1$, on obtient ainsi la première équation linéaire du système de stabilité

$$C_\infty + M_{1,0}^\lambda C_0 + M_{1,1}^\lambda C_1 = 0,$$

où le coefficient $M_{1,2} = 1$ et

$$M_{1,0}^\lambda = \mathbb{E} \left[\sum_{k \geq 0} \frac{e^{-X_k}}{\pi_k} \right]$$

$$M_{1,1}^\lambda = \mathbb{E} \left[\sum_{k \geq 0} \frac{e^{-X_k}(X_k - 1)}{\pi_k} \right]$$

Une deuxième équation linéaire est due à l'équation (5.7) $\mathbb{E}(\phi_{\Delta C}(X_\infty)) = 0$, ce qui implique que

$$M_{2,0}^\lambda C_0 + M_{2,1}^\lambda C_1 = 0,$$

où

$$M_{2,0}^\lambda = -\mathbb{E}(e^{-X_\infty}), \quad M_{2,1}^\lambda = \mathbb{E}(e^{-X_\infty}) - \mathbb{E}(X_\infty e^{-X_\infty}).$$

Or, $C_0 = 2\phi_\lambda(\lambda)$ et $C_1 = 2\phi_\lambda(\lambda) + \phi'_\lambda(\lambda)$, alors on obtient que

$$\phi'_\lambda(\lambda) = -2 \left(\frac{M_{2,0}^\lambda}{M_{2,1}^\lambda} + 1 \right) \phi_\lambda(\lambda) = 2 \left(\frac{\mathbb{E}(e^{-X_\infty})}{\mathbb{E}(e^{-X_\infty}) - \mathbb{E}(X_\infty e^{-X_\infty})} - 1 \right) \phi_\lambda(\lambda). \quad (5.17)$$

Rappelons que, si $(A)_{i \geq 0}$ désigne une suite *i.i.d.* de loi \mathcal{W} , alors $X_\infty = \lambda \sum_{k \geq 0} A_0 \dots A_{k-1}$. Dans le cas particulier où le processus de décomposition est symétrique, *i.e.* $p = q = 1/2$, X_∞ est déterministe et vaut 2λ .

La relation (5.17) correspond à celle établie par Fayolle *et al.* [35]

$$\phi'_\lambda(\lambda) = 2(K - 1) \phi_\lambda(\lambda),$$

où

$$\begin{aligned} K &= -\frac{e^{-\lambda/p} - e^{-\lambda/q}}{\frac{\lambda}{p}e^{-\lambda/p} - \frac{\lambda}{q}e^{-\lambda/q}} \quad \text{si } p \neq 1/2, \\ &= 1/(1 - 2\lambda) \quad \text{si } p = q = 1/2. \end{aligned}$$

La troisième et dernière équation linéaire est due à l'égalité

$$\phi_\lambda(\lambda) = 1 + \lambda C_\infty + \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k} [\phi_C(X_k) - \phi_C(\pi_k \lambda + X_k)] \right).$$

On obtient alors la relation suivante

$$M_{3,0}^\lambda C_0 + M_{3,1}^\lambda C_1 + M_{3,2}^\lambda C_\infty = -1,$$

où

$$\begin{aligned} M_{3,0}^\lambda &= -\frac{1}{2} + \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k} [e^{-X_k} - e^{-(\pi_k \lambda + X_k)}] \right) \\ M_{3,1}^\lambda &= \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k} [X_k e^{-X_k} - (\pi_k \lambda + X_k) e^{-(\pi_k \lambda + X_k)}] \right) \\ M_{3,2}^\lambda &= \lambda. \end{aligned}$$

Si $\mathcal{D}(\lambda)$ désigne le déterminant de la matrice \mathcal{M}_λ définie par

$$\mathcal{M}_\lambda = \begin{pmatrix} M_{1,0}^\lambda & M_{1,1}^\lambda & 1 \\ M_{2,0}^\lambda & M_{2,1}^\lambda & 0 \\ M_{3,0}^\lambda & M_{3,1}^\lambda & \lambda \end{pmatrix}$$

alors, la condition de stabilité de l'*algorithme en arbre binaire* est donnée par

$$\mathcal{D}(\lambda) \neq 0,$$

ainsi, la région de stabilité est l'intervalle $[0, \lambda_{\mathcal{W}}[$, où $\lambda_{\mathcal{W}} = \inf\{\lambda \geq 0; \mathcal{D}(\lambda) = 0\}$. Remarquons que, dans le cas *statique* $\lambda = 0$, on a bien $\mathcal{D}(0) = -1/2$. De plus, l'unique solution du système linéaire traduisant la stabilité de l'*algorithme* est donnée par le vecteur : $(C_0, C_1, C_\infty) = (2, 2, 0)$. Le coût moyen de l'*algorithme binaire statique* est alors donné, pour tout $n \geq 0$, par

$$\mathbb{E}(R_n^0) = 1 + 2 \sum_{k \geq 0} \mathbb{E} \left(\frac{1}{\pi_k} \mathbf{1}_{(U_{2,n} \leq \pi_k)} \right),$$

où, pour $k \geq 1$, $\pi_k = A_0 \dots A_{k-1}$ avec $\pi_0 = 1$.

Conclusion

Tout au long de ce mémoire, une classe d’algorithmes ayant une structure en arbre a été étudiée d’un point de vue *taille* et *hauteur* de l’arbre associé. L’approche construite et les outils utilisés se veulent simples et robustes à la fois, au sens où différents algorithmes, *statiques* ou *dynamiques*, soient traités selon une même méthodologie. Cette étude est un tour d’horizon de quelques unes parmi les questions qu’il serait possible d’étudier suivant cette approche telles que l’analyse exacte ou asymptotique de différents coûts d’algorithmes, l’impact d’une possible perte lors du processus de décomposition associé, la stabilité d’algorithmes dits *dynamiques* ou encore le lien entre *algorithmes en arbre* et *processus de branchement*. L’objectif principal étant l’analyse du comportement asymptotique des ces algorithmes, le cas dynamique est un travail en cours avec Philippe Robert. En effet, il existe un théorème limite, le *théorème de renouvellement*, qui “connaît” les trajectoires les plus significatives dans les compositions des opérateurs fonctionnels qui interviennent dans l’expression établie du coût moyen d’un tel algorithme (voir l’expression (4.7) par exemple) et évite ainsi l’utilisation de différentes estimations assez techniques nécessaires pour en tirer les termes significatifs asymptotiquement comme il est le cas dans [33]. Dans le cas *dynamique*, l’application de ce théorème est basée sur une propriété d’indépendance asymptotique entre le processus auto-régressif $(X_k)_{k \geq 0}$ d’une part et la marche aléatoire associée à l’algorithme d’autre part. Ceci constitue une suite naturelle au travail présenté dans ce mémoire.

Annexe A

Théorème de renouvellement

A.1 Lois et Processus de *Poisson*

On note par $(t_n)_{n \geq 1}$ une suite non décroissante de variables aléatoires telles que

- t_1 suit une distribution exponentielle de paramètre 1,
- $(t_{n+1} - t_n)$ est une suite *i.i.d.* de variables aléatoires exponentiellement distribuées de paramètre 1.

Pour $x \geq 0$, soit $\mathcal{N}([0, x])$ le nombre des points t_n dans l'intervalle $[0, x]$

$$\mathcal{N}([0, x]) \stackrel{\text{déf.}}{=} \sum_{n=1}^{\infty} \mathbf{1}_{(t_n \leq x)}.$$

Pour $x > 0$ fixé, $\mathcal{N}([0, x])$ suit une loi de *Poisson* de paramètre x , *i.e.*

$$\mathbb{P}(\mathcal{N}([0, x]) = k) = \frac{x^k}{k!} e^{-x}.$$

Le processus qui à $x > 0$ associe $\mathcal{N}([0, x])$ est un *processus de Poisson* de paramètre 1 sur $[0, x]$.

Définition 9 (Transformée de *Poisson*). La transformée de Poisson d'une suite positive (a_n) est définie par

$$\sum_{n \geq 0} a_n \frac{x^n}{n!} e^{-x} = \mathbb{E}(a_{\mathcal{N}([0, x])}). \quad (\text{A.1})$$

Le résultat suivant établit la loi jointe des points (t_i) d'un processus de *Poisson* $\mathcal{N}([0, x])$ sachant que leur nombre sur l'intervalle $[0, x]$ est fixé. Voir Robert [94].

Proposition 25 (Lois Conditionnelles). Pour $n \geq 1$, conditionnellement à l'événement $(\mathcal{N}([0, x]) = n)$, l'ensemble des variables (t_1, t_2, \dots, t_n) a la même loi que n variables indépendantes uniformément distribuées sur $[0, x]$.

A.2 Quelques Résultats autour du Théorème de Renouvellement

Quelques définitions et résultats concernant le théorème de renouvellement sont brièvement rappelés ici. Voir Grimmett et Stirzaker [45] par exemple.

Définition 10. Distribution Arithmétique. *Une distribution μ sur \mathbb{R}_+ est dite arithmétique s'il existe un réel $\lambda > 0$, tel que $\mu(\{n\lambda : n \in \mathbb{N}\}) = 1$. Le plus grand λ satisfaisant cette propriété est appelé pas de la distribution.*

A.2.1 Processus de Renouvellement Continu

Si (τ_i) est une suite *i.i.d.*, pour $x > 0$, la variable ν_x est définie par le temps d'atteinte de l'ensemble $[x, +\infty[$, ou encore celui du dépassement de la barrière x , par la marche aléatoire associée à (τ_i) ,

$$\nu_x = \inf\{n : S_n \geq x\},$$

où, pour $n \geq 0$, $S_n = \tau_1 + \tau_2 + \dots + \tau_n$.

La variable $t_0^x = S_{\nu_x} - x$ est la valeur du dépassement du niveau x par la marche aléatoire S_k . Pour $k \geq 0$, on définit t_k^x comme la position du k ème point de la marche avant x , $t_k^x = S_{\nu_x - k} - x$, avec la convention $S_i = -\infty$ pour $i < 0$.

Théorème 8 (Théorème de Renouvellement Continu). *Si la distribution de τ_0 est non-arithmétique, alors les variables aléatoires*

$$\left((t_k^x - t_{k+1}^x, k \geq 1), t_1^x, t_0^x \right)$$

convergent en distribution vers une suite $((\tilde{\tau}_i, i \geq 2), \tau_1^, \tau^*)$, où la suite $(\tilde{\tau}_i, i \geq 2)$ est *i.i.d.* de même distribution que τ_0 et indépendante du couple de variables (τ_1^*, τ^*) dont la distribution est donnée par*

$$\mathbb{E}(f(\tau_1^*, \tau^*)) = \frac{1}{\mathbb{E}(\tau_0)} \mathbb{E} \left(\int_0^{\tau_0} f(-u, \tau_0 - u) du \right), \quad (\text{A.2})$$

pour toute fonction borélienne positive f sur \mathbb{R}^2 .

A.2.2 Processus de Renouvellement Discret

Si (C_i) est une suite *i.i.d.* de variables aléatoires à valeurs entières telle que $\mathbb{P}(C_1 = 1) > 0$ et

$$\tau_n = \inf \left\{ k : \sum_{i=1}^k C_i \geq n \right\}, \text{ alors}$$

Théorème 9 (Théorème de Renouvellement Discret). *Les variables aléatoires*

$$\left((C_{\tau_n - k}, 1 \leq k \leq \tau_n), n - \sum_{i=1}^{\tau_n - 1} C_i \right)$$

convergent en distribution vers $((\tilde{C}_i, i \geq 2), C_1^*)$, où la suite $(\tilde{C}_i, i \geq 2)$ est i.i.d. de même distribution que C_1 et indépendantes de la variable C_1^* dont la distribution est donnée par

$$\mathbb{P}(C_1^* = n) = \frac{1}{\mathbb{E}(C_1)} \mathbb{P}(C_1 \geq n), \quad n \geq 1.$$

Annexe B

Élection du leader : preuve du lemme 3

On présente ici la preuve du lemme 3 du chapitre 3 de la première partie de ce mémoire.

Démonstration du lemme 3 page 56. Rappelons la suite aléatoire (γ_i) définie par

$$(\gamma_i)_{i \in \mathbb{N}} = \{j \geq 0 : B_j = p\}.$$

Pour étudier le temps d'arrêt τ , notons que

$$\begin{aligned} (\nu(x) \geq 2 + \gamma_0) &\Leftrightarrow (\gamma_0 \geq \lfloor \log_p(x) \rfloor) \\ (\mu(y) \leq \gamma_0) &\Leftrightarrow (\gamma_0 \geq \lceil \log_p(y) \rceil). \end{aligned}$$

Comme $0 < x < y$,

$$\log_p(y) < \log_p(x).$$

On distingue deux cas :

1. Si $\exists n \in \mathbb{N}$ tel que

$$\log_p(y) < n < \log_p(x),$$

alors

$$\lceil \log_p(y) \rceil \leq \lfloor \log_p(x) \rfloor.$$

Une discussion selon la position de γ_0 par rapport à ces deux nombres implique que :

- si $\gamma_0 \geq \lfloor \log_p(x) \rfloor$, alors $\tau(x, y) = \mu(y) \leq \gamma_0$,
- si $\gamma_0 < \lceil \log_p(y) \rceil$, alors

$$\begin{cases} \mu(y) \geq 2 + \gamma_0, \\ \nu(x) = 1 + \gamma_0, \end{cases} \text{ et donc } \tau(x, y) = \nu(x) = 1 + \gamma_0.$$

- si $\lceil \log_p(y) \rceil \leq \gamma_0 < \lfloor \log_p(x) \rfloor$, alors

$$\begin{cases} \mu(y) \leq \gamma_0, \\ \nu(x) = 1 + \gamma_0, \end{cases} \text{ et donc } \tau(x, y) = \mu(y) \leq \gamma_0.$$

2. si $[\log_p(y), \log_p(x)] \cap \mathbb{N} = \emptyset$, alors

$$\lceil \log_p(y) \rceil = \lceil \log_p(x) \rceil.$$

De même, une discussion selon la position de γ_0 par rapport à $\lceil \log_p(y) \rceil$ implique que :

- si $\gamma_0 \geq \lceil \log_p(y) \rceil$, alors $\tau(x, y) = \mu(y) \leq \gamma_0$.
- si $\gamma_0 < \lceil \log_p(y) \rceil$, alors

$$\begin{cases} \mu(y) \geq 2 + \gamma_0, \\ \nu(x) = 1 + \gamma_0, \end{cases} \text{ et donc } \tau(x, y) = \nu(x) = 1 + \gamma_0.$$

- si $(\gamma_0 = \lceil \log_p(y) \rceil)$, alors

$$\begin{cases} \mu(y) \geq 2 + \gamma_0, \\ \nu(x) \geq 2 + \gamma_0, \end{cases} \text{ et donc } \tau(x, y) \geq 2 + \gamma_0.$$

Notons par Ω_0 l'ensemble suivant

$$\Omega_0 = \Omega(x, y) = \{(x, y) \in (]0, 1])^2 : \lceil \log_p(y) \rceil = \lceil \log_p(x) \rceil\}.$$

En décomposant la fonction Φ par rapport à Ω_0 , on obtient que

$$\begin{aligned} \mathbb{E} \left(\sum_{i=0}^{\tau(x,y)-1} \frac{1}{\pi_i} \right) &= \mathbb{E} \left(\sum_{i=0}^{\gamma_0} \frac{1}{p^i} \mathbf{1}_{\{\gamma_0 < \lceil \log_p(y) \rceil\}} \right) \\ &+ \mathbb{P}(\gamma_0 = \lceil \log_p(y) \rceil) \sum_{i=0}^{\lceil \log_p(y) \rceil} \frac{1}{p^i} \\ &+ \mathbb{E} \left(\sum_{i=0}^{\mu(y)-1} \frac{1}{p^i} \mathbf{1}_{\{\gamma_0 \geq \lceil \log_p(y) \rceil\}} \right) \\ &+ \mathbb{E} \left(\left(\sum_{i=\lceil \log_p(y) \rceil}^{\tau(x,y)-1} \frac{1}{\pi_i} \right) \mathbf{1}_{\{\gamma_0 = \lceil \log_p(y) \rceil\}} \right) \mathbf{1}_{\Omega_0}. \end{aligned}$$

Il est clair que la variable aléatoire γ_0 admet une densité géométrique de paramètre q . Intéressons-nous au terme

$$\mathbb{E} \left(\sum_{i=0}^{\mu(y)-1} \frac{1}{p^i} \mathbf{1}_{\{\gamma_0 \geq \lceil \log_p(y) \rceil\}} \right).$$

Comme

$$(\gamma_0 \geq \lceil \log_p(y) \rceil) \Leftrightarrow (\mu(y) \leq \gamma_0),$$

alors

$$\mu(y) | (\gamma_0 \geq \lceil \log_p(y) \rceil) = \inf\{i \geq 1, p^i < y\} = \lceil \log_p(y) \rceil.$$

Ceci implique que

$$\mathbb{E} \left(\sum_{i=0}^{\mu(y)-1} \frac{1}{p^i} \mathbf{1}_{\{\gamma_0 \geq \lceil \log_p(y) \rceil\}} \right) = \frac{p}{q} (1 - p^{\lceil \log_p(y) \rceil}).$$

Par des simples calculs, on obtient que

$$\mathbb{E} \left(\sum_{i=0}^{\tau(x,y)-1} \frac{1}{\pi_i} \right) = \lceil \log_p(y) \rceil + \mathbb{E} \left(\sum_{i=\lceil \log_p(y) \rceil}^{\tau(x,y)-1} \frac{1}{\pi_i} \mathbf{1}_{\{\gamma_0 = \lceil \log_p(y) \rceil\}} \right) \mathbf{1}_{\Omega_0}.$$

Une seconde discussion sur γ_1 implique que, sur l'ensemble $(\gamma_0 = \lceil \log_p(y) \rceil, \Omega_0)$, on a

$$\begin{aligned} (\nu(x) \geq 2 + \gamma_1) &\Leftrightarrow (\gamma_1 \geq \lceil \log_p(\rho(\log_p(x))x) \rceil) \\ (\mu(y) \leq \gamma_1) &\Leftrightarrow (\gamma_1 \geq 1 + \lceil \log_p(\rho(\log_p(y))y) \rceil), \end{aligned}$$

où

$$\rho(z) = \frac{1 - p^{1-\{z\}}}{1 - p}$$

est une fonction périodique d'amplitude 1 et $\{z\}$ désigne la partie fractionnaire de z . De plus, comme ρ est décroissante sur $[0, 1[$, alors, sur l'ensemble Ω_0 , on a

$$\rho(\log_p(x))x < \rho(\log_p(y))y.$$

Suivant le même raisonnement que la première discussion sur γ_0 , on définit

$$\Omega_1 = \Omega(\rho(\log_p(x))x, \rho(\log_p(y))y).$$

Alors, sur l'ensemble $(\gamma_0 = \lceil \log_p(y) \rceil, \Omega_0)$, on a

$$\begin{aligned} \sum_{i=\lceil \log_p(y) \rceil}^{\tau(x,y)-1} \frac{1}{\pi_i} &= \left(\sum_{i=\lceil \log_p(y) \rceil}^{\gamma_1} \frac{1}{\pi_i} \right) \mathbf{1}_{\{\gamma_1 \leq \lceil \log_p(\rho(\log_p(y))y) \rceil\}} \\ &+ \left(\sum_{i=\lceil \log_p(y) \rceil}^{\lceil \log_p(\rho(\log_p(y))y) \rceil} \frac{1}{\pi_i} \right) \mathbf{1}_{\{\gamma_1 = \lceil \log_p(\rho(\log_p(y))y) \rceil\}} \\ &+ \left(\sum_{i=\lceil \log_p(y) \rceil}^{\mu(y)-1} \frac{1}{\pi_i} \right) \mathbf{1}_{\{\gamma_1 \geq 1 + \lceil \log_p(\rho(\log_p(y))y) \rceil\}} \\ &+ \left(\sum_{i=1 + \lceil \log_p(\rho(\log_p(y))y) \rceil}^{\tau(x,y)-1} \frac{1}{\pi_i} \right) \mathbf{1}_{\{\gamma_1 = \lceil \log_p(\rho(\log_p(y))y) \rceil\}} \mathbf{1}_{\Omega_1} \end{aligned}$$

Il est simple de vérifier que, si

$$\gamma_1 \geq 1 + \lceil \log_p(\rho(\log_p(y))y) \rceil, \text{ alors } \mu(y) = 1 + \lceil \log_p(\rho(\log_p(y))y) \rceil.$$

D'autre part, $\gamma_1 = 1 + \gamma_0 + Geo(q)$ où, pour tout $k \in \mathbb{N}$,

$$Geo(q) = (k) = q p^k,$$

d'où

$$\begin{aligned} \mathbb{E} \left(\left(\sum_{i=\lceil \log_p(y) \rceil}^{\gamma_1} \frac{1}{\pi_i} \right) \mathbf{1}_{\{\gamma_0 = \lfloor \log_p(y) \rfloor; \gamma_1 \leq \lfloor \log_p(\rho(\log_p(y))y) \rfloor\}} \right) &= (\lfloor \log_p(\rho(\log_p(y))y) \rfloor - \lfloor \log_p(y) \rfloor) \\ &\quad - \frac{p}{q} (1 - p^{\lfloor \log_p(\rho(\log_p(y))y) \rfloor - \lfloor \log_p(y) \rfloor}) \end{aligned}$$

$$\mathbb{P}(\gamma_0 = \lfloor \log_p(y) \rfloor; \gamma_1 = \lceil \log_p(\rho(\log_p(y))y) \rceil) \sum_{i=\lceil \log_p(y) \rceil}^{\lceil \log_p(\rho(\log_p(y))y) \rceil} \frac{1}{\pi_i} = 1 - p^{\lceil \log_p(\rho(\log_p(y))y) \rceil - \lfloor \log_p(y) \rfloor}$$

$$\mathbb{E} \left(\left(\sum_{i=\lceil \log_p(y) \rceil}^{\mu(y)-1} \frac{1}{\pi_i} \right) \mathbf{1}_{\{\gamma_0 = \lfloor \log_p(y) \rfloor; \gamma_1 \geq 1 + \lceil \log_p(\rho(\log_p(y))y) \rceil\}} \right) = \frac{p}{q} (1 - p^{\lceil \log_p(\rho(\log_p(y))y) \rceil - \lfloor \log_p(y) \rfloor}).$$

Comme

$$\Omega_0 \cap \Omega_1 = \Omega_0,$$

alors

$$\begin{aligned} \mathbb{E} \left(\sum_{i=0}^{\tau(x,y)-1} \frac{1}{\pi_i} \right) &= \lfloor \log_p(y) \rfloor + (\lceil \log_p(\rho(\log_p(y))y) \rceil - \lfloor \log_p(y) \rfloor) \mathbf{1}_{\Omega(x,y)} \\ &\quad + \mathbb{E} \left(\sum_{i=1+\lceil \log_p(\rho(\log_p(y))y) \rceil}^{\tau(x,y)-1} \frac{1}{\pi_i} \mathbf{1}_{\{\gamma_0 = \lfloor \log_p(y) \rfloor; \gamma_1 = \lceil \log_p(\rho(\log_p(y))y) \rceil\}} \right) \mathbf{1}_{\Omega(x,y)}. \end{aligned}$$

Ceci termine la preuve. □

Table des figures

1	Exemple d'arbre binaire.	xii
2	Élection du <i>leader</i> A ; $H_4 = 4$. Structure d'arbre incomplet.	xiv
3	Exemple d'arbre binaire de recherche sur la chaîne : $\{7,3,10,14,1,6,7,4,13\}$	xvi
4	Trie binaire pour 5 chaînes de bits 000011, 000111, 001010, 100100, 110011.	xix
5	Arbre PATRICIA : suppression d'un nœud interne père à fils unique	xx
6	Algorithme en arbre avec $D = 2$, deux ensembles de poids aléatoires $(V_{1,2}, V_{2,2})$ et $(V_{1,3}, V_{2,3}, V_{1,3})$, G une variable aléatoire à valeurs dans $\{2, 3\}$, le groupe initial étant $\{A, B, C, D, E\}$	xxiii
7	Algorithme en arbre binaire avec arrivées libres aléatoires à valeurs dans $\{0, 1, 2\}$, le groupe initial étant $\{A, B, C\}$	xxiv
1.1	Algorithme en arbre avec $D = 2$, deux vecteurs de poids aléatoires $(V_{1,2}, V_{2,2})$ et $(V_{1,3}, V_{2,3}, V_{3,3})$, G une variable aléatoire à valeurs dans $\{2, 3\}$, le groupe initial étant $\{A, B, C, D, E, F\}$	3
1.2	Algorithme binaire : oscillations de la suite $n \mapsto \mathbb{E}(R_n)/n$	16
1.3	Décomposition de $[0, 1]$: transition entre algorithme en arbre et arbre généalogique.	27
3.1	Élection du <i>leader</i> A ; $H_4 = 1 + P_4 = 4$	43
3.2	Cas non biaisé : simulations de $x \mapsto \mathbb{E}(4^{\tau(x,x)})$	57
3.3	Cas biaisé : $p = 0.2$, $p = 0.8$. Simulations de $x \mapsto \mathbb{E}((\frac{1}{\delta^2})^{\tau(x,x)})$	58
3.4	Élection du <i>leader</i> A ; $H_4^0 = 4$, $H_2^1 = 3$, $H_2^2 = 2$, $H_1^3 = 1$	60
4.1	Premier niveau d'un algorithme en arbre dynamique décomposant $n \geq D$, $n_1 + \dots + n_G \leq n$ et avec arrivées (A_i)	69

Bibliographie

- [1] David Aldous, *Ultimate instability of exponential back-off protocol for acknowledgement-based transmission of control of random access communication channels*, IEEE Transactions on Information Theory **33** (1987), 219–223.
- [2] W. Szpankowski B. Rais, P. Jacquet, *A limiting distribution for the depth in patricia tries*, SIAM Journal on Algebraic and Discrete Methods **6** (1993), 193–213.
- [3] Julien Barral, *Moments, continuité et analyse multifractale des martingales de Mandelbrot*, Probability Theory and Related Fields **113** (1999), no. 4, 582–597.
- [4] Tim Bedford, *Applications of dynamical systems theory to fractals—a study of cookie-cutter Cantor sets*, Fractal geometry and analysis (Montreal, PQ, 1989), NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., vol. 346, Kluwer Acad. Publ., Dordrecht, 1991, pp. 1–44.
- [5] T. Berger, *The Poisson multiple-access conflict resolution problem*, Multi-User Communication Systems (G. Longo, ed.), CISM Courses and Lectures, vol. 265, Springer Verlag, 1981, pp. 1–28.
- [6] Jean Bertoin, *Homogeneous fragmentation processes*, Probability Theory and Related Fields **121** (2001), 301–318.
- [7] Jean Bertoin and Servet Martinez, *Fragmentation energy*, Advances in Applied Probability **37** (2005), no. 2, 553–570.
- [8] Dimitri Bertsekas and Robert Gallager, *Data networks*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1991, 2nd Edition.
- [9] Patrick Billingsley, *Ergodic theory and information*, Robert E. Krieger Publishing Co., Huntington, N.Y., 1978, Reprint of the 1965 original.
- [10] ———, *Convergence of probability measures*, second ed., Wiley Series in Probability and Statistics : Probability and Statistics, John Wiley & Sons Inc., New York, 1999, A Wiley-Interscience Publication.
- [11] Jérémie Bourdon, Markus Nebel, and Brigitte Vallée, *On the stack-size of general tries*, Theoretical Informatics and Applications **35** (2001), no. 2, 163–185.
- [12] Onno J. Boxma, Dee Denteneer, and Jacques Resing, *Delay models for contention trees in closed populations*, Performance Evaluation **53** (2003), no. 3-4, 169–185.
- [13] K. Brink, *The expected performance of traversal algorithms in binary trees*, The Computer Journal **28** (1985), 426–432.
- [14] ———, *On deletion in threaded binary trees*, Journal of Algorithms **7** (1985), 395–411.

- [15] G. Brown and B. Shubert, *On random binary trees*, Mathematics of Operations Research **9** (1984), 43–65.
- [16] John I. Capetanakis, *Tree algorithms for packet broadcast channels*, IEEE Transactions on Information Theory **25** (1979), no. 5, 505–515.
- [17] Philippe Carmona, Frédérique Petit, and Marc Yor, *On the distribution and asymptotic results for exponential functionals of Lévy processes*, Exponential functionals and principal values related to Brownian motion, Rev. Mat. Iberoam., Madrid, 1997, pp. 73–130.
- [18] Joseph T. Chang, *Inequalities for the overshoot*, The Annals of Applied Probability **4** (1994), no. 4, 1223–1233.
- [19] Bogdan S. Chlebus, *Randomized communication in radio networks*, Handbook of randomized computing, Vol. I, II, Comb. Optim., vol. 9, Kluwer Acad. Publ., Dordrecht, 2001, pp. 401–456.
- [20] J. Clément, P. Flajolet, and B. Vallée, *Dynamical sources in information theory : a general analysis of trie structures*, Algorithmica **29** (2001), no. 1-2, 307–369, Average-case analysis of algorithms (Princeton, NJ, 1998).
- [21] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, *Algorithms*, MIT Press, Cambridge, Massachusetts, 1990.
- [22] Hubert Delange, *Sur la formule sommatoire de la fonction somme des chiffres*, L'enseignement Mathématique **XXI** (1975), no. 1, 31–47.
- [23] M. Delasnerie, *Flot mélangeant et mesure de Palm*, Annales de l'institut Henri Poincaré **XIII** (1977), no. 4, 357–369.
- [24] Y. Derriennic, *On the integrability of the supremum of ergodic ratios*, Annals of Probability **1** (1973), no. 2, 338–340.
- [25] Luc Devroye, *A note on the height of binary search trees*, Journal of the ACM **33** (1986), 489–498.
- [26] ———, *Universal limit laws for depths in random trees*, SIAM Journal on Computing **28** (1998), no. 2, 409–432.
- [27] ———, *Universal asymptotics for random tries and Patricia trees*, Algorithmica **42** (2005), no. 1, 11–29.
- [28] Vincent Dumas, Fabrice Guillemin, and Philippe Robert, *A Markovian analysis of Additive-Increase Multiplicative-Decrease (AIMD) algorithms*, Advances in Applied Probability **34** (2002), no. 1, 85–111.
- [29] Richard Durrett, *Probability : theory and examples*, second ed., Duxbury Press, Belmont, CA, 1996.
- [30] Anthony Ephremides and Bruce Hajek, *Information theory and communication networks : an unconsummated union*, IEEE Transactions on Information Theory **44** (1998), no. 6, 1–20.
- [31] P. Erdős, A. Hildeb, P. Pudaite, and B. Reznick, *The asymptotic behavior of a family of sequences*, Pacific Journal of Mathematics **126** (1987), no. 2, 227–241.

-
- [32] Kenneth Falconer, *Techniques in fractal geometry*, John Wiley and Sons, 1997.
- [33] Guy Fayolle, Philippe Flajolet, and Micha Hofri, *On a functional equation arising in the analysis of a protocol for a multi-access broadcast channel*, *Advances in Applied Probability* **18** (1986), 441–472.
- [34] Guy Fayolle, Philippe Flajolet, and Micha Hofri, *On a functional equation arising in the analysis of a protocol for a multi-access broadcast channel*, *Adv. in Appl. Probab.* **18** (1986), no. 2, 441–472.
- [35] Guy Fayolle, Philippe Flajolet, Micha Hofri, and Philippe Jacquet, *Analysis of a stack algorithm for random multiple-access communication*, Institute of Electrical and Electronics Engineers. *Transactions on Information Theory* **31** (1985), no. 2, 244–254.
- [36] W. Feller, *An introduction to probability theory and its applications*, Wiley, New York, 1970.
- [37] A.F. Filippov, *On the distribution of the sizes of particles which undergo splitting*, *Theory of Probability and its Applications* **6** (1961), no. 3, 275–294.
- [38] James Allen Fill, Hosam M. Mahmoud, and Wojciech Szpankowski, *On the distribution for the duration of a randomized leader election algorithm*, *Ann. Appl. Probab.* **6** (1996), no. 4, 1260–1283.
- [39] Philippe Flajolet, *Approximate counting : a detailed analysis*, *BIT. Nordisk Tidskrift for Informationsbehandling (BIT)* **25** (1985), no. 1, 113–134.
- [40] Philippe Flajolet, Xavier Gourdon, and Philippe Dumas, *Mellin transforms and asymptotics : harmonic sums*, *Theoretical Computer Science* **144** (1995), no. 1-2, 3–58, Special volume on mathematical analysis of algorithms.
- [41] Philippe Flajolet and Philippe Jacquet, *Analytic models for tree communication protocols*, NATO Advance Science Institute Series (A. R. Odoni L. Bianco and G. Szegö, eds.), Series F : Computer and Systems Sciences, vol. 38, Springer Verlag, 1987, pp. 223–234.
- [42] Philippe Flajolet and Nigel Martin, *Probabilistic counting for data base applications*, *Journal of Computer Systems* **31** (1985), 182–209.
- [43] Dimitris Gatzouras, *On the lattice case of an almost-sure renewal theorem for branching random walks*, *Adv. Appl. Prob.* **32** (2000), 720–737.
- [44] Charles M. Goldie and Maller Ross A., *Stability of perpetuities*, *The Annals of Probability*. **28** (2000), no. 3, 1195–1218.
- [45] G. R. Grimmett and D. R. Stirzaker, *Probability and random processes*, second ed., The Clarendon Press Oxford University Press, New York, 1992.
- [46] B.M. Hambly and Michel L. Lapidus, *Random fractal strings : their Zeta functions, complex dimensions and spectral asymptotics*, *Transactions of the AMS*, To appear.
- [47] Micha Hofri, *Analysis of algorithms*, The Clarendon Press Oxford University Press, New York, 1995, Computational methods & mathematical tools.
- [48] P. Jacquet, W. Szpankowski, and J. Tang, *Average profile of the Lempel-Ziv parsing scheme for a Markovian source*, *Algorithmica* **31** (2001), no. 3, 318–360, Mathematical analysis of algorithms.

- [49] Philippe Jacquet and Mireille Régnier, *Trie partitioning process : Limiting distributions*, Lecture Notes in Computer Science, vol. 214, Springer Verlag, New York, 1986, pp. 196–210.
- [50] Philippe Jacquet and Wojciech Szpankowski, *Analytical de-Poissonization and its applications*, Theoret. Comput. Sci. **201** (1998), no. 1-2, 1–62.
- [51] Philippe Jacquet and Wojciech Szpankowski, *Analytical depoissonization and its applications*, Theoretical Computer Science **201** (1998), no. 1–2, 1–62.
- [52] P. Jagers, *General branching processes with biological applications*, John Wiley & Sons, 1975, A Wiley-Interscience Publication.
- [53] Svante Janson and Wojciech Szpankowski, *Analysis of an asymmetric leader election algorithm*, Electron. J. Combin. **4** (1997), no. 1, Research Paper 17, 16 pp. (electronic).
- [54] J.P. Kahane and J. Peyrière, *Sur certaines martingales de Benoît Mandelbrot*, Advances in Mathematics **22** (1976), 131–145.
- [55] J. F. C. Kingman, *Poisson processes*, Oxford Studies in Probability, 1993.
- [56] Peter Kirschenhofer, Helmut Prodinger, and Wojciech Szpankowski, *Analysis of a splitting process arising in probabilistic counting and other related algorithms*, Random Structures and Algorithms **9** (1996), no. 4, 379–401.
- [57] Charles Knessl, *Asymptotic and numerical studies of the leader election algorithm*, European J. Appl. Math. **12** (2001), no. 6, 645–664.
- [58] Charles Knessl and Wojciech Szpankowski, *Limit laws for the height in PATRICIA tries*, Journal of Algorithms **44** (2002), no. 1, 63–97, Analysis of algorithms.
- [59] Donald E. Knuth, *The art of computer programming. Volume 3*, Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1973, Sorting and searching, Addison-Wesley Series in Computer Science and Information Processing.
- [60] Steven P. Lalley, *Renewal theorems in symbolic dynamics, with applications to geodesic flows, non-Euclidean tessellations and their fractal limits*, Acta Mathematica **163** (1989), no. 1-2, 1–55.
- [61] ———, *Probabilistic methods in certain counting problems of ergodic theory*, Ergodic theory, symbolic dynamics, and hyperbolic spaces (Trieste, 1989), Oxford Sci. Publ., Oxford Univ. Press, New York, 1991, pp. 223–258.
- [62] M.L. Lapidus and M. van Frankenhuysen, *Fractal geometry and number theory*, Birkhäuser, Boston, 2000.
- [63] Christian Lavault and Guy Louchard, *Asymptotic analysis of a leader election algorithm*, submitted to Theoretical Computer Science, Elsevier, 2005.
- [64] Christian Lavault, Jean-Francois Marckert, and Vlady Ravelomanana, *Quasi-optimal leader election algorithms in radio networks with log-logarithmic awake time slots*, Proc. IEEE ICT'2003 **2** (2003), 1113–1119.
- [65] Nelly Litvak and Willem R. van Zwet, *On the minimal travel time needed to collect n items on a circle*, Annals of Applied Probability **14** (2004), no. 2, 881–902.

-
- [66] Q. Liu, *On generalized multiplicative cascades*, Stochastic Processes and their Applications **86** (2000), 263–286.
- [67] Gary Lorden, *On excess over the boundary*, Annals of Mathematical Statistics **41** (1970), 520–527.
- [68] Guy Louchard, *Exact and asymptotic distributions in digital and binary search trees*, Theoretical Informatics and Applications **21** (1987), 479–495.
- [69] Guy Louchard and Helmut Prodinger, *The moments problem of extreme-value related distribution functions*, 2004, Preprint.
- [70] ———, *The asymmetric leader election algorithm*, Annals of combinatorics (2006), to appear.
- [71] W. Lynch, *More combinatorial problems on certain trees*, The Computer Journal, Volume 7 (1965), 299–302.
- [72] Hosam Mahmoud, *On the average internal path length of m -ary search trees*, Acta Informatica **23** (1986), 111–117.
- [73] ———, *Evolution of random search trees*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons Inc., New York, 1992, A Wiley-Interscience Publication.
- [74] Navneet Malpani, Jennifer L. Welch, and Nitin Vaidya, *Leader election algorithms for mobile ad hoc networks*, DIALM '00 : Proceedings of the 4th international workshop on Discrete algorithms and methods for mobile computing and communications (New York, NY, USA), ACM Press, 2000, pp. 96–103.
- [75] Benoît Mandelbrot, *Multiplications aléatoires et distributions invariantes par moyennes pondérées*, Comptes Rendus de l'Académie des Sciences Série I **278** (1974), 289–292 and 355–358.
- [76] J.L. Massey, *Collision-resolution algorithms and random access communication*, Multi-User Communication Systems (G. Longo, ed.), CISM Courses and Lectures, vol. 265, Springer Verlag, 1981, pp. 73–140.
- [77] P. Mathys and P. Flajolet, *Q -ary collision resolution algorithms in random access systems with free or blocked channel access*, IEEE Transactions on Information Theory **31** (1985), 244–254, Special Issue on Random Access Communication.
- [78] R. Daniel Mauldin and S.C. Williams, *Random recursive constructions : Asymptotic geometric and topological properties*, Transactions of the AMS **295** (1986), no. 1, 325–346.
- [79] R. Metcalfe and D. Boggs, *Ethernet : Distributed packet switching for local computer networks*, Communications of the ACM **19** (1976), no. 7, 395–403.
- [80] Grégory Miermont, *Coalescence et fragmentation stochastique, arbres aléatoires et processus de Lévy*, Ph.D. thesis, Université de Paris VI, December 2003.
- [81] Hanène Mohamed and Philippe Robert, *A probabilistic analysis of some tree algorithms*, Annals of Applied Probability **15** (2005), no. 4, 2445–2471.
- [82] ———, *A probabilistic analysis of some tree algorithms (II) : dynamic case*, in preparation, 2006.

- [83] Olle Nerman, *On the convergence of supercritical general (C-M-J) branching processes*, Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete **57** (1981), no. 3, 365–395.
- [84] Jacques Neveu, *Processus ponctuels*, École d'Été de Probabilités de Saint-Flour (P.-L. Hennequin, ed.), Lecture Notes in Math., vol. 598, Springer-Verlag, Berlin, 1977, pp. 249–445.
- [85] Jacques Neveu, *Arbres et processus de galton-watson*, Annales de l'institut Henri Poincaré, Série B **22** (1986), 199–207.
- [86] William Parry, *Topics in ergodic theory*, Cambridge Tracts in Mathematics, vol. 75, Cambridge University Press, Cambridge, 1981.
- [87] Nicholas Pippenger, *An elementary approach to some analytic asymptotics*, SIAM Journal on Mathematical Analysis **24** (1993), no. 5, 1361–1377.
- [88] Jim Pitman, *Exchangeable and partially exchangeable random partitions*, Probability Theory and Related Fields **102** (1995), 145–158.
- [89] B. Pittel, *Asymptotical growth of a class of random trees*, Annals of Probability **13** (1985), no. 2, 414–427.
- [90] Helmut Prodinger, *How to select a loser*, Discrete Math. **120** (1993), no. 1-3, 149–159.
- [91] Danny Raz, Yuval Shavitt, and Lixia Zhang, *Distributed council election*, IEEE/ACM Transactions on Networking (2004), –.
- [92] Philippe Robert, *Stochastic networks and queues*, Stochastic Modelling and Applied Probability Series, vol. 52, Springer, New-York, June 2003.
- [93] ———, *On the asymptotic behavior of some algorithms*, Random Structures and Algorithms **27** (2005), no. 2, 235–250. MR 2155708
- [94] Philippe Robert, *Réseaux de communication, algorithmes et probabilités*, École Polytechnique, CMAP, Route de Saclay, Palaiseau 91120, Septembre 2005.
- [95] Salvador Roura, *Improved master theorems for divide-and-conquer recurrences*, Journal of the ACM **48** (2001), no. 2, 170–205.
- [96] M. Régnier, *A limiting distribution for quicksort*, RAIRO, Theoretical Informatics and Applications **23** (1989), 335–343.
- [97] Mireille Régnier and Philippe Jacquet, *New results on the size of tries*, IEEE Transactions on Information Theory **35** (1989), 203–205.
- [98] U. Rösler, *A limit theorem for "QUICKSORT"*, RAIRO, Theoretical Informatics and Applications **25** (1991), 85–100.
- [99] Robert Sedgewick and Philippe Flajolet, *Introduction to the analysis of algorithms*, Addison-Wesley, 1995.
- [100] Wojciech Szpankowski, *Average case analysis of algorithms on sequences*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley-Interscience, New York, 2001.
- [101] B. S. Tsybakov and V. A. Mikhaïlov, *Free synchronous packet access in a broadcast channel with feedback*, Problems Inform. Transmission **14** (1978), no. 4, 32–59.

-
- [102] Tsybakov, B.S. and Vvedenskaya, N.D., *Random multiple-access stack algorithm.*, Probl. Inf. Transm. **16** (1981), 230–243.
- [103] B. Vallée, *Dynamical sources in information theory : fundamental intervals and word prefixes*, Algorithmica **29** (2001), no. 1-2, 262–306, Average-case analysis of algorithms (Princeton, NJ, 1998).
- [104] B. Van Houdt and C. Blondia, *Throughput of Q-ary splitting algorithms for contention resolution in communication networks*, Communications in Information and Systems **4** (2005), no. 2, 135–164.
- [105] J. Van Velthoven, B. Van Houdt, and C. Blondia, *Transient analysis of tree-like processes and its application to random access systems*, SIGMETRICS Perform. Eval. Rev. **34** (2006), no. 1, 181–190.
- [106] Mark Daniel Ward and Wojciech Szpankowski, *Analysis of a randomized selection algorithm motivated by the LZ'77 Scheme*, Proc. of the First Workshop on Analytic Algorithmics and Combinatorics (ANALCO04), 2004.
- [107] Edward C. Waymire and S.C. Williams, *A general decomposition theory for random cascades*, Bulletin of the AMS **31** (1994), no. 2, 216–222.
- [108] Dan E. Willard, *Log-logarithmic protocols for resolving ethernet and semaphore conflicts*, STOC '84 : Proceedings of the sixteenth annual ACM symposium on Theory of computing (New York, NY, USA), ACM Press, 1984, pp. 512–521.
- [109] Jack K. Wolf, *Born again group testing : multiaccess communications*, IEEE Transactions on Information Theory **31** (1985), no. 2, 185–191.

Résumé

Cette thèse est dédiée à l'étude d'une large classe d'algorithmes, appelés *algorithmes en arbre*. En utilisant une représentation probabiliste appropriée, le comportement asymptotique de tels algorithmes est analysé. L'approche unifie les études faites sur ces algorithmes ainsi que simplifie et généralise certains résultats établis dans le domaine.

Mots-clés: Algorithmes en Arbre. Diviser pour Régner. Étrange Loi des Grands Nombres. Phénomène d'Oscillations Asymptotiques. Structures de Données. Tries. Théorème de Renouvellement.

Abstract

In this thesis a general class of tree algorithms is analyzed. It is shown that, by using an appropriate probabilistic representation of the quantities of interest, the asymptotic behavior of these algorithms can be obtained quite easily. This approach gives a unified probabilistic treatment of these questions. It simplifies and extends some of the results known in this domain.

Keywords: Splitting Algorithms. Divide and Conquer Algorithms. Unusual Laws of Large Numbers. Asymptotic Oscillating Behavior. Data Structures. Tries. Renewal Theorem.

